Learning Equilibria with Bandit Feedback

Maryam Kamgarpour
École Polytechnique Fédérale de Lausanne, Switzerland

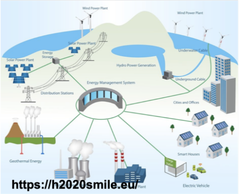
Workshop on Learning in Games, Toulouse, France

02.07.2024

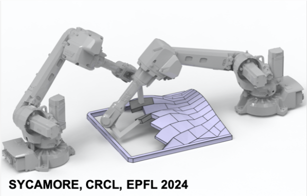
Background - Control systems

From . . .

to . . .



https://h2020smile.eu/



SYCAMORE, CRCL, EPFL 2024

MIRA Ltd. 2014

Problem of interest - Learning in games

Player i does not know J^i but can query it

How do players learn to optimize their decisions?

Introduction

Learning Nash equilibria
- Normal form games
- Markov games

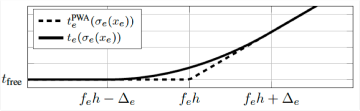
No-regret learning
- Normal form games
- Markov games

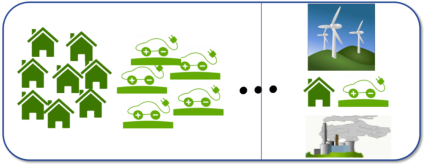
Conclusions

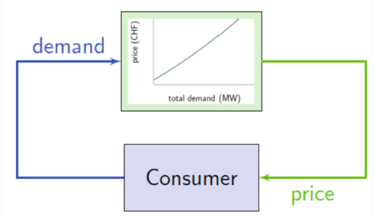
Outline

Convex games

- $J^i(a^i, a^{-i})$: convex in a^i, continuously differentiable
- $a^i \in A^i \subset \mathbb{R}^d$: convex and compact
- Examples
  - mixed strategy extension of a finite action game
  - traffic networks, electricity market







Nash equilibrium as a desirable solution outcome

$\boldsymbol{a}^* = (a^{*1}, a^{*2}, \ldots, a^{*N})$ is a Nash equilibrium if for every player i

$$J^i(a^{*i}, a^{*-i}) = \min_{a^i} J^i(a^i, a^{*-i})$$

- characterized by the <u>pseudo-gradient</u>: $\boldsymbol{M} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$

$$\boldsymbol{M}(\boldsymbol{a}) = [\nabla_i J^i(a^i, a^{-i})]_{i=1}^N$$

$\boldsymbol{a}^*$ Nash equilibrium $\iff \boldsymbol{M}(\boldsymbol{a}^*)^T(\boldsymbol{a} - \boldsymbol{a}^*) \geq 0, \forall \boldsymbol{a} \in \boldsymbol{A}$

Learning in convex games

Player i does not know J^i but can query it

Independent payoff-based approach:

$$\theta_{t+1}^i = \mathsf{Proj}_{A^i}(\theta_t^i - \eta_t \nabla_{\theta^i} \widehat{J^i(\theta_t^i, \theta_t^{-i})})$$

Learning in convex games

Player i does not know J^i but can query it

Independent payoff-based approach:

$$\theta^i_{t+1} = \mathsf{Proj}_{A^i}(\theta^i_t - \eta_t \nabla_{\theta^i} \widehat{J^i(\theta^i_t, \theta^{-i}_t}))$$

Challenges compared to the single agent setting:

1. How can agent i estimate $\nabla_{\theta^i} J^i(\boldsymbol{\theta})$ without knowing $\boldsymbol{\theta}$?
2. Under which conditions do we have convergence?

Independent estimation of local gradients

- Finite difference: $\widehat{\nabla_{\theta^i} J^i(\boldsymbol{\theta})} \approx \frac{J^i(\theta^i, {\color{red}\theta^{-i}}) - J^i(\theta^i + \delta, {\color{red}\theta^{-i}})}{\delta}$
  - requires others to stay with their action $\implies$ coordination

- approach: randomize query $\delta^i \sim \mathcal{N}(0, \sigma^2)$

- Construct $\widehat{\nabla_{\theta^i} J^i(\boldsymbol{\theta})}$ with one function evaluation
  - bias: $O(\sigma)$, variance $O(\frac{1}{\sigma^2})$ [Nesterov, Spokoiny 2019]

Alternatively, uniform distribution sampling [Flaxman et al. 2004

The game pseudo-gradient

Consider known gradients, unconstrained. Learning dynamics:

$$\begin{bmatrix} \theta^1_{t+1} \\ \vdots \\ \theta^N_{t+1} \end{bmatrix} = \begin{bmatrix} \theta^1_{t} \\ \vdots \\ \theta^N_{t} \end{bmatrix} - \eta_t \underbrace{\begin{bmatrix} \nabla_{\theta^1} J^1(\boldsymbol{\theta}_t) \\ \vdots \\ \nabla_{\theta^N} J^N(\boldsymbol{\theta}_t) \end{bmatrix}}_{\neq \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})}$$

- ex: $J^1(\boldsymbol{\theta}) = \theta^1\theta^2 = -J^2(\boldsymbol{\theta})$, $\begin{bmatrix} \nabla_{\theta^1} J^1(\boldsymbol{\theta}) \\ \nabla_{\theta^2} J^2(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix}$
- single agent analysis don't generally work

Sufficient conditions for convergence

$\boldsymbol{a}^*$ is strongly variationally stable: $\exists \nu > 0$:

$$\boldsymbol{M}(\boldsymbol{a})^T(\boldsymbol{a} - \boldsymbol{a}^*) > \nu \|\boldsymbol{a} - \boldsymbol{a}^*\|^2, \ \forall \boldsymbol{a} \in \boldsymbol{A}$$

- example $J^i(\boldsymbol{a}) = a^1 a^2 a^3 + (a^i)^2$, $a^i \in [-1, 2], i \in \{1, 2, 3\}$

Sufficient conditions for convergence

$\boldsymbol{a}^*$ is strongly variationally stable: $\exists \nu > 0$:

$$\boldsymbol{M}(\boldsymbol{a})^T(\boldsymbol{a} - \boldsymbol{a}^*) > \nu \|\boldsymbol{a} - \boldsymbol{a}^*\|^2, \ \forall \boldsymbol{a} \in \boldsymbol{A}$$

- example $J^i(\boldsymbol{a}) = a^1 a^2 a^3 + (a^i)^2$, $a^i \in [-1, 2], i \in \{1, 2, 3\}$

Algorithm: $\boldsymbol{\theta}_{t+1} = \mathsf{Proj}_{\boldsymbol{A}}\big(\boldsymbol{\theta}_t - \eta_t \hat{\boldsymbol{M}}(\boldsymbol{\theta}_t)\big)$

Theorem

Assume $\boldsymbol{M}$ Lipschitz and $\boldsymbol{a}^*$ strongly VS. For $\sum_t \eta_t = \infty$, $\sum_t \frac{\eta_t^2}{\sigma_t^2} < \infty$, $\boldsymbol{\theta}_t$ converges almost surely to $\boldsymbol{a}^*$

Payoff-based learning leverages pseudo-gradient properties

Recent progress

- Mere monotonicity of $\boldsymbol{M}(\boldsymbol{a}) \supseteq$ zero-sum matrix games:
  extra-gradient, optimistic gradient descent-ascent, Tikhonov regularization, ...
- Local variational stability $\implies$ local convergence
- Convergence rates

[Tatarenko, **MK**, IEEE TAC 2019, IEEE TCNS 2024, ECC 2024]

[Bravo et al., 2018], [Mertikopoulos et al. 2018], [Gao, Pavel, 2022], ...

Challenge: many games including Markov games do not satisfy above conditions

Markov games

- Dynamics: $s_{h+1} \sim P(.|s_h, a_h^1, \ldots, a_h^N)$
- Policy $\pi^i : S \to \Delta(A^i)$
- $V_s^i(\pi^i, \pi^{-i}) = \mathbb{E}_{P,\boldsymbol{\pi}} \sum_{h=0}^{\infty} \gamma^t R^i(s_h, \pi^1(s_h), \ldots, \pi^N(s_h))$
- Nash equilibrium:

$$V_s^i(\pi^{*i}, \pi^{*-i}) \geq V_s^i(\pi^i, \pi^{*-i}), \ \forall \pi^i, \ \forall i$$

note change of notation: from costs to rewards and value function for players

Multiagent reinforcement learning approach

~~Given $s_{h+1} \sim P(.|s_h, a_h^1, \dots, a_h^N)$~~

- Parametrize a policy $a_t^i \sim \pi_{\theta^i}(.|s_h)$, $\theta^i \in \mathbb{R}^d$
- Find equilibrium $\boldsymbol{\theta}^* = (\theta^1, \dots, \theta^N)$ by interacting with the system

Policy gradient class of algorithms

Single agent RL: $V(\theta) = \mathbb{E}_{P,\pi} \sum_{h=0}^{\infty} \gamma^t R(s_h, \pi(s_h))$

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta V(\theta_t)$$

- convergence under *gradient dominance* condition [Agarwal et al., 2021],[Hu et al. 2023], [Bhandari et al. 2024], ...

Multiagent RL: $V^i(\theta^i, \theta^{-i}) = \mathbb{E}_{P,\boldsymbol{\pi}} \sum_{h=0}^{\infty} \gamma^t R^i(s_h, \pi^1(s_h), \dots, \pi^N(s_h))$

$$\theta_{t+1}^i = \theta_t^i - \eta_t \nabla_{\theta^i} V^i(\theta_t^i, \theta_t^{-i})$$

- generally non-convergent

Challenging even in linear quadratic setting

single agent

$$J(\theta) = \mathbb{E}_{s_0}[\sum_{h=0}^{\infty} s_h^T Q s_h + a_h^T R a_h]$$

$$s_{h+1} = A s_h + B a_h$$

$$a_h = \theta^T s_h, \; s_0 \sim \mathcal{D}$$

Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator

Maryam Fazel [*1] **Rong Ge** [*2] **Sham M. Kakade** [*1] **Mehran Mesbahi** [*1]

Abstract

Direct policy gradient methods for reinforcement learning and continuous control problems are a popular approach for a variety of reasons: 1) they

2016) and Atari game playing (Mnih et al., 2015). Deep reinforcement learning (DeepRL) is becoming increasingly popular for tackling such challenging sequential decision making problems.

multiagent

$$J^i(\boldsymbol{\theta}) = \mathbb{E}_{s_0}[\sum_{h=0}^{\infty} s_h^T Q^i s_h + (a^i)_h^T R^i a_h^i]$$

$$s_{h+1} = A s_h + \sum_{i=1}^{N} B a_h^i$$

$$a_h^i = (\theta^i)^T s_h, \; x_0 \sim \mathcal{D}$$

Policy-Gradient Algorithms Have No Guarantees of Convergence in Linear Quadratic Games

Eric Mazumdar
University of California, Berkeley
Berkeley, CA
mazumdar@berkeley.edu

Lillian J. Ratliff
University of Washington
Seattle, WA
ratliffl@uw.edu

Michael I. Jordan
University of California, Berkeley
Berkeley, CA
jordan@cs.berkeley.edu

S. Shankar Sastry
University of California, Berkeley
Berkeley, CA
sastry@coe.berkeley.edu

ABSTRACT

We show by counterexample that policy-gradient algorithms have no guarantees of even local convergence to Nash equilibria in continuous action and state space multi-agent settings. To do so, we analyze gradient-play in N–player general-sum linear quadratic games, a classic game setting which is recently emerging as a bench-

of multi-agent reinforcement learning have made use of policy optimization algorithms such as multi-agent actor-critic [13, 17, 30], multi-agent proximal policy optimization [2], and even simple multi-agent policy-gradients [15] in problems where the various agents have high-dimensional continuous state and action spaces like StarCraft II [32].

Multiagent policy gradient convergence condition

Results on subclasses of Markov games or depend on equilibria

- Zero-sum [Daskalakis et al. 2020], [Wei et al. 2021], [Cen et al. 2021], [K. Zhang et al. 2023], ...
- Potential [Leonardos et al. 2022], [R. Zhang et al. 2021], [Ding et al. 2022]
- Variationally stable equilibrium [Giannou et al. 2022] $\Longrightarrow$ local convergence

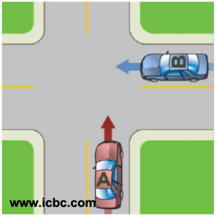
Our focus: presented as posters here

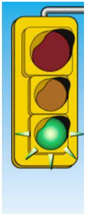
- Linear quadratic setting: conditions to be a potential game, characterizing number of equilibria
- Zero-sum Markov games: relaxing past assumptions while strengthening convergence result

Relaxing the equilibrium notion

A *probability distribution* $\mathcal{P}^*$ on $\boldsymbol{A}$ is an equilibrium

$$\forall i \quad \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{P}^*}[J^i(\boldsymbol{\theta})] \leq \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{P}^*}[J^i(\tilde{\theta}^i, \theta^{-i})],\ \forall \tilde{\theta}^i$$

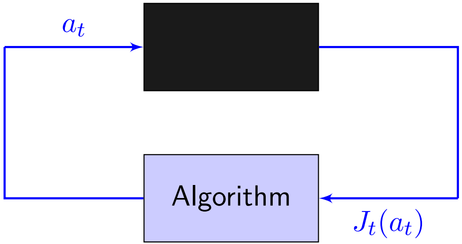




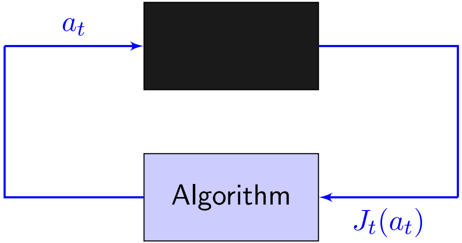
- Focus: learning algorithms that scale with number of agents

Outline

Game as an adversarial bandit problem



Game as an adversarial bandit problem

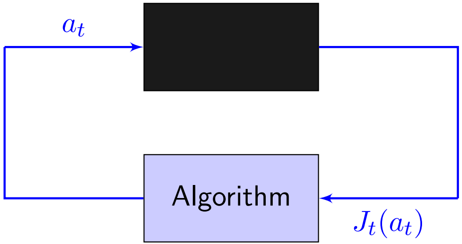


In a game: $J_t(.) := J^i(., a_t^{-i})$ for player i

Benchmark: no-regret

- Regret: $R(T) = \underbrace{\sum_{t=0}^{T} J_t(a_t)}_{\text{incurred cost}} - \underbrace{\min_a \sum_{t=0}^{T} J_t(a)}_{\text{best cost}}$

Game as an adversarial bandit problem



In a game: $J_t(.) := J^i(., a_t^{-i})$ for player i
Benchmark: no-regret

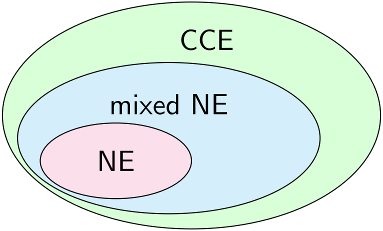
- Regret: $R(T) = \underbrace{\sum_{t=0}^{T} J_t(a_t)}_{\text{incurred cost}} - \underbrace{\min_a \sum_{t=0}^{T} J_t(a)}_{\text{best cost}}$

Algorithm is no-regret: $R(T)/T \to 0$

No-regret learning and equilibria

Let each player adopt a no-regret algorithm

- empirical distribution of actions $\rightarrow$ *coarse-correlated equilibrium*



Remark

- CCEs may have better efficiency but
- CCEs can have weight on strictly dominated actions

Multiplicative weight algorithms for no-regret

Player i's actions $\{1, 2, \ldots, n\}$, unknown cost: $J_t(.)$

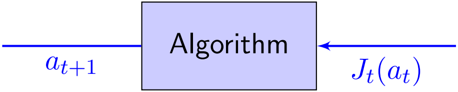
Probability distribution on actions: w_t

- sample: $a_t \sim w_t$
- play the action: $J_t(a_t)$
- update probabilities w_{t+1}, based on $J_t(a_t)$
  - bandit feedback: $w_{t+1}(k) = w_t(k) \exp^{-\eta_t J_t(k)}$, for $k = a_t$
  - full feedback: $w_{t+1}(k) = w_t(k) \exp^{-\eta_t J_t(k)}$, for $\forall k$

Optimal regret rates based on player's feedback

n: number of actions for player, T: number of iterations

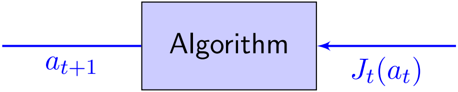
- Bandit feedback [Auer et al. 2003]



Optimal regret rates based on player's feedback

n: number of actions for player, T: number of iterations

- Bandit feedback [Auer et al. 2003]

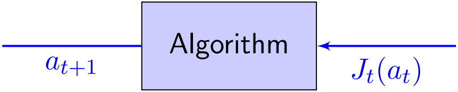


  - Regret $R(T)$ grows as $\sqrt{Tn \log n}$

Optimal regret rates based on player's feedback

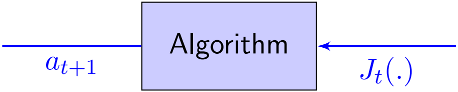
n: number of actions for player, T: number of iterations

- Bandit feedback [Auer et al. 2003]



  - Regret $R(T)$ grows as $\sqrt{T{\color{red}n}\log n}$

- Full feedback [Freund et al.1997]

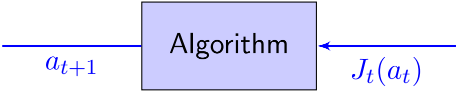


  - Regret $R(T)$ grows as $\sqrt{T\log n}$

Optimal regret rates based on player's feedback

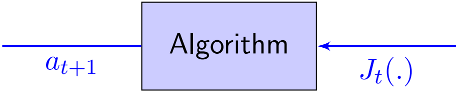
n: number of actions for player, T: number of iterations

- Bandit feedback [Auer et al. 2003]



  - Regret $R(T)$ grows as $\sqrt{Tn\log n}$

- Full feedback [Freund et al.1997]



  - Regret $R(T)$ grows as $\sqrt{T\log n}$

Can we improve the dependence on n?

Idea: mimic full feedback

Notice: $J^i(., a_t^{-i})$ is a static function
Algorithms achieving optimal regret rate:

- bandit: $w_{t+1}(k) = w_t(k) \exp^{-\eta_t J_t(k)}$
- full: $w_{t+1}(.) = w_t(.) \exp^{-\eta_t J_t(.)}$

Player i estimates its cost from past data $\hat{J}_t^i(a_t^i, a_t^{-i})$, a_t^i, a_t^{-i}

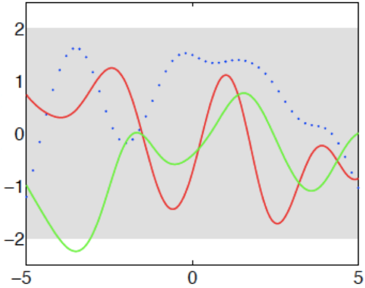
- mimic full: $w_{t+1}(.) = w_t(.) \exp^{-\eta_t \hat{J}_t^i(., a_t^{-i})}$

Modeling class for cost function

J has a bounded norm in a reproducing Kernel space $\implies$
J can be modeled by a Gaussian process

- $J(\boldsymbol{a}) \sim \mathcal{GP}\big(\mu(\boldsymbol{a}), k(\boldsymbol{a}, \boldsymbol{a}')\big)$
- μ: mean, k: covariance (kernel)
- examples of covariance function:

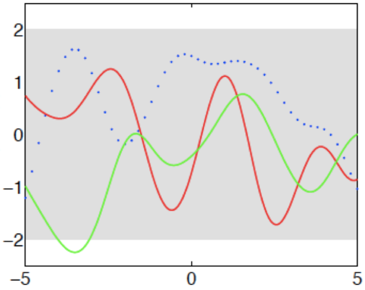
$$k_{poly}(\boldsymbol{a}, \boldsymbol{a}') = \left(l + \boldsymbol{a}^\top \boldsymbol{a}'\right)^d, \; k_{SE}(\boldsymbol{a}, \boldsymbol{a}') = \exp\left(-\frac{\|\boldsymbol{a} - \boldsymbol{a}'\|^2}{l^2}\right)$$



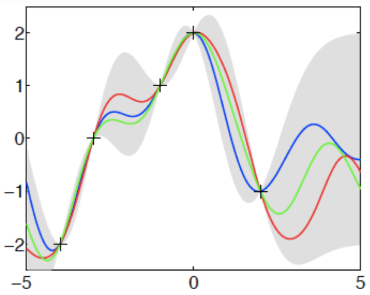
Estimating the cost function distribution

$$J(\boldsymbol{a}) \sim \mathcal{GP}\big(\mu(\boldsymbol{a}), k(\boldsymbol{a}, \boldsymbol{a}')\big)$$

- observe: costs $J(\boldsymbol{a}_l)$, actions $\boldsymbol{a}_l$, $\quad l = 1, \ldots, t$
- obtain posterior distribution of $J(.)$
  - analytic formula for updating mean $\mu_t(.)$ and variance $\sigma_t(.)$



prior

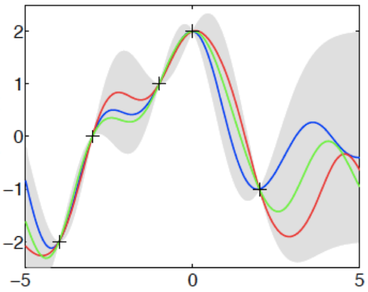


posterior

Confidence bounds on the estimated cost

$$\hat{J}_t(\boldsymbol{a}) := \mu_t(\boldsymbol{a}) - \beta_t \sigma_t(\boldsymbol{a})$$

- $\hat{J}_t(\boldsymbol{a})$ small $\implies$ cost low or uncertainty high
- $\beta_t > 0$ chosen to ensure $\hat{J}_t(\boldsymbol{a}) \leq J(\boldsymbol{a})$ with high probability



confidence bound on GP

Gaussian process multiplicative weight algorithm (GPMW)

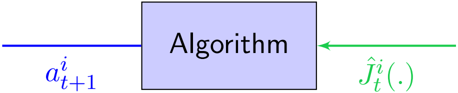
Player i's actions $\{1, 2, \ldots, n\}$, unknown cost: $J(a^i, a^{-i})$

Optimistic cost estimate at time t: $\hat{J}^i_t(\boldsymbol{a}) := \mu_t(\boldsymbol{a}) - \beta_t \sigma_t(\boldsymbol{a})$
Probability distribution on actions: w_t

- sample: $a \sim w_t$
- observe: $J^i(a, a_t^{-i})$ and a_t^{-i}
- update $\hat{J}^i_t(.)$
  - $w_{t+1}(k) = w_t(k) \exp^{-\eta_t \hat{J}^i_t(\boldsymbol{a})}$, $\forall k$

GPMW regret rates

- Mimic full feedback by observing others' actions

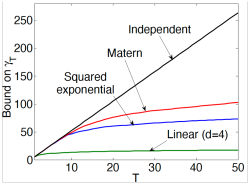


Theorem

Assume: player's cost from a GP prior

- *Regret grows as:* $\left(\sqrt{T \log n} + \gamma_T \sqrt{T}\right)$

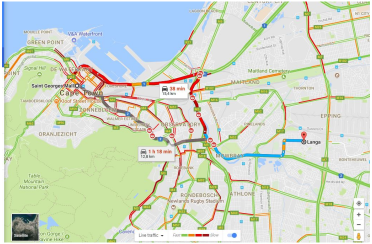
[Sessa, Bogunovic, MK, Krause, NeurIPS 2019]



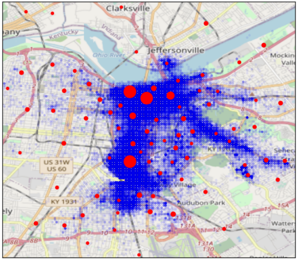
bound on γ_T based on the kernel [Srinivas et al. 2010]

Extensions of GP multi-agent learning

- contextual games [NeurIPS 2020],[AISTATS 2024], equilibria efficiency and game design [AISTAT2019, ICML2021]
- transportation network, resource allocation, electricity auctions, autonomous driving, energy management



Reducing congestion on road networks [NeurIPS 2020]



Balancing bike distribution to maximize utility [ICML 2021]

Extension to multi-agent reinforcement learning (MARL)

- Dynamics $s_{h+1} = f(s_h, a_h^1, a_h^2 \ldots, a_h^N) + \omega_h$
  - $s_h \in S \subset \mathbb{R}^p$, $a_h^i \in A^i \subset \mathbb{R}^q$
- Objective $V^i(\pi^i, \pi^{-i}) = \mathrm{E}[\sum_{h=0}^{H-1} r^i(s_h, \pi^i(s_h), \pi^{-i}(s_h))]$

Approach: estimate the transition function f via its posterior mean $\mu_t(s, \boldsymbol{a}) \in \mathbb{R}^p$ and confidence functions $\Sigma_t(s, \boldsymbol{a}) \in \mathbb{R}^{p \times p}$

Approach: model-based learning of equilibrium distribution

- Initialize $\mathcal{P}_0$. For $t = 0, 1, \ldots$
  - sample $(\pi_t^1, \ldots, \pi_t^N) \sim \mathcal{P}_t$

$$(\pi_t^1, \ldots, \pi_t^N) \longrightarrow \blacksquare \longrightarrow s_0, \mathbf{a}_0, s_1, \ldots, s_{H-1}$$

  - estimate $P(.|s_h, a_h^1, \ldots, a_h^N) \to \{\bar{V}_t^i(\boldsymbol{\theta}\ )\}_{i=1}^N$
  - compute $\mathcal{P}_{t+1}$ as the equilibrium distribution of $\{\bar{V}_t^i(\boldsymbol{\theta}\ )\}_{i=1}^N$
- $\bar{V}_t^i(.)$: optimistic estimate of $V^i(.)$ at iteration t

Regret of the MARL algorithm

Dynamic regret

$$R^i(T) := \sum_{t=1}^{T} \max_{\pi \in \Pi^i} \mathbb{E}_{\pi_t^{-i}} \left[V^i(\pi, \pi_t^{-i}) \right] - \mathbb{E}_{\boldsymbol{\pi}_t} \left[V^i(\boldsymbol{\pi}_t) \right]$$

Theorem

Under Lipschitz continuity of f, $\{r^i, \pi^i\}_{i=1}^N$

$$R^i(T) = \mathcal{O}\left(LH^{1/2}\sqrt{T\mathcal{I}_T}\right) + \sum_{t=0}^{T} \epsilon_t$$

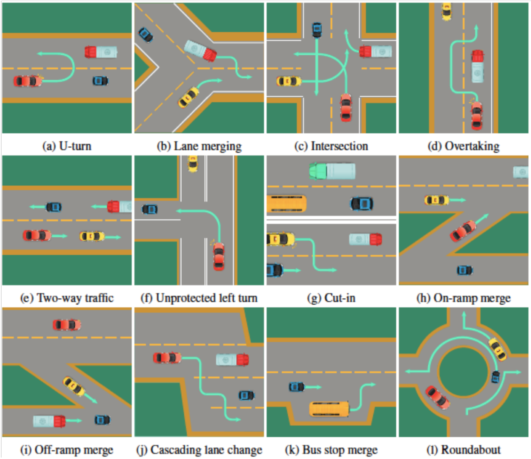
- $\mathcal{I}_T(p, H, \gamma_{HT})$: *information gain*
- ϵ_t: *approximate CCE for* $\{\bar{V}_t^i(\boldsymbol{\theta}\ )\}_{i=1}^N$

[Sessa, **MK**, Krause, ICML 2022]

Example: Multi-agent RL in autonomous driving

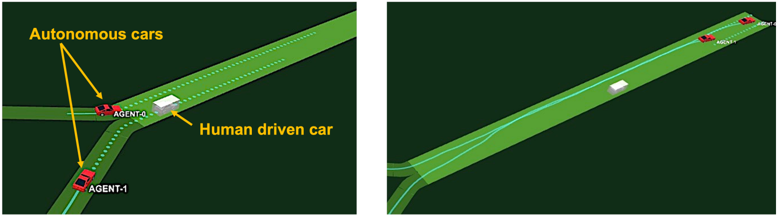
SMARTS autonomous car simulation environment [Zhou et al. 2021]

- testing multi-agent RL algorithms for autonomous driving
- realistic traffic data and car dynamics



Multiagent reinforcement learning for autonomous driving

- Objective: progress towards the goal, avoid collision
- Dynamics: $P(.|s_h, a_h^1, a_h^2)$
  - s: positions and velocities of cars
  - a^i: heading and speed, $i = 1, 2$
  - $\pi_{\theta^i}(s)$: parametrized by neural networks, $i = 1, 2$

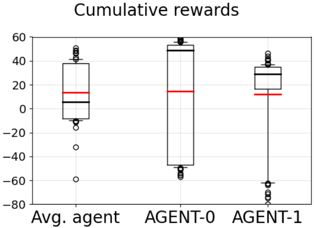


The autonomous cars can coordinate and overtake the human-driven car

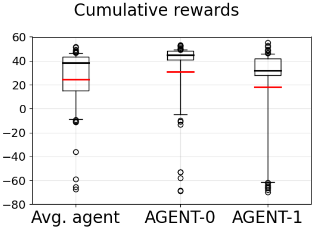
Implementation on multiagent autonomous car simulation environment [Zhou et al. 2021]

Learning to coordinate

Learning to coordinate $\implies$ less breaking, more successful merges

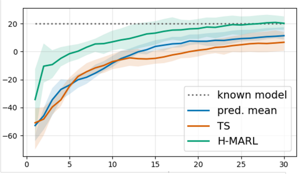


Single-agent optima



Multi-agent equilibrium

Average rewards for the agents



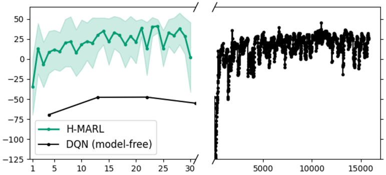


Figure: left: value of optimism, right: value of learning the model

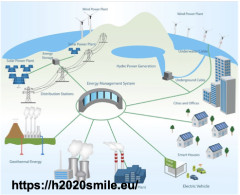
Outline

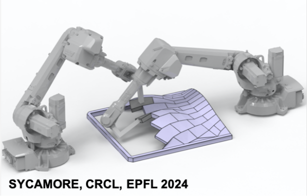
Summary

- Payoff-based learning of Nash equilibria
  - require assumptions on pseudo-gradient or the equilibrium
  - challenging to extend to Markov games
- No-regret learning
  - tractable and ensure convergence to CCEs
  - can improve rates using a model-based approach

Outlook

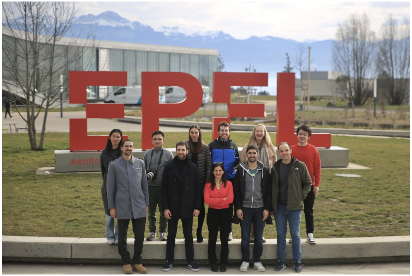
- Learning equilibria in Markov games under coupling constraint
- Provable algorithms under partial and asymmetric information
- Learning of “good” equilibria, mechanism design
- Applications: power markets, robotics, autonomous driving





Acknowledgements

- Former and current students and postdocs: O Karaca, L Furieri, P Giuseppe Sessa, A Maddux, G Salizzoni, S Hosseinirad, R Ouhamma
- Collaborators: T Tatarenko, A Krause, Bugonovic
- Funding : ERC, NSERC Canada, Swiss National Fund, NCCR Automation



https://www.epfl.ch/labs/sycamore/