Learning Methods in Mean Field Games
Parts 1 & 2

Mathieu LAURIÈRE

NYU Shanghai

"Numerical methods for optimal transport problems, mean field games, and multi-agent dynamics"
January 8-12, 2024
Universidad Técnica Federico Santa María, Valparaíso, Chile

Survey paper: `arXiv:2205.12944`

Questions, comments or suggestions are most welcome.

Based on joint works with many people, including:

Andrea Angiuli, Olivier Bachem, Tamer Basar, Theophile Cabannes, René Carmona, Gökçe Dayanikli, Romuald Élie, Jean-Pierre Fouque, Matthieu Geist, Maximilien Germain, Sertan Girgin, Kenza Hamidouche, Ruimeng Hu, Ayush Jain, Alec Koppel, Raphael Marinier, Paul Muller, Rémi Munos, Julien Pérolat, Sarah Perrin, Huyên Pham, Olivier Pietquin, Georgios Piliouras, Mark Rowland, Zongjun Tan, Karl Tuyls, Muhammad Aneeq uz Zaman, . . .

as well as other people's works

Outline

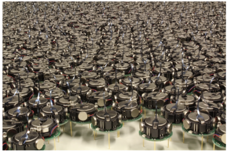
Motivations

Flocking

Crowd motion

Traffic flow

Collective AI

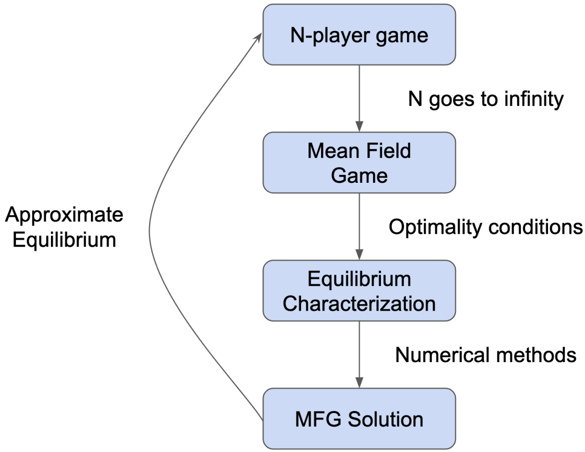


[Image credits: Unsplash, Wikimedia Commons (Kilobots)]

Some other existing approaches *("What MFGs are* ***not****")*

- Dynamical systems:
  - describe the dynamics of one or many agents, sometimes mean field
  - but usually **no rationality** (optimization)
- Agent based models (ABM):
  - "Agent-based models are a kind of **microscale model** that simulate the simultaneous operations and interactions of multiple agents in an attempt to re-create and predict the appearance of complex phenomena. "
  - "Individual agents are typically characterized as **boundedly rational**, presumed to be acting in what they perceive as their own interests, such as reproduction, economic benefit, or social status, using heuristics or simple decision-making rules." (Wikipedia)
- Game theory
  - optimization aspects
  - notion of Nash equilibrium, social optimum, ...
  - but usually limited to a **finite (small) number of agents**
- Evolutionary game theory (EGT)
  - "application of game theory to evolving populations in biology"
  - "an evolutionary version of game theory **does not require players to act rationally** – only that they have a strategy" (Wikipedia)
- Non-atomic anonymous games
  - continuum of rational players; each player has her **own index** and own strategy
  - mostly limited to static games; difficulties for dynamic, stochastic games

MFG paradigm in a nutshell



Some References

- Introduction to Mean Field Games:
  - Pierre-Louis Lions' lectures at Collège de France (`https://www.college-de-france.fr/`)
  - Pierre Cardaliaguet's notes (2013):
    `https://www.ceremade.dauphine.fr/ cardaliaguet/MFG20130420.pdf`
  - Gomes, D. A., & Saúde, J. (2014). Mean field games models—a brief survey. Dynamic Games and Applications, 4, 110-154.
  - Cardaliaguet, P., & Porretta, A. (2020). An Introduction to Mean Field Game Theory. In *Mean Field Games* (pp. 1-158). Springer, Cham.
  - Carmona, Delarue, Graves, Lacker, Laurière, Malhamé & Ramanan: Lecture notes of the 2020 AMS Short Course on Mean Field Games (American Mathematical Society), organized by François Delarue
  - Achdou, Y., Cardaliaguet, P., Delarue, F., Porretta, A., & Santambrogio, F. (2021). Mean Field Games: Cetraro, Italy 2019 (Vol. 2281). Springer Nature.
  - Delarue, F. (Ed.). (2021). Mean Field Games (Vol. 78). American Mathematical Society.

Some References

- Monographs on Mean Field Games and Mean Field Control:
  - Bensoussan, A., Frehse, J., & Yam, P. (2013). *Mean field games and mean field type control theory* (Vol. 101). New York: Springer.
  - Gomes, D. A., Pimentel, E. A., & Voskanyan, V. (2016). *Regularity theory for mean-field game systems. New York: Springer.*
  - Carmona, R., & Delarue, F. (2018). *Probabilistic Theory of Mean Field Games with Applications I: Mean Field FBSDEs, Control, and Games* (Vol. 83). Springer.
  - Carmona, R., & Delarue, F. (2018). *Probabilistic Theory of Mean Field Games with Applications II: Mean Field Games with Common Noise and Master Equations* (Vol. 84). Springer.

Some References

- Surveys about numerical methods for MFGs:

- Achdou, Y. (2013). Finite difference methods for mean field games. In *Hamilton-Jacobi equations: approximations, numerical analysis and applications* (pp. 1-47). Springer, Berlin, Heidelberg.
- Achdou, Y., & Laurière, M. (2020). Mean Field Games and Applications: Numerical Aspects. *Mean Field Games: Cetraro, Italy 2019, 2281*, 249.
- Laurière, M. (2021). Numerical Methods for Mean Field Games and Mean Field Type Control. Lecture notes for the AMS'20 short course. arXiv preprint arXiv:2106.06231.
- Carmona, R., & Laurière, M. (2021). Deep Learning for Mean Field Games and Mean Field Control with Applications to Finance. arXiv preprint arXiv:2107.04568.
- Hu, R., & Laurière, M. (2023). Recent developments in machine learning methods for stochastic control and games. arXiv preprint arXiv:2303.10257.
- Laurière, M., Perrin, S., Geist, M., & Pietquin, O. (2022). Learning mean field games: A survey. arXiv preprint arXiv:2205.12944.

MFG & (R)L: Motivations

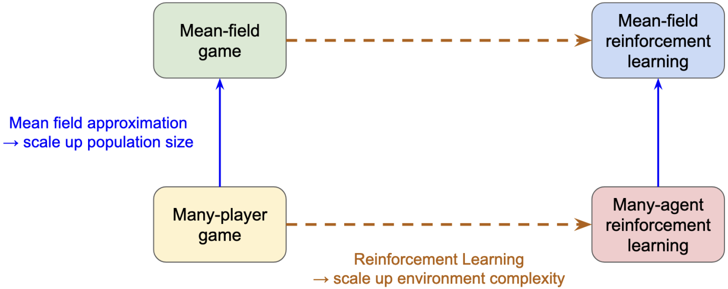
Main motivation: real-world applications require methods for large-scale problems

- Scaling up **population size** $\rightarrow$ **Mean Field Games**
  - Initial papers: Lasry & Lions; Caines, Huang & Malhamé (2006-2007)
  - Books: Bensoussan, Frehse & Yam; Carmona & Delarue; ...
- Scaling up **environment complexity** $\rightarrow$ (model-free) **Reinforcement Learning**
  - Book: Sutton & Barto; ...
  - Applications: Robotics, language processing, games, ...

MFG & (R)L: Motivations

Main motivation: real-world applications require methods for large-scale problems

- Scaling up **population size** $\rightarrow$ **Mean Field Games**
  - Initial papers: Lasry & Lions; Caines, Huang & Malhamé (2006-2007)
  - Books: Bensoussan, Frehse & Yam; Carmona & Delarue; ...
- Scaling up **environment complexity** $\rightarrow$ (model-free) **Reinforcement Learning**
  - Book: Sutton & Barto; ...
  - Applications: Robotics, language processing, games, ...



Motivations behind this overview

Rapidly growing literature

Goal: overview of the landscape & codes to make this topic more easily accessible

A few key aspects:

1. Problem setting

   $\rightarrow$ *continuous / discrete time & space, ...*

Motivations behind this overview

Rapidly growing literature

Goal: overview of the landscape & codes to make this topic more easily accessible

A few key aspects:

1. Problem setting

   $\rightarrow$ *continuous / discrete time & space, ...*

2. Solution concept

   $\rightarrow$ *cooperative / non-cooperative, ...*

Motivations behind this overview

Rapidly growing literature

Goal: overview of the landscape & codes to make this topic more easily accessible

A few key aspects:

1. Problem setting

   $\rightarrow$ *continuous / discrete time & space, . . .*
2. Solution concept

   $\rightarrow$ *cooperative / non-cooperative, . . .*
3. Iterative learning methods

   $\rightarrow$ *learning solution with "ideal" updates*

Motivations behind this overview

Rapidly growing literature

Goal: overview of the landscape & codes to make this topic more easily accessible

A few key aspects:

1. Problem setting
   $\rightarrow$ *continuous / discrete time & space, ...*
2. Solution concept
   $\rightarrow$ *cooperative / non-cooperative, ...*
3. Iterative learning methods
   $\rightarrow$ *learning solution with "ideal" updates*
4. Reinforcement learning
   $\rightarrow$ *learning solution with model-free updates*

Motivations behind this overview

Rapidly growing literature

Goal: overview of the landscape & codes to make this topic more easily accessible

A few key aspects:

1. Problem setting

   $\rightarrow$ *continuous / discrete time & space, ...*
2. Solution concept

   $\rightarrow$ *cooperative / non-cooperative, ...*
3. Iterative learning methods

   $\rightarrow$ *learning solution with "ideal" updates*
4. Reinforcement learning

   $\rightarrow$ *learning solution with model-free updates*
5. Implementation

   $\rightarrow$ *code samples (OpenSpiel, ...)*

Learning

Recent successes of learning in games, e.g.:

Go [SHM+16, SSS+17, SHS+18], Chess [CHJH02], Checkers [SBB+07], Hex [ATB17], Starcraft II [VBC+19], poker games [BS17, BS19, MSB+17, BBJT15], Stratego [MLFB20], . . .

Learning

Recent successes of learning in games, e.g.:

Go [SHM$^+$16, SSS$^+$17, SHS$^+$18], Chess [CHJH02], Checkers [SBB$^+$07], Hex [ATB17], Starcraft II [VBC$^+$19], poker games [BS17, BS19, MSB$^+$17, BBJT15], Stratego [MLFB20], ...

At least **two interpretations** of "learning":

- Game theory, economics, ...:
  Fudenberg & Levine [FL09][1]: *"The theory of learning in games [...] examines how, which, and what kind of equilibrium might arise as a consequence of a long-run nonequilibrium process of learning, adaptation, and/or imitation"*

- Machine Learning, Reinforcement Learning, ...:
  Mitchell [M$^+$97][2]: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*

[1] Fudenberg, D., & Levine, D. K. (2009). Learning and equilibrium. *Annu. Rev. Econ.*, 1(1), 385-420.

[2] Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill. ISBN: 978-0-07-042807-2

Outline

N-Player Stochastic Differential Game

For now, continuous time and continuous space:

- N players
- Player i's state is $X_t^i \in \mathbb{R}^d$
- with dynamics:

$$dX_t^i = b(t, X_t^i, \alpha_t^i, \mu_t^N)dt + \sigma dW_t^i, \qquad X_0^i \sim m^0$$

- W^i is an idiosyncratic (individual) noise, independent from other W^j's
- The empirical state distribution is: $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{X_t^j}$

N-Player Stochastic Differential Game

For now, continuous time and continuous space:

- N players
- Player i's state is $X_t^i \in \mathbb{R}^d$
- with dynamics:

$$dX_t^i = b(t, X_t^i, \alpha_t^i, \mu_t^N)dt + \sigma dW_t^i, \qquad X_0^i \sim m^0$$

- W^i is an idiosyncratic (individual) noise, independent from other W^j's
- The empirical state distribution is: $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{X_t^j}$
- Instantaneous cost function f and terminal cost function g
- Goal for player i: minimize over α^i the total expected cost:

$$J(\alpha^i, \alpha^{-i}) = \mathbb{E}\left[\int_0^T f(t, X_t^i, \alpha_t^i, \mu_t^N)dt + g(X_T^i, \mu_T^N)\right]$$

N-Player Stochastic Differential Game: Solution Concepts

Two concepts:

- **Nash equilibrium** $(\hat{\alpha}^1, \dots, \hat{\alpha}^N)$: for all $i = 1, \dots, N$ and all α^i,

$$J(\hat{\alpha}^i, \hat{\alpha}^{-i}) \leq J(\alpha^i, \hat{\alpha}^{-i})$$

  $\rightarrow$ no incentive for unilateral deviations

  $\rightarrow$ **fixed point** problem

N-Player Stochastic Differential Game: Solution Concepts

Two concepts:

- **Nash equilibrium** $(\hat{\alpha}^1, \dots, \hat{\alpha}^N)$: for all $i = 1, \dots, N$ and all α^i,

$$J(\hat{\alpha}^i, \hat{\alpha}^{-i}) \leq J(\alpha^i, \hat{\alpha}^{-i})$$

  $\rightarrow$ no incentive for unilateral deviations
  $\rightarrow$ **fixed point** problem

- **Social optimum** $(\alpha^{*1}, \dots, \alpha^{*N})$: for all $i = 1, \dots, N$ and all $(\alpha^1, \dots, \alpha^N)$,

$$\bar{J}(\alpha^{*1}, \dots, \alpha^{*N}) = \frac{1}{N} \sum_{i=1} J(\alpha^{*i}, \alpha^{*-i}) \leq \bar{J}(\alpha^1, \dots, \alpha^N) = \frac{1}{N} \sum_{i=1} J(\alpha^i, \alpha^{-i})$$

  $\rightarrow$ no incentive for joint deviations
  $\rightarrow$ **optimization** problem

In general, they are different, which leads to the notion of Price of Anarchy

Mean Field Limit

Pass to the limit $N \to +\infty$?

Key assumptions:

- homogeneity: all the agents have the same f, g, b, σ
- symmetry/anonymity: interactions are only through the empirical distribution

Mean Field Limit

Pass to the limit $N \to +\infty$?

Key assumptions:

- homogeneity: all the agents have the same f, g, b, σ
- symmetry/anonymity: interactions are only through the empirical distribution

In the limit, we expect to have: the cost for one representative player is:

$$J(\alpha, \mu) = \mathbb{E}\left[\int_0^T f(t, X_t, \alpha_t, \mu_t)dt + g(X_T, \mu_T)\right]$$

with the dynamics:

$$dX_t = b(t, X_t, \alpha_t, \mu_t) + \sigma dW_t$$

where

- X and α are respectively the state and the control of the representative player,
- μ is the first marginal (state-only distribution)

Mean Field Solution Concepts

Here again, two concepts:

- **Nash equilibrium** $(\hat{\alpha}, \hat{\mu})$:
  - Optimality: for all α,
    $$J(\hat{\alpha}, \hat{\mu}) \leq J(\alpha, \hat{\mu})$$
  - Consistency: $\hat{\mu}_t = \mathcal{L}(X_t^{\hat{\alpha}})$

  $\rightarrow$ no incentive for unilateral deviations

  $\rightarrow$ **fixed point** problem over the mean field flow μ

Mean Field Solution Concepts

Here again, two concepts:

- **Nash equilibrium** $(\hat{\alpha}, \hat{\mu})$:
  - Optimality: for all α,
    $$J(\hat{\alpha}, \hat{\mu}) \leq J(\alpha, \hat{\mu})$$
  - Consistency: $\hat{\mu}_t = \mathcal{L}(X_t^{\hat{\alpha}})$

  $\rightarrow$ no incentive for unilateral deviations

  $\rightarrow$ **fixed point** problem over the mean field flow μ

- **Social optimum** α^*: for all α,
  $$J(\alpha^*, \mu^{\alpha^*}) \leq J(\alpha, \mu^{\alpha})$$
  where $\mu_t^{\alpha} = \mathcal{L}(X_t^{\alpha})$

  $\rightarrow$ no incentive for joint deviations

  $\rightarrow$ **optimization** problem for $\alpha \mapsto J(\alpha, \mu^{\alpha})$

Optimality Conditions

Large(st) part of the MFG literature focuses on equations of the form:

INTRODUCTION

This paper is devoted to the analysis of second order mean field games systems with a local coupling. The general form of these systems is:

$$\begin{cases} (i) & -\partial_t \phi - A_{ij}\partial_{ij}\phi + H(x, D\phi) = f(x, m(x,t)) \\ (ii) & \partial_t m - \partial_{ij}(A_{ij}m) - \operatorname{div}(m D_p H(x, D\phi)) = 0 \\ (iii) & m(0) = m_0, \ \phi(x,T) = \phi_T(x) \end{cases} \tag{1}$$

Source: Cardaliaguet, P., Graber, P.J., Porretta, A. and Tonon, D., 2015. Second order mean field games with degenerate diffusion and local coupling. Nonlinear Differential Equations and Applications NoDEA, 22(5), pp.1287-1317.

In a nutshell, the probabilistic approach to the solution of the mean-field game problem results in the solution of a FBSDE of the McKean–Vlasov type

$$\begin{cases} dX_t = b\big(t, X_t, \mathbb{P}_{X_t}, \hat{\alpha}(t, X_t, \mathbb{P}_{X_t}, Y_t)\big)dt + \sigma dW_t, \\ dY_t = -\partial_x H\big(t, X_t, \mathbb{P}_{X_t}, Y_t, \hat{\alpha}(t, X_t, \mathbb{P}_{X_t}, Y_t)\big)dt + Z_t dW_t, \end{cases} \tag{3.1}$$

with the initial condition $X_0 = x_0 \in \mathbb{R}^d$, and terminal condition $Y_T = \partial_x g(X_T, \mathbb{P}_{X_T})$.

Source: Carmona, R. and Delarue, F., 2013. Probabilistic analysis of mean-field games. SIAM Journal on Control and Optimization, 51(4), pp.2705-2734.

$\rightarrow$ Theory: derivation, analysis, . . .

“Classical” Numerical Methods for MFG: Some references

Some methods based on the deterministic approach to MFG/MFC:

- Finite difference & Newton method: [ACD10], [ACCD12], ...
- (Semi-)Lagrangian approach: [CS14, CS15], [CS18], [CCS22], ...
- Augmented Lagrangian & ADMM: [BC15], [And17a], [AL16], ...
- Primal-dual algo.: [BnAKS18], [BnAKK+19], ...
- Gradient descent based methods [LP16], [Pfe16], [LP22], ...
- Monotone operators [AFG17], [GS18], [GY20], ...
- Policy iteration [CCG21a], [CK21a], [CT22], [TS22], [LST23], ...
- Finite elements [BC15], [And17b], ...
- Gaussian processes [MYZ22], ...
- Kernel-based representation [LJL+21], ...
- Fourier approximation [N+19], ...

“Classical” Numerical Methods for MFG: Some references

Some methods based on the probabilistic approach to MFG/MFC:

- Cubature [dRT15], ...
- Markov chain approximation: [BBC18], ...
- Probabilistic approach and Picard: [CCD19], [AGL+19], ...
- Probabilistic approach and regression: [BHL+19], ...
- ...

“Classical” Numerical Methods for MFG: Shortcomings

Many of these methods are very efficient and have been analyzed in detail

However, they are usually limited to problems with:

- (relatively) small dimension
- (relatively) simple structure

$\Rightarrow$ motivations to develop deep learning methods

Deep Learning Numerical Methods for MFG

- DL for direct approach for MFG [FZ20], [CL22], ...
- DL for McKean-Vlasov FBSDEs [FZ20], [CL22], [GMW22], ...
- DL for PDE system [AACN+19], [CL21], [ROL+20], [CGL20], ...
- DL for Master equations [GLPW22], [Lau21, Section 7.2], ...

Pros & Cons:

- Scalability in terms of dimension
- Much less understood than classical methods

  $\Rightarrow$ Lots of open questions for mathematicians!

Extensions

From the modeling viewpoint, many possible extensions:

- More settings, e.g. MFG with **ergodic** cost [CLLP12], [Fel13], [BP14], [ABC17b], [AKS23], ...
- Interactions through the **action distribution** ("extended MFGs", "MFGs of controls", ...): [GPV14], [GV16], [CL18], [AK20], [LT22], [Kob22], ...
- **Common noise**: in the continuous space case see [CD18] and references therein; in the finite state case, see e.g. [BLL19], [BCCD21], ...
- **Several populations** MFGs: [HMC+06b], [Fel13], [Cir15], [ABC17a], [BHL18], ...
- **Mean field type games**: [DTT17], [BGT21] and references therein; [MP19a], [CP19], [CLT19a], ...
- **Mean field control games**: [ADF+22b], [ADF+22a]

Extensions

- **Major player**: [CZ16], [CK16], [CW17], [LL18], [CCP20], [CD21], [CDL22], ...
- **Stackelberg** MFGs [BCY15], [MB18], [EMP19], [FSJ21], [ACDL22b], [VB22], [GHZ22], [DL23],...
- **Graphon** games [PO19], [CH19], [CH21], [LS22], [GTC20], [VMV21], [CCGL22], [ACL22], [ACDL22a], [BWZ23], ...
- **Correlated** equilibria [CF22], [MRE+21], [MER+22], ...
- ...

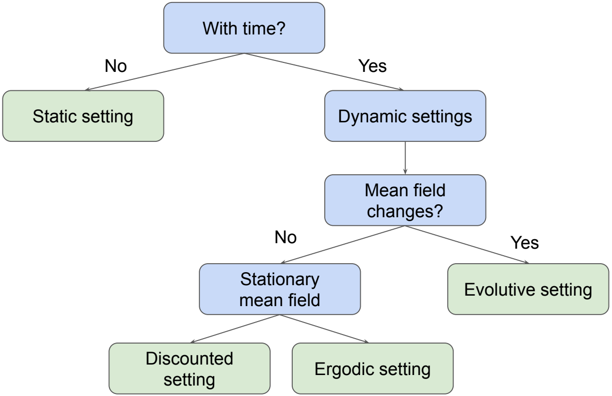
For simplicity, in most of the presentation, we will consider

- “plain” MFGs/MFCs,
- with discrete time and spaces

but many ideas can be extended in a (more or less) straightforward way.

Outline

Settings: Intuition



Settings

4 different settings:

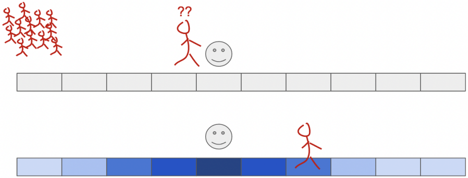
- **Static:**
  - **No states** (normal-form game): each player chooses an **action** $a \sim \pi(\cdot)$
  - Reward: depends on own action & population's action distribution
  - Examples: towel on the beach, urban settlement, ...
- **Evolutive:**
  - One-step reward: depends on own state, action & population's (state,action) distribution.
  - Fixed initial state distribution; finite or infinite time horizon.
  - Policy: **time-dependant policy** $\pi_n(\cdot|x)$
  - Examples: crowd motion, traffic routing, ...
- **Infinite horizon discounted & stationary:**
  - One-step reward: similar to Evolutive case.
  - Total reward: infinite horizon discounted sum.
  - Initial state distribution = stationary distribution induced by the population's policy.
  - Policy: **stationary policy** $\pi(\cdot|x)$
  - Examples: player joining a crowd already in a steady state
- **Ergodic:**
  - Similar to infinite horizon discounted & stationary.
  - But: Total reward = long time average.
- Other settings: asymptotic, γ-discounted, ...

Outline

Static game

Example: Population distribution (towel on the beach, . . .)

- action: choice of position
- reward: depends on my position and on the density of people



Source: unsplash

Static setting

- Finite action set A *(e.g., beach = possible towels' positions)*
- Player's behavior $\pi \in \Delta_A = \mathcal{P}(A)$
- Population's behavior $\xi \in \Delta_A$
- Player's reward: for player policy $\pi \in \Delta_A$ and population behavior $\xi \in \Delta_A$,

$$J(\pi; \xi) = \mathbb{E}_{a \sim \pi}\left[r(a, \xi)\right]$$

(e.g., crowd aversion, ice cream stall attraction, ...)

Static game: Solution concepts

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best response: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi J(\pi; \hat{\xi})$
  2. Consistency: $\hat{\xi} = \hat{\pi}$
- **Static MFC Social optimum:** $\pi^* \in \Delta_A$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_\pi J(\pi; \pi)$

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best response: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi J(\pi; \hat{\xi})$
  2. Consistency: $\hat{\xi} = \hat{\pi}$
- **Static MFC Social optimum:** $\pi^* \in \Delta_A$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_\pi J(\pi; \pi)$
- Note: at social optimum, the population distribution is $\xi^* = \pi^*$
- But in general $\pi^* \neq \hat{\pi}$ so $\hat{\xi} \neq \xi^*$

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}, \qquad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}, \qquad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Then:

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best resp.: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi J(\pi; \hat{\xi}) = \pi(1)(c - \hat{\xi}(1)) + \pi(2)(-\hat{\xi}(2))$
  2. Consistency: $\hat{\xi} = \hat{\pi}$

  Is $\xi = (\xi(1), \xi(2)) = (1, 0)$ be a Nash equilibrium? Then $c - \xi(1) = c - 1 < 0 = -\xi(2)$ so $\pi = (\pi(1), \pi(2)) = (0, 1)$ would be *the* BR. Contradiction!

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}, \qquad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Then:

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best resp.: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi\, J(\pi; \hat{\xi}) = \pi(1)(c - \hat{\xi}(1)) + \pi(2)(-\hat{\xi}(2))$
  2. Consistency: $\hat{\xi} = \hat{\pi}$

  Is $\xi = (\xi(1), \xi(2)) = (1, 0)$ be a Nash equilibrium? Then
  $c - \xi(1) = c - 1 < 0 = -\xi(2)$ so $\pi = (\pi(1), \pi(2)) = (0, 1)$ would be *the* BR.
  Contradiction!
  So at equilibrium both actions are optimal: $c - \hat{\xi}(1) = -\hat{\xi}(2)$
  Since $\hat{\xi}(1) + \hat{\xi}(2) = 1$, the equilibrium distrib. is: $\hat{\xi} = (\hat{\xi}(1), \hat{\xi}(2)) = (\frac{1+c}{2}, \frac{1-c}{2})$

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}, \qquad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Then:

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best resp.: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi\, J(\pi; \hat{\xi}) = \pi(1)(c - \hat{\xi}(1)) + \pi(2)(-\hat{\xi}(2))$
  2. Consistency: $\hat{\xi} = \hat{\pi}$

  Is $\xi = (\xi(1), \xi(2)) = (1, 0)$ be a Nash equilibrium? Then
  $c - \xi(1) = c - 1 < 0 = -\xi(2)$ so $\pi = (\pi(1), \pi(2)) = (0, 1)$ would be *the* BR.
  Contradiction!
  So at equilibrium both actions are optimal: $c - \hat{\xi}(1) = -\hat{\xi}(2)$
  Since $\hat{\xi}(1) + \hat{\xi}(2) = 1$, the equilibrium distrib. is: $\hat{\xi} = (\hat{\xi}(1), \hat{\xi}(2)) = (\frac{1+c}{2}, \frac{1-c}{2})$
- **Static MFC Social optimum:** $\pi^* \in \Delta_A$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_\pi\, J(\pi; \pi) = \pi(1)(c - \pi(1)) + \pi(2)(-\pi(2))$

  and since $\pi(2) = 1 - \pi(1)$, $J(\pi; \pi) = -1 + (2 + c)\pi(1) - 2\pi(1)^2$ First order optimality condition gives:
  $0 = \frac{d}{d\pi(1)}[-1 + (2 + c)\pi^*(1) - 2\pi^*(1)^2] = (2 + c) - 4\pi^*(1)$
  so the socially optimum distribution is: $\xi^* = (\xi^*(1), \xi^*(2)) = (\frac{2+c}{4}, \frac{2-c}{4})$

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}, \qquad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Then:

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best resp.: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi J(\pi; \hat{\xi}) = \pi(1)(c - \hat{\xi}(1)) + \pi(2)(-\hat{\xi}(2))$
  2. Consistency: $\hat{\xi} = \hat{\pi}$

  Is $\xi = (\xi(1), \xi(2)) = (1, 0)$ be a Nash equilibrium? Then
  $c - \xi(1) = c - 1 < 0 = -\xi(2)$ so $\pi = (\pi(1), \pi(2)) = (0, 1)$ would be *the* BR.
  Contradiction!
  So at equilibrium both actions are optimal: $c - \hat{\xi}(1) = -\hat{\xi}(2)$
  Since $\hat{\xi}(1) + \hat{\xi}(2) = 1$, the equilibrium distrib. is: $\hat{\xi} = (\hat{\xi}(1), \hat{\xi}(2)) = (\frac{1+c}{2}, \frac{1-c}{2})$
- **Static MFC Social optimum:** $\pi^* \in \Delta_A$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_\pi J(\pi; \pi) = \pi(1)(c - \pi(1)) + \pi(2)(-\pi(2))$

  and since $\pi(2) = 1 - \pi(1)$, $J(\pi; \pi) = -1 + (2 + c)\pi(1) - 2\pi(1)^2$ First order optimality condition gives:
  $0 = \frac{d}{d\pi(1)}[-1 + (2 + c)\pi^*(1) - 2\pi^*(1)^2] = (2 + c) - 4\pi^*(1)$
  so the socially optimum distribution is: $\xi^* = (\xi^*(1), \xi^*(2)) = (\frac{2+c}{4}, \frac{2-c}{4})$
- So, in this example with $c \in (0, 1)$, $\hat{\xi} \neq \xi^*$

Nash Equilibrium vs Social Optimum: Example

Consider: $A = \{1, 2\}$, $\quad r(a, \xi) = c\mathbf{1}_{a=1} - \xi(a)$ where

- the constant $c \in (0, 1)$ gives some attraction to action $a = 1$
- $-\xi(a)$ is a repulsion term (crowd aversion)

Then:

- **Static MFG Nash equilibrium:** $(\hat{\pi}, \hat{\xi}) \in \Delta_A \times \Delta_A$ s.t.
  1. Best resp.: $\hat{\pi} \in \mathrm{BR}(\hat{\xi}) := \mathrm{argmax}_\pi \, J(\pi; \hat{\xi}) = \pi(1)(c - \hat{\xi}(1)) + \pi(2)(-\hat{\xi}(2))$
  2. Consistency: $\hat{\xi} = \hat{\pi}$

  Is $\xi = (\xi(1), \xi(2)) = (1, 0)$ be a Nash equilibrium? Then
  $c - \xi(1) = c - 1 < 0 = -\xi(2)$ so $\pi = (\pi(1), \pi(2)) = (0, 1)$ would be *the* BR.
  Contradiction!
  So at equilibrium both actions are optimal: $c - \hat{\xi}(1) = -\hat{\xi}(2)$
  Since $\hat{\xi}(1) + \hat{\xi}(2) = 1$, the equilibrium distrib. is: $\hat{\xi} = (\hat{\xi}(1), \hat{\xi}(2)) = (\frac{1+c}{2}, \frac{1-c}{2})$
- **Static MFC Social optimum:** $\pi^* \in \Delta_A$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_\pi \, J(\pi; \pi) = \pi(1)(c - \pi(1)) + \pi(2)(-\pi(2))$

  and since $\pi(2) = 1 - \pi(1)$, $J(\pi; \pi) = -1 + (2 + c)\pi(1) - 2\pi(1)^2$ First order optimality condition gives:
  $0 = \frac{d}{d\pi(1)}[-1 + (2 + c)\pi^*(1) - 2\pi^*(1)^2] = (2 + c) - 4\pi^*(1)$
  so the socially optimum distribution is: $\xi^* = (\xi^*(1), \xi^*(2)) = (\frac{2+c}{4}, \frac{2-c}{4})$
- So, in this example with $c \in (0, 1)$, $\hat{\xi} \neq \xi^*$
- Nash equilibrium is more concentrated on action 1 than MFC (“selfishness”)

Nash Equilibrium vs Social Optimum: Potential case

- In *some cases*, the two notions coincide.
- Example: Potential MFG with reward: $r(a, \xi) = \nabla F(\xi)(a)$ for some $F : \Delta_A \to \mathbb{R}$
- The average cost is: $J(\pi, \xi) = \mathbb{E}_{a\sim\pi}[r(a, \xi)] = \sum_a \pi(a)\nabla F(\xi)(a) = \pi \cdot \nabla F(\xi)$

Nash Equilibrium vs Social Optimum: Potential case

- In *some cases*, the two notions coincide.
- Example: Potential MFG with reward: $r(a,\xi) = \nabla F(\xi)(a)$ for some $F : \Delta_A \to \mathbb{R}$
- The average cost is: $J(\pi,\xi) = \mathbb{E}_{a\sim\pi}[r(a,\xi)] = \sum_a \pi(a)\nabla F(\xi)(a) = \pi \cdot \nabla F(\xi)$
- Assuming the potential F concave, we have the equivalence:

$$
\begin{aligned}
\hat{\pi} \text{ is a NE} &\Leftrightarrow J(\pi,\hat{\pi}) - J(\hat{\pi},\hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow (\pi - \hat{\pi}) \cdot \nabla F(\hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow \nabla F(\hat{\pi}) = 0 \\
&\Leftrightarrow \hat{\pi} \text{ is a maximizer of } F \\
&\Leftrightarrow \hat{\pi} \text{ is a social optimum}
\end{aligned}
$$

Nash Equilibrium vs Social Optimum: Potential case

- In *some cases*, the two notions coincide.
- Example: Potential MFG with reward: $r(a, \xi) = \nabla F(\xi)(a)$ for some $F : \Delta_A \to \mathbb{R}$
- The average cost is: $J(\pi, \xi) = \mathbb{E}_{a \sim \pi}[r(a, \xi)] = \sum_a \pi(a) \nabla F(\xi)(a) = \pi \cdot \nabla F(\xi)$
- Assuming the potential F concave, we have the equivalence:

$$\begin{aligned}
\hat{\pi} \text{ is a NE} &\Leftrightarrow J(\pi, \hat{\pi}) - J(\hat{\pi}, \hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow (\pi - \hat{\pi}) \cdot \nabla F(\hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow \nabla F(\hat{\pi}) = 0 \\
&\Leftrightarrow \hat{\pi} \text{ is a maximizer of } F \\
&\Leftrightarrow \hat{\pi} \text{ is a social optimum}
\end{aligned}$$

- Example: (negative of) entropy: $F(\xi) = -\sum_a \xi(a) \log(\xi(a))$: encourages agent to spread throughout the action space A

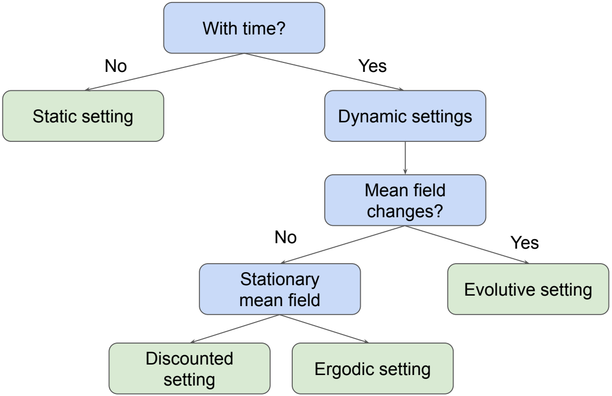
Nash Equilibrium vs Social Optimum: Potential case

- In *some cases*, the two notions coincide.
- Example: Potential MFG with reward: $r(a, \xi) = \nabla F(\xi)(a)$ for some $F : \Delta_A \to \mathbb{R}$
- The average cost is: $J(\pi, \xi) = \mathbb{E}_{a\sim\pi}[r(a, \xi)] = \sum_a \pi(a)\nabla F(\xi)(a) = \pi \cdot \nabla F(\xi)$
- Assuming the potential F concave, we have the equivalence:

$$\begin{aligned}
\hat{\pi} \text{ is a NE} &\Leftrightarrow J(\pi, \hat{\pi}) - J(\hat{\pi}, \hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow (\pi - \hat{\pi}) \cdot \nabla F(\hat{\pi}) \leq 0, \quad \forall \pi \\
&\Leftrightarrow \nabla F(\hat{\pi}) = 0 \\
&\Leftrightarrow \hat{\pi} \text{ is a maximizer of } F \\
&\Leftrightarrow \hat{\pi} \text{ is a social optimum}
\end{aligned}$$

- Example: (negative of) entropy: $F(\xi) = -\sum_a \xi(a) \log(\xi(a))$: encourages agent to spread throughout the action space A
- Note: the link between potential MFGs and MFC can be exploited to design numerical methods

Settings: Intuition – Reminder



Outline

Notation for Dynamic Settings

- State $x \in S$, action $a \in A$ (S, A finite for most of this presentation)
- Mean field state $\mu \in \Delta_S = \mathcal{P}(S)$ (extensions: state-action distrib.)
- Discrete time $n \in \mathbb{N}$
- Player's transition probability: $p(\cdot|x, a, \mu)$
- Player's reward: $r(x, a, \mu)$
- One-step policy: $\pi \in \Pi := (\Delta_A)^S$, functions $S \to \Delta_A$
- One-step mean field transition matrix: $P_{\mu,\pi}(x, y) = \sum_{a \in A} \pi(a|x) p(y|x, a, \mu)$

Notation for Dynamic Settings

- State $x \in S$, action $a \in A$ (S, A finite for most of this presentation)
- Mean field state $\mu \in \Delta_S = \mathcal{P}(S)$ (extensions: state-action distrib.)
- Discrete time $n \in \mathbb{N}$
- Player's transition probability: $p(\cdot|x, a, \mu)$
- Player's reward: $r(x, a, \mu)$
- One-step policy: $\pi \in \Pi := (\Delta_A)^S$, functions $S \to \Delta_A$
- One-step mean field transition matrix: $P_{\mu,\pi}(x, y) = \sum_{a \in A} \pi(a|x) p(y|x, a, \mu)$
- What happens in one time step?
  - "Each" player selects an action (we focus on one "representative" player)
  - "Each" player gets a reward
  - "Each" player state is updated
  - Mean field is updated
- Mathematically: with policy π_n and mean field μ_n

$$a_n \sim \pi_n(\cdot|x_n)$$

$$r(x_n, a_n, \mu_n)$$

$$x_{n+1} \sim p(\cdot|x_n, a_n, \mu_n)$$

$$\mu_{n+1} = P^\top_{\mu_n,\pi_n} \mu_n = \sum_{y \in S} \mu_n(y) \sum_{a \in A} \pi_n(a|y) p(\cdot|y, a, \mu_n)$$

Stationary setting

Stationary game

Example: joining a population in a stationary regime (flocking, economics, ...)

- the population is at equilibrium $\rightarrow$ MF distribution is stationary
- a player wants to join $\rightarrow$ optimal control problem
- but the distribution is the result of the agents' decisions $\rightarrow$ fixed point problem

Source: unsplash

Stationary setting

- Stationary setting: $N_T = \infty$
- No fixed initial m_0 but a stationary distribution
- Notation: $\mathrm{MF}(\pi) :=$ stationary distribution when using policy π:

$$\mu = P^{\top}_{\mu,\pi}\mu =: \mathcal{P}^{\pi}(\mu)$$

- Player's reward: for player's policy $\pi \in \Delta_A$ and mean field $\mu \in \Delta_S$,

$$J(\pi; \mu) = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r(x_n, a_n, \mu)\right]$$

where $\gamma \in (0, 1)$ is a discount parameter, and

$$a_n \sim \pi(\cdot|x_n), \qquad x_0 \sim \mu, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \mu), n \geq 0$$

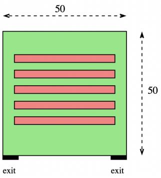
Stationary setting: Solution concepts

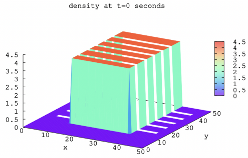
- **Stationary MFG Nash equilibrium:** $(\hat{\pi}, \hat{\mu}) \in \Pi \times \Delta_{S \times A}$ s.t.
  1. Best response: $\hat{\pi} \in \mathrm{BR}(\hat{\mu}) := \mathrm{argmax}_{\pi} J(\pi; \hat{\mu})$
  2. Mean field state: $\hat{\mu} = \mathrm{MF}(\hat{\pi})$
- Fixed point: $\hat{\mu} \in \mathrm{MF}(\mathrm{BR}(\hat{\mu}))$

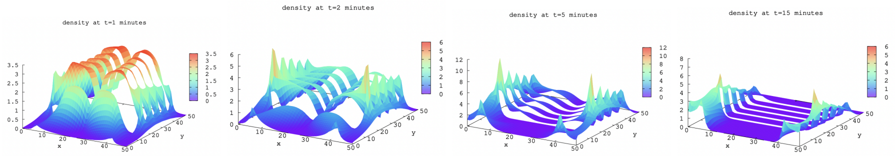
- **Stationary MFC Social optimum:** $\pi^* \in \Pi$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_{\pi^*} J(\pi^*; \mu^{\pi^*})$ where $\mu^{\pi^*} = \mathrm{MF}(\pi^*)$

Evolutive setting

Example: Crowd exiting a room [AL15]







Evolutive setting

- Horizon: $N_T \in \mathbb{N}$ (extensions: p, r depending on n; infinite horizon)
- Fixed initial state distribution: $m_0 \in \Delta_S$
- The MF evolves in time: $\boldsymbol{\mu} = (\boldsymbol{\mu}_n)_{n=0,\dots,N_T} \in \Delta_S^{N_T}$
- Notation $\mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}) :=$ generated by policy $\boldsymbol{\pi}$ starting from m_0:

$$\begin{cases} \boldsymbol{\mu}_0 = m_0, \\ \boldsymbol{\mu}_{n+1} = P^\top_{\boldsymbol{\mu}_n, \boldsymbol{\pi}_n} \boldsymbol{\mu}_n, \quad n \geq 0 \end{cases}$$

- Player's reward: for player's policy $\boldsymbol{\pi} \in \Pi^{N_T}$ and mean field $\boldsymbol{\mu} \in \Delta_S^{N_T}$,

$$J(\boldsymbol{\pi}; \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{n=0}^{N_T} r(x_n, a_n, \boldsymbol{\mu}_n)\right]$$

where

$$a_n \sim \boldsymbol{\pi}_n(\cdot|x_n), \qquad x_0 \sim m_0, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \boldsymbol{\mu}_n), n \geq 0$$

Evolutive setting: Solution concepts

- **Evolutive MFG Nash equilibrium:** $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}) \in \Pi^{N_T} \times \Delta_S^{N_T}$ s.t.
  1. Best response: $\hat{\boldsymbol{\pi}} \in \mathrm{BR}(\hat{\boldsymbol{\mu}}) := \mathrm{argmax}_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; \hat{\boldsymbol{\mu}})$
  2. Mean field flow: $\hat{\boldsymbol{\mu}} = \mathrm{MF}_{m_0, N_T}(\hat{\boldsymbol{\pi}})$
- Fixed point: $\hat{\boldsymbol{\mu}} \in \mathrm{MF}_{m_0, N_T}(\mathrm{BR}(\hat{\boldsymbol{\mu}}))$

- **Evolutive MFC Social optimum:** $\boldsymbol{\pi}^* \in \Pi^{N_T}$ s.t.
  - Optimality: $\boldsymbol{\pi}^* \in \mathrm{argmax}_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; \boldsymbol{\mu}^{\boldsymbol{\pi}})$ where $\boldsymbol{\mu}^{\boldsymbol{\pi}} = \mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi})$

Outline

Value function: stationary case

- Value function of a (stationary) policy π given a (stationary) mean field μ: $V^{\mu,\pi}(x) := \mathbb{E}_\pi\left[\sum_{n\geq 0}\gamma^n r(x_n, a_n, \mu)\right]$ satisfies:

$$V^{\mu,\pi}(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}\Big[\underbrace{r(x,a,\mu) + \gamma\mathbb{E}_{x'\sim p(\cdot|x,a,\mu)}[V^{\mu,\pi}(x')]}_{Q^{\mu,\pi}(x,a)}\Big]$$

$$V^{\mu,\pi} = \mathcal{T}^{\mu,\pi}V^{\mu,\pi}$$
$$Q^{\mu,\pi} = \mathcal{B}^{\mu,\pi}Q^{\mu,\pi}$$

- Optimal value function given a mean field μ: $V^{\mu,*}(x) = \max_\pi V^{\mu,\pi}(x)$:

$$V^{\mu,*}(x) = \max_\pi \mathbb{E}_{a\sim\pi(\cdot|x))}\Big[\underbrace{r(x,a,\mu) + \gamma\mathbb{E}_{x'\sim p(\cdot|x,a,\mu)}[V^{\mu,*}(x')]}_{Q^{\mu,*}(x,a)}\Big]$$

$$V^{\mu,*} = \mathcal{T}^{\mu,*}V^{\mu,*}$$
$$Q^{\mu,*} = \mathcal{B}^{\mu,*}Q^{\mu,*}$$

- Optimal policy given a mean field μ: single player's problem:

$$\mathrm{supp}(\pi^*(\cdot|x)) \subseteq \operatorname*{argmax}_{a\in A} Q^{\mu,*}(x,a)$$

Value function: stationary case

- Value function of a (stationary) policy π given a (stationary) mean field μ:
  $V^{\mu,\pi}(x) := \mathbb{E}_\pi\left[\sum_{n\geq 0}\gamma^n r(x_n, a_n, \mu)\right]$ satisfies:

$$V^{\mu,\pi}(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}\Big[\underbrace{r(x,a,\mu) + \gamma\mathbb{E}_{x'\sim p(\cdot|x,a,\mu)}[V^{\mu,\pi}(x')]}_{Q^{\mu,\pi}(x,a)}\Big]$$

$$V^{\mu,\pi} = \mathcal{T}^{\mu,\pi}V^{\mu,\pi}$$

$$Q^{\mu,\pi} = \mathcal{B}^{\mu,\pi}Q^{\mu,\pi}$$

- Optimal value function given a mean field μ: $V^{\mu,*}(x) = \max_\pi V^{\mu,\pi}(x)$:

$$V^{\mu,*}(x) = \max_\pi \mathbb{E}_{a\sim\pi(\cdot|x))}\Big[\underbrace{r(x,a,\mu) + \gamma\mathbb{E}_{x'\sim p(\cdot|x,a,\mu)}[V^{\mu,*}(x')]}_{Q^{\mu,*}(x,a)}\Big]$$

$$V^{\mu,*} = \mathcal{T}^{\mu,*}V^{\mu,*}$$

$$Q^{\mu,*} = \mathcal{B}^{\mu,*}Q^{\mu,*}$$

- Optimal policy given a mean field μ: single player's problem:

$$\mathrm{supp}(\pi^*(\cdot|x)) \subseteq \operatorname*{argmax}_{a\in A} Q^{\mu,*}(x,a)$$

- Bellman equations are **fixed point equations**

Value function: finite horizon evolutive case

Finite horizon evolutive case ($N_T < +\infty$):

- Value function of a policy $\boldsymbol{\pi}$ given a mean field $\boldsymbol{\mu}$: $V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) := \mathbb{E}_{\boldsymbol{\pi}}[\sum_{n'=n}^{N_T} r(x_{n'}, a_{n'}, \boldsymbol{\mu}_{n'}) | x_n = x]$ satisfies:

$$\begin{cases} V_{N_T+1}^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) = 0 \\ V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) = \mathbb{E}_{a \sim \boldsymbol{\pi}_n(\cdot|x)} \Big[\underbrace{r(x, a, \boldsymbol{\mu}_n) + \mathbb{E}_{x' \sim p(\cdot|x,a,\boldsymbol{\mu}_n)}[V_{n+1}^{\boldsymbol{\mu},\boldsymbol{\pi}}(x')]}_{Q_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x,a)} \Big], \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad n = N_T - 1, \dots, 0 \end{cases}$$

- Optimal value function given a mean field $\boldsymbol{\mu}$:

$$V_n^{\boldsymbol{\mu},*}(x) = \max_{\boldsymbol{\pi}} V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x)$$

$$Q_n^{\boldsymbol{\mu},*}(x,a) = \max_{\boldsymbol{\pi}} Q_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x,a)$$

- Optimal policy given a mean field $\boldsymbol{\mu}$: single player's problem:

$$\mathrm{supp}(\boldsymbol{\pi}_n^*(\cdot|x)) \subseteq \operatorname*{argmax}_{a \in A} Q_n^{\boldsymbol{\mu},*}(x,a)$$

Value function: finite horizon evolutive case

Finite horizon evolutive case ($N_T < +\infty$):

- Value function of a policy $\boldsymbol{\pi}$ given a mean field $\boldsymbol{\mu}$:
  $V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) := \mathbb{E}_{\boldsymbol{\pi}}[\sum_{n'=n}^{N_T} r(x_{n'}, a_{n'}, \boldsymbol{\mu}_{n'})|x_n = x]$ satisfies:

$$\begin{cases} V_{N_T+1}^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) = 0 \\ V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x) = \mathbb{E}_{a\sim\boldsymbol{\pi}_n(\cdot|x)}\Big[\underbrace{r(x,a,\boldsymbol{\mu}_n) + \mathbb{E}_{x'\sim p(\cdot|x,a,\boldsymbol{\mu}_n)}[V_{n+1}^{\boldsymbol{\mu},\boldsymbol{\pi}}(x')]}_{Q_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x,a)}\Big], \\ \hfill n = N_T - 1, \dots, 0 \end{cases}$$

- Optimal value function given a mean field $\boldsymbol{\mu}$:

$$V_n^{\boldsymbol{\mu},*}(x) = \max_{\boldsymbol{\pi}} V_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x)$$

$$Q_n^{\boldsymbol{\mu},*}(x,a) = \max_{\boldsymbol{\pi}} Q_n^{\boldsymbol{\mu},\boldsymbol{\pi}}(x,a)$$

- Optimal policy given a mean field $\boldsymbol{\mu}$: single player's problem:

$$\mathrm{supp}(\boldsymbol{\pi}_n^*(\cdot|x)) \subseteq \operatorname*{argmax}_{a\in A} Q_n^{\boldsymbol{\mu},*}(x,a)$$

- Bellman equations are **backward induction equations**

Outline

Outline

We are going to focus mostly on MFG Nash equilibria computation

MFG Equilibrium Computation: General Principles

We are going to focus mostly on MFG Nash equilibria computation

Two main objects: policy π and population distribution μ

Most basic idea: alternate

1. Update of the policy
2. Update of the population's distribution

MFG Equilibrium Computation: General Principles

We are going to focus mostly on MFG Nash equilibria computation

Two main objects: policy π and population distribution μ

Most basic idea: alternate

1. Update of the policy
2. Update of the population's distribution

Many other possibilities using optimality conditions, e.g.

- traditional methods such as Newton's method for the PDE system [ACCD12]
- deep learning methods for PDE/FBSDE system, see [HL22]

But cannot be directly adapted to the model-free RL setting.

Updating the policy

For standard MDPs:

- Bellman operators
  - Optimal Bellman operator:

$$\mathcal{B}^* : (Q(x,a))_{x,a} \mapsto \mathcal{B}^* Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a)} [\max_{a'} Q(x', a')] \right)_{x,a}$$

  - Bellman operator associated to a policy π:

$$\mathcal{B}^\pi : (Q(x,a))_{x,a} \mapsto \mathcal{B}^\pi Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a), a' \sim \pi} [Q(x, a')] \right)_{x,a}$$

Updating the policy

For standard MDPs:

- Bellman operators
  - Optimal Bellman operator:

$$\mathcal{B}^* : (Q(x,a))_{x,a} \mapsto \mathcal{B}^* Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a)}[\max_{a'} Q(x',a')] \right)_{x,a}$$

  - Bellman operator associated to a policy π:

$$\mathcal{B}^\pi : (Q(x,a))_{x,a} \mapsto \mathcal{B}^\pi Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a), a' \sim \pi}[Q(x,a')] \right)_{x,a}$$

- Iterative learning methods:
  - **Value iteration:**

$$Q^{k+1} = \mathcal{B}^* Q^k$$

  - **Policy iteration:**

$$\begin{cases} Q^{k+1} = Q^{\pi^k} & \text{(policy evaluation)} \\ \pi^{k+1} \in \operatorname{argmax} Q^{k+1} & \text{(policy improvement)} \end{cases}$$

where the policy evaluation can be done by applying $\mathcal{B}^{\pi^k}$ many times

Updating the policy

For standard MDPs:

- Bellman operators
  - Optimal Bellman operator:

$$\mathcal{B}^* : (Q(x,a))_{x,a} \mapsto \mathcal{B}^* Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a)} [\max_{a'} Q(x',a')] \right)_{x,a}$$

  - Bellman operator associated to a policy π:

$$\mathcal{B}^\pi : (Q(x,a))_{x,a} \mapsto \mathcal{B}^\pi Q = \left(r(x,a) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a), a' \sim \pi} [Q(x,a')] \right)_{x,a}$$

- Iterative learning methods:
  - **Value iteration:**

$$Q^{k+1} = \mathcal{B}^* Q^k$$

  - **Policy iteration:**

$$\begin{cases} Q^{k+1} = Q^{\pi^k} & \text{(policy evaluation)} \\ \pi^{k+1} \in \operatorname{argmax} Q^{k+1} & \text{(policy improvement)} \end{cases}$$

where the policy evaluation can be done by applying $\mathcal{B}^{\pi^k}$ many times

$\to$ For MFG: intertwine applications of $\mathcal{B}^{\mu,*}$ or $\mathcal{B}^{\mu,\pi}$ with MF updates

Iterative methods for MFG: Stationary case

Goal: find MFG Nash equilibirum $(\hat{\pi}, \hat{\mu}) \in \Pi \times \Delta_S$

- Iterations based on **Best response computation**:
  1. Compute best response: $\pi^{k+1} = \mathrm{BR}(\mu^k)$:
     - 1.1 Compute the optimal value function: $Q^{\mu^k,*} = \mathcal{B}^{\mu^k,*} Q^{\mu^k,*}$
     - 1.2 Let: $\pi^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q^{\mu^k,*}(x,a)$
  2. Compute stationary MF: $\mu^{k+1} = \mathrm{MF}(\pi^{k+1})$: $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}} \mu^{k+1}$
- Iterations based on **Policy evaluation** ("policy iteration"):
  1. Update policy:
     - 1.1 Evaluate policy: $Q^{\mu^k,\pi^k} = \mathcal{B}^{\mu^k,\pi^k} Q^{\mu^k,\pi^k}$
     - 1.2 Let: $\pi^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q^{\mu^k,\pi^k}(x,a)$
  2. Compute stationary MF: $\mu^{k+1} = \mathrm{MF}(\pi^{k+1})$: $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}}(\mu^{k+1})$

Iterative methods for MFG: Stationary case

Goal: find MFG Nash equilibirum $(\hat{\pi}, \hat{\mu}) \in \Pi \times \Delta_S$

- Iterations based on **Best response computation**:
  1. Compute best response: $\pi^{k+1} = \mathrm{BR}(\mu^k)$:
     - 1.1 Compute the optimal value function: $Q^{\mu^k,*} = \mathcal{B}^{\mu^k,*} Q^{\mu^k,*}$
     - 1.2 Let: $\pi^{k+1}(\cdot|x) \in \mathrm{argmax}_a \, Q^{\mu^k,*}(x,a)$
  2. Compute stationary MF: $\mu^{k+1} = \mathrm{MF}(\pi^{k+1})$: $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}} \mu^{k+1}$
- Iterations based on **Policy evaluation** ("policy iteration"):
  1. Update policy:
     - 1.1 Evaluate policy: $Q^{\mu^k,\pi^k} = \mathcal{B}^{\mu^k,\pi^k} Q^{\mu^k,\pi^k}$
     - 1.2 Let: $\pi^{k+1}(\cdot|x) \in \mathrm{argmax}_a \, Q^{\mu^k,\pi^k}(x,a)$
  2. Compute stationary MF: $\mu^{k+1} = \mathrm{MF}(\pi^{k+1})$: $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}}(\mu^{k+1})$

Sometimes: one application of fixed point operator instead of true fixed point:

- $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}}(\mu^k)$ instead of μ^{k+1} s.t. $\mu^{k+1} = \mathcal{P}^{\pi^{k+1}}(\mu^{k+1})$
- Learning step $\approx$ time step in the game

Iterative methods for MFG: Evolutive case

Goal: find MFG Nash equilibrium $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}) \in \Pi^{N_T} \times \Delta_S^{N_T}$

- Iterations based on **Best response computation**:
  1. Compute best response: $\boldsymbol{\pi}^{k+1} = \mathrm{BR}(\boldsymbol{\mu}^k)$:
     - 1.1 Compute the optimal value function: $Q^{\boldsymbol{\mu}^k,*}$
     - 1.2 Let: $\boldsymbol{\pi}_n^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q_n^{\boldsymbol{\mu}^k,*}(x,a)$
  2. Compute MF flow: $\boldsymbol{\mu}^{k+1} = \mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}^{k+1})$
- Iterations based on **Policy evaluation** ("policy iteration"):
  1. Update policy:
     - 1.1 Evaluate policy: $Q^{\boldsymbol{\mu}^k,\boldsymbol{\pi}^k}$
     - 1.2 Let: $\boldsymbol{\pi}_n^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q_n^{\boldsymbol{\mu}^k,\boldsymbol{\pi}^k}(x,a)$
  2. Compute MF flow: $\boldsymbol{\mu}^{k+1} = \mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}^{k+1})$

Iterative methods for MFG: Evolutive case

Goal: find MFG Nash equilibrium $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}) \in \Pi^{N_T} \times \Delta_S^{N_T}$

- Iterations based on **Best response computation**:
  1. Compute best response: $\boldsymbol{\pi}^{k+1} = \mathrm{BR}(\boldsymbol{\mu}^k)$:
     1.1 Compute the optimal value function: $Q^{\boldsymbol{\mu}^k,*}$
     1.2 Let: $\boldsymbol{\pi}_n^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q_n^{\boldsymbol{\mu}^k,*}(x,a)$
  2. Compute MF flow: $\boldsymbol{\mu}^{k+1} = \mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}^{k+1})$
- Iterations based on **Policy evaluation** (“policy iteration”):
  1. Update policy:
     1.1 Evaluate policy: $Q^{\boldsymbol{\mu}^k,\boldsymbol{\pi}^k}$
     1.2 Let: $\boldsymbol{\pi}_n^{k+1}(\cdot|x) \in \mathrm{argmax}_a\, Q_n^{\boldsymbol{\mu}^k,\boldsymbol{\pi}^k}(x,a)$
  2. Compute MF flow: $\boldsymbol{\mu}^{k+1} = \mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}^{k+1})$

Backward equations instead of fixed point equations as in stationary case

Challenges

Potential issues (for both stationary and evolutive settings):

- Non-uniqueness of the equilibrium MF $\hat{\mu}$ or $\hat{\boldsymbol{\mu}}$

Challenges

Potential issues (for both stationary and evolutive settings):

- Non-uniqueness of the equilibrium MF $\hat{\mu}$ or $\hat{\boldsymbol{\mu}}$
- Non-uniqueness of the Best Response $\pi \in \mathrm{BR}(\hat{\boldsymbol{\mu}})$
  (even though there might be a unique equilibrium policy $\hat{\pi}$!)

Challenges

Potential issues (for both stationary and evolutive settings):

- Non-uniqueness of the equilibrium MF $\hat{\mu}$ or $\hat{\boldsymbol{\mu}}$
- Non-uniqueness of the Best Response $\pi \in \mathrm{BR}(\hat{\boldsymbol{\mu}})$
  (even though there might be a unique equilibrium policy $\hat{\pi}$!)
- Lack of convergence (typically if $\mathrm{MF} \circ \mathrm{BR}$ is not a strict contraction)

  $\Rightarrow$ Oscillations / instabilities

Challenges

Potential issues (for both stationary and evolutive settings):

- Non-uniqueness of the equilibrium MF $\hat{\mu}$ or $\hat{\boldsymbol{\mu}}$
- Non-uniqueness of the Best Response $\pi \in \mathrm{BR}(\hat{\boldsymbol{\mu}})$
  (even though there might be a unique equilibrium policy $\hat{\pi}$!)
- Lack of convergence (typically if $\mathrm{MF} \circ \mathrm{BR}$ is not a strict contraction)

  $\Rightarrow$ Oscillations / instabilities

Several variations / improvements have been studied

Outline

Damping / Averaging

Damping / smoothing:

- for policies: instead of:
  $$\pi^{k+1} = \mathrm{BR}(\mu^k)$$
  use:
  $$\bar{\pi}^{k+1} = \sum_{i=1}^{k} \alpha_i \mathrm{BR}(\mu^i)$$
  for some coefficients $(\alpha_i)_i$, and then:
  $$\mu^{k+1} = \mathrm{MF}(\bar{\pi}^{k+1})$$
- and/or average mean fields, value functions, ...

Damping / Averaging

Damping / smoothing:

- for policies: instead of:
$$\pi^{k+1} = \mathrm{BR}(\mu^k)$$
use:
$$\bar\pi^{k+1} = \sum_{i=1}^{k} \alpha_i \mathrm{BR}(\mu^i)$$
for some coefficients $(\alpha_i)_i$, and then:
$$\mu^{k+1} = \mathrm{MF}(\bar\pi^{k+1})$$
- and/or average mean fields, value functions, …
- tends to avoid oscillations
- helps to learn a mixed policy even if every BR is pure
- slower convergence if small α's

Damping / Averaging

Damping / smoothing:

- for policies: instead of:

$$\pi^{k+1} = \mathrm{BR}(\mu^k)$$

  use:

$$\bar{\pi}^{k+1} = \sum_{i=1}^{k} \alpha_i \mathrm{BR}(\mu^i)$$

  for some coefficients $(\alpha_i)_i$, and then:

$$\mu^{k+1} = \mathrm{MF}(\bar{\pi}^{k+1})$$

- and/or average mean fields, value functions, ...
- tends to avoid oscillations
- helps to learn a mixed policy even if every BR is pure
- slower convergence if small α's

$\rightarrow$ Encompasses many possible variants such as:

- Fixed point iteration / value iteration (no damping): e.g. [HMC06a, GHXZ19, AKS20b] ...
- Fictitious Play: e.g. [CH17, Had17, MJMdC18, PPL+20, MH21, DV21] ...
- Policy Iteration: e.g. [CCG21b, CT21, LST21] ...
- Online Mirror Descent (OMD): e.g. [Had17, Had18, PPE+21a] ...

Smooth policies

Class of smooth(er) policies:

- E.g. softmax/Botzmann policies: instead of

$$\pi^{k+1}(\cdot|x) \in \operatorname{argmax} Q^k(x,\cdot)$$

use:

$$\pi^{k+1}(\cdot|x) = \text{softmax}_\tau Q^k(x,\cdot) = \frac{e^{\frac{1}{\tau}Q(x,\cdot)}}{\sum_a e^{\frac{1}{\tau}Q(x,a)}}$$

Smooth policies

Class of smooth(er) policies:

- E.g. softmax/Botzmann policies: instead of

$$\pi^{k+1}(\cdot|x) \in \operatorname{argmax} Q^k(x,\cdot)$$

  use:

$$\pi^{k+1}(\cdot|x) = \operatorname{softmax}_\tau Q^k(x,\cdot) = \frac{e^{\frac{1}{\tau}Q(x,\cdot)}}{\sum_a e^{\frac{1}{\tau}Q(x,a)}}$$

- forces to play every action with a positive probability
- temperature τ can be decreased progressively if needed
- solves the problem of ambiguity among possible elements of argmax
- but the equilibrium policy $\hat{\pi}$ is not necessarily of $\operatorname{softmax}$ form!

Reward regularization

Reward regularization:

- Modify the reward with a regularizing penalty
- For instance, entropy penalty: instead of:

$$r(x, a, \mu)$$

use:

$$r(x, a, \mu) - \eta \log \left(\frac{\pi(a|x)}{\tilde{\pi}(a|x)} \right)$$

where $\tilde{\pi}$ is a reference policy (e.g., uniform)

Reward regularization

Reward regularization:

- Modify the reward with a regularizing penalty
- For instance, entropy penalty: instead of:

$$r(x, a, \mu)$$

  use:

$$r(x, a, \mu) - \eta \log\left(\frac{\pi(a|x)}{\tilde{\pi}(a|x)}\right)$$

  where $\tilde{\pi}$ is a reference policy (e.g., uniform)
- it depends on the whole policy $\pi(\cdot|x)$ and not just on the action played
- helps to ensure uniqueness of the equilibrium and the BR
- but only for the *modified* game $\neq$ original game

Some Canonical Examples

Algorithm: Fixed point iter.

input : Initial policy π^0

$\mu^0 := \mu^{\pi^0}$;
for $k = 1, \dots, K$ **do**
  $\pi^k :=$ BR against μ^{k-1};
  $\mu^k := \mu^{\pi^k}$;
return π^K, μ^K

$\downarrow$

Algorithm: Fictitious Play

input : Initial policy π^0

$\bar{\pi}^0 := \pi^0$;
$\bar{\mu}^0 := \mu^{\bar{\pi}^0}$;
for $k = 1, \dots, K$ **do**
  $\pi^k :=$ BR against $\bar{\mu}^{k-1}$;
  $\bar{\mu}^k := \frac{k}{k+1}\bar{\mu}^{k-1} + \frac{1}{k+1}\mu^{\pi^k}$;
  $\bar{\pi}^k :=$ policy giving $\bar{\mu}^k$;
return $\bar{\pi}^K$, $\bar{\mu}^K$

Algorithm: Policy iter.

input : Initial policy π^0

$\mu^0 := \mu^{\pi^0}$;
for $k = 1, \dots, K$ **do**
  $Q^k :=$ Q-func. for π^{k-1} given μ^{k-1};
  $\pi^k := \mathrm{argmax}\ Q^k$;
  $\mu^k := \mu^{\pi^k}$;
return π^K, μ^K

$\downarrow$

Algorithm: OMD

input : Initial policy π^0

$\mu^0 := \mu^{\pi^0}$;
for $k = 1, \dots, K$ **do**
  $Q^k :=$ Q-func. for π^{k-1} given μ^{k-1};
  $\bar{Q}^k := \bar{Q}^{k-1} + \alpha Q^k$;
  $\pi^k := \mathrm{softmax}_\tau\ \bar{Q}^k$;
  $\mu^k := \mu^{\pi^k}$;
return π^K, μ^K

Assumptions and convergence guarantees

Several classes of assumptions to guarantee convergence of the iterations:

1. "**Quantitative**" assumptions:
   - small Lipschitz constants / short time
   - proof by strict contraction
   - Ex: [HMC06a, GHXZ19, AKS20b, LST21] . . .

2. "**Qualitative**/structural" assumptions:
   - potential structure / monotonicity
   - proof by Lyapunov stability
   - Ex: [CH17, Had17, Had18, MJMdC18, PPL+20, PPE+21a] . . .

Convergence?

How can we check whether the algorithm has converged?

Beware:

- Total reward of a player is not a good indicator of convergence
- Distance between π and $\hat{\pi}$ is not necessarily meaningful

Convergence?

How can we check whether the algorithm has converged?

Beware:

- Total reward of a player is not a good indicator of convergence
- Distance between π and $\hat{\pi}$ is not necessarily meaningful

$\rightarrow$ **Exploitability:**

- Evaluates the quality of a policy in a game [ZJBP07, LWZB09]
- *How "far" π is from being a Nash equilibrium policy?*

Convergence?

How can we check whether the algorithm has converged?

Beware:

- Total reward of a player is not a good indicator of convergence
- Distance between π and $\hat{\pi}$ is not necessarily meaningful

$\rightarrow$ **Exploitability:**

- Evaluates the quality of a policy in a game [ZJBP07, LWZB09]
- *How "far" π is from being a Nash equilibrium policy?*

In the context of MFGs:

- Definition: The **exploitability** $\mathcal{E}(\pi)$ of a policy π is defined as:

$$\mathcal{E}(\pi) := \max_{\pi'} J(\pi', \mu^{\pi}) - J(\pi, \mu^{\pi})$$

- Interpretation: $\mathcal{E}(\pi)$ quantifies the average gain for a representative player to replace its policy by a best response, while the rest of the population plays with policy π.
- If $\mathcal{E}(\pi) = 0$, then π is a Nash equilibrium policy.

Outline

OpenSpiel

- Open source framework for research in learning in games
- Main motivation: multi-agent reinforcement learning (MARL)
- Marc Lanctot (Google DeepMind) + many contributors
- Mostly in C++ and Python; APIs in Julia, ...
- Various games including zero-sum games, N-player games, imperfect information, ...
- Chess, Blackjack, Atari, Kuhn poker, Go, ...
- And also: Mean field games

OpenSpiel

Introduction to OpenSpiel:

- https://openspiel.readthedocs.io/en/latest/intro.html
- Python notebook: https://colab.research.google.com/github/deepmind/open_spiel/blob/master/open_spiel/colabs/OpenSpielTutorial.ipynb
- Tutorial by Marc Lanctot available online: https://www.youtube.com/watch?v=8NCPqtPwlFQ
- Paper [LLL+19]
- Two big components:
  - Games
  - Algorithms

MFG in OpenSpiel

- Julien Pérolat, Raphael Marinier, Sertan Girgin & growing number of contributors Théophille Cabannes, Sarah Perrin, Paul Muller, . . .
- For today, three main questions:
  - How to use the existing material?
  - How to define a new MFG model (environment/game)?
  - How to define a new algorithm to learn the MFG solution?

Existing codes for MFG in OpenSpiel

- MFG models in C++: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/games/mfg`
- MFG models in Python: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/games`
  - Crowd modeling 1D illustrated in [PPL+20]
  - Crowd modeling 2D illustrated in [PPL+20, GPL+22]
  - Dynamic routing illustrated in [CLP+22]
  - Linear quadratic (1D) illustrated in [LPG+22]
  - Predator prey (multi-population 2D) illustrated in [PPE+21b]

Existing codes for MFG in OpenSpiel

- MFG models in C++: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/games/mfg`
- MFG models in Python: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/games`
  - Crowd modeling 1D illustrated in [PPL+20]
  - Crowd modeling 2D illustrated in [PPL+20, GPL+22]
  - Dynamic routing illustrated in [CLP+22]
  - Linear quadratic (1D) illustrated in [LPG+22]
  - Predator prey (multi-population 2D) illustrated in [PPE+21b]
- MFG algorithms in Python: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/algorithms`
  - Deep fictitious play [LPG+22]
  - Boltzmann policy iteration [CK21a]
  - Fictitious play [PPL+20], ...
  - Fixed point
  - Mirror descent [PPE+21b]
  - Munchausen deep mirror descent [LPG+22]
  - Munchausen mirror descent

  as well as codes for policies and an evaluation metric: exploitability (nash_conv)

Existing codes for MFG in OpenSpiel

- MFG models in C++: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/games/mfg`
- MFG models in Python: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/games`
  - Crowd modeling 1D illustrated in [PPL+20]
  - Crowd modeling 2D illustrated in [PPL+20, GPL+22]
  - Dynamic routing illustrated in [CLP+22]
  - Linear quadratic (1D) illustrated in [LPG+22]
  - Predator prey (multi-population 2D) illustrated in [PPE+21b]
- MFG algorithms in Python: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/algorithms`
  - Deep fictitious play [LPG+22]
  - Boltzmann policy iteration [CK21a]
  - Fictitious play [PPL+20], . . .
  - Fixed point
  - Mirror descent [PPE+21b]
  - Munchausen deep mirror descent [LPG+22]
  - Munchausen mirror descent

  as well as codes for policies and an evaluation metric: exploitability (nash_conv)
- Some examples: `https://github.com/deepmind/open_spiel/tree/master/open_spiel/python/mfg/examples`

More to come soon. Contributions are welcome!

Q1. *How to use existing material?*

- Install & imports
- Creating a game (e.g., grid world)
- Running a learning algorithm (e.g., fictitious play)
- Plotting the results (e.g., exploitability and distribution)

Tutorial 1: Introduction to MFG in OpenSpiel

Code

Sample code to illustrate: IPython notebook

https://colab.research.google.com/drive/16p95oXZGdhzCAX9MTPlcMNnsD3dyW9ur?usp=sharing

- Installation and imports
- Creating a game
- Running an algorithm
- Visualizing the results

* Special thanks to Marc Lanctot, Julien Pérolat, Raphael Marinier, Sertan Girgin, Sarah Perrin and Kai Shao for this notebook

Tutorial 2: Comparing Learning Algorithms

Another example of game: 2D crowd modeling in a grid world but with obstacles (4 connected rooms). The performance of several algorithms are compared.

Code

Sample code to illustrate: IPython notebook

https://colab.research.google.com/drive/1L1MIVba_2Wm534TDcGL35W2D5vxCsFeo?usp=sharing

- Four room grid world
- Running multiple pre-defined algorithms
- Comparing their exploitabilities

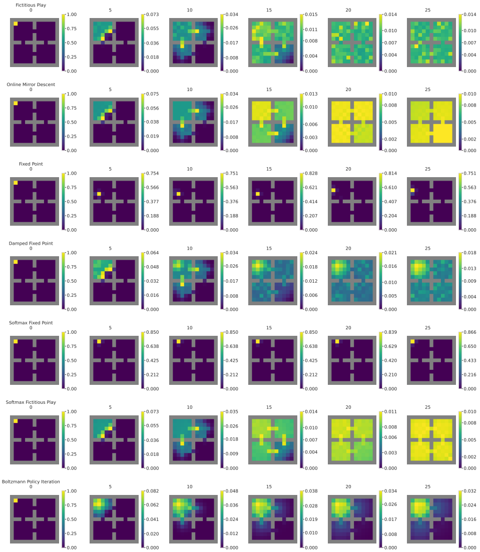
* Special thanks to Marc Lanctot, Julien Pérolat, Raphael Marinier, Sertan Girgin, Sarah Perrin and Kai Shao for this notebook

Comparing Learning Algorithms – Results

Game: crowd aversion in a four-room grid world

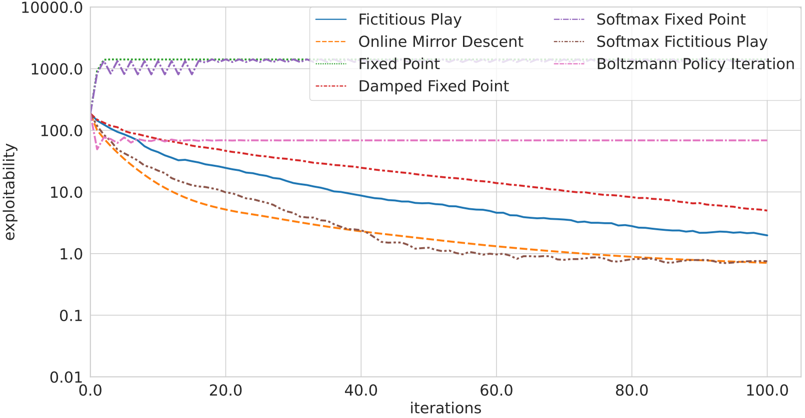
Test case 1: Noise level = 0.2

State distribution at different time steps (columns) for different algorithms (rows):



Comparing Learning Algorithms – Results

Game: crowd aversion in a four-room grid world
Test case 1: Noise level = 0.2
Exploitability vs number of steps:

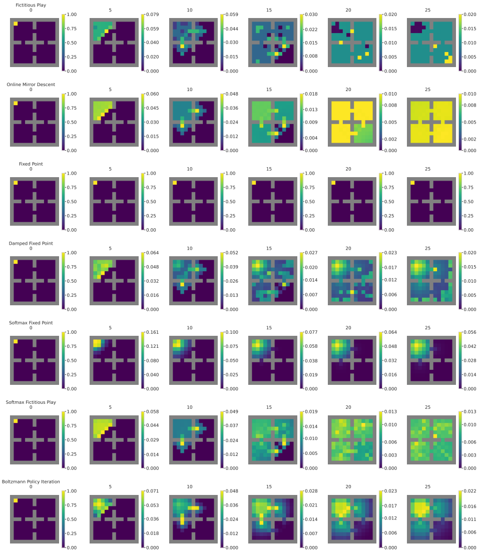


Comparing Learning Algorithms – Results

Game: crowd aversion in a four-room grid world

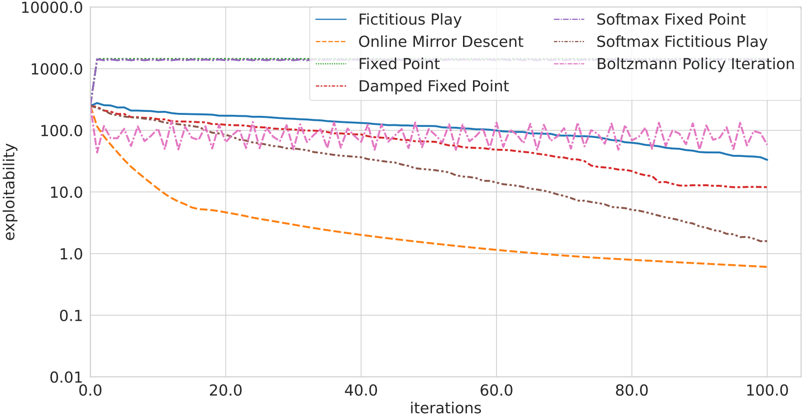
Test case 2: Noise level = 0

State distribution at different time steps (columns) for different algorithms (rows):



Comparing Learning Algorithms – Results

Game: crowd aversion in a four-room grid world
Test case 2: Noise level = 0
Exploitability vs number of steps:



MFG model in OpenSpiel: State

Q2. *How to define a new MFG model?*

- State of the game = all the information required to describe the current stage
- In an MFG: representative player's state and mean field state
- Evolution of the state:
  - Players play in turn
  - Every change to the state occurs through a node
  - Each node has a set of possible actions and a probability to pick each action

MFG model in OpenSpiel: State

Q2. *How to define a new MFG model?*

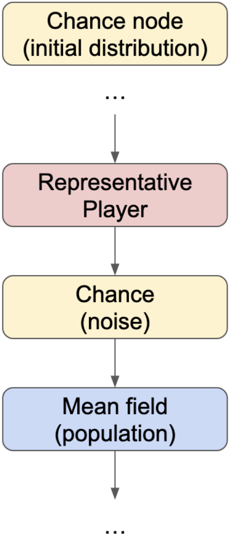
- State of the game = all the information required to describe the current stage
- In an MFG: representative player's state and mean field state
- Evolution of the state:
  - Players play in turn
  - Every change to the state occurs through a node
  - Each node has a set of possible actions and a probability to pick each action
  - So: the representative player is a node
  - the "mean field" is viewed as a node
  - and the "noise" is viewed as a node too

MFG model in OpenSpiel: State

Q2. *How to define a new MFG model?*

- State of the game = all the information required to describe the current stage
- In an MFG: representative player's state and mean field state
- Evolution of the state:
  - Players play in turn
  - Every change to the state occurs through a node
  - Each node has a set of possible actions and a probability to pick each action
  - So: the representative player is a node
  - the "mean field" is viewed as a node
  - and the "noise" is viewed as a node too
  - Time is part of the state: (t, x)
- The state evolves along a tree of possibilities

MFG model in OpenSpiel: State evolution



MFG model in OpenSpiel: State types

- Initial **chance** node:
  - actions: possible states
  - probabilities: given by the initial state distribution

MFG model in OpenSpiel: State types

- Initial **chance** node:
  - actions: possible states
  - probabilities: given by the initial state distribution
- **Player:**
  - actions: set of possible (“legal”) actions for the player
  - probabilities: given by the policy used by this player

MFG model in OpenSpiel: State types

- Initial **chance** node:
  - actions: possible states
  - probabilities: given by the initial state distribution
- **Player:**
  - actions: set of possible (“legal”) actions for the player
  - probabilities: given by the policy used by this player
- **Chance:**
  - can be viewed as a player with a fixed policy
  - actions: set of possible values for the noise impacting the dynamics
  - probabilities: distribution of the noise values

MFG model in OpenSpiel: State types

- Initial **chance** node:
  - actions: possible states
  - probabilities: given by the initial state distribution
- **Player:**
  - actions: set of possible (“legal”) actions for the player
  - probabilities: given by the policy used by this player
- **Chance:**
  - can be viewed as a player with a fixed policy
  - actions: set of possible values for the noise impacting the dynamics
  - probabilities: distribution of the noise values
- **Mean field:** no actions

MFG in OpenSpiel: Distribution

- The distribution is something specific to MFGs (compared with other games in OpenSpiel)
- Remember that time is part of the state object. Evaluating the distribution at a given state means evaluating the distribution at (t, x).
- `master/open_spiel/python/mfg/algorithms/distribution.py`
  - Computes the distribution of a policy
  - `DistributionPolicy`
    - `evaluate`: based on the logic behind nodes
    - `_one_forward_step`
- `master/open_spiel/python/mfg/distribution.py`
  - Representation of a distribution for a game
  - `Distribution`
- `master/open_spiel/python/mfg/tabular_distribution.py`
  - Tabular representation of a distribution for a game
  - `TabularDistribution`

MFG model in OpenSpiel: Example

We take a concrete example: crowd modeling in 1D with a grid world

`master/open_spiel/python/mfg/games/crowd_modelling.py`

3 main classes

- `MFGCrowdModellingGame`:
  - `__init__`: initialization
  - `new_initial_state`: generate new initial state
- `MFGCrowdModellingState`:
  - `__init__`: initialization
  - `_legal_actions`: actions that are valid
  - `chance_outcomes`: distribution over values of the noise in the dynamics
  - `_apply_action`: will be called at each node to modify the state based on the action
  - `_rewards`: representative player's reward
- `Observer`:
  - defines an observation, here basically t and x

Q3. *How to define a new algorithm?*

Simplest one: Fixed point
`master/open_spiel/python/mfg/algorithms/fixed_point.py`

A bit more involved: Fictitious play
`master/open_spiel/python/mfg/algorithms/fictitious_play.py`

- Main class `FictitiousPlay`
- Main method `iteration`
  - Compute the distribution (sequence) associated to the current policy
  - Update the policy (using fictitious play rule); this uses an auxiliary class `MergedPolicy` to mix the previous policy and the new one
- `get_policy`: returns the current policy

Tutorial 3: Game construction

Code

Sample code to illustrate: IPython notebook

https://colab.research.google.com/drive/1uIcDYxQ9f7ngqIOo7ittZ4jEmXFOOZs9?usp=sharing

- Details of the definition of an MFG game in OpenSpiel
- Modification of an existing game
- Reward function, transitions, . . .

* Special thanks to Marc Lanctot, Julien Pérolat, Raphael Marinier, Sertan Girgin, Sarah Perrin and Kai Shao for this notebook

Outline

Outline

Revisiting Dynamic Programming: Classical Setup

Classical MDP (S, A, p, r, γ):

$$Q^\pi(x,a) = (\mathcal{B}^\pi Q^\pi)(x,a) = r(x,a) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a), \\ a' \sim \pi(\cdot|x)}} \Big[Q^\pi(x',a') \Big]$$

$\rightarrow$ Can be computed by applying repeatedly $\mathcal{B}^\pi$

Revisiting Dynamic Programming: Classical Setup

Classical MDP (S, A, p, r, γ):

$$Q^\pi(x,a) = (\mathcal{B}^\pi Q^\pi)(x,a) = r(x,a) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a),\\ a' \sim \pi(\cdot|x)}} \Big[Q^\pi(x',a') \Big]$$

$\rightarrow$ Can be computed by applying repeatedly $\mathcal{B}^\pi$

$\rightarrow$ *But what if* p *&* r *are unknown and we can only observe samples* $(x', r(x,a))$*?*

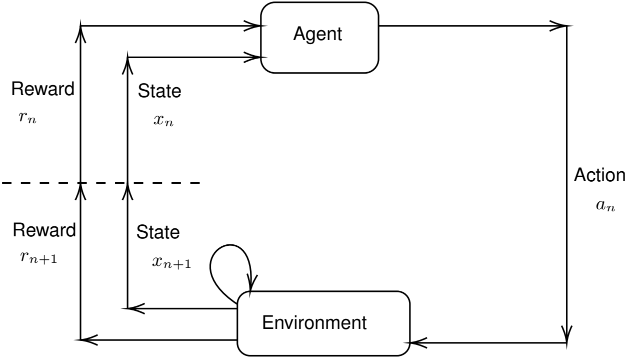
Revisiting Dynamic Programming: Classical Setup

Classical MDP (S, A, p, r, γ):

$$Q^\pi(x,a) = (\mathcal{B}^\pi Q^\pi)(x,a) = r(x,a) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a), \\ a' \sim \pi(\cdot|x)}} \left[Q^\pi(x', a') \right]$$

$\rightarrow$ Can be computed by applying repeatedly $\mathcal{B}^\pi$

$\rightarrow$ *But what if p & r are unknown and we can only observe samples $(x', r(x,a))$?*



See e.g. [SB18]

Revisiting Dynamic Programming: Mean Field Game Setup

MDP parameterized by mean field term $(S, A, p(\cdot|\cdot,\cdot,\mu), r(\cdot|\cdot,\cdot,\mu), \gamma)$:

$$Q^{\mu,\pi}(x,a) = (\mathcal{B}^{\mu,\pi} Q^{\mu,\pi})(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

MDP parameterized by mean field term $(S, A, p(\cdot|\cdot,\cdot,\mu), r(\cdot|\cdot,\cdot,\mu), \gamma)$:

$$Q^{\mu,\pi}(x,a) = (\mathcal{B}^{\mu,\pi} Q^{\mu,\pi})(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu),\\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

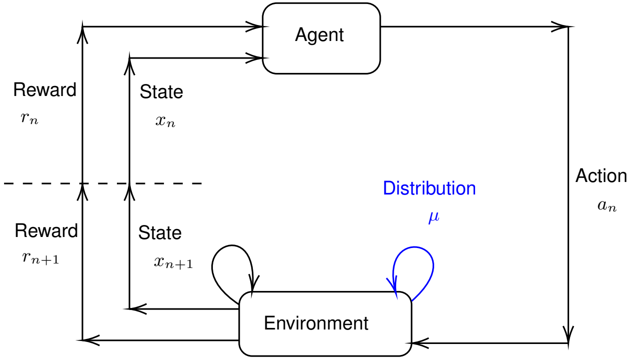
$\rightarrow$ *What if p & r are unknown and we can only observe samples $(x', r(x,a,\mu))$?*

Revisiting Dynamic Programming: Mean Field Game Setup

MDP parameterized by mean field term $(S, A, p(\cdot|\cdot,\cdot,\mu), r(\cdot|\cdot,\cdot,\mu), \gamma)$:

$$Q^{\mu,\pi}(x,a) = (\mathcal{B}^{\mu,\pi} Q^{\mu,\pi})(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu),\\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

$\rightarrow$ *What if p & r are unknown and we can only observe samples $(x', r(x,a,\mu))$?*

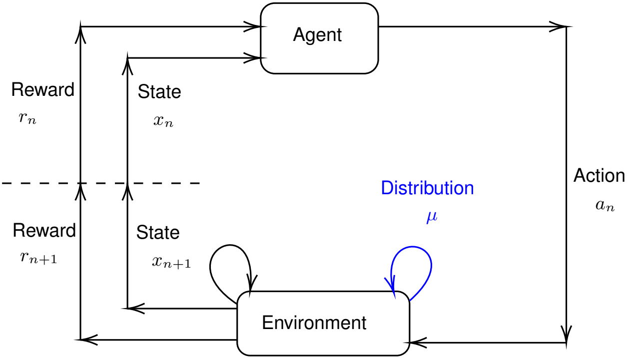


Revisiting Dynamic Programming: Mean Field Game Setup

MDP parameterized by mean field term $(S, A, p(\cdot|\cdot,\cdot,\mu), r(\cdot|\cdot,\cdot,\mu), \gamma)$:

$$Q^{\mu,\pi}(x,a) = (\mathcal{B}^{\mu,\pi} Q^{\mu,\pi})(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

$\to$ *What if p & r are unknown and we can only observe samples $(x', r(x,a,\mu))$?*



Note: the agent does not need to observe μ, but it is part of the environment.

Revisiting Dynamic Programming: Mean Field Game Setup

How to deal with μ in practice? To implement the simulator, we can for instance:

- Vector (if finite S); updates using transition matrix
- Empirical distribution μ^N; updates using individual transitions
- Neural network (e.g., normalizing flow); updates by training
- ...

Outline

Policy evaluation

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Policy evaluation

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume we can compute the expectation perfectly.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$:

- With tabular representation: pointwise update for (x,a)

$$Q^{k+1}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{k}(x',a') \right]$$

- With function approximation: Q^{k+1} parameterized by θ^{k+1} minimizing

$$\mathbb{E}\left[\left| Q_{\theta^{k+1}}(x,a) - r(x,a,\mu) - \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q_{\theta^k}(x',a') \right] \right|^2 \right]$$

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume only samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$:

- Observe $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment
- Approximate $\mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$ by Monte Carlo
- Use similar updates as before (in the ideal case)? For instance with tabular representation: at a given k, for all (x,a) compute:

$$Q^{k+1}(x,a) = r(x,a,\mu) + \gamma \tilde{\mathbb{E}}^B_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{k}(x',a') \right]$$

  where $\tilde{\mathbb{E}}^B$ is an empirical expectation based on a batch of B i.i.d samples.
- This is model-free (= purely based on samples from the environment) ...

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume only samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$:

- Observe $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment
- Approximate $\mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$ by Monte Carlo
- Use similar updates as before (in the ideal case)? For instance with tabular representation: at a given k, for all (x,a) compute:

  $$Q^{k+1}(x,a) = r(x,a,\mu) + \gamma \tilde{\mathbb{E}}^B_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{k}(x',a') \right]$$

  where $\tilde{\mathbb{E}}^B$ is an empirical expectation based on a batch of B i.i.d samples.
- This is model-free (= purely based on samples from the environment) . . .
- But this requires: *many* samples for *every* (x,a) at *every* iteration k . . .

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume we only have access to samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$: In practice:

- asynchronous updates: follow a trajectory $(x^k, a^k)_{k \geq 0}$:

$$Q^{k+1}(x^k, a^k) = r(x^k, a^k, \mu) + \gamma Q^k(x^{k+1}, a^{k+1})$$

where $x^{k+1} \sim p(\cdot|x^k, a^k, \mu)$, $a^{k+1} \sim \pi(\cdot|x^k)$
$\rightarrow$ addresses the previous point . . . but very unstable

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu),\\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume we only have access to samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$: In practice:

- asynchronous updates: follow a trajectory $(x^k, a^k)_{k\geq 0}$:

$$Q^{k+1}(x^k,a^k) = r(x^k,a^k,\mu) + \gamma Q^k(x^{k+1},a^{k+1})$$

  where $x^{k+1} \sim p(\cdot|x^k,a^k,\mu)$, $a^{k+1} \sim \pi(\cdot|x^k)$
  $\rightarrow$ addresses the previous point . . . but very unstable

- learning rate:

$$Q^{k+1}(x^k,a^k) = (1-\alpha) Q^k(x^k,a^k) + \alpha \Big[r(x^k,a^k,\mu) + \gamma Q^k(x^{k+1},a^{k+1}) \Big]$$

Policy evaluation: Model-free

Policy evaluation: given μ, π, evaluate

$$Q^{\mu,\pi}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{\substack{x' \sim p(\cdot|x,a,\mu), \\ a' \sim \pi(\cdot|x)}} \left[Q^{\mu,\pi}(x',a') \right]$$

Assume we only have access to samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,\pi}$: In practice:

- asynchronous updates: follow a trajectory $(x^k, a^k)_{k \geq 0}$:

  $$Q^{k+1}(x^k, a^k) = r(x^k, a^k, \mu) + \gamma Q^k(x^{k+1}, a^{k+1})$$

  where $x^{k+1} \sim p(\cdot|x^k, a^k, \mu)$, $a^{k+1} \sim \pi(\cdot|x^k)$

  $\rightarrow$ addresses the previous point ... but very unstable

- learning rate:

  $$Q^{k+1}(x^k, a^k) = (1-\alpha) Q^k(x^k, a^k) + \alpha \left[r(x^k, a^k, \mu) + \gamma Q^k(x^{k+1}, a^{k+1}) \right]$$

- many extra tricks (replay buffer, policy parameterization, ...)

Optimal policy computation

Best response computation: given μ, compute

$$Q^{\mu,*}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)}\left[\max_{a'} Q^{\mu,*}(x',a')\right]$$

Optimal policy computation

Best response computation: given μ, compute

$$Q^{\mu,*}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)} \left[\max_{a'} Q^{\mu,*}(x',a') \right]$$

Assume we can compute the expectation perfectly.

Repeatedly improve estimate Q^k of $Q^{\mu,*}$:

- With tabular representation: pointwise update for (x,a)

$$Q^{k+1}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)} \left[\max_{a'} Q^k(x',a') \right]$$

- With function approximation: Q^{k+1} parameterized by θ^{k+1} minimizing

$$\left\| (x,a) \mapsto Q_{\theta^{k+1}}(x,a) - r(x,a,\mu) - \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)} \left[\max_{a'} Q_{\theta^k}(x',a') \right] \right\|_2$$

Optimal policy computation: Model-free

Best response computation: given μ, compute

$$Q^{\mu,*}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)} \left[\max_{a'} Q^{\mu,*}(x',a') \right]$$

Assume only samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,*}$:

- similar as evaluation, using MC samples
- computation of $\max$ (and argmax to recover an optimal policy) possible by exhaustive search if the action space A is finite *and small*
- tabular Q-learning [WD92] (with extra μ in the environment):

$$Q^{k+1}(x^k,a^k) = (1-\alpha)Q^k(x^k,a^k) + \alpha \left[r(x^k,a^k,\mu) + \gamma \max_{a'} Q^k(x^{k+1},a') \right]$$

  where $x^{k+1} \sim p(\cdot|x^k,a^k,\mu)$ and $a^k \sim$ some policy

Optimal policy computation: Model-free

Best response computation: given μ, compute

$$Q^{\mu,*}(x,a) = r(x,a,\mu) + \gamma \mathbb{E}_{x' \sim p(\cdot|x,a,\mu)} \left[\max_{a'} Q^{\mu,*}(x',a') \right]$$

Assume only samples $x' \sim p(\cdot|x,a,\mu), r(x,a,\mu)$ from the environment.

Repeatedly improve estimate Q^k of $Q^{\mu,*}$:

- similar as evaluation, using MC samples
- computation of $\max$ (and argmax to recover an optimal policy) possible by exhaustive search if the action space A is finite *and small*
- tabular Q-learning [WD92] (with extra μ in the environment):

  $$Q^{k+1}(x^k,a^k) = (1-\alpha)Q^k(x^k,a^k) + \alpha\left[r(x^k,a^k,\mu) + \gamma \max_{a'} Q^k(x^{k+1},a')\right]$$

  where $x^{k+1} \sim p(\cdot|x^k,a^k,\mu)$ and $a^k \sim$ some policy
- otherwise: learn an optimal parameterized policy
  - either along the way, with the Q-function $\Rightarrow$ actor-critic methods
  - only the parameterized policy $\Rightarrow$ policy gradient methods
- Ex: DQN, SAC, PPO, . . .

MFG Nash equilibrium: Model-free

- Above: enables the computation of a Best Response
  (using for instance model-free versions of value iteration and policy iteration)

MFG Nash equilibrium: Model-free

- Above: enables the computation of a Best Response
  (using for instance model-free versions of value iteration and policy iteration)

- Good but not enough for Nash equilibrium!

- "Outer loop" to **update the mean field μ**

MFG Nash equilibrium: Model-free

- Above: enables the computation of a Best Response
  (using for instance model-free versions of value iteration and policy iteration)
- Good but not enough for Nash equilibrium!
- “Outer loop” to **update the mean field μ**
- Various options, depending on the MFG setting:
  - Stationary setting: one or many applications of the transition matrix (ideally: computation of the stationary distribution)
  - Evolutive setting: application of the transition matrix for each of the time steps (computation of the MF sequence)

MFG Nash equilibrium: Model-free

- Above: enables the computation of a Best Response
  (using for instance model-free versions of value iteration and policy iteration)
- Good but not enough for Nash equilibrium!
- "Outer loop" to **update the mean field** μ
- Various options, depending on the MFG setting:
  - Stationary setting: one or many applications of the transition matrix (ideally: computation of the stationary distribution)
  - Evolutive setting: application of the transition matrix for each of the time steps (computation of the MF sequence)
- If applying the transition matrix is not an option (e.g., continuous spaces), one can for instance use an empirical distribution obtained by simulating N agents

MFG algorithms in OpenSpiel: Reinforcement Learning – Principles

OpenSpiel also contains RL codes for MFGs

Two main building blocks:

- **Environment** (in the sense of RL): in charge of updating the State based on the based on the Game
- **Agent**: in charge of training the policy by interacting with the environment

MFG algorithms in OpenSpiel: Reinforcement Learning – Examples

Policy update: best respond computation for instance through DQN:

- DQN is a variant of Q-learning with a neural network for Q [MKS+15]
- Implementation: `open_spiel/python/mfg/examples/mfg_dqn_jax.py`
- neural network implementation through JAX
- see the source code for details (hyperparameters etc.)

MFG algorithms in OpenSpiel: Reinforcement Learning – Examples

Policy update: best respond computation for instance through DQN:

- DQN is a variant of Q-learning with a neural network for Q [MKS+15]
- Implementation: `open_spiel/python/mfg/examples/mfg_dqn_jax.py`
- neural network implementation through JAX
- see the source code for details (hyperparameters etc.)

Mean field update: Example of DQN embedded in Fictitious Play [LPG+22]:

- Train a NN for the *average* policy across iterations
- Implem.: `open_spiel/python/mfg/examples/mfg_dqn_fp_jax.py`
- Key steps:
  - `fp.iteration(br_policy=joint_avg_policy)`: performs one iteration of fictitious play (updates the policy and the distribution)
  - `distrib = distribution.DistributionPolicy(game, fp.get_policy())`: get the distribution induced by the new policy, just computed by fictitious play iteration
  - `env.update_mfg_distribution(distrib)`: update the environment's distribution using the one obtained from the fictitious play iteration
  - `agents[p].step(time_step)`: train the agent

MFG algorithms in OpenSpiel: Reinforcement Learning – Examples

Policy update: best respond computation for instance through DQN:

- DQN is a variant of Q-learning with a neural network for Q [MKS+15]
- Implementation: `open_spiel/python/mfg/examples/mfg_dqn_jax.py`
- neural network implementation through JAX
- see the source code for details (hyperparameters etc.)

Mean field update: Example of DQN embedded in Fictitious Play [LPG+22]:

- Train a NN for the *average* policy across iterations
- Implem.: `open_spiel/python/mfg/examples/mfg_dqn_fp_jax.py`
- Key steps:
  - `fp.iteration(br_policy=joint_avg_policy)`: performs one iteration of fictitious play (updates the policy and the distribution)
  - `distrib = distribution.DistributionPolicy(game, fp.get_policy())`: get the distribution induced by the new policy, just computed by fictitious play iteration
  - `env.update_mfg_distribution(distrib)`: update the environment's distribution using the one obtained from the fictitious play iteration
  - `agents[p].step(time_step)`: train the agent

Alternative: Munchausen Deep Mirror Descent [LPG+22]:

- Train a NN for the cumulative Q-function
- Implem.: `open_spiel/python/mfg/examples/munchausen_deep_mirror_descent.py`

Tutorial 4: Deep RL for MFG in OpenSpiel

Code

Sample code to illustrate: IPython notebook

`https://colab.research.google.com/drive/1rF9DpjO_xTpbBC2Y-6h_7yQ80j75eb6j?usp=sharing`

- Installation and imports for DRL in OpenSpiel
- Munchausen Deep Mirror Descent
- Average Network Fictitious Play

* Special thanks to Marc Lanctot, Julien Pérolat, Raphael Marinier, Sertan Girgin, Sarah Perrin and Kai Shao for this notebook

A (Non-exhaustive) Glance at the literature: RL for MFG

RL for Mean Field Game:

- MARL with mean field approximation: Yang et al. [YLL+18]
- Inverse RL: Yang et al. [YYT+17], Chen et al. [CLK21]
- Multi-time scales: Subramanian et al. [SM19], Angiuli et al. [AFL20, AFLZ20, AH21]
- Fictitious Play with tabular RL: Pérolat et al. [PPL+20], with deep RL: Elie et al. [EPL+20, CK21b] and distribution embedding: Perrin et al. [PLP+21b]
- Fixed point iterations with Q-learning and variants: Guo et al. [GHXZ19, GHXZ20], Anahtarci et al. [AKS19, AKS21], Xie et al. [XYWM21]
- Entropy regularization: Anahtarci et al. [AKS20a], Cui et al. [CK21b]
- LQ MFG with actor-critic: [FYCW19, uZZMB20], or policy gradient: Wang et al. [WHYW21]
- RL for partially observable MFG: Subramanian et al. [STCP20]
- Mean field RL for multiple types: Subramabian et al. [SPTH20, uZMB22]
- Learning Master policies with deep RL: Perrin et al. [PLP+21a]
- Learning with a single agent: [AFL20, ZKBB23]
- ...

Outline

Outline

Stationary Setting – Reminder

Setting:

- Stationary setting: $N_T = \infty$
- No fixed initial m_0 but a stationary distribution
- Notation: $\mathrm{MF}(\pi) :=$ stationary distribution when using policy π:

$$\mu = P_{\mu,\pi}^{\top} \mu =: \mathcal{P}^{\pi}(\mu)$$

- Player's reward: for player's policy $\pi \in \Delta_A$ and mean field $\mu \in \Delta_S$,

$$J(\pi; \mu) = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r(x_n, a_n, \mu)\right]$$

where $\gamma \in (0, 1)$ is a discount parameter, and

$$a_n \sim \pi(\cdot|x_n), \qquad x_0 \sim \mu, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \mu), n \geq 0$$

Solution concepts:

- **Stationary MFG Nash equilibrium:** $(\hat{\pi}, \hat{\mu}) \in \Pi \times \Delta_{S \times A}$ s.t.
  1. Best response: $\hat{\pi} \in \mathrm{BR}(\hat{\mu}) := \mathrm{argmax}_{\pi} J(\pi; \hat{\mu})$
  2. Mean field state: $\hat{\mu} = \mathrm{MF}(\hat{\pi})$
- Fixed point: $\hat{\mu} \in \mathrm{MF}(\mathrm{BR}(\hat{\mu}))$

- **Stationary MFC Social optimum:** $\pi^* \in \Pi$ s.t.
  - Optimality: $\pi^* \in \mathrm{argmax}_{\pi^*} J(\pi^*; \mu^{\pi^*})$ where $\mu^{\pi^*} = \mathrm{MF}(\pi^*)$

Evolutive Setting – Reminder

Setting:

- Horizon: $N_T \in \mathbb{N}$ (extensions: p, r depending on n; infinite horizon)
- Fixed initial state distribution: $m_0 \in \Delta_S$
- The MF evolves in time: $\boldsymbol{\mu} = (\boldsymbol{\mu}_n)_{n=0,\dots,N_T} \in \Delta_S^{N_T}$
- Notation $\mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi}) :=$ generated by policy $\boldsymbol{\pi}$ starting from m_0:

$$\boldsymbol{\mu}_0 = m_0, \qquad \boldsymbol{\mu}_{n+1} = P^\top_{\boldsymbol{\mu}_n, \boldsymbol{\pi}_n} \boldsymbol{\mu}_n, \quad n \geq 0$$

- Player's reward: for player's policy $\boldsymbol{\pi} \in \Pi^{N_T}$ and mean field $\boldsymbol{\mu} \in \Delta_S^{N_T}$,

$$J(\boldsymbol{\pi}; \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{n=0}^{N_T} r(x_n, a_n, \boldsymbol{\mu}_n)\right]$$

where $a_n \sim \boldsymbol{\pi}_n(\cdot|x_n), \qquad x_0 \sim m_0, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \boldsymbol{\mu}_n), n \geq 0$

Solution concepts:

- **Evolutive MFG Nash equilibrium:** $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}) \in \Pi^{N_T} \times \Delta_S^{N_T}$ s.t.
  1. Best response: $\hat{\boldsymbol{\pi}} \in \mathrm{BR}(\hat{\boldsymbol{\mu}}) := \mathrm{argmax}_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; \hat{\boldsymbol{\mu}})$
  2. Mean field flow: $\hat{\boldsymbol{\mu}} = \mathrm{MF}_{m_0,N_T}(\hat{\boldsymbol{\pi}})$
- Fixed point: $\hat{\boldsymbol{\mu}} \in \mathrm{MF}_{m_0,N_T}(\mathrm{BR}(\hat{\boldsymbol{\mu}}))$

- **Evolutive MFC Social optimum:** $\boldsymbol{\pi}^* \in \Pi^{N_T}$ s.t.
  - Optimality: $\boldsymbol{\pi}^* \in \mathrm{argmax}_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; \boldsymbol{\mu}^{\boldsymbol{\pi}})$ where $\boldsymbol{\mu}^{\boldsymbol{\pi}} = \mathrm{MF}_{m_0,N_T}(\boldsymbol{\pi})$

Evolutive Setting – Infinite Horizon Discounted

Setting:

- Horizon: $N_T = +\infty$; discount $\gamma \in (0, 1)$

Evolutive Setting – Infinite Horizon Discounted

Setting:

- Horizon: $N_T = +\infty$; discount $\gamma \in (0, 1)$
- Notation $\mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi})$:= as before generated by policy $\boldsymbol{\pi}$ starting from m_0 (but now: infinite sequence, $N_T = \infty$)
- Player's reward: for player's policy $\boldsymbol{\pi} \in \Pi^\infty$ and mean field $\boldsymbol{\mu} \in \Delta_S^\infty$,

$$J(\boldsymbol{\pi}; \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{n=0}^{+\infty} \gamma^n r(x_n, a_n, \boldsymbol{\mu}_n)\right]$$

where $a_n \sim \boldsymbol{\pi}_n(\cdot|x_n), \qquad x_0 \sim m_0, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \boldsymbol{\mu}_n), n \geq 0$

Setting:

- Horizon: $N_T = +\infty$; discount $\gamma \in (0, 1)$
- Notation $\mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi}) :=$ as before generated by policy $\boldsymbol{\pi}$ starting from m_0 (but now: infinite sequence, $N_T = \infty$)
- Player's reward: for player's policy $\boldsymbol{\pi} \in \Pi^\infty$ and mean field $\boldsymbol{\mu} \in \Delta_S^\infty$,

$$J(\boldsymbol{\pi}; \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{n=0}^{+\infty} \gamma^n r(x_n, a_n, \boldsymbol{\mu}_n)\right]$$

where $a_n \sim \boldsymbol{\pi}_n(\cdot|x_n), \qquad x_0 \sim m_0, \quad x_{n+1} \sim p(\cdot|x_n, a_n, \boldsymbol{\mu}_n), n \geq 0$

Solution concepts: as before (with infinite sequences, $N_T = \infty$)

From MFC to MFMDP

Let:

$$J^{MFC}(\boldsymbol{\pi}) := J(\boldsymbol{\pi}; \mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi}))$$

MFC problem:

$$\boldsymbol{\pi}^* \in \operatorname*{argmax}_{\boldsymbol{\pi}} J^{MFC}(\boldsymbol{\pi})$$

From MFC to MFMDP

Let:

$$J^{MFC}(\boldsymbol{\pi}) := J(\boldsymbol{\pi}; \mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi}))$$

MFC problem:

$$\boldsymbol{\pi}^* \in \operatorname*{argmax}_{\boldsymbol{\pi}} J^{MFC}(\boldsymbol{\pi})$$

Note: in the definition of J using policy $\boldsymbol{\pi}$,

$$\boldsymbol{\mu}_n = \mathcal{L}(x_n)$$

so

$$J^{MFC}(\boldsymbol{\pi}) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

where $\bar{r}(\bar{a}_n, \boldsymbol{\mu}_n) := \mathbb{E}_{x_n \sim \boldsymbol{\mu}_n, a_n \sim \boldsymbol{\pi}_n(\cdot|x_n)}[r(x_n, a_n, \boldsymbol{\mu}_n)]$

From MFC to MFMDP

Let:

$$J^{MFC}(\boldsymbol{\pi}) := J(\boldsymbol{\pi}; \mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi}))$$

MFC problem:

$$\boldsymbol{\pi}^* \in \operatorname*{argmax}_{\boldsymbol{\pi}} J^{MFC}(\boldsymbol{\pi})$$

Note: in the definition of J using policy $\boldsymbol{\pi}$,

$$\boldsymbol{\mu}_n = \mathcal{L}(x_n)$$

so

$$J^{MFC}(\boldsymbol{\pi}) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

where $\bar{r}(\bar{a}_n, \boldsymbol{\mu}_n) := \mathbb{E}_{x_n \sim \boldsymbol{\mu}_n, a_n \sim \boldsymbol{\pi}_n(\cdot|x_n)}[r(x_n, a_n, \boldsymbol{\mu}_n)]$

Intuitively, this is an MDP with state = mean field $\boldsymbol{\mu}_n$: **Mean Field MDP**

From MFC to MFMDP

Let:

$$J^{MFC}(\boldsymbol{\pi}) := J(\boldsymbol{\pi}; \mathrm{MF}_{m_0, N_T}(\boldsymbol{\pi}))$$

MFC problem:

$$\boldsymbol{\pi}^* \in \operatorname*{argmax}_{\boldsymbol{\pi}} J^{MFC}(\boldsymbol{\pi})$$

Note: in the definition of J using policy $\boldsymbol{\pi}$,

$$\boldsymbol{\mu}_n = \mathcal{L}(x_n)$$

so

$$J^{MFC}(\boldsymbol{\pi}) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

where $\bar{r}(\bar{a}_n, \boldsymbol{\mu}_n) := \mathbb{E}_{x_n \sim \boldsymbol{\mu}_n, a_n \sim \boldsymbol{\pi}_n(\cdot|x_n)}[r(x_n, a_n, \boldsymbol{\mu}_n)]$

Intuitively, this is an MDP with state = mean field $\boldsymbol{\mu}_n$: **Mean Field MDP**

Extensions:

- common noise: evolution of $\boldsymbol{\mu}_n$ becomes stochastic
- $\boldsymbol{\pi}$ population-dependent policies: $\boldsymbol{\pi}(\cdot|x_n, \boldsymbol{\mu}_n)$
- **common randomization** [CLT23]: $\boldsymbol{\pi}$ itself can be random, picked according to a central planner's policy $\bar{\boldsymbol{\pi}}$

MFMDP

MFMDP problem:

$$\bar{\boldsymbol{\pi}}^* \in \operatorname*{argmax}_{\bar{\boldsymbol{\pi}}} \bar{J} J(\bar{\boldsymbol{\pi}}; m_0)$$

where

$$\bar{J}(\bar{\boldsymbol{\pi}}; m_0) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

subject to:

$$\boldsymbol{\mu}_0 = m_0, \qquad \boldsymbol{\mu}_{n+1} = P^\top_{\boldsymbol{\mu}_n, \bar{\boldsymbol{\pi}}_n} \boldsymbol{\mu}_n \text{ (+ noise)}, \quad n \geq 0$$

MFMDP

MFMDP problem:

$$\bar{\boldsymbol{\pi}}^* \in \operatorname*{argmax}_{\bar{\boldsymbol{\pi}}} \bar{J}J(\bar{\boldsymbol{\pi}}; m_0)$$

where

$$\bar{J}(\bar{\boldsymbol{\pi}}; m_0) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

subject to:

$$\boldsymbol{\mu}_0 = m_0, \qquad \boldsymbol{\mu}_{n+1} = P^\top_{\boldsymbol{\mu}_n, \bar{\boldsymbol{\pi}}_n} \boldsymbol{\mu}_n \text{ (+ noise)}, \quad n \geq 0$$

Value functions:

- $\bar{V}^*(\mu)$ and $\bar{Q}^*(\mu, \bar{a})$
- Dynamic programming equations [CLT23] (see also [GGWX23] without common noise, and [MP19b] with common noise but no common randomization)
- Need to properly define the class of actions and policies [omitted here; see e.g. [CLT23] for details]

MFMDP

MFMDP problem:

$$\bar{\boldsymbol{\pi}}^* \in \operatorname*{argmax}_{\bar{\boldsymbol{\pi}}} \bar{J}J(\bar{\boldsymbol{\pi}}; m_0)$$

where

$$\bar{J}(\bar{\boldsymbol{\pi}}; m_0) = \sum_{n=0}^{+\infty} \gamma^n \bar{r}(\bar{a}_n, \boldsymbol{\mu}_n)$$

subject to:

$$\boldsymbol{\mu}_0 = m_0, \qquad \boldsymbol{\mu}_{n+1} = P^{\top}_{\boldsymbol{\mu}_n, \bar{\boldsymbol{\pi}}_n} \boldsymbol{\mu}_n \text{ (+ noise)}, \quad n \geq 0$$

Value functions:

- $\bar{V}^*(\mu)$ and $\bar{Q}^*(\mu, \bar{a})$
- Dynamic programming equations [CLT23] (see also [GGWX23] without common noise, and [MP19b] with common noise but no common randomization)
- Need to properly define the class of actions and policies [omitted here; see e.g. [CLT23] for details]

RL:

- From here, we can re-use existing RL methods for this MDP of mean-field type
- *Question 1: What is the environment?*
- *Question 2: How to deal with the (continuous) state?*

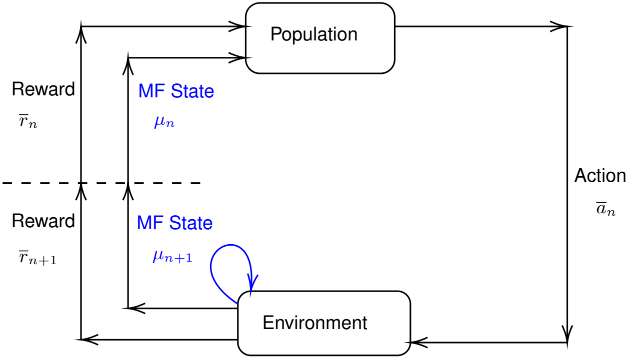
Outline

MFMDP: Environment

Mean field MDP $(\bar{S} = \Delta_S, \bar{A} = \Delta_A^S, \bar{p}, \bar{r}, \gamma)$:

$$\bar{Q}^{\bar{\pi}}(\mu, \bar{a}) = (\bar{\mathcal{B}}^{\mu,\bar{\pi}} \bar{Q}^{\bar{\pi}})(\mu, \bar{a}) = \bar{r}(\mu, \bar{a}) + \gamma \mathbb{E}_{\substack{\mu' \sim \bar{p}(\cdot|\mu, \bar{a}), \\ \bar{a}' \sim \bar{\pi}(\cdot|\mu)}} \left[\bar{Q}^{\bar{\pi}}(\mu', \bar{a}') \right]$$

$\rightarrow$ *What if $\bar{p}$ & $\bar{r}$ are unknown and we can only observe samples $(x', \bar{r}(\mu, \bar{a}))$?*



Mean Field State

How to deal with the MFMDP value functions (and policies)?

- Difficulty: μ_n takes continuous values in Δ_S (likewise for actions $\bar{a}_n$)

Mean Field State

How to deal with the MFMDP value functions (and policies)?

- Difficulty: μ_n takes continuous values in Δ_S (likewise for actions $\bar{a}_n$)
- Option 1: discretize the simplex(es) and then use tabular RL methods

Mean Field State

How to deal with the MFMDP value functions (and policies)?

- Difficulty: μ_n takes continuous values in Δ_S (likewise for actions $\bar{a}_n$)
- Option 1: discretize the simplex(es) and then use tabular RL methods
- Option 2: function approximation $Q_\theta(\mu, \bar{a})$ and then use deep RL methods

Mean Field State

How to deal with the MFMDP value functions (and policies)?

- Difficulty: μ_n takes continuous values in Δ_S (likewise for actions $\bar{a}_n$)
- Option 1: discretize the simplex(es) and then use tabular RL methods
- Option 2: function approximation $Q_\theta(\mu, \bar{a})$ and then use deep RL methods
- Remarks on **policy randomization:**
  - Randomization at the agent level is useful to allow agents to have different trajectories even when start at the same state
  - There exists an optimal policy which is pure at the pop. level [CLT23]
  - But **common randomization** (at the pop. level) helps with exploration

A (Non-exhaustive) Glance at the literature: RL for MFC

RL for Mean Field Control:

- Early works on MDP viewpoint: Gast et al. [GG11, GGLB12]
- Policy optimization for stationary MFC: Subramanian et al. [SM19]
- Policy gradient for LQ MFC [CLT19b, WHYW21] and zero sum mean field type game [CHLT20]
- Multi-time scale for MFC (and MFG): Angiuli et al. [AFL20, AFLZ20, AH21]:
- Mean field MDP: dynamic programming and RL [CLT23, GGWX23, MP19b, GGWX20, CTSK21]
- Decentralized network approach [GGWX21]
- Model based RL for MFC: Pasztor et al. [PBK21]
- . . .

Tabular Q-Learning for MFMDP: Numerical Illustration

Cyber-security example of [KB16]

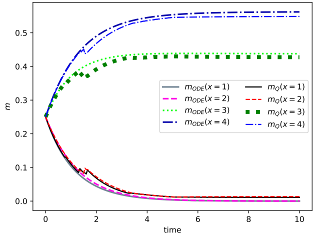
- MFC viewpoint, MF Q-learning
- pure (population and individual) strategies
- discretization of $\bar{S} = \Delta_S, \bar{A} = \Delta_{S \times A}$

Tabular Q-Learning for MFMDP: Numerical Illustration

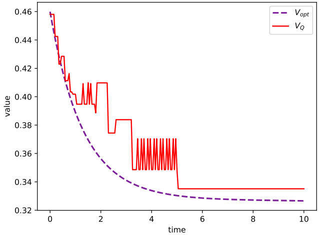
Cyber-security example of [KB16]

- MFC viewpoint, MF Q-learning
- pure (population and individual) strategies
- discretization of $\bar{S} = \Delta_S, \bar{A} = \Delta_{S \times A}$

Test 1: $m_0 = (1/4, 1/4, 1/4, 1/4)$



Evolution of m^{m_0} optimally controlled (m_{ODE}) or controlled using the approximate Q-function (m_Q)



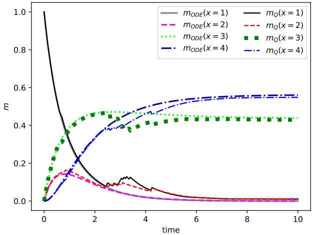
V function (V_{opt}) and approximate Q-function (V_Q) along the optimal flow.

(See section 8.1 of [Lau21] and section 6.1 of [CLT23])

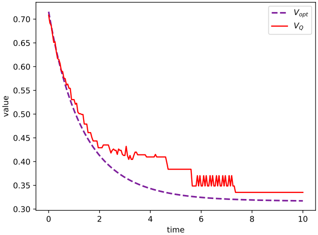
Cyber-security example of [KB16]

- MFC viewpoint, MF Q-learning
- pure (population and individual) strategies
- discretization of $\bar{S} = \Delta_S, \bar{A} = \Delta_{S \times A}$

Test 2: $m_0 = (1, 0, 0, 0)$



Evolution of m^{m_0} optimally controlled (m_{ODE}) or controlled using the approximate Q-function (m_Q)



V function (V_{opt}) and approximate Q-function (V_Q) along the optimal flow.

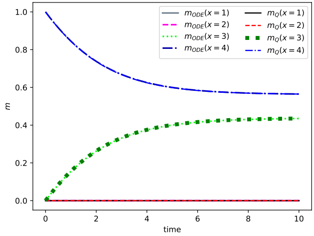
(See section 8.1 of [Lau21] and section 6.1 of [CLT23])

Tabular Q-Learning for MFMDP: Numerical Illustration

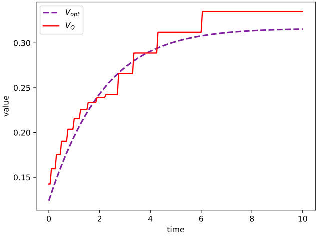
Cyber-security example of [KB16]

- MFC viewpoint, MF Q-learning
- pure (population and individual) strategies
- discretization of $\bar{S} = \Delta_S, \bar{A} = \Delta_{S \times A}$

Test 3: $m_0 = (0, 0, 0, 1)$



Evolution of m^{m_0} optimally controlled (m_{ODE}) or controlled using the approximate Q-function (m_Q)



V function (V_{opt}) and approximate Q-function (V_Q) along the optimal flow.

(See section 8.1 of [Lau21] and section 6.1 of [CLT23])

- Tabular RL is easy to implement and well understood (convergence, etc.)
- But:
  - leads to errors due to projections on the discretized state space
  - not feasible if the number $|S|$ of (individual) states is large, because μ becomes high dimensional

From Tabular to Deep RL for MFMDP

- Tabular RL is easy to implement and well understood (convergence, etc.)
- But:
  - leads to errors due to projections on the discretized state space
  - not feasible if the number $|S|$ of (individual) states is large, because μ becomes high dimensional
- Instead of discretizing the distribution, we can:
  - replace $\bar{Q}^*$ by a parameterized function, e.g., neural network
  - train it using a deep RL algorithm, e.g., DDPG, ...
- Deep RL for MFMDP: See sections 6.1, 6.2 and 6.3 of [CLT23]

Sample code

Code

Sample code to illustrate: IPython notebook

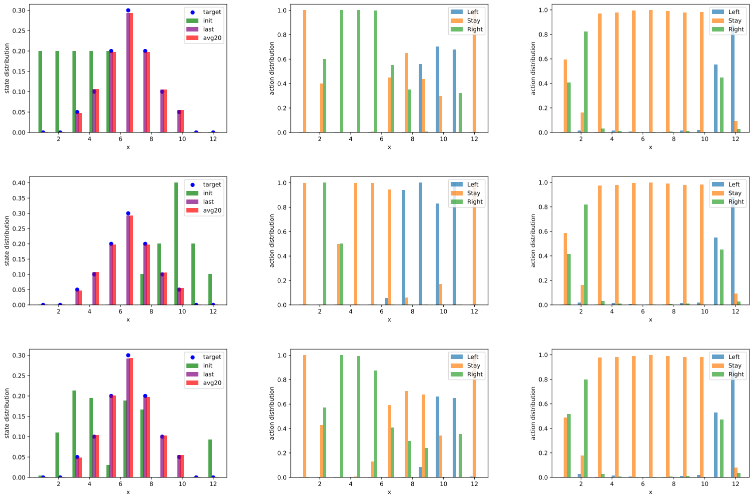
https://colab.research.google.com/drive/1W8H4EM0bx0RFQFzIaNEcPiEYzG02b0jb?usp=sharing

- Same example as above: MFC for cybersecurity
- Solved using deep RL with population-dependent controls

Another Example: Distribution Planning

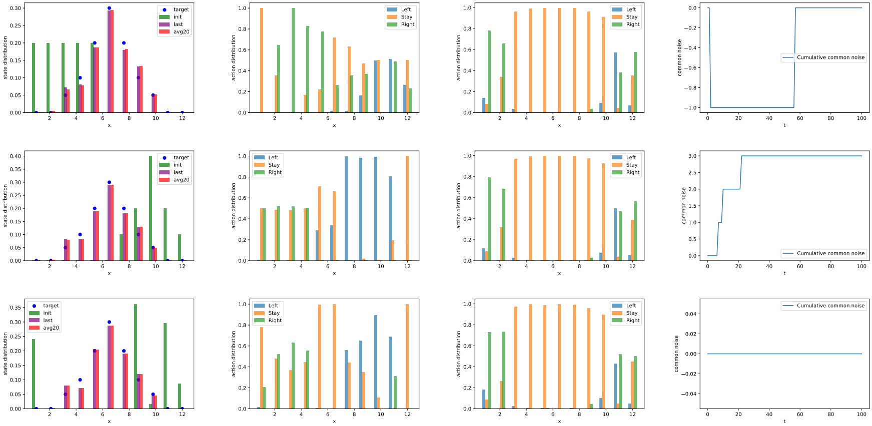
- Goal: match a target distribution.
- $S = \{1, \ldots, 10\}$ and $A = \{-1, 0, +1\}$.
- Transitions: $F(x, a, \mu, e, e^0) = x + a + e^0$.
- Cost:
$$f(x, a, \mu) = |a| + \sum_i |\mu(i) - \mu_{\text{target}}(i)|^2.$$
- Here we chose: $\mu_{\text{target}} = (0, 0, 0.05, 0.1, 0.2, 0.3, 0.2, 0.1, 0.05, 0, 0)$.
- No idiosyncratic noise.
- Hence in general it is not possible to match the target distribution unless the agents are allowed to randomize their actions at the individual level.
- We use $(\Delta_A)^S$ for the level-1 action space.
- Without or with common noise $\varepsilon_n^0 \in A$.
- It is not feasible to rely on a tabular method. We show deep RL results.

Another Example: Distribution Planning



More details in [CLT23]

Another Example: Distribution Planning with Common Noise



More details in [CLT23]

Some Proofs of Convergence

Proof of convergence of RL methods for MFMDP?

- Tabular Q-learning after simplex discretization [CLT23]
- Policy gradient for LQ MFC [CLT19a]
- Still a lot of open questions to study

Outline

Reminder:

- **MFGame:** Fix a distribution μ, compute best response π^{μ}, update μ, ...
- **MFControl:** Fix a policy π, compute induced distribution μ^{π}, update π, ...

Unification of Iterations via Two timescales [AFL20]

Reminder:

- **MFGame:** Fix a distribution μ, compute best response π^{μ}, update μ, ...
- **MFControl:** Fix a policy π, compute induced distribution μ^{π}, update π, ...

Relaxation: using two-timescale idea

- computing best response $\pi^{\mu} \approx$ many steps of policy improvement
- computing stationary distribution $\mu^{\pi} \approx$ many steps of evolution
- rewrite each scheme with 2 nested loops

Unification of Iterations via Two timescales [AFL20]

Reminder:

- **MFGame:** Fix a distribution μ, compute best response π^μ, update μ, ...
- **MFControl:** Fix a policy π, compute induced distribution μ^π, update π, ...

Relaxation: using two-timescale idea

- computing best response $\pi^\mu \approx$ many steps of policy improvement
- computing stationary distribution $\mu^\pi \approx$ many steps of evolution
- rewrite each scheme with 2 nested loops
- replace “many steps” of inner loop by “one step but with a **larger learning rate**”

Unification of Iterations via Two timescales [AFL20]

Reminder:

- **MFGame:** Fix a distribution μ, compute best response π^{μ}, update μ, ...
- **MFControl:** Fix a policy π, compute induced distribution μ^{π}, update π, ...

Relaxation: using two-timescale idea

- computing best response $\pi^{\mu} \approx$ many steps of policy improvement
- computing stationary distribution $\mu^{\pi} \approx$ many steps of evolution
- rewrite each scheme with 2 nested loops
- replace “many steps” of inner loop by “one step but with a **larger learning rate**”

Unification: update both π, μ simultaneously but at different rates ρ^{π}, ρ^{μ}

Unification of Iterations via Two timescales [AFL20]

Reminder:

- **MFGame:** Fix a distribution μ, compute best response π^{μ}, update μ, ...
- **MFControl:** Fix a policy π, compute induced distribution μ^{π}, update π, ...

Relaxation: using two-timescale idea

- computing best response $\pi^{\mu} \approx$ many steps of policy improvement
- computing stationary distribution $\mu^{\pi} \approx$ many steps of evolution
- rewrite each scheme with 2 nested loops
- replace "many steps" of inner loop by "one step but with a **larger learning rate**"

Unification: update both π, μ simultaneously but at different rates ρ^{π}, ρ^{μ}

- $\rho^{\pi} < \rho^{\mu} \Rightarrow \pi$ evolves slowly $\Rightarrow$ MFControl
- $\rho^{\pi} > \rho^{\mu} \Rightarrow \mu$ evolves slowly $\Rightarrow$ MFGame

Policy improvement can be implemented through the Q-function for instance:

$$Q(x,a) = f(x,\mu,a) + \sum_{x' \in \mathcal{X}} p(x'|x,\mu,a) \max_{a'} Q(x',a').$$

Policy improvement can be implemented through the Q-function for instance:

$$Q(x,a) = f(x,\mu,a) + \sum_{x' \in \mathcal{X}} p(x'|x,\mu,a) \max_{a'} Q(x',a').$$

The scheme (using ideal updates) can be written as: for $k \geq 0$

$$\begin{cases} Q_{k+1} & = Q_k + \rho_k^Q \, \mathcal{T}(Q_k, \mu_k) \\ \mu_{k+1} & = \mu_k + \rho_k^\mu \, \mathcal{P}(Q_k, \mu_k), \end{cases}$$

where

$$\begin{cases} \mathcal{T}(Q,\mu)(x,a) = f(x,a,\mu) + \gamma \sum_{x'} p(x'|x,a,\mu) \max_{a'} Q(x',a') - Q(x,a), \\ \mathcal{P}(Q,\mu)(x) = (\mu P^{Q,\mu})(x) - \mu(x), \qquad \text{with } P^{Q,\mu}(x,x') = p(x'|x,\hat{\pi}_Q(x),\mu) \end{cases}$$

Policy improvement can be implemented through the Q-function for instance:

$$Q(x,a) = f(x,\mu,a) + \sum_{x' \in \mathcal{X}} p(x'|x,\mu,a) \max_{a'} Q(x',a').$$

The scheme (using ideal updates) can be written as: for $k \geq 0$

$$\begin{cases} Q_{k+1} & = Q_k + \rho_k^Q \, \mathcal{T}(Q_k, \mu_k) \\ \mu_{k+1} & = \mu_k + \rho_k^\mu \, \mathcal{P}(Q_k, \mu_k), \end{cases}$$

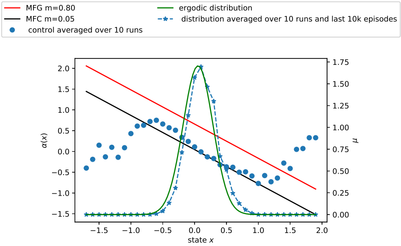
where

$$\begin{cases} \mathcal{T}(Q,\mu)(x,a) = f(x,a,\mu) + \gamma \sum_{x'} p(x'|x,a,\mu) \max_{a'} Q(x',a') - Q(x,a), \\ \mathcal{P}(Q,\mu)(x) = (\mu P^{Q,\mu})(x) - \mu(x), \qquad \text{with } P^{Q,\mu}(x,x') = p(x'|x,\hat{\pi}_Q(x),\mu) \end{cases}$$

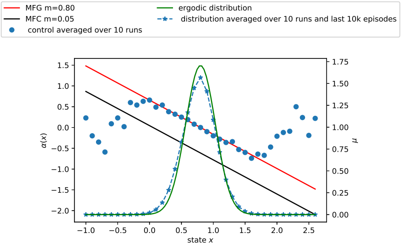
Extension: **sample-based asynchronous** (stochastic approximation [Bor09])

Numerical illustration: Linear-quadratic example

- fixed (quadratic) reward function and (linear) drift function
- the two notions of solutions (MFG/MFC) are different



MFC solution ($\rho^Q < \rho^\mu$)



MFG solution ($\rho^Q > \rho^\mu$)

Comments

- The distribution can be estimated along the way, using a single agent's sample (without "mean field oracle" in the environment)

Comments

- The distribution can be estimated along the way, using a single agent's sample (without "mean field oracle" in the environment)
- Theory: Proof of convergence [AFLZ23]
- Application: Tuning properly the two learning rates is not trivial!

Comments

- The distribution can be estimated along the way, using a single agent's sample (without "mean field oracle" in the environment)
- Theory: Proof of convergence [AFLZ23]
- Application: Tuning properly the two learning rates is not trivial!
- Extension: this approach also works for other models, such as **mean field control games (MFCG)** [ADF$^+$22b, ADF$^+$22a]

  $\rightarrow$ MFG where each agent is of mean field type (solves an MFC)

  $\rightarrow$ 3 time scales instead of 2

Outline

Summary

- Settings (static, stationary, evolutive, . . .)
- Solution concepts (Nash, Social opt., . . .)
- Iterative learning methods for MFG (fixed point, fictitious play, . . .)
- Model-free RL methods for MFG (intuition, implementation in OpenSpiel, . . .)
- MFC and Mean Field MDP
- Tabular and Deep RL for MFMDP

Future Directions

Lot of work to be done! Feel free to reach out if you're interested in contributing.

Future Directions

Lot of work to be done! Feel free to reach out if you're interested in contributing.

- **Theory:**
  - Convergence of iterative methods in more general settings (e.g., Fictitious Play)
  - Convergence rates for iterative methods
  - Same questions for tabular RL algorithms (sample complexity, exploration/exploitation, ...)
  - ... for for deep RL algorithms
  - Extension beyond "plain" MFG/MFC

Future Directions

Lot of work to be done! Feel free to reach out if you're interested in contributing.

- **Theory:**
  - Convergence of iterative methods in more general settings (e.g., Fictitious Play)
  - Convergence rates for iterative methods
  - Same questions for tabular RL algorithms (sample complexity, exploration/exploitation, ...)
  - ... for for deep RL algorithms
  - Extension beyond "plain" MFG/MFC
- **Applications:**
  - More efficient implementation of existing methods
  - Contributing to OpenSpiel (more algorithms, more environments, ...)
  - Real-world applications (more realistic model, real data, ...)

Thank you!

[AACN+19] Ali Al-Aradi, Adolfo Correia, Danilo de Frietas Naiff, Gabriel Jardim, and Yuri Saporito, *Applications of the deep galerkin method to solving partial integro-differential and hamilton-jacobi-bellman equations*, arXiv preprint arXiv:1912.01455 (2019).

[ABC17a] Yves Achdou, Martino Bardi, and Marco Cirant, *Mean field games models of segregation*, Mathematical Models and Methods in Applied Sciences **27** (2017), no. 01, 75–113.

[ABC17b] Ari Arapostathis, Anup Biswas, and Johnson Carroll, *On solutions of mean field games with ergodic cost*, Journal de Mathématiques Pures et Appliquées **107** (2017), no. 2, 205–251.

[ACCD12] Yves Achdou, Fabio Camilli, and Italo Capuzzo-Dolcetta, *Mean field games: numerical methods for the planning problem*, SIAM J. Control Optim. **50** (2012), no. 1, 77–109. MR 2888257

[ACD10] Yves Achdou and Italo Capuzzo-Dolcetta, *Mean field games: numerical methods*, SIAM J. Numer. Anal. **48** (2010), no. 3, 1136–1162. MR 2679575

[ACDL22a] Alexander Aurell, René Carmona, Gökçe Dayanıklı, and Mathieu Laurière, *Finite state graphon games with applications to epidemics*, Dynamic Games and Applications **12** (2022), no. 1, 49–81.

References II

[ACDL22b] Alexander Aurell, Rene Carmona, Gokce Dayanikli, and Mathieu Lauriere, *Optimal incentives to mitigate epidemics: a stackelberg mean field game approach*, SIAM Journal on Control and Optimization **60** (2022), no. 2, S294–S322.

[ACL22] Alexander Aurell, René Carmona, and Mathieu Lauriere, *Stochastic graphon games: Ii. the linear-quadratic case*, Applied Mathematics & Optimization **85** (2022), no. 3, 39.

[ADF+22a] Andrea Angiuli, Nils Detering, Jean-Pierre Fouque, Mathieu Lauriere, and Jimin Lin, *Reinforcement learning algorithm for mixed mean field control games*, arXiv preprint arXiv:2205.02330 (2022).

[ADF+22b] Andrea Angiuli, Nils Detering, Jean-Pierre Fouque, Mathieu Laurière, and Jimin Lin, *Reinforcement learning for intra-and-inter-bank borrowing and lending mean field control game*, Proceedings of the Third ACM International Conference on AI in Finance, 2022, pp. 369–376.

[AFG17] Noha Almulla, Rita Ferreira, and Diogo Gomes, *Two numerical approaches to stationary mean-field games*, Dyn. Games Appl. **7** (2017), no. 4, 657–682. MR 3698446

[AFL20] Andrea Angiuli, Jean-Pierre Fouque, and Mathieu Laurière, *Unified reinforcement q-learning for mean field game and control problems*, arXiv preprint arXiv:2006.13912 (2020).

References III

[AFLZ20] Andrea Angiuli, Jean-Pierre Fouque, Mathieu Laurière, and Mengrui Zhang, *Convergence of two-timescale stochastic approximation for learning MFG and MFC*, In preparation., 2020.

[AFLZ23] ______, *Convergence of multi-scale reinforcement q-learning algorithms for mean field game and control problems*, arXiv preprint arXiv:2312.06659 (2023).

[AGL+19] Angiuli, Andrea, Graves, Christy V., Li, Houzhi, Chassagneux, Jean-François, Delarue, François, and Carmona, René, *Cemracs 2017: numerical probabilistic approach to mfg*, ESAIM: ProcS **65** (2019), 84–113.

[AH21] Andrea Angiuli and Ruimeng Hu, *Deep reinforcement learning for mean field games and mean field control problems in continuous spaces*, In preparation., 2021.

[AK20] Yves Achdou and Ziad Kobeissi, *Mean field games of controls: Finite difference approximations*, arXiv preprint arXiv:2003.03968 (2020).

[AKS19] Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi, *Fitted q-learning in mean-field games*, arXiv preprint arXiv:1912.13309 (2019).

[AKS20a] Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi, *Q-learning in regularized mean-field games*, 2020.

[AKS20b] Berkay Anahtarcı, Can Deha Karıksız, and Naci Saldi, *Value iteration algorithm for mean-field games*, Systems & Control Letters **143** (2020), 104744.

References IV

[AKS21] Berkay Anahtarcı, Can Deha Karıksız, and Naci Saldi, *Learning in discounted-cost and average-cost mean-field games*, 2021.

[AKS23] Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi, *Learning mean-field games with discounted and average costs*, Journal of Machine Learning Research **24** (2023), no. 17, 1–59.

[AL15] Yves Achdou and Mathieu Laurière, *On the system of partial differential equations arising in mean field type control*, Discrete Contin. Dyn. Syst. **35** (2015), no. 9, 3879–3900. MR 3392611

[AL16] Yves Achdou and Mathieu Laurière, *Mean Field Type Control with Congestion (II): An augmented Lagrangian method*, Appl. Math. Optim. **74** (2016), no. 3, 535–578. MR 3575615

[And17a] Roman Andreev, *Preconditioning the augmented Lagrangian method for instationary mean field games with diffusion*, SIAM J. Sci. Comput. **39** (2017), no. 6, A2763–A2783. MR 3731033

[And17b] ______, *Preconditioning the augmented lagrangian method for instationary mean field games with diffusion*, SIAM Journal on Scientific Computing **39** (2017), no. 6, A2763–A2783.

[ATB17] Thomas Anthony, Zheng Tian, and David Barber, *Thinking fast and slow with deep learning and tree search*, Proceedings of NeurIPS, 2017.

References V

[BBC18] Erhan Bayraktar, Amarjit Budhiraja, and Asaf Cohen, *A numerical scheme for a mean field game in some queueing systems based on markov chain approximation method*, SIAM Journal on Control and Optimization **56** (2018), no. 6, 4017–4044.

[BBJT15] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin, *Heads-up limit hold'em poker is solved*, Science **347** (2015), no. 6218.

[BC15] Jean-David Benamou and Guillaume Carlier, *Augmented lagrangian methods for transport optimization, mean field games and degenerate elliptic equations*, Journal of Optimization Theory and Applications **167** (2015), no. 1, 1–26.

[BCCD21] Erhan Bayraktar, Alekos Cecchin, Asaf Cohen, and François Delarue, *Finite state mean field games with wright–fisher common noise*, Journal de Mathématiques Pures et Appliquées **147** (2021), 98–162.

[BCY15] Alain Bensoussan, Michael HM Chau, and Sheung Chi Phillip Yam, *Mean field stackelberg games: Aggregation of delayed instructions*, SIAM Journal on Control and Optimization **53** (2015), no. 4, 2237–2266.

[BGT21] Julian Barreiro-Gomez and Hamidou Tembine, *Mean-field-type games for engineers*, CRC Press, 2021.

[BHL18] Alain Bensoussan, Tao Huang, and Mathieu Laurière, *Mean field control and mean field game models with several populations*, Minimax Theory and its Applications **3** (2018), no. 2, 173–209.

References VI

[BHL+19] Alessandro Balata, Côme Huré, Mathieu Laurière, Huyên Pham, and Isaque Pimentel, *A class of finite-dimensional numerically solvable mckean-vlasov control problems*, ESAIM: Proceedings and Surveys **65** (2019), 114–144.

[BLL19] Charles Bertucci, Jean-Michel Lasry, and Pierre-Louis Lions, *Some remarks on mean field games*, Communications in Partial Differential Equations **44** (2019), no. 3, 205–227.

[BnAKK+19] Luis M. Briceño Arias, Dante Kalise, Ziad Kobeissi, Mathieu Laurière, Álvaro Mateos González, and Francisco J. Silva, *On the implementation of a primal-dual algorithm for second order time-dependent mean field games with local couplings*, ESAIM: ProcS **65** (2019), 330–348.

[BnAKS18] Luis M. Briceño Arias, Dante Kalise, and Francisco J. Silva, *Proximal methods for stationary mean field games with local couplings*, SIAM J. Control Optim. **56** (2018), no. 2, 801–836. MR 3772008

[Bor09] Vivek S Borkar, *Stochastic approximation: a dynamical systems viewpoint*, vol. 48, Springer, 2009.

[BP14] Martino Bardi and Fabio S Priuli, *Linear-quadratic n-person and mean-field games with ergodic cost*, SIAM Journal on Control and Optimization **52** (2014), no. 5, 3022–3052.

[BS17] Noam Brown and Tuomas Sandholm, *Superhuman AI for heads-up no-limit poker: Libratus beats top professionals*, Science **360** (2017), no. 6385.

References VII

[BS19] Noam Brown and Tuomas Sandholm, *Superhuman AI for multiplayer poker*, Science **365** (2019), no. 6456.

[BWZ23] Erhan Bayraktar, Ruoyu Wu, and Xin Zhang, *Propagation of chaos of forward–backward stochastic differential equations with graphon interactions*, Applied Mathematics & Optimization **88** (2023), no. 1, 25.

[CCD19] Jean-François Chassagneux, Dan Crisan, and François Delarue, *Numerical method for FBSDEs of McKean-Vlasov type*, Ann. Appl. Probab. **29** (2019), no. 3, 1640–1684. MR 3914553

[CCG21a] Simone Cacace, Fabio Camilli, and Alessandro Goffi, *A policy iteration method for mean field games*, ESAIM: Control, Optimisation and Calculus of Variations **27** (2021), 85.

[CCG21b] Cacace, Simone, Camilli, Fabio, and Goffi, Alessandro, *A policy iteration method for mean field games*, ESAIM: COCV **27** (2021), 85.

[CCGL22] René Carmona, Daniel B Cooney, Christy V Graves, and Mathieu Lauriere, *Stochastic graphon games: I. the static case*, Mathematics of Operations Research **47** (2022), no. 1, 750–778.

[CCP20] Pierre Cardaliaguet, Marco Cirant, and Alessio Porretta, *Remarks on Nash equilibria in mean field game models with a major player*, Proceedings of the American Mathematical Society **148** (2020), no. 10, 4241–4255.

References VIII

[CCS22] Elisa Calzola, Elisabetta Carlini, and Francisco J Silva, *A high-order lagrange-galerkin scheme for a class of fokker-planck equations and applications to mean field games*, arXiv preprint arXiv:2207.08463 (2022).

[CD18] René Carmona and François Delarue, *Probabilistic theory of mean field games with applications. II*, Probability Theory and Stochastic Modelling, vol. 84, Springer, Cham, 2018, Mean field games with common noise and master equations. MR 3753660

[CD21] René Carmona and Gökçe Dayanıklı, *Mean field game model for an advertising competition in a duopoly*, International Game Theory Review **23** (2021), no. 04, 2150024.

[CDL22] René Carmona, Gökçe Dayanıklı, and Mathieu Laurière, *Mean field models to regulate carbon emissions in electricity production*, Dynamic Games and Applications **12** (2022), no. 3, 897–928.

[CF22] Luciano Campi and Markus Fischer, *Correlated equilibria and mean field games: a simple model*, Mathematics of Operations Research **47** (2022), no. 3, 2240–2259.

[CGL20] Haoyang Cao, Xin Guo, and Mathieu Laurière, *Connecting gans, mfgs, and ot*, arXiv preprint arXiv:2002.04112 (2020).

[CH17] Pierre Cardaliaguet and Saeed Hadikhanloo, *Learning in mean field games: the fictitious play*, ESAIM Cont. Optim. Calc. Var. (2017). MR 3608094

[CH19] Peter E Caines and Minyi Huang, *Graphon mean field games and the GMFG equations: ε-Nash equilibria*, 2019 IEEE 58th conference on decision and control (CDC), IEEE, 2019, pp. 286–292.

[CH21] ______ , *Graphon mean field games and their equations*, SIAM Journal on Control and Optimization **59** (2021), no. 6, 4373–4399.

[CHJH02] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu, *Deep Blue*, Artificial intelligence **134** (2002), no. 1-2.

[CHLT20] René Carmona, Kenza Hamidouche, Mathieu Laurière, and Zongjun Tan, *Policy optimization for linear-quadratic zero-sum mean-field type games*, 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 1038–1043.

[Cir15] Marco Cirant, *Multi-population mean field games systems with neumann boundary conditions*, Journal de Mathématiques Pures et Appliquées **103** (2015), no. 5, 1294–1315.

[CK16] Peter E Caines and Arman C Kizilkale, *ϵ-Nash equilibria for partially observed lqg mean field games with a major player*, IEEE Transactions on Automatic Control **62** (2016), no. 7, 3225–3234.

[CK21a] Kai Cui and Heinz Koeppl, *Approximately solving mean field games via entropy-regularized deep reinforcement learning*, International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1909–1917.

[CK21b] ______ , *Approximately solving mean field games via entropy-regularized deep reinforcement learning*, Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Arindam Banerjee and Kenji Fukumizu, eds.), Proceedings of Machine Learning Research, vol. 130, PMLR, 13–15 Apr 2021, pp. 1909–1917.

[CL18] Pierre Cardaliaguet and Charles-Albert Lehalle, *Mean field game of controls and an application to trade crowding*, Mathematics and Financial Economics **12** (2018), 335–363.

[CL21] René Carmona and Mathieu Laurière, *Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: The ergodic case*, SIAM Journal on Numerical Analysis **59** (2021), no. 3, 1455–1485.

[CL22] ______ , *Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: Ii—the finite horizon case*, The Annals of Applied Probability **32** (2022), no. 6, 4065–4105.

[CLK21] Yang Chen, Jiamou Liu, and Bakhadyr Khoussainov, *Maximum entropy inverse reinforcement learning for mean field games*, arXiv preprint arXiv:2104.14654 (2021).

[CLLP12] Pierre Cardaliaguet, Jean-Michael Lasry, Pierre-Louis Lions, and Alessio Porretta, *Long time average of mean field games.*, Networks & Heterogeneous Media **7** (2012), no. 2.

[CLP+22] Theophile Cabannes, Mathieu Laurière, Julien Perolat, Raphael Marinier, Sertan Girgin, Sarah Perrin, Olivier Pietquin, Alexandre M Bayen, Eric Goubault, and Romuald Elie, *Solving n-player dynamic routing games with congestion: A mean-field approach*, Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, 2022, pp. 1557–1559.

[CLT19a] René Carmona, Mathieu Laurière, and Zongjun Tan, *Linear-quadratic mean-field reinforcement learning: convergence of policy gradient methods*, arXiv preprint arXiv:1910.04295 (2019).

[CLT19b] ______, *Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods*, Preprint, September 2019.

[CLT23] ______, *Model-free mean-field reinforcement learning: mean-field mdp and mean-field q-learning*, The Annals of Applied Probability **33** (2023), no. 6B, 5334–5381.

[CP19] Andrea Cosso and Huyên Pham, *Zero-sum stochastic differential games of generalized mckean–vlasov type*, Journal de Mathématiques Pures et Appliquées **129** (2019), 180–212.

[CS14] Elisabetta Carlini and Francisco J. Silva, *A fully discrete semi-Lagrangian scheme for a first order mean field game problem*, SIAM J. Numer. Anal. **52** (2014), no. 1, 45–67. MR 3148086

[CS15] ______, *A semi-Lagrangian scheme for a degenerate second order mean field game system*, Discrete Contin. Dyn. Syst. **35** (2015), no. 9, 4269–4292. MR 3392626

[CS18] Elisabetta Carlini and Francisco J Silva, *On the discretization of some nonlinear fokker–planck–kolmogorov equations and applications*, SIAM Journal on Numerical Analysis **56** (2018), no. 4, 2148–2177.

[CT21] Fabio Camilli and Qing Tang, *Rates of convergence for the policy iteration method for mean field games systems*, arXiv preprint arXiv:2108.00755 (2021).

[CT22] ______, *Rates of convergence for the policy iteration method for mean field games systems*, Journal of Mathematical Analysis and Applications **512** (2022), no. 1, 126138.

[CTSK21] Kai Cui, Anam Tahir, Mark Sinzger, and Heinz Koeppl, *Discrete-time mean field control with environment states*, 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 5239–5246.

[CW17] Rene Carmona and Peiqi Wang, *An alternative approach to mean field game with major and minor players, and applications to herders impacts*, Applied Mathematics & Optimization **76** (2017), 5–27.

[CZ16] René Carmona and Xiuneng Zhu, *A probabilistic approach to mean field games with major and minor players.*, Annals of applied probability: an official journal of the Institute of Mathematical Statistics **26** (2016), no. 3, 1535–1580.

[DL23] Gokce Dayanikli and Mathieu Lauriere, *A machine learning method for stackelberg mean field games*, arXiv preprint arXiv:2302.10440 (2023).

[dRT15] PE Chaudru de Raynal and CA Garcia Trillos, *A cubature based algorithm to solve decoupled mckean–vlasov forward–backward stochastic differential equations*, Stochastic Processes and their Applications **125** (2015), no. 6, 2206–2255.

[DTT17] Boualem Djehiche, Alain Tcheukam, and Hamidou Tembine, *Mean-field-type games in engineering*, AIMS Electronics and Electrical Engineering **1** (2017), no. 1, 18–73.

[DV21] François Delarue and Athanasios Vasileiadis, *Exploration noise for learning linear-quadratic mean field games*, arXiv preprint arXiv:2107.00839 (2021).

[EMP19] Romuald Elie, Thibaut Mastrolia, and Dylan Possamaï, *A tale of a principal and many, many agents*, Mathematics of Operations Research **44** (2019), no. 2, 440–467.

[EPL+20] Romuald Elie, Julien Perolat, Mathieu Laurière, Matthieu Geist, and Olivier Pietquin, *On the convergence of model free learning in mean field games*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 7143–7150.

[Fel13] Ermal Feleqi, *The derivation of ergodic mean field game equations for several populations of players*, Dynamic Games and Applications **3** (2013), 523–536.

[FL09] Drew Fudenberg and David K Levine, *Learning and equilibrium*, Annu. Rev. Econ. **1** (2009), no. 1, 385–420.

[FSJ21] Dena Firoozi, Arvind V Shrivats, and Sebastian Jaimungal, *Principal agent mean field games in rec markets*, arXiv preprint arXiv:2112.11963 (2021).

[FYCW19] Zuyue Fu, Zhuoran Yang, Yongxin Chen, and Zhaoran Wang, *Actor-critic provably finds nash equilibria of linear-quadratic mean-field games*, 2019.

[FZ20] Jean-Pierre Fouque and Zhaoyu Zhang, *Deep learning methods for mean field control problems with delay*, Frontiers in Applied Mathematics and Statistics **6** (2020), 11.

[GG11] Nicolas Gast and Bruno Gaujal, *A mean field approach for optimization in discrete time*, Discrete Event Dynamic Systems **21** (2011), no. 1, 63–101.

[GGLB12] Nicolas Gast, Bruno Gaujal, and Jean-Yves Le Boudec, *Mean field for markov decision processes: from discrete to continuous optimization*, IEEE Transactions on Automatic Control **57** (2012), no. 9, 2266–2280.

[GGWX20] Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu, *Mean-field controls with q-learning for cooperative marl: Convergence and complexity analysis*, arXiv preprint arXiv:2002.04131 (2020).

[GGWX21] ______, *Mean-field multi-agent reinforcement learning: A decentralized network approach*, arXiv preprint arXiv:2108.02731 (2021).

References XV

[GGWX23] ———, *Dynamic programming principles for mean-field controls with learning*, Operations Research (2023).

[GHXZ19] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang, *Learning mean-field games*, proc. of NeurIPS, 2019.

[GHXZ20] ———, *A general framework for learning mean-field games*, arXiv preprint arXiv:2003.06069 (2020).

[GHZ22] Xin Guo, Anran Hu, and Jiacheng Zhang, *Optimization frameworks and sensitivity analysis of stackelberg mean-field games*, arXiv preprint arXiv:2210.04110 (2022).

[GLPW22] Maximilien Germain, Mathieu Laurière, Huyên Pham, and Xavier Warin, *Deepsets and their derivative networks for solving symmetric pdes*, Journal of Scientific Computing **91** (2022), no. 2, 63.

[GMW22] Maximilien Germain, Joseph Mikael, and Xavier Warin, *Numerical resolution of mckean-vlasov fbsdes using neural networks*, Methodology and Computing in Applied Probability **24** (2022), no. 4, 2557–2586.

[GPL+22] Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Oliver Bachem, Rémi Munos, and Olivier Pietquin, *Concave utility reinforcement learning: The mean-field game viewpoint*, Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, 2022, pp. 489–497.

References XVI

[GPV14] Diogo A Gomes, Stefania Patrizi, and Vardan Voskanyan, *On the existence of classical solutions for stationary extended mean field games*, Nonlinear Analysis: Theory, Methods & Applications **99** (2014), 49–79.

[GS18] Diogo A. Gomes and João Saúde, *Numerical methods for finite-state mean-field games satisfying a monotonicity condition*, Applied Mathematics & Optimization (2018).

[GTC20] Shuang Gao, Rinel Foguen Tchuendom, and Peter E Caines, *Linear quadratic graphon field games*, arXiv preprint arXiv:2006.03964 (2020).

[GV16] Diogo A Gomes and Vardan K Voskanyan, *Extended deterministic mean-field games*, SIAM Journal on Control and Optimization **54** (2016), no. 2, 1030–1055.

[GY20] Diogo A Gomes and Xianjin Yang, *The hessian riemannian flow and newton's method for effective hamiltonians and mather measures*, ESAIM: Mathematical Modelling and Numerical Analysis **54** (2020), no. 6, 1883–1915.

[Had17] Saeed Hadikhanloo, *Learning in anonymous nonatomic games with applications to first-order mean field games*, arXiv preprint arXiv:1704.00378 (2017).

[Had18] ______, *Learning in mean field games*, Ph.D. thesis, PSL Research University, 2018.

[HL22] Ruimeng Hu and Mathieu Laurière, *Recent developments in machine learning methods for stochastic control and games*, SSRN preprint:4096569 (2022).

[HMC06a] Minyi Huang, Roland P. Malhamé, and Peter E. Caines, *Nash certainty equivalence in large population stochastic dynamic games: connections with the physics of interacting particle systems*, Proceedings of the 45th IEEE conference on decision and control, IEEE, 2006, pp. 4921–4926.

[HMC+06b] Minyi Huang, Roland P Malhamé, Peter E Caines, et al., *Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle*, Communications in Information & Systems **6** (2006), no. 3, 221–252.

[KB16] Vassili N. Kolokoltsov and Alain Bensoussan, *Mean-field-game model for botnet defense in cyber-security*, Appl. Math. Optim. **74** (2016), no. 3, 669–692. MR 3575619

[Kob22] Ziad Kobeissi, *Mean field games with monotonous interactions through the law of states and controls of the agents*, Nonlinear Differential Equations and Applications NoDEA **29** (2022), no. 5, 52.

[Lau21] Mathieu Laurière, *Numerical methods for mean field games and mean field type control*, arXiv preprint arXiv:2106.06231 (2021).

[LJL+21] Siting Liu, Matthew Jacobs, Wuchen Li, Levon Nurbekyan, and Stanley J Osher, *Computational methods for first-order nonlocal mean field games with applications*, SIAM Journal on Numerical Analysis **59** (2021), no. 5, 2639–2668.

[LL18] Jean-Michel Lasry and Pierre-Louis Lions, *Mean-field games with a major player*, Comptes Rendus Mathematique **356** (2018), no. 8, 886–890.

[LLL+19] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al., *Openspiel: A framework for reinforcement learning in games*, arXiv preprint arXiv:1908.09453 (2019).

[LP16] Mathieu Laurière and Olivier Pironneau, *Dynamic programming for mean-field type control*, J. Optim. Theory Appl. **169** (2016), no. 3, 902–924. MR 3501391

[LP22] Pierre Lavigne and Laurent Pfeiffer, *Generalized conditional gradient and learning in potential mean field games*, arXiv preprint arXiv:2209.12772 (2022).

[LPG+22] Mathieu Laurière, Sarah Perrin, Sertan Girgin, Paul Muller, Ayush Jain, Theophile Cabannes, Georgios Piliouras, Julien Pérolat, Romuald Elie, Olivier Pietquin, et al., *Scalable deep reinforcement learning algorithms for mean field games*, International Conference on Machine Learning, PMLR, 2022, pp. 12078–12095.

[LS22] Daniel Lacker and Agathe Soret, *A label-state formulation of stochastic graphon games and approximate equilibria on large networks*, Mathematics of Operations Research (2022).

[LST21] Mathieu Laurière, Jiahao Song, and Qing Tang, *Policy iteration method for time-dependent mean field games systems with non-separable hamiltonians*, arXiv preprint arXiv:2110.02552 (2021).

[LST23] ______, *Policy iteration method for time-dependent mean field games systems with non-separable hamiltonians*, Applied Mathematics & Optimization **87** (2023), no. 2, 17.

[LT22] Mathieu Laurière and Ludovic Tangpi, *Convergence of large population games to mean field games with interaction through the controls*, SIAM Journal on Mathematical Analysis **54** (2022), no. 3, 3535–3574.

[LWZB09] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling, *Monte carlo sampling for regret minimization in extensive games*, vol. 22, 2009, pp. 1078–1086.

[M+97] Tom M Mitchell et al., *Machine learning*.

[MB18] Jun Moon and Tamer Başar, *Linear quadratic mean field stackelberg differential games*, Automatica **97** (2018), 200–213.

[MER+22] Paul Muller, Romuald Elie, Mark Rowland, Mathieu Lauriere, Julien Perolat, Sarah Perrin, Matthieu Geist, Georgios Piliouras, Olivier Pietquin, and Karl Tuyls, *Learning correlated equilibria in mean-field games*, arXiv preprint arXiv:2208.10138 (2022).

[MH21] Ming Min and Ruimeng Hu, *Signatured deep fictitious play for mean field games with common noise*, 2021.

[MJMdC18] David Mguni, Joel Jennings, and Enrique Munoz de Cote, *Decentralised learning in systems with many, many strategic agents*, Proceedings of AAAI, 2018.

[MKS^{+}15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., *Human-level control through deep reinforcement learning*, nature **518** (2015), no. 7540, 529–533.

[MLFB20] Stephen McAleer, John Lanier, Roy Fox, and Pierre Baldi, *Pipeline PSRO: A scalable approach for finding approximate Nash equilibria in large games*, Proceedings of NeurIPS, 2020.

[MP19a] Enzo Miller and Huyen Pham, *Linear-quadratic mckean-vlasov stochastic differential games*, Modeling, Stochastic Control, Optimization, and Applications (2019), 451–481.

[MP19b] Médéric Motte and Huyên Pham, *Mean-field markov decision processes with common noise and open-loop controls*, arXiv preprint arXiv:1912.07883 (2019).

[MRE^{+}21] Paul Muller, Mark Rowland, Romuald Elie, Georgios Piliouras, Julien Perolat, Mathieu Lauriere, Raphael Marinier, Olivier Pietquin, and Karl Tuyls, *Learning equilibria in mean-field games: Introducing mean-field PSRO*, arXiv preprint arXiv:2111.08350 (2021).

[MSB^{+}17] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling, *Deepstack: Expert-level artificial intelligence in heads-up no-limit poker*, Science **356** (2017), no. 6337.

[MYZ22] Chenchen Mou, Xianjin Yang, and Chao Zhou, *Numerical methods for mean field games based on gaussian processes and fourier features*, Journal of Computational Physics **460** (2022), 111188.

[N+19] Levon Nurbekyan et al., *Fourier approximation methods for first-order nonlocal mean-field games*, Portugaliae Mathematica **75** (2019), no. 3, 367–396.

[PBK21] Barna Pasztor, Ilija Bogunovic, and Andreas Krause, *Efficient model-based multi-agent mean-field reinforcement learning*, arXiv preprint arXiv:2107.04050 (2021).

[Pfe16] Laurent Pfeiffer, *Numerical methods for mean-field type optimal control problems*, Pure Appl. Funct. Anal. **1** (2016), no. 4, 629–655. MR 3619691

[PLP+21a] Sarah Perrin, Mathieu Laurière, Julien Pérolat, Romuald Élie, Matthieu Geist, and Olivier Pietquin, *Generalization in mean field games by learning master policies*, arXiv preprint arXiv:2109.09717 (2021).

[PLP+21b] Sarah Perrin, Mathieu Laurière, Julien Pérolat, Matthieu Geist, Romuald Élie, and Olivier Pietquin, *Mean field games flock! the reinforcement learning way*, arXiv preprint arXiv:2105.07933 (2021).

[PO19] Francesca Parise and Asuman Ozdaglar, *Graphon games*, Proceedings of the 2019 ACM Conference on Economics and Computation, 2019, pp. 457–458.

[PPE+21a] Julien Perolat, Sarah Perrin, Romuald Elie, Mathieu Laurière, Georgios Piliouras, Matthieu Geist, Karl Tuyls, and Olivier Pietquin, *Scaling up mean field games with online mirror descent*, arXiv preprint arXiv:2103.00623 (2021).

[PPE+21b] ______, *Scaling up mean field games with online mirror descent*, arXiv preprint arXiv:2103.00623 (2021).

[PPL+20] Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin, *Fictitious play for mean field games: Continuous time analysis and applications*, Advances in Neural Information Processing Systems (2020).

[ROL+20] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung, *A machine learning framework for solving high-dimensional mean field game and mean field control problems*, Proceedings of the National Academy of Sciences **117** (2020), no. 17, 9183–9193.

[SB18] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An introduction*, 2nd ed., The MIT Press, 2018.

[SBB+07] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen, *Checkers is solved*, Science **317** (2007), no. 5844.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., *Mastering the game of Go with deep neural networks and tree search*, Nature **529** (2016), no. 7587.

[SHS+18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis, *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*, Science **632** (2018), no. 6419.

[SM19] Jayakumar Subramanian and Aditya Mahajan, *Reinforcement learning in stationary mean-field games*, AAMAS, 2019.

[SPTH20] Sriram Ganapathi Subramanian, Pascal Poupart, Matthew E. Taylor, and Nidhi Hegde, *Multi type mean field reinforcement learning*, CoRR **abs/2002.02513** (2020).

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al., *Mastering the game of Go without human knowledge*, Nature **550** (2017), no. 7676.

References XXIV

[STCP20] Sriram Ganapathi Subramanian, Matthew E. Taylor, Mark Crowley, and Pascal Poupart, *Partially observable mean field reinforcement learning*, CoRR **abs/2012.15791** (2020).

[TS22] Qing Tang and Jiahao Song, *Learning optimal policies in potential mean field games: Smoothed policy iteration algorithms*, arXiv preprint arXiv:2212.04791 (2022).

[uZMB22] Muhammad Aneeq uz Zaman, Erik Miehling, and Tamer Başar, *Reinforcement learning for non-stationary discrete-time linear–quadratic mean-field games in multiple populations*, Dynamic Games and Applications (2022), 1–47.

[uZZMB20] Muhammad Aneeq uz Zaman, Kaiqing Zhang, Erik Miehling, and Tamer Başar, *Reinforcement learning in non-stationary discrete-time linear-quadratic mean-field games*, 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 2278–2284.

[VB22] Deepanshu Vasal and Randall Berry, *Master equation for discrete-time stackelberg mean field games with a single leader*, 2022 IEEE 61st Conference on Decision and Control (CDC), IEEE, 2022, pp. 5529–5535.

[VBC+19] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al., *Grandmaster level in StarCraft II using multi-agent reinforcement learning*, Nature **575** (2019), no. 7782.

References XXV

[VMV21] Deepanshu Vasal, Rajesh Mishra, and Sriram Vishwanath, *Sequential decomposition of graphon mean field games*, 2021 American Control Conference (ACC), IEEE, 2021, pp. 730–736.

[WD92] Christopher JCH Watkins and Peter Dayan, *Q-learning*, Machine learning **8** (1992), 279–292.

[WHYW21] Weichen Wang, Jiequn Han, Zhuoran Yang, and Zhaoran Wang, *Global convergence of policy gradient for linear-quadratic mean-field control/game in continuous time*, International Conference on Machine Learning, PMLR, 2021, pp. 10772–10782.

[XYWM21] Qiaomin Xie, Zhuoran Yang, Zhaoran Wang, and Andreea Minca, *Learning while playing in mean-field games: Convergence and optimality*, Proceedings of the 38th International Conference on Machine Learning (Marina Meila and Tong Zhang, eds.), Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 11436–11447.

[YLL+18] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang, *Mean field multi-agent reinforcement learning*, Proceedings of ICML, 2018.

[YYT+17] Jiachen Yang, Xiaojing Ye, Rakshit Trivedi, Huan Xu, and Hongyuan Zha, *Deep mean field games for learning optimal behavior policy of large populations*, CoRR **abs/1711.03156** (2017).

[ZJBP07] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione, *Regret minimization in games with incomplete information*, vol. 20, 2007, pp. 1729–1736.

[ZKBB23] Muhammad Aneeq Uz Zaman, Alec Koppel, Sujay Bhatt, and Tamer Basar, *Oracle-free reinforcement learning in mean-field games along a single sample path*, International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 10178–10206.