RLChina 2020

Multi-agent Reinforcement Learning: From a Mean-Field Perspective

Renyuan Xu
Mathematical Institute, University of Oxford

August 8, 2020

A Toy Example: When Does RLChina Lecture Start?

Guéant, Lasry, and Lions (2011)

- The RLChina lectures are scheduled to start at $t = 7$PM but often start some time later than t, say T
- The actual starting time T depends on the arrivals of participants
- A rule is imposed saying that the lecture starts at time t or after 90% of the participants have arrived, whichever is earlier.
- Very large number of participants: we agree to consider them as a continuum of agents
- **Question:** What is the actual starting time T?

A Toy Example: When Does RLChina Lecture Start?

- τ_i: time at which participant i decides to arrive
- $\tilde{\tau}_i$: time at which participant i actually arrives:

$$\tilde{\tau}_i = \tau_i + \sigma_i \epsilon_i$$

where

- $\sigma_i \epsilon_i$ is the uncertainty, $\sigma_i \sim m_0$, $\epsilon_i \sim \mathcal{N}(0, 1)$

A Toy Example: When Does RLChina Lecture Start?

Each participant makes decision upon minimizing the expectation of the total cost

$$c(t, T, \tilde{\tau}_i) = c_1(t, T, \tilde{\tau}_i) + c_2(t, T, \tilde{\tau}_i) + c_3(t, T, \tilde{\tau}_i)$$

- Lateness compared to the scheduled time t

$$c_1(t, T, \tilde{\tau}_i) = \alpha[\tilde{\tau}_i - t]_+$$

- Lateness compared to the actually time T

$$c_2(t, T, \tilde{\tau}_i) = \beta[\tilde{\tau}_i - T]_+$$

- Waiting time

$$c_3(t, T, \tilde{\tau}_i) = \gamma[T - \tilde{\tau}_i]_+$$

A Toy Example: When Does RLChina Lecture Start?

Resolution:

1. **Anticipate an actual starting time T, solve for an optimal τ_i**:

$$\alpha\mathcal{N}\left(\frac{\tau_i - t}{\sigma_i}\right) + (\beta + \gamma)\,\mathcal{N}\left(\frac{\tau_i - T}{\sigma_i}\right) = \gamma$$

   where $\mathcal{N}$ is the cumulative distribution of a standard normal.

2. **From τ^i, together with the noise and the rule, find the corresponding actual starting time T^***
   - "Continuum of participants": 90% quantile of a distribution
   - N participants: ordered statistics (intractable)
3. **Show that the mapping $\Gamma : T \mapsto T^*$ has a fixed point**:
   - Banach Fixed Point Theorem

A Toy Example: When Does RLChina Lecture Start?

Resolution:

1. **Anticipate an actual starting time T, solve for an optimal τ_i:**

$$\alpha\mathcal{N}\left(\frac{\tau_i - t}{\sigma_i}\right) + (\beta + \gamma)\,\mathcal{N}\left(\frac{\tau_i - T}{\sigma_i}\right) = \gamma$$

   where $\mathcal{N}$ is the cumulative distribution of a standard normal.

2. **From τ^i, together with the noise and the rule, find the corresponding actual starting time T^***
   - "Continuum of participants": 90% quantile of a distribution
   - N participants: ordered statistics (intractable)
3. **Show that the mapping $\Gamma : T \mapsto T^*$ has a fixed point:**
   - Banach Fixed Point Theorem

A Toy Example: When Does RLChina Lecture Start?

Resolution:

1. **Anticipate an actual starting time T, solve for an optimal τ_i**:

$$\alpha\mathcal{N}\left(\frac{\tau_i - t}{\sigma_i}\right) + (\beta + \gamma)\,\mathcal{N}\left(\frac{\tau_i - T}{\sigma_i}\right) = \gamma$$

   where $\mathcal{N}$ is the cumulative distribution of a standard normal.

2. **From τ^i, together with the noise and the rule, find the corresponding actual starting time T^***
   - "Continuum of participants": 90% quantile of a distribution
   - N participants: ordered statistics (intractable)
3. **Show that the mapping $\Gamma : T \mapsto T^*$ has a fixed point**:
   - Banach Fixed Point Theorem

Schedule

Mean-field approximation for MARL with **large population**

- (45 mins) Non-cooperative games $\Rightarrow$ mean-field game
- (45 mins) Cooperative games $\Rightarrow$ mean-field control

Learning in Mean-field Games

Motivation: A Sequential Auction Game

Figure: An Overview of Taobao Display Advertising System.

Motivation: A Sequential Auction Game

Ad auction problem for advertisers:

- Ad auction: a stochastic game on an ad exchange platform among a large number of players (the advertisers)
- Environment: in each round, a web user requests a page, and then a (Vickrey-type) *second-best-price* auction is run to incentivize advertisers to bid for a slot to display advertisement
- Characteristics:
  - partial information (unknown conversion of clicks)
  - large population

Question: From an individual bidder's perspective, how to bid in this sequential game with a **large** population of competing bidders and **unknown** distributions of the conversion of clicks/rewards?

Motivation: A Sequential Auction Game

Reinforcement Learning

Attempt: the simultaneous-learning-and-decision-making problem in a sequential auction with a large number of homogeneous bidders.

- **Full model** approach: solve it as an N-player reinforcement learning problem
  - Curse of "many agents" when N is large
    - intractable interactions on individual levels
    - computational complexity growths exponentially with respect to the number of players
  - Structure-dependent policies: Lauer and Riedmiller (2000), Qu, Wierman and Li (2019).
  - Provable algorithms for two-player games
  - No theoretical guarantee for general non-zero sum MARL

Motivation: sequential auction game

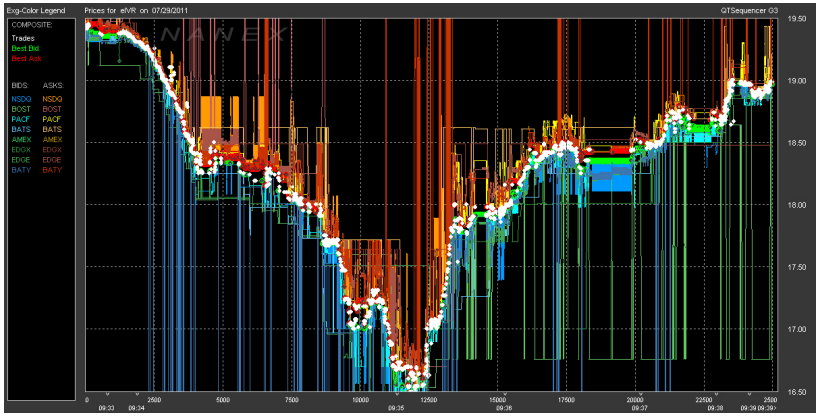
Reinforcement Learning

Attempt: the simultaneous learning and decision-making problem in a sequential auction with a large number of homogeneous bidders.

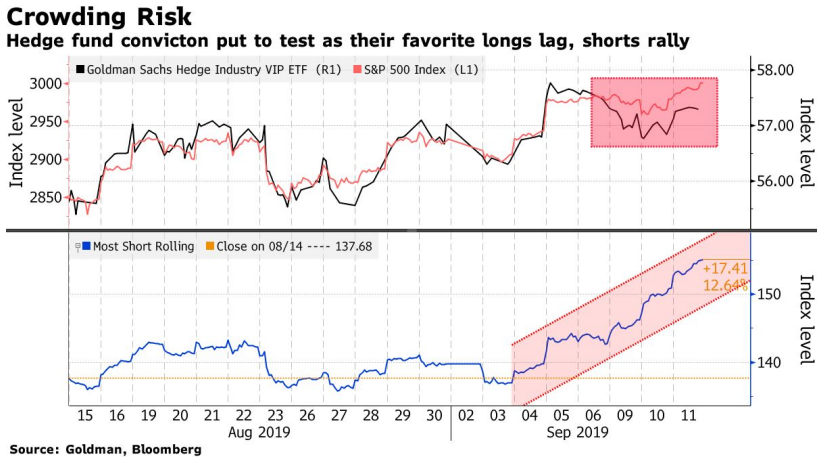
Mean-Field Games

- **Approximation** approach: mean-field approximation
  - When N is large, consider instead the "aggregated" version of the N-player game by letting $N \to \infty$
  - About small interacting individuals, with each player choosing optimal strategy in view of the macroscopic information (mean field)
  - By (f)SLLN, the aggregated version becomes an "approximation" of the N-player game
  - The aggregated version, mean-field approximation, is analytically feasible and easy to learn

Other Applications: HFT and Crowded Shorts



(a) High-frequency Trading



(b) Crowded Shorts

Outline

- **When mean-field approximation works?**
  - **Model set-up for mean-field games (MFG)**
  - Existence and uniqueness of MFG solutions
- How to apply mean-field theory in MARL algorithmic design?
  - Q-learning for GMFG
  - Smoothing and stabilizing techniques
  - Convergence and complexity analysis
  - More general results: policy-based and value-based algorithms

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

subject to $\quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{\boldsymbol{P}} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

subject to $\quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{P} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

subject to $\quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{P} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

subject to $\quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{P} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

subject to $\quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{\boldsymbol{P}} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

N-player Games

N-player game

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right]$$

$$\text{subject to} \quad s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t).$$

- N players, state space $\mathcal{S}$, action space $\mathcal{A}$
- $\boldsymbol{s}_t = (s^1_t, \ldots, s^N_t) \in \mathcal{S}^N$ is the state profile
- $\boldsymbol{a}_t = (a^1_t, \ldots, a^N_t) \in \mathcal{A}^N$ is the action profile
- r^i is the reward function and P^i is the transition kernel of player i
- $(\boldsymbol{s}_t, \boldsymbol{a}_t) \xrightarrow{\boldsymbol{P}} \boldsymbol{s}_{t+1}$
- admissible policy $\pi^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures over $\mathcal{A}$

From N-player Game to MFG

N-player games

$$\begin{aligned} &\text{maximize}_{\pi^i} && V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s}\right] \\ &\text{subject to} && s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t) \end{aligned}$$

- Mean-field approximation will not work for all N-player games
- Need assumptions to embrace **law of large number** and **theory of propagation of chaos**

Assume

1. **Homogeneity**: Agents are identical, indistinguishable and interchangeable
2. **Weak interactions**: player i depends on other agents through empirical measure $(\mu_t^{-i}(\cdot), \alpha_t^{-i}(\cdot)) := \left(\frac{\sum_{j\neq i} I(s_t^j = \cdot)}{N-1}, \frac{\sum_{j\neq i} I(a_t^j = \cdot)}{N-1}\right)$
3. **Local policy**: $a_t^i \sim \pi_t^i(s_t^i)$ here

From N-player Game to MFG

N-player games

$$\begin{aligned} &\text{maximize}_{\pi^i} && V^i(\boldsymbol{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s}\right] \\ &\text{subject to} && s^i_{t+1} \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad a^i_t \sim \pi^i_t(\boldsymbol{s}_t) \end{aligned}$$

- Mean-field approximation will not work for all N-player games
- Need assumptions to embrace **law of large number** and **theory of propagation of chaos**

Assume

1. **Homogeneity**: Agents are identical, indistinguishable and interchangeable
2. **Weak interactions**: player i depends on other agents through empirical measure $(\mu_t^{-i}(\cdot), \alpha_t^{-i}(\cdot)) := \left(\frac{\sum_{j\neq i} I(s_t^j = \cdot)}{N-1}, \frac{\sum_{j\neq i} I(a_t^j = \cdot)}{N-1}\right)$
3. **Local policy**: $a^i_t \sim \pi^i_t(s^i_t)$ here

From N-player Game to MFG

A smaller class of N-player games

$$
\begin{aligned}
& V^i(s^i, \mu^{-i}, \pi^i, \alpha^{-i}) := \\
& \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(s_t^i, \mu_t^{-i}, \pi_t^i, \alpha_t^{-i}) \,\middle|\, (s_0^i, \mu_0^{-i}, \alpha_0^{-i}) = (s^i, \mu^{-i}, \alpha^{-i})\right] \\
\text{subject to} \quad & s_{t+1}^i \sim P^i(s_t^i, \mu_t^{-i}, \pi_t^i, \alpha_t^{-i}), \quad a_t^i \sim \pi_t^i(s_t^i)
\end{aligned}
$$

- Homogeneous agents $\Rightarrow$ look at a representative agent is enough
- A game between agent i and the empirical measure of other agents (μ^{-i}, α^{-i})

When the number of players goes to infinity, view the limit of $(\mu_t^{-i}(\cdot), \alpha_t^{-i}(\cdot)) := \left(\frac{\sum_{j\neq i} I(s_t^j = \cdot)}{N-1}, \frac{\sum_{j\neq i} I(a_t^j = \cdot)}{N-1}\right)$ as population state-action joint distribution L_t

From N-player Game to MFG

A smaller class of N-player games

$$
\begin{aligned}
& V^i(s^i, \mu^{-i}, \pi^i, \alpha^{-i}) := \\
& \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(s_t^i, \mu_t^{-i}, \pi_t^i, \alpha_t^{-i}) \,\middle|\, (s_0^i, \mu_0^{-i}, \alpha_0^{-i}) = (s^i, \mu^{-i}, \alpha^{-i})\right] \\
\text{subject to} \quad & s_{t+1}^i \sim P^i(s_t^i, \mu_t^{-i}, \pi_t^i, \alpha_t^{-i}), \quad a_t^i \sim \pi_t^i(s_t^i)
\end{aligned}
$$

- Homogeneous agents $\Rightarrow$ look at a representative agent is enough
- A game between agent i and the empirical measure of other agents (μ^{-i}, α^{-i})

When the number of players goes to infinity, view the limit of $(\mu_t^{-i}(\cdot), \alpha_t^{-i}(\cdot)) := \left(\frac{\sum_{j\neq i} I(s_t^j = \cdot)}{N-1}, \frac{\sum_{j\neq i} I(a_t^j = \cdot)}{N-1}\right)$ as population state-action joint distribution L_t

General MFG

MFG (Representative agent)

$$V(s, \pi, \{L_t\}_{t=0}^{\infty}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, L_t) | s_0 = s\right]$$

subject to $s_{t+1} \sim P(s_t, a_t, L_t),\ a_t \sim \pi_t(s_t)$

- $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and action of a representative agent at time t
- r is the reward function, P is the transition dynamics
- $L_t \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ is the population state-action distribution at time t, with state marginal μ_t and action marginal α_t
- admissible policy $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ here

General MFG

MFG (Representative agent)

$$V(s, \pi, \{L_t\}_{t=0}^{\infty}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, L_t) | s_0 = s\right]$$

subject to $\quad s_{t+1} \sim P(s_t, a_t, L_t),\ a_t \sim \pi_t(s_t)$

- $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and action of a representative agent at time t
- r is the reward function, P is the transition dynamics
- $L_t \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ is the population state-action distribution at time t, with state marginal μ_t and action marginal α_t
- admissible policy $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ here

General MFG

MFG (Representative agent)

$$V(s, \pi, \{L_t\}_{t=0}^{\infty}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, L_t) | s_0 = s\right]$$

subject to $s_{t+1} \sim P(s_t, a_t, L_t),\ a_t \sim \pi_t(s_t)$

- $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and action of a representative agent at time t
- r is the reward function, P is the transition dynamics
- $L_t \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ is the population state-action distribution at time t, with state marginal μ_t and action marginal α_t
- admissible policy $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ here

Nash Equilibrium in GMFGs

Parallel definition of NE for N-player game

Denote

- Policy profile $\pi = \{\pi_t\}_{t=0}^{\infty}$
- Population profile $L = \{L_t\}_{t=0}^{\infty}$

Definition (NE for GMFGs)

In GMFGs, a policy-population pair $(\pi^\star, L^\star)$ *is called a NE if*

1. *(***Representative agent side***) Fix* $L^\star$*, for any policy* π *and any initial state* $s \in \mathcal{S}$*,*

$$V\left(s, \pi^\star, L^\star\right) \geq V\left(s, \pi, L^\star\right).$$

2. *(***Population side***)* $\mathbb{P}_{s_t, a_t} = L_t^\star$ *for all* $t \geq 0$*, where* $\{s_t, a_t\}_{t=0}^{\infty}$ *is the dynamics under control* $\pi^\star$*, with* $a_t \sim \pi_t^\star(s_t)$*,* $s_{t+1} \sim P(\cdot|s_t, a_t, L^\star)$*.*

Nash Equilibrium in GMFG

Parallel Definitions

Definition (NE for GMFGs)

In GMFGs, a policy-population pair $(\pi^\star, L^\star)$ *is called a NE if*

1. *(***Representative agent side***)* *Fix* $L^\star$*, for any policy* π *and any initial state* $s \in \mathcal{S}$*,*

$$V\left(s, \pi^\star, L^\star\right) \geq V\left(s, \pi, L^\star\right).$$

2. *(***Population side***)* $\mathbb{P}_{s_t, a_t} = L_t^\star$ *for all* $t \geq 0$*, where* $\{s_t, a_t\}_{t=0}^{\infty}$ *is the dynamics under control* $\pi^\star$*, with* $a_t \sim \pi_t^\star(s_t)$*,* $s_{t+1} \sim P(\cdot|s_t, a_t, L^\star)$*.*

Definition (NE for N-player games)

In N-player game, a policy profile $\boldsymbol{\pi}^*$ *is called a NE if*

1. *(***One agent side***)* *Fix* $\boldsymbol{\pi}^{*,-i}$*, for any policy* π^i *and any initial state* $\boldsymbol{s} \in \mathcal{S}^N$*,*

$$V^i\left(\boldsymbol{s}, \boldsymbol{\pi}^*\right) \geq V^i\left(\boldsymbol{s}, (\boldsymbol{\pi}^{*,-i}, \pi^i),\right).$$

2. *(***Population side***)* *Condition* **1** *holds for all agents.*

Nash Equilibrium in GMFG

Stationary Solution

Definition (NE for GMFGs)

In GMFGs, a policy-population pair $(\pi^\star, L^\star)$ *is called a NE if*

1. *(***Representative agent side***) Fix* $L^\star$*, for any policy* π *and any initial state* $s \in \mathcal{S}$*,*

$$V\left(s, \pi^\star, L^\star\right) \geq V\left(s, \pi, L^\star\right).$$

2. *(***Population side***)* $\mathbb{P}_{s_t, a_t} = L_t^\star$ *for all* $t \geq 0$*, where* $\{s_t, a_t\}_{t=0}^{\infty}$ *is the dynamics under control* $\pi^\star$*, with* $a_t \sim \pi_t^\star(s_t)$*,* $s_{t+1} \sim P(\cdot|s_t, a_t, L^\star)$*.*

- **Stationary solution**: If there exists $(L^\star, \pi^\star)$ independent of time
  - For single-agent RL: optimal stationary policy always exists for infinite time horizon problem
  - For game: it is more difficult due to the competition

Outline

- **When mean-field approximation works?**
  - Model set-up for mean-field games (MFG)
  - **Existence and uniqueness of MFG solutions**
- How to apply mean-field theory in MARL algorithmic design?
  - Q-learning for GMFG
  - Smoothing and stabilizing techniques
  - General policy-based and value-based algorithms

Fixed point/Three-step approach

Model is fully known

- Step 1 (Γ_1): given L, solve the stochastic control problem to get $\pi_L^\star$:

$$\begin{aligned} \text{maximize}_\pi \quad & V(s,\pi|L) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t|L)|s_0 = s\right] \\ \text{subject to} \quad & s_{t+1} \sim P(s_t, a_t, L) \end{aligned}$$

$$\Gamma_1(L) = \pi_L^\star$$

- Step 2 (Γ_2): given $\pi_L^\star$, update from L for one time step to get L' following the dynamics

$$\Gamma_2(L, \pi_L^\star) = L'$$

- Step 3: Check whether L' matches L, and repeat

$$\Gamma_2 \circ \Gamma_1(L) = L'$$

Remark. Well-definedness; Algorithm design; Convergence analysis

Existence and Uniqueness

Theorem 1 (Guo, Hu, X. & Zhang, 2019 here)

1. *Under some "small parameter" conditions,* $\Gamma_2 \circ \Gamma_1$ *is contractive;*
2. *For any GMFG, if* $\Gamma_2 \circ \Gamma_1$ *is contractive, then there exists a unique stationary NE. In addition, the three-step approach converges.*

- Uniqueness is in the sense of L
- Explicit model conditions to guarantee "small parameter" conditions
- Parallel results for both stationary and non-stationary MFG

Existence and Uniqueness

Theorem 1 (Guo, Hu, X. & Zhang, 2019 here)

1. *Under some "small parameter" conditions,* $\Gamma_2 \circ \Gamma_1$ *is contractive;*
2. *For any GMFG, if* $\Gamma_2 \circ \Gamma_1$ *is contractive, then there exists a unique stationary NE. In addition, the three-step approach converges.*

- Uniqueness is in the sense of L
- Explicit model conditions to guarantee "small parameter" conditions
- Parallel results for both stationary and non-stationary MFG

Existence and Uniqueness

Theorem 1 (Guo, Hu, X. & Zhang, 2019 here)

1. *Under some "small parameter" conditions,* $\Gamma_2 \circ \Gamma_1$ *is contractive;*
2. *For any GMFG, if* $\Gamma_2 \circ \Gamma_1$ *is contractive, then there exists a unique stationary NE. In addition, the three-step approach converges.*

- Uniqueness is in the sense of L
- Explicit model conditions to guarantee "small parameter" conditions
- Parallel results for both stationary and non-stationary MFG

Existence and Uniqueness

Theorem 1 (Guo, Hu, X. & Zhang, 2019 here)

1. *Under some "small parameter" conditions,* $\Gamma_2 \circ \Gamma_1$ *is contractive;*
2. *For any GMFG, if* $\Gamma_2 \circ \Gamma_1$ *is contractive, then there exists a unique stationary NE. In addition, the three-step approach converges.*

- Uniqueness is in the sense of L
- Explicit model conditions to guarantee "small parameter" conditions
- Parallel results for both stationary and non-stationary MFG

(Partial) References on MFGs

Model fully known

- **Existence**:
  - PDE approach: Huang, Malhamé & Caines (2006), Lasry & Lions (2006)
  - Probability approach: Carmona & Delarue (2013), Carmona & Lacker (2014)
  - Weak solution approach: Lacker (2015)
- **Uniqueness**:
  - Small parameter condition: Huang, Malhamé & Caines (2006)
  - Monotonicity condition: Lasry & Lions (2006)
- **Convergence (value/policy)**: Lacker (2015, 2018), Fischer (2017), Cardaliaguet, Delarue, Lasry, and Lions (2019)
- **Reachability when multiple equilibrium**: Cecchin, Fischer, and Pelino (2018), Delarue and Tchuendom (2018), Nutz, Martin, and Tan (2019)
- **Tutorials**: Guéant, Lasry, and Lions (2011); Daniel Lacker (IPAM Notes, 2018); Carmona & Delarue (Two volume books)

Outline

- When mean-field approximation works?
  - Model set-up for mean-field games (MFG)
  - Existence and uniqueness of MFG solutions
- **How to apply mean-field theory in MARL algorithmic design?**
  - **Q-learning for GMFG**
  - **Smoothing and stabilizing techniques**
  - Convergence and complexity analysis
  - More general results: policy-based and value-based algorithms

Bridge MFG with RL: Finding NE

Three-step approach revisited:

- Step 1: given L, solve the stochastic control problem to get $\pi_L^\star$:

$$\begin{aligned} \text{maximize}_\pi \quad & V(s,\pi,L) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t,L) | s_0 = s\right], \\ \text{subject to} \quad & s_{t+1} \sim P(s_t,a_t,L) \end{aligned}$$

- Step 2: given $\pi_L^\star$, update from L for one time step to get L' following the dynamics
- Step 3: Check whether L' matches L

Bridge MFG with RL: Finding NE

Three-step approach revisited (when P and distr of r are unknown):

- **Step 1** [Inner Iteration]: given L, perform Q-learning with transition $P_L(s'|s,a) := P(s'|s,a,L)$ and reward $r_L(s,a) := r(s,a,L)$ $\Rightarrow Q_L^*(s,a)$
- Step 2 [Outer Iteration]: given $\pi_L^\star = \textbf{argmax-e}(Q_L^*(s,a))$, update from L for one time step to get L' following the dynamics
- Step 3: Check whether L' matches L

Remark: $\pi_L^\star(s) \in \textbf{argmax}_a\ Q_L^\star(s,a)$. When **argmax** is non-unique, replace it with **argmax-e**, which assigns equal probability to the maximizers.

Naive RL Algorithm for GMFG

Algorithm 1 Naive Q-learning for GMFGs

1: **Input**: Initial population state-action pair L_0
2: **for** $k = 0, 1, \cdots$ **do**
3: Given L_k, perform Q-learning to find the Q-function $Q_k^\star(s,a) = Q_{L_k}^\star(s,a)$ of an MDP with dynamics $P_{L_k}(s'|s,a)$ and reward distributions $R_{L_k}(s,a)$.
4: Solve $\pi_k \in \Pi$ with $\pi_k(s) = \textbf{argmax-e}\,(Q_k^\star(s,\cdot))$.
5: Sample $s \sim \mu_k$, where μ_k is the population state marginal of L_k, and obtain L_{k+1} from $\mathcal{G}(s, \pi_k, L_k)$.
6: **end for**

Step 3: Fix L

- $$Q_L^{t+1}(s,a) \leftarrow Q_L^t(s,a) + \beta_t(s,a)\left[r(s,a,L) + \gamma \max_{a'} Q_L^t(s',a') - Q_L^t(s,a)\right],$$
- $Q_L^t \to Q_L^\star$, with
$$Q_L^\star(s,a) := r_L(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_L(s'|s,a) V_L^\star(s').$$

Naive RL Algorithm for GMFG

Algorithm 2 Naive Q-learning for GMFGs

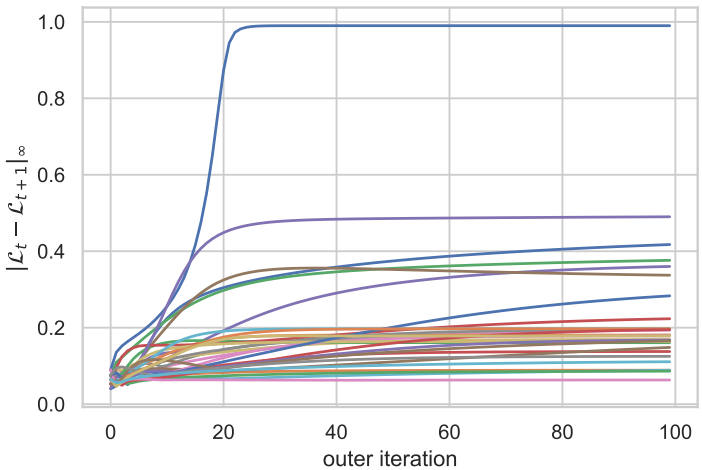
1: **Input**: Initial population state-action pair L_0
2: **for** $k = 0, 1, \cdots$ **do**
3: Given L_k, perform Q-learning to find the Q-function $Q_k^\star(s,a) = Q_{L_k}^\star(s,a)$ of an MDP with dynamics $P_{L_k}(s'|s,a)$ and reward distributions $R_{L_k}(s,a)$.
4: Solve $\pi_k \in \Pi$ with $\pi_k(s) = \textbf{argmax-e}\,(Q_k^\star(s,\cdot))$.
5: Sample $s \sim \mu_k$, where μ_k is the population state marginal of L_k, and obtain L_{k+1} from $\mathcal{G}(s, \pi_k, L_k)$.
6: **end for**

Step 3: Fix L

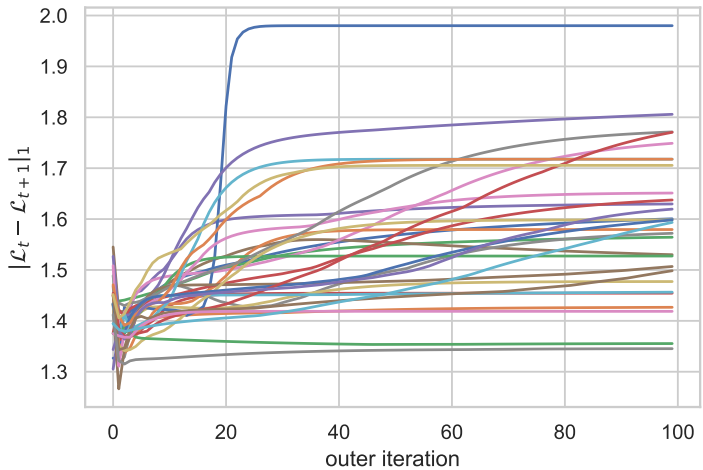
- $$Q_L^{t+1}(s,a) \leftarrow Q_L^t(s,a) + \beta_t(s,a)\left[r(s,a,L) + \gamma \max_{a'} Q_L^t(s',a') - Q_L^t(s,a)\right],$$
- $Q_L^t \to Q_L^\star$, with
$$Q_L^\star(s,a) := r_L(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_L(s'|s,a) V_L^\star(s').$$

Failure of the Naive Algorithm

Failure examples:



(c) Fluctuation in l_∞: $|L_k - L_{k+1}|_\infty$



(d) Fluctuation in l_1: $|L_k - L_{k+1}|_{l_1}$

Figure: Fluctuations of Naive Algorithm (30 sample paths).

Problems in the Naive Algorithm: Approximation Errors

Algorithm 1 Naive Q-learning for GMFGs

1: **Input**: Initial population state-action pair L_0
2: **for** $k = 0, 1, \cdots$ **do**
3: Perform Q-learning to find the Q-function $\overbrace{Q_k^\star(s,a) = Q_{L_k}^\star(s,a)}^{\text{impossible}}$ of an MDP with dynamics $P_{L_k}(s'|s,a)$ and reward distributions $R_{L_k}(s,a)$.
4: Solve $\pi_k \in \Pi$ with $\pi_k(s) = \overbrace{\textbf{argmax-e}}^{\text{unstable}}(Q_k^\star(s,\cdot))$.
5: Sample $s \sim \mu_k$, where μ_k is the population state marginal of L_k, and obtain $\underbrace{L_{k+1}}_{\text{unstable}}$ from $\mathcal{G}(s, \pi_k, L_k)$.
6: **end for**

- Inner iteration: error control
- **argmax-e**: not continuous
- The update from L_k to L_{k+1} is not controlled

Instability of **argmax-e**:

Magnify the Approximation Errors

argmax-e is not continuous:

- $x = (1, 1)$, then **argmax-e**$(x) = (1/2, 1/2)$
- $y = (1, 1 - \epsilon)$, then for any $\epsilon > 0$, **argmax-e**$(y) = (1, 0)$
- $\|$**argmax-e**$(x) -$ **argmax-e**$(y)\|_2 / \|x - y\|_2 = \frac{1}{\sqrt{2}\epsilon}$

Stable Algorithm for GMFG (GMF-Q)

Algorithm 2 Q-learning for GMFGs (GMF-Q)

1: **Input**: Initial L_0, tolerance $\epsilon > 0$.
2: **for** $k = 0, 1, \cdots$ **do**
3: Perform Q-learning for $\boldsymbol{T_k}$ iterations to find the approximate Q-function $\hat{Q}^{\star}_k(s,a) = \hat{Q}^{\star}_{L_k}(s,a)$ of an MDP with dynamics $P_{L_k}(s'|s,a)$ and reward distributions $R_{L_k}(s,a)$.
4: Compute $\pi_k \in \Pi$ with $\pi_k(s) = \textbf{softmax}_c(\hat{Q}^{\star}_k(s,\cdot))$.
5: Sample $s \sim \mu_k$, where μ_k is the population state marginal of L_k, and obtain $\tilde{L}_{k+1}$ from $\mathcal{G}(s, \pi_k, L_k)$.
6: Find $L_{k+1} = \textbf{Proj}_{S_\epsilon}(\tilde{L}_{k+1})$
7: **end for**

1. T_k: carefully chosen in the PAC
2. **argmax-e** to **softmax**: $\textbf{softmax}_c(x)_i = \frac{\exp(cx_i)}{\sum_{j=1}^{n}\exp(cx_j)}$
3. Projection on to S_ϵ: a ϵ-net (finite cover) of L

Outline

- When mean-field approximation works?
  - Model set-up for mean-field games (MFG)
  - Existence and uniqueness of MFG solutions
- **How to apply mean-field theory in MARL algorithmic design?**
  - Q-learning for GMFG
  - Smoothing and stabilizing techniques
  - **Convergence and complexity analysis**
  - More general results: policy-based and value-based algorithms

Convergence and Complexity of GMF-Q

Theorem 2 (Guo, Hu, X. & Zhang, 2019)

Given the same assumptions in the existence and uniqueness theorem, for any specified tolerances $\epsilon,\ \delta > 0$, *set* T_k, c *and* S_ϵ *appropriately. Then with probability at least* $1 - 2\delta$, $W_1(L_{K_\epsilon}, L^\star) = O(\epsilon)$, *and the total number of iterations* $T = \sum_{k=0}^{K_\epsilon - 1} T_k$ *is bounded by*

$$T = O\left(K_\epsilon^{19/3} \left(\log(K_\epsilon/\delta)\right)^{41/3}\right).$$

Here $K_\epsilon := \left\lceil 2\max\left\{(\eta\epsilon)^{-1/\eta}, \log_d(\epsilon/\max\{\mathit{diam}(\mathcal{S})\mathit{diam}(\mathcal{A}), 1\}) + 1\right)\right\}\right\rceil$ *is the number of outer iterations.*

Here W_1 is the ℓ_1 Wasserstein distance.

Outline

- When mean-field approximation works?
  - Model set-up for mean-field games (MFG)
  - Existence and uniqueness of MFG solutions
- **How to apply mean-field theory in MARL algorithmic design?**
  - Q-learning for GMFG
  - Smoothing and stabilizing techniques
  - Convergence and complexity analysis
  - **More general results: policy-based and value-based algorithms**

More General Results

Guo, Hu, X., Zhang (2020)

Key take-away: **"Three-step approach + Smoothing+ Stabalizing"** provides a **meta framework** for learning MFG:

- For inner iterations, any single-agent RL with finite sample bound can be used
  - Value-based algorithms: Q-learning
  - Policy-based algorithms: Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Conservative Policy Iteration (CPI)
- For outer iterations, different choices of smoothing method
  - Boltzmann (softmax)
  - MellowMax: $MM_c(\boldsymbol{x}) = \frac{\log(\frac{1}{n}\sum_{i=1}^{n}\exp(cx_i))}{c}$
  - Momentum: linear combination of L_t and L_{t-1}

More General Results

Guo, Hu, X., Zhang (2020)

Key take-away: **"Three-step approach + Smoothing+ Stabalizing"** provides a **meta framework** for learning MFG:

- For inner iterations, any single-agent RL with finite sample bound can be used
  - Value-based algorithms: Q-learning
  - Policy-based algorithms: Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Conservative Policy Iteration (CPI)
- For outer iterations, different choices of smoothing method
  - Boltzmann (softmax)
  - MellowMax: $MM_c(\boldsymbol{x}) = \frac{\log(\frac{1}{n}\sum_{i=1}^{n}\exp(cx_i))}{c}$
  - Momentum: linear combination of L_t and L_{t-1}

More General Results

Guo, Hu, X., Zhang (2020)

Key take-away: **"Three-step approach + Smoothing+ Stabalizing"** provides a **meta framework** for learning MFG:

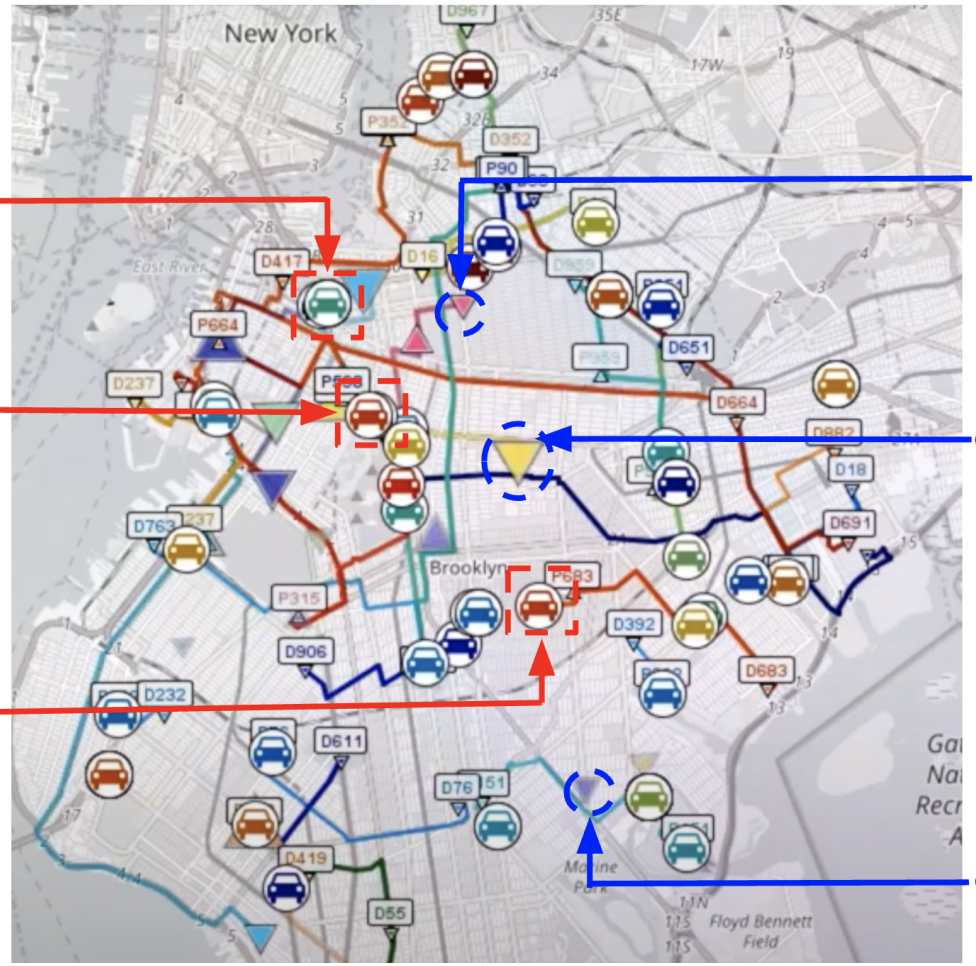
- For inner iterations, any single-agent RL with finite sample bound can be used
  - Value-based algorithms: Q-learning
  - Policy-based algorithms: Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Conservative Policy Iteration (CPI)
- For outer iterations, different choices of smoothing method
  - Boltzmann (softmax)
  - MellowMax: $MM_c(\boldsymbol{x}) = \frac{\log(\frac{1}{n}\sum_{i=1}^{n}\exp(cx_i))}{c}$
  - Momentum: linear combination of L_t and L_{t-1}

(Partial) Related Literature on Learning MFG

- Mean Field Multi-Agent Reinforcement Learning: Yang, Luo, Li, Zhou, Zhang, and Wang (2018))
  - First paper of applying mean-field approximation to MARL
  - Interaction through actions
- Model-based framework (linear-quadratic): Fu, Yang, Chen, Wang (2019)
- Local NE: Subramanian and Mahajan (2019)
- Deep Deterministic Policy Gradient (continuous action): Elie, Pérolat, Laurière, Geist, Pietquin (2019)
- Variants of Q-learning algorithms:
  - Fitted Q: Berkay Anahtarcı, Can Deha Karıksız, Naci Saldi (2019)
  - Regularized Q: Berkay Anahtarci, Can Deha Kariksiz, Naci Saldi (2020)

Learning Mean-Field Controls

Motivating Example: Ride-sharing Order Dispatch



available driver$_1$ location$_1$

available driver$_2$ location$_2$

...

available driver$_N$ location$_N$

order$_1$ = (origin$_1$, destination$_1$)

order$_2$ = (origin$_2$, destination$_2$)

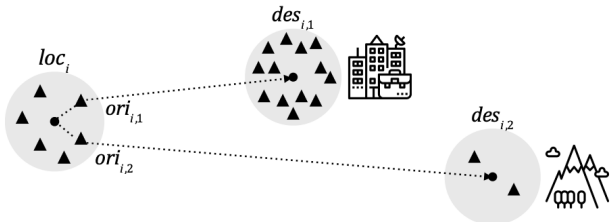
...

order$_M$ = (origin$_M$, destination$_M$)

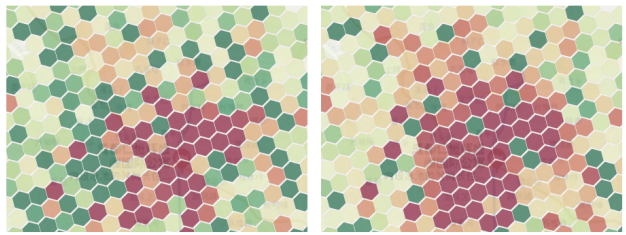
Motivating Example: Ride-sharing Order Dispatch

Real-time (driver,order)-pair match to maximize the combined rewards:

- **Rider**: short waiting time
- **Driver**:
  - short-term reward $\sim$ distance(origin,destination)
  - long-term reward:



- **Global supply-demand balance**:



(a) COD (b) REV

Figure 4: An example of the demand-supply gap in the city center during peak hours. Grids with more drivers are shown in green (in red if opposite) and the gap is proportional to the shade of colors.

Li, Qin, Jiao, Yang, Gong, Wang, Wang, Wu and Ye (2019)

Motivating Example: Ride-sharing Order Dispatch

Key features:

- Cooperative games: optimization from platform's perspective
- Large population: N and M are large
- Interaction with (partially) unknown environment: traffic condition, stochastic and time-dependent demand, and etc
- Demand for (fast) real-time resolution: short waiting time

Solution: MARL with mean-field approximation (Li, Qin, Jiao, Yang, Gong, Wang, Wang, Wu, Ye (2019))

- Cooperative order dispatching algorithm
- Mean-field approximation for local dynamics
- Centralized training with decentralized execution
- Convergence proof of Q-learning

Motivating Example: Ride-sharing Order Dispatch

Key features:

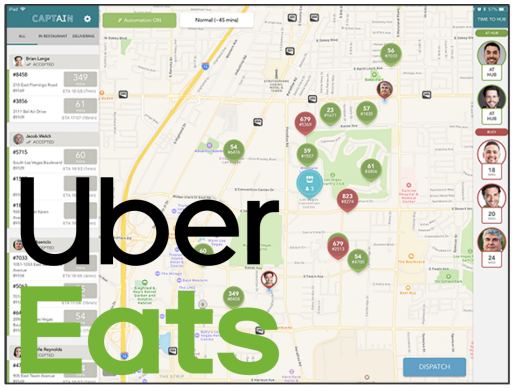
- Cooperative games: optimization from platform's perspective
- Large population: N and M are large
- Interaction with (partially) unknown environment: traffic condition, stochastic and time-dependent demand, and etc
- Demand for (fast) real-time resolution: short waiting time

Solution: MARL with mean-field approximation (Li, Qin, Jiao, Yang, Gong, Wang, Wang, Wu, Ye (2019))

- Cooperative order dispatching algorithm
- Mean-field approximation for local dynamics
- Centralized training with decentralized execution
- Convergence proof of Q-learning

Other Examples

(a) Data Routing



(b) Food Delivery

(c) Autonomous Driving

Outline

- **Mean-field control formulation**
  - **N-player cooperative game**
  - **Pareto optimality condition**
  - $N \to \infty$
- Dynamic Programming Principle (DPP) on the probability measure space
- Q-learning algorithm on the probability measure space

Cooperative MARL

At the beginning of each round t, for each agent i

- State: agent i is at $s_t^i \in \mathcal{S}$
- Action: he/she takes an action $a_t^i \in \mathcal{A}$ such that $a_t^i \sim \pi^i$
- Policy: $\pi^i : \mathcal{S} \to \mathcal{P}(\mathcal{A})$
- Reward: receive a reward $\tilde{r}^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$
- Value function: $V^i(\boldsymbol{s}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s}\right]$
- Transition $s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$: the state will change to s_{t+1}^i according to a transition probability function $P^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$

Pareto Optimality Condition

Pareto optimality

If $\boldsymbol{\pi}^*$ is Pareto optimal, then there does not exist $\boldsymbol{\pi}$ such that

- $V^i(\boldsymbol{s}, \boldsymbol{\pi}) \geq V^i(\boldsymbol{s}, \boldsymbol{\pi}^*)$ for all i and $\boldsymbol{s}$
- There exist j such that $V^j(\boldsymbol{s}, \boldsymbol{\pi}) > V^j(\boldsymbol{s}, \boldsymbol{\pi}^*)$ for all $\boldsymbol{s}$

- No individual can be better off without making at least one individual worse off
- Central controller's solution is pareto optimal

Cooperative MARL

Central controller

$$V(\boldsymbol{s}, \boldsymbol{\pi}) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s} \right]$$

subject to $s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad (i = 1, 2, \cdots, N)$

- Cooperative game $\Rightarrow$ agreement among agents $\Rightarrow$ auxiliary central controller
- Central controller:
  - Observe joint state $\boldsymbol{s}_t$
  - Decide $a_t^i \sim \pi_t^i(\boldsymbol{s}_t)$ for each agent i from a global perspective
  - Observe joint reward $\tilde{\boldsymbol{r}}_t$ as feedback
  - $\boldsymbol{P}$ and the distribution of $\tilde{\boldsymbol{r}}$ are unknown
- **Curse of dimensionality** Vanilla Q-learning algorithm for central controller

$$\Omega\left(\text{poly}\left((|\mathcal{S}||\mathcal{A}|)^N \cdot \frac{N}{\epsilon} \cdot \ln(\frac{1}{\delta\epsilon}) \right) \right).$$

Cooperative MARL

Central controller

$$V(\boldsymbol{s}, \boldsymbol{\pi}) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s} \right]$$

$$\text{subject to} \quad s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad (i = 1, 2, \cdots, N)$$

- Cooperative game $\Rightarrow$ agreement among agents $\Rightarrow$ auxiliary central controller
- Central controller:
  - Observe joint state $\boldsymbol{s}_t$
  - Decide $a_t^i \sim \pi_t^i(\boldsymbol{s}_t)$ for each agent i from a global perspective
  - Observe joint reward $\tilde{\boldsymbol{r}}_t$ as feedback
  - $\boldsymbol{P}$ and the distribution of $\tilde{\boldsymbol{r}}$ are unknown
- **Curse of dimensionality** Vanilla Q-learning algorithm for central controller

$$\Omega\left(\text{poly}\left((|\mathcal{S}||\mathcal{A}|)^N \cdot \frac{N}{\epsilon} \cdot \ln\left(\frac{1}{\delta\epsilon}\right) \right) \right).$$

Cooperative MARL

Central controller

$$V(\boldsymbol{s}, \boldsymbol{\pi}) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s} \right]$$

$$\text{subject to} \quad s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t), \quad (i = 1, 2, \cdots, N)$$

- Cooperative game $\Rightarrow$ agreement among agents $\Rightarrow$ auxiliary central controller
- Central controller:
  - Observe joint state $\boldsymbol{s}_t$
  - Decide $a_t^i \sim \pi_t^i(\boldsymbol{s}_t)$ for each agent i from a global perspective
  - Observe joint reward $\tilde{\boldsymbol{r}}_t$ as feedback
  - $\boldsymbol{P}$ and the distribution of $\tilde{\boldsymbol{r}}$ are unknown
- **Curse of dimensionality** Vanilla Q-learning algorithm for central controller

$$\Omega\left(\text{poly}\left((|\mathcal{S}||\mathcal{A}|)^N \cdot \frac{N}{\epsilon} \cdot \ln(\frac{1}{\delta\epsilon}) \right) \right).$$

Mean-field Approximation

Assumptions

- **Homogeneity**: Agents are identical and interchangeable.
- **Weak interaction**: Each agent depends on all other agents only through the empirical distributions of states and actions.

When $N \to \infty$, this becomes a **mean-field control** problem.

Mean-field Approximation

Mean-field control

$$\begin{array}{ll} \sup_\pi & V(\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(s_t, a_t, \mu_t) \,\middle|\, s_0 \sim \mu, \mu_0 = \mu\right] \\ \text{subject to} & s_{t+1} \sim P(s_t, a_t, \mu_t). \end{array}$$

- $s_t \in \mathcal{S}$: dynamics of representative agent with randomized initial state $s_0 \sim \mu$
- $a_t \in \mathcal{A}$: action
- $\mu_t = \text{Law}(s_t) \in \mathcal{P}(\mathcal{S})$: limit of $\mu_t^N(\cdot) = \sum_{j=1}^N \frac{1(s_t^j = \cdot)}{N}$
- $\tilde{r}$: reward of the representative agent
- Admissible policy $\pi_t(s, \mu) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$

Mean-field Approximation

Mean-field control

$$\begin{aligned} \sup_{\pi} \quad & V(\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(s_t,a_t,\mu_t) \,\middle|\, s_0 \sim \mu, \mu_0 = \mu\right] \\ \text{subject to} \quad & s_{t+1} \sim P(s_t,a_t,\mu_t). \end{aligned}$$

- $s_t \in \mathcal{S}$: dynamics of representative agent with randomized initial state $s_0 \sim \mu$
- $a_t \in \mathcal{A}$: action
- $\mu_t = \text{Law}(s_t) \in \mathcal{P}(\mathcal{S})$: limit of $\mu_t^N(\cdot) = \sum_{j=1}^{N} \frac{1(s_t^j = \cdot)}{N}$
- $\tilde{r}$: reward of the representative agent
- Admissible policy $\pi_t(s,\mu) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$

Mean-field Approximation

Mean-field control

$$\begin{aligned} \sup_{\pi} \quad & V(\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(s_t,a_t,\mu_t) \,\middle|\, s_0 \sim \mu, \mu_0 = \mu\right] \\ \text{subject to} \quad & s_{t+1} \sim P(s_t,a_t,\mu_t). \end{aligned}$$

- $s_t \in \mathcal{S}$: dynamics of representative agent with randomized initial state $s_0 \sim \mu$
- $a_t \in \mathcal{A}$: action
- $\mu_t = \text{Law}(s_t) \in \mathcal{P}(\mathcal{S})$: limit of $\mu_t^N(\cdot) = \sum_{j=1}^{N} \frac{1(s_t^j = \cdot)}{N}$
- $\tilde{r}$: reward of the representative agent
- Admissible policy $\pi_t(s,\mu) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$

Mean-field Approximation

Mean-field control

$$\begin{aligned} \sup_{\pi} \quad & V(\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(s_t,a_t,\mu_t) \,\middle|\, s_0 \sim \mu, \mu_0 = \mu\right] \\ \text{subject to} \quad & s_{t+1} \sim P(s_t,a_t,\mu_t). \end{aligned}$$

- $s_t \in \mathcal{S}$: dynamics of representative agent with randomized initial state $s_0 \sim \mu$
- $a_t \in \mathcal{A}$: action
- $\mu_t = \text{Law}(s_t) \in \mathcal{P}(\mathcal{S})$: limit of $\mu_t^N(\cdot) = \sum_{j=1}^{N} \frac{1(s_t^j = \cdot)}{N}$
- $\tilde{r}$: reward of the representative agent
- Admissible policy $\pi_t(s,\mu) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$

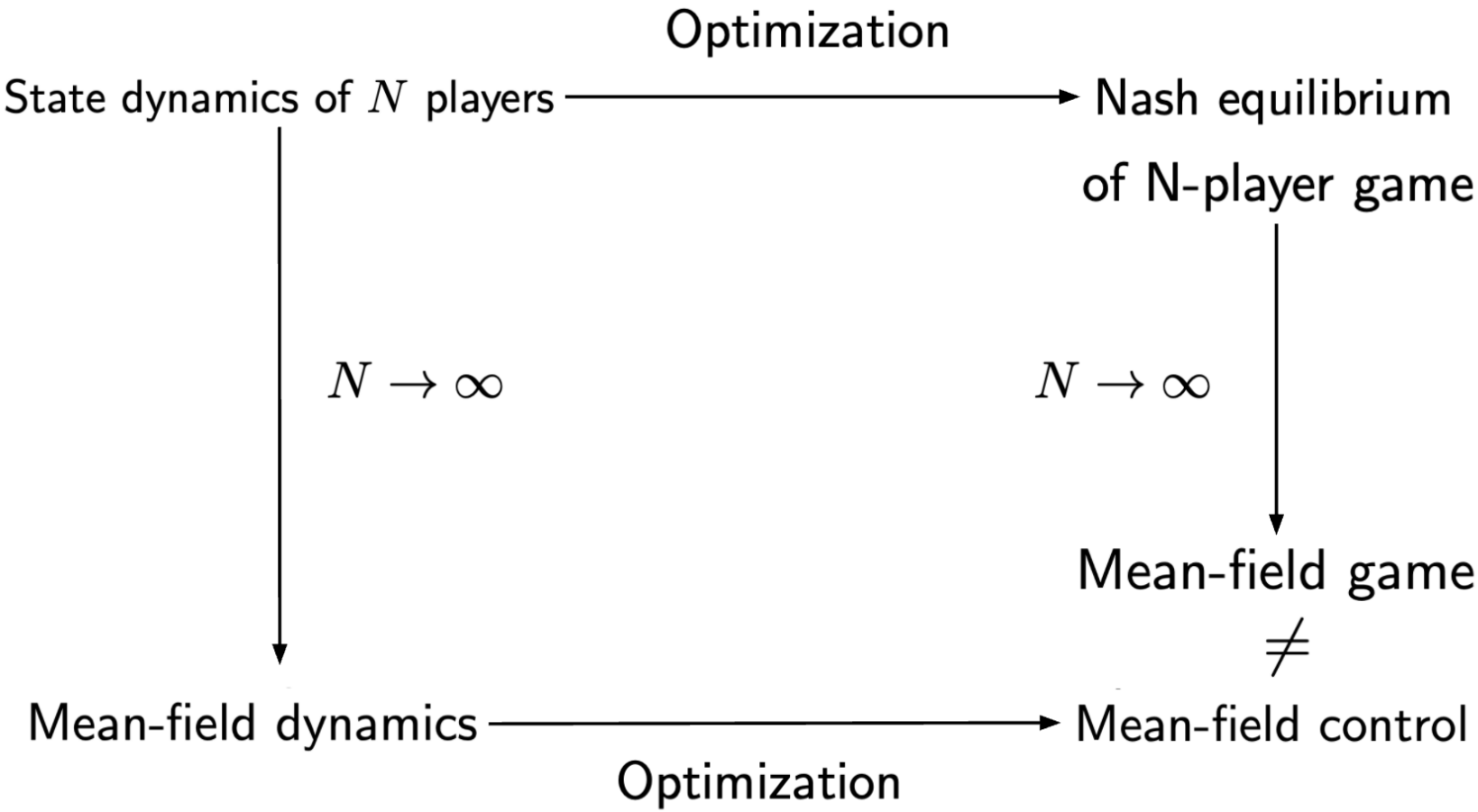
Mean-field Approximation

Mean-field control

$$\begin{aligned} \sup_{\pi} \quad & V(\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(s_t,a_t,\mu_t) \,\middle|\, s_0 \sim \mu, \mu_0 = \mu\right] \\ \text{subject to} \quad & s_{t+1} \sim P(s_t,a_t,\mu_t). \end{aligned}$$

- $s_t \in \mathcal{S}$: dynamics of representative agent with randomized initial state $s_0 \sim \mu$
- $a_t \in \mathcal{A}$: action
- $\mu_t = \text{Law}(s_t) \in \mathcal{P}(\mathcal{S})$: limit of $\mu_t^N(\cdot) = \sum_{j=1}^{N} \frac{1(s_t^j = \cdot)}{N}$
- $\tilde{r}$: reward of the representative agent
- Admissible policy $\pi_t(s,\mu) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$

Comparison between MFG and MFC



Carmona, Delarue & Lachapelle (2013)

Outline

- Mean-field control formulation
  - N-player cooperative game
  - Pareto optimality condition
  - $N \to \infty$
- **Dynamic Programming Principle (DPP) on the probability measure space**
- Q-learning algorithm on the probability measure space

Time Consistency (Dynamic Programming Principle)

Dynamic Programming is an umbrella encompassing many algorithms (single agent RL and MARL)

- Value-based method: Q-learning
- Policy-based method: Actor-Critic Algorithm
- Planning: Value Iteration or Policy Iteration

Dynamic Programming does not hold **for free** with infinite number of players: need to work with the **correct "state" and "action" spaces**.

- Pham and Wei (2016), Pham and Wei (2017), and Wu and Zhang (2019) for value function

Time Consistency (Dynamic Programming Principle)

Dynamic Programming is an umbrella encompassing many algorithms (single agent RL and MARL)

- Value-based method: Q-learning
- Policy-based method: Actor-Critic Algorithm
- Planning: Value Iteration or Policy Iteration

Dynamic Programming does not hold **for free** with infinite number of players: need to work with the **correct "state" and "action" spaces**.

- Pham and Wei (2016), Pham and Wei (2017), and Wu and Zhang (2019) for value function

Q-function on the probability measure space

Define the Q-function for MFC:

$$Q(\mu, h) := \underbrace{\mathbb{E}\left[\tilde{r}(s_0, a_0, \mu) \middle| s_0 \sim \mu, a_0 \sim h\right]}_{\text{Reward of taking } a_0 \sim h}$$
$$+ \underbrace{\mathbb{E}_{s_1 \sim P(s_0, a_0, \mu)}\left[\sum_{t=1}^{\infty} \gamma^t \tilde{r}(s_t, a_t, \mu_t) \middle| a_t \sim \pi_t^*\right]}_{\text{Reward of playing optimal afterwards } a_t \sim \pi_t^*}$$

- local policy $h : \mathcal{S} \to \mathcal{P}(\mathcal{A})$
- $V^*(\mu) = \max_h Q(\mu, h)$

Bellman Equation for Integrated Q-function

Theorem 3 (Bellman Equation for IQ (Gu, Guo, Wei and X., 2019))

For any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$,

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

- $\mathcal{H} := \{h : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$: set of "local" policies
- **Aggregated Dynamics**
  - $\Phi(\mu, h) := \sum_{s,a} P(s, \mu, a)\mu(s)h(s, a)$: aggregated dynamics
  - $\mu_{t+1} = \Phi(\mu_t, h)$: distribution at time $t + 1$, **flow property**
- **Aggregated Reward**: $r(\mu, h) := \sum_{s,a} \mu(s)h(s, a)\tilde{r}(s, a, \mu)$

Bellman Equation for Integrated Q-function

Theorem 3 (Bellman Equation for IQ (Gu, Guo, Wei and X., 2019))

For any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$,

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

- $\mathcal{H} := \{h : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$: set of "local" policies
- **Aggregated Dynamics**
  - $\Phi(\mu, h) := \sum_{s,a} P(s, \mu, a)\mu(s)h(s, a)$: aggregated dynamics
  - $\mu_{t+1} = \Phi(\mu_t, h)$: distribution at time $t + 1$, **flow property**
- **Aggregated Reward**: $r(\mu, h) := \sum_{s,a} \mu(s)h(s, a)\tilde{r}(s, a, \mu)$

Bellman Equation for Integrated Q-function

Theorem 3 (Bellman Equation for IQ (Gu, Guo, Wei and X., 2019))

For any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$,

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

- $\mathcal{H} := \{h : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$: set of "local" policies
- **Aggregated Dynamics**
  - $\Phi(\mu, h) := \sum_{s,a} P(s, \mu, a)\mu(s)h(s, a)$: aggregated dynamics
  - $\mu_{t+1} = \Phi(\mu_t, h)$: distribution at time $t + 1$, **flow property**
- **Aggregated Reward**: $r(\mu, h) := \sum_{s,a} \mu(s)h(s, a)\tilde{r}(s, a, \mu)$

Bellman Equation for Integrated Q-function

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

- $\mathcal{H}$ is the **minimum** space under which the Bellman equation holds
- Similar results for finite-time horizon; Bellman equation for value function; Including law of actions

Outline

- Mean-field control formulation
  - N-player cooperative game
  - Pareto optimality condition
  - $N \to \infty$
- Dynamic Programming Principle (DPP) on the probability measure space
- **Q-learning algorithm on the probability measure space**

Q-learning Algorithm on Probability Measure Space

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

Properties of $Q(\mu, h)$:

- Defined on the probability measure space
- Continuous μ and h even when s and a are discrete
- Deterministic dynamics: $\mu_{t+1} = \Phi(\mu_t, h)$

$\Longrightarrow$ Q-learning algorithm with **continuous state-action** and **deterministic dynamics**

Q-learning Algorithm on Probability Measure Space

$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

Properties of $Q(\mu, h)$:

- Defined on the probability measure space
- Continuous μ and h even when s and a are discrete
- Deterministic dynamics: $\mu_{t+1} = \Phi(\mu_t, h)$

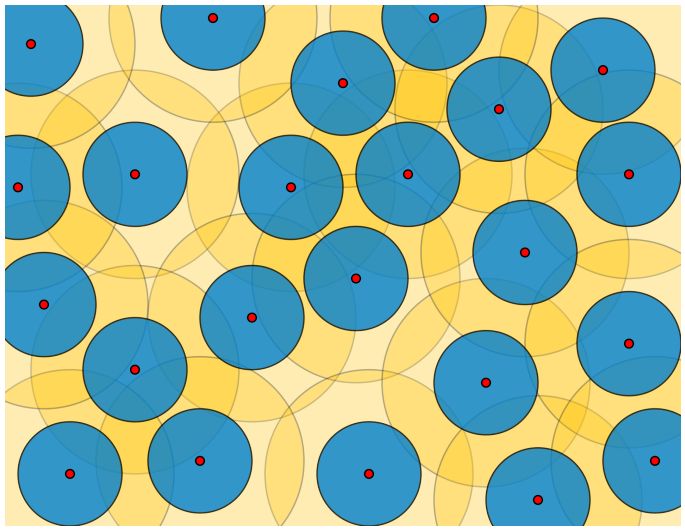
$\Longrightarrow$ Q-learning algorithm with **continuous state-action** and **deterministic dynamics**

Q-learning Algorithm on Probability Measure Space

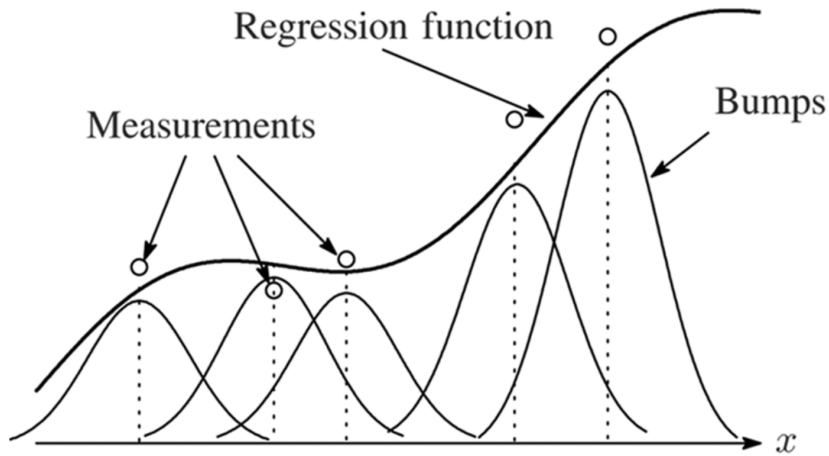
$$Q(\mu, h) \quad = \quad r(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h').$$

- **Continuous state space**: kernel regression
- **Continuous action space**: only optimize on a finite action space
- **Deterministic dynamic**: take advantage of it to reduce the sample complexity

ϵ-net and kernel regression



(d) ϵ-net



(e) Kernel regression

- $\mathcal{C}_\epsilon$: ϵ-net on $\mathcal{C} := \mathcal{P}(\mathcal{S}) \times \mathcal{H}$
- $\mathcal{H}_\epsilon$: induced ϵ-net on $\mathcal{H}$
- Kernel regression on $\mathcal{C}_\epsilon$

Approximated Bellman Operator

The original **Bellman operator** $B : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for the problem is

$$(B\,q)(c) := r(c) + \gamma \sup_{\tilde{h} \in \mathcal{H}} q(\Phi(c), \tilde{h}).$$

- $c \in \mathcal{C} := \mathcal{P}(\mathcal{S}) \times \mathcal{H}$

To facilitate the algorithm design, we introduce an **approximated Bellman operator** $B_K : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ such that

$$(B_K\,q)(c^i) = r(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q(\Phi(c^i), \tilde{h}),$$

- Γ_K: kernel operator

Approximated Bellman Operator

The original **Bellman operator** $B : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for the problem is

$$(B\,q)(c) := r(c) + \gamma \sup_{\tilde{h} \in \mathcal{H}} q(\Phi(c), \tilde{h}).$$

- $c \in \mathcal{C} := \mathcal{P}(\mathcal{S}) \times \mathcal{H}$

To facilitate the algorithm design, we introduce an **approximated Bellman operator** $B_K : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ such that

$$(B_K\,q)(c^i) = r(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q(\Phi(c^i), \tilde{h}),$$

- Γ_K: kernel operator

Sketch of the Algorithm

The algorithm consists of 2 steps:

1. **Get sampled data for an ϵ-net**
   - Initial μ, $\frac{\epsilon}{2}$-net $\mathcal{C}_{\epsilon/2}$, exploration policy π taking actions from $\mathcal{H}_{\epsilon/2}$
   - At the current state μ_t, act h_t according to π, observe $\mu_{t+1} = \Phi(\mu_t, h_t)$ and $r_t = r(\mu_t, h_t)$
2. **Find the fixed point of B_K**

$$q_{l+1}(\mu_t, h_t) \leftarrow \left(r(\mu_t, h_t) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q_l(\Phi(\mu_t, h_t), \tilde{h}) \right)$$

Comment: Visit $\mathcal{C}_{\epsilon/2}$ **once** is enough $\Rightarrow$ Low sample-complexity and efficient

Sketch of the Algorithm

The algorithm consists of 2 steps:

1. **Get sampled data for an ϵ-net**
   - Initial μ, $\frac{\epsilon}{2}$-net $\mathcal{C}_{\epsilon/2}$, exploration policy π taking actions from $\mathcal{H}_{\epsilon/2}$
   - At the current state μ_t, act h_t according to π, observe $\mu_{t+1} = \Phi(\mu_t, h_t)$ and $r_t = r(\mu_t, h_t)$
2. **Find the fixed point of B_K**

$$q_{l+1}(\mu_t, h_t) \leftarrow \left(r(\mu_t, h_t) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q_l(\Phi(\mu_t, h_t), \tilde{h}) \right)$$

Comment: Visit $\mathcal{C}_{\epsilon/2}$ **once** is enough $\Rightarrow$ Low sample-complexity and efficient

Sketch of the Algorithm

The algorithm consists of 2 steps:

1. **Get sampled data for an ϵ-net**
   - Initial μ, $\frac{\epsilon}{2}$-net $\mathcal{C}_{\epsilon/2}$, exploration policy π taking actions from $\mathcal{H}_{\epsilon/2}$
   - At the current state μ_t, act h_t according to π, observe $\mu_{t+1} = \Phi(\mu_t, h_t)$ and $r_t = r(\mu_t, h_t)$
2. **Find the fixed point of B_K**

$$q_{l+1}(\mu_t, h_t) \leftarrow \left(r(\mu_t, h_t) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q_l(\Phi(\mu_t, h_t), \tilde{h}) \right)$$

Comment: Visit $\mathcal{C}_{\epsilon/2}$ **once** is enough $\Rightarrow$ Low sample-complexity and efficient

Convergence and Complexity Results

Theorem(Gu, Guo, Wei, X. 2020)

Assume mild assumptions, for any $\epsilon' > 0$, under the ϵ'-greedy policy, with probability $1 - \delta$, for any initial state distribution μ and ϵ-net, after

$$\Omega(\text{poly}((1/\epsilon) \cdot (1/\epsilon') \cdot \log(1/\delta)))$$

samples, MFC-K-Q converges linearly to some function $\hat{Q}_{\mathcal{C}_\epsilon}$; and the sup distance between $\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}$ and $Q_{\mathcal{C}}$ is upper bounded by $C\epsilon$, where C is a constant independent of ϵ, ϵ' and δ.

Comment: There are 3 sources of the approximation error:

- kernel regression
- discretized action space
- estimated data (for both dynamics and rewards)

Comparison of Complexity Results

Work	MFC/N-player	Sample Complexity Guarantee
Our work	MFC	$\Omega(T_{cov} \cdot \log(1/\delta))$
Carmona et al., 2019	MFC	$\Omega((T_{cov} \cdot \log(1/\delta))^{l} \cdot \text{poly}(\log(1/(\delta\epsilon))/\epsilon))$
Vanilla N-player	N-player	$\Omega(\text{poly}((\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{N} \cdot \log(1/(\delta\epsilon)) \cdot N/\epsilon))$
Qu & Li, 2019	N-player	$\Omega(\text{poly}((\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{f(\log(1/\epsilon))} \cdot \log(1/\delta) \cdot N/\epsilon))$

Table: Comparison of algorithms

- Here T_{cov} is the covering time of the exploration policy.
- $l = \max\{3 + 1/\kappa, 1/(1-\kappa)\} > 4$ for some $\kappa \in (0.5, 1)$.
- $f(\log(1/\epsilon))$ is a structure-dependent quantity and can scale as N when agents are not interacting locally.

Mean-Field Theory for Machine Learning

Mean-Field Theory for Machine Learning

On the theory front

- Theoretical development is (sometimes) easier on the lifted probability measure space
- Address the issues of curse of dimensionality

Applications

- **Training Neural Network**: Mei, Montenari, and Nguyen (2018); E, Han and Li (2018); Sirignano and Spiliopoulos (2019); Hu, Ren, Siska, and Szpruch (2019); Bo, Capponi and Liao (2019); Lu, Ma, Lu, Lu, and Ying (2020); Wojtowytsch and E (2020)
- **GAN**: Cao, Guo Lauriere (2019), Lin, Fung, Li, Nurbekyan, and Osher (2020); Conforti, Kazeykina, and Ren (2020); Domingo-Enrich, Jelassi, Mensch, Rotskoff, and Bruna (2020)
- **MARL**: (as mentioned before)

Mean-Field Theory for Machine Learning

On the theory front

- Theoretical development is (sometimes) easier on the lifted probability measure space
- Address the issues of curse of dimensionality

Applications

- **Training Neural Network**: Mei, Montenari, and Nguyen (2018); E, Han and Li (2018); Sirignano and Spiliopoulos (2019); Hu, Ren, Siska, and Szpruch (2019); Bo, Capponi and Liao (2019); Lu, Ma, Lu, Lu, and Ying (2020); Wojtowytsch and E (2020)
- **GAN**: Cao, Guo Lauriere (2019), Lin, Fung, Li, Nurbekyan, and Osher (2020); Conforti, Kazeykina, and Ren (2020); Domingo-Enrich, Jelassi, Mensch, Rotskoff, and Bruna (2020)
- **MARL**: (as mentioned before)

Thank you!

References: Mean-field Games

- Large Population Stochastic Dynamic Games: Closed-loop McKean-Vlasov Systems and the Nash Certainty Equivalence Principle (Huang, Malhamé, & Caines, 2006)
- Mean Field Games (Lasry & Lions, 2007)
- Mean Field Games and Applications (Guéant, Lasry & Lions, 2011)
- Probabilistic Theory of Mean Field Games with Applications I&II (Carmona & Delarue, 2018)
- Probabilistic Analysis of Mean-field Games (Carmona & Delarue, 2013)
- A Probabilistic Weak Formulation of Mean Field Games and Applications (Carmona & Lacker, 2014)
- A General Characterization of the Mean Field Limit for Stochastic Differential Games (Lacker, 2014)

References: Mean-field Games

- On the Convergence of Closed-loop Nash Equilibria to the Mean Field Game Limit (Lacker, 2018)
- Mean Field Games via Controlled Martingale Problems: Existence of Markovian Equilibria (Lacker, 2018)
- On the Connection Between Symmetric n-player Games and Mean Field games (Fischer, 2017)
- The Master Equation and the Convergence Problem in Mean Field Games (Cardaliaguet, Delarue, Lasry, and Lions, 2015)
- On the Convergence Problem in Mean Field Games: A Two State Model without Uniqueness (Cecchin, Pra, Fischer, and Pelino, 2018)
- Selection of Equilibria in a Linear quadratic Mean-field Game (Delarue and Tchuendom, 2018)
- Convergence to the Mean Field Game Limit: A Case Study (Nutz, Martin and Tan, 2019)

References: Mean-field Games

- Mean Field Multi-agent Reinforcement Learning (Yang, Luo, Li, Zhou, Zhang, and Wang, 2018)
- Actor-Critic Provably Finds Nash Equilibria of Linear-Quadratic Mean-Field Games (Fu, Yang, Chen, and Wang, 2019)
- Mean Field Equilibria of Dynamic Auctions with Learning (Iyer, Johari and Sundararajan, 2014)
- Reinforcement Learning in Stationary Mean-field Games (Subramanian and Mahajan, 2019)
- Approximate Fictitious Play for Mean Field Games (Elie, Pérolat, Laurière, Geist, Pietquin, 2019)
- Fitted Q-Learning in Mean-field Games: (Anahtarcı, Deha Karıksız, Saldi, 2019)
- Q-Learning in Regularized Mean-field Games: (Anahtarcı, Deha Karıksız, Saldi, 2020)

References: Mean-field Controls

- Efficient Ridesharing Order Dispatching with Mean Field Multi-agent Reinforcement learning (Li, Qin, Jiao, Yang, Gong, Wang, Wang, Wu and Ye, 2019)
- Dynamic Programming for Optimal Control of Stochastic McKean–Vlasov Dynamics (Pham and Wei, 2017)
- Bellman Equation and Viscosity Solutions for Mean-field Stochastic Control Problem (Pham and Wei, 2018)
- Viscosity Solutions to Parabolic Master Equations and McKean–Vlasov SDEs with Closed-loop Controls (Zhang and Wu, 2020)
- Model-Free Mean-Field Reinforcement Learning: Mean-Field MDP and Mean-Field Q-Learning (Carmona, Laurière and Tan, 2019)
- Scalable Reinforcement Learning of Localized Policies for Multi-Agent Networked Systems (Qu, Wierman, Li, 2020)

References: Mean-field Theory for Deep Learning

- A Mean Field View of the Landscape of Two-Layers Neural Networks (Mei, Montenari, and Nguyen, 2018)
- Mean Field Analysis of Neural Networks: A Law of Large Numbers (Sirignano and Spiliopoulos, 2019)
- Mean-Field Langevin Dynamics and Energy Landscape of Neural Networks (Hu, Ren, Siska and Szpruch, 2020)
- Relaxed Control and Gamma-Convergence of Stochastic Optimization Problems with Mean Field (Bo, Capponi and Liao, 2019)
- A Mean-field Analysis of Deep ResNet and Beyond: Towards Provable Optimization Via Overparameterization From Depth (Lu, Ma, Lu, Lu, and Ying ,2020)
- Can Shallow Neural Networks Beat the Curse of Dimensionality? A mean field training perspective (Wojtowytsch and E, 2020)
- Connecting GANs, MFGs, and OT (Cao, Guo and Laurière, 2020)
- APAC-Net: Alternating the Population and Agent Control via Two Neural Networks to Solve High-Dimensional Stochastic Mean Field Games (Lin, Fung, Li, Nurbekyan, Osher, 2020)
- A Mean-field Analysis of Two-player Zero-sum Games (Jelassi, Mensch, Rotskoff, and Bruna, 2020)

References

- Guo, Hu, X., and Zhang (2019).
  Learning Mean-Field Games.
  NeurIPS 2019.
- Guo, Hu, X., and Zhang (2020).
  A General Framework for Learning Mean-Field Games.
  https://arxiv.org/pdf/2003.06069.pdf.
- Gu, Guo, Wei, X., (2019).
  Dynamic Programming Principles for Learning Mean-Field Controls.
  arxiv.org/abs/1911.07314.
- Gu, Guo, Wei, X., (2020).
  Q-Learning Algorithm for Mean-Field Controls, with Convergence and Complexity Analysis.
  https://arxiv.org/pdf/2002.04131.pdf.

Local Policy vs General Policy

For player i,

- Local policy: $\pi^i(s^i) : \mathcal{X} \to \mathcal{P}(\mathcal{A})$
  - Individual agent learns NE: population state distribution is not observable
  - Example: sequential ad auction
- General (non-local) policy: $\pi^i(\boldsymbol{s}) : \mathcal{X}^N \to \mathcal{P}(\mathcal{A})$
  - Individual agent learns NE: population states are observable
  - Platform coordinates agents to learn NE: population state distribution is observable
  - Example: routing recommendation
  - Our framework also works under general policy

back

Local Policy vs General Policy

For representative agent,

- Local policy: $\pi(s) : \mathcal{X} \to \mathcal{P}(\mathcal{A})$
  - Individual agent learns NE: population state distribution is not observable
  - Example: sequential ad auction
- General (non-local) policy: $\pi(s, \mu) : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{A})$
  - Individual agent learns NE: population states are observable
  - Platform coordinates agents to learn NE: population state distribution is observable
  - Example: routing recommendation
  - Our framework also works under general policy

back

"Small parameter" condition

Assumption 1

There exists a constant $d_1 \geq 0$, *such that for any* $\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{L}}' \in \{\mathcal{P}(\mathcal{S} \times \mathcal{A})\}_{t=0}^{\infty}$,

$$D(\Gamma_1(\boldsymbol{\mathcal{L}}), \Gamma_1(\boldsymbol{\mathcal{L}}')) \leq d_1 \mathcal{W}_1(\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{L}}'), where$$

$$D(\boldsymbol{\pi}, \boldsymbol{\pi}') := \sup_{s \in \mathcal{S}} \mathcal{W}_1(\boldsymbol{\pi}(s), \boldsymbol{\pi}'(s)), \mathcal{W}_1(\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{L}}') := \sup_{t \in \mathbb{N}} W_1(\mathcal{L}_t, \mathcal{L}'_t),$$

and W_1 *is the* ℓ_1*-Wasserstein distance.*

Assumption 2

There exist constants $d_2,\ d_3 \geq 0$, *such that for any admissible policy sequences* $\boldsymbol{\pi}, \boldsymbol{\pi}^1, \boldsymbol{\pi}^2$ *and joint distribution sequences* $\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{L}}^1, \boldsymbol{\mathcal{L}}^2$,

$$\mathcal{W}_1(\Gamma_2(\boldsymbol{\pi}^1, \boldsymbol{\mathcal{L}}), \Gamma_2(\boldsymbol{\pi}^2, \boldsymbol{\mathcal{L}})) \leq d_2 D(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2),$$

$$\mathcal{W}_1(\Gamma_2(\boldsymbol{\pi}, \boldsymbol{\mathcal{L}}^1), \Gamma_2(\boldsymbol{\pi}, \boldsymbol{\mathcal{L}}^2)) \leq d_3 \mathcal{W}_1(\boldsymbol{\mathcal{L}}^1, \boldsymbol{\mathcal{L}}^2).$$

Assume

$$d_1 d_2 + d_3 < 1.$$