Learning Mean-Field Games

Xin Guo [1] **Anran Hu** [1] **Renyuan Xu** [1] **Junzi Zhang** [2]

Abstract

In this paper, we consider the problem of simultaneous learning and decision-making in a stochastic game setting with a large population. We formulate this problem as a generalized mean-field game (GMFG). We first analyze the existence of the solution to this GMFG, and show that naively combining Q-learning with the three-step fixed-point approach in classical MFGs yields unstable algorithms. We then propose an alternating approximating Q-learning algorithm with Boltzmann policy (MF-AQ) and establish its convergence property. The numerical performance of this MF-AQ algorithm on repeated Ad auction problem shows superior computational efficiency, when compared with existing algorithms for multi-agent reinforcement learning (MARL).

1. Introduction

Motivating example. This paper is motivated by the following Ad auction problem for an advertiser. An Ad auction is a stochastic game on an ad exchange platform among a large number of players, the advertisers. In between the time a web user requests a page and the time the page is displayed, usually within a millisecond, a Vickrey-type of second-best-price auction is run to incentivize interested advertisers to bid for an Ad slot to display advertisement. Each advertiser has limited information before each bid: first, her own *valuation* for a slot depends on an unknown conversion of clicks for the item; secondly, she, should she win the bid, only knows the reward *after* the user's activities on the website are finished. In addition, she has a budget constraint in this repeated auction.

The question is, how should she bid in this online sequential repeated game when there is a *large* population of bidders competing on the Ad platform, with *unknown* distributions of the conversion of clicks and rewards?

Our work. Motivated by this example of Ad auction, we consider the general problem of simultaneous learning and decision-making in a stochastic game setting with a large population. We formulate this type of games with unknown rewards and dynamics as a generalized mean-field-game (GMFG), with incorporation of action distributions. It can also be viewed as a general version of MFGs of McKean-Vlasov type (Acciaio et al., 2018), which is a different paradigm from the classical MFG. It is also beyond the scope of the existing Q-learning framework for Markov decision problem (MDP) with unknown distributions, as MDP is technically equivalent to a single player stochastic game.

On the theory front, we establish under appropriate technical conditions, the existence and uniqueness of the NE solution to this (GMFG). On the computational front, we show that naively combining Q-learning with the three-step fixed-point approach in classical MFGs yields unstable algorithms. We then propose an alternating approximating Q-learning algorithm with Boltzmann policy (MF-AQ) and establish its convergence property. This MF-AQ algorithm is then applied to analyze the Ad auction problem. Compared with the MF-VI algorithm with known rewards and dynamics and existing algorithms for multi-agent reinforcement learning (MARL), MF-AQ demonstrates superior numerical performance in terms of computational efficiency.

Related works. On learning large population games with mean-field approximations, (Yang et al., 2017) focused on inverse reinforcement learning for MFGs without decision making, (Yang et al., 2018) studied an MARL problem with a mean-field approximation term modeling the interaction between one agent and all the other finite agents, and (Kizilkale & Caines, 2013) and (Yin et al., 2014) considered model-based adaptive learning for MFGs. For learning large population games without mean-field approximation, see (Kapoor, 2018; Hernandez-Leal et al., 2018) and the references therein.

In the specific topic of learning auctions with a large number of advertisers, (Cai et al., 2017) and (Jin et al., 2018) explored reinforcement learning techniques to search for social optimal solutions with real-word data, and (Iyer et al.,

[1]Industrial Engineering and Operations Research Department, UC Berkeley [2]Institute for Computational & Mathematical Engineering at Stanford University. Correspondence to: Anran Hu <anran_hu@berkeley.edu>, Xin Guo <xinguo@berkeley.edu>, Renyuan Xu <renyuanxu@berkeley.edu>, Junzi Zhang <junziz@stanford.edu>.

2011) used MFGs to model the auction system with unknown conversion of clicks within a Bayesian framework.

On the subject of reinforcement learning, Q-learning and its variants have been widely applied to various problems with empirical success. In particular, the combination of deep neural networks and Q-learning has been the fundamental building block for some of the most successful human-level AIs (Mnih et al., 2015; Haarnoja et al., 2017). Apart from Q-learning, there are also policy gradient (Sutton et al., 2000) and actor-critic algorithms (Konda & Tsitsiklis, 2000)). On the theoretical side, several algorithms have also been proposed with near-optimal regret bounds. (See (Osband et al., 2013) and (Jaksch et al., 2010).)

However, none of these works formulated the problem of simultaneous learning and decision-making in the MFG framework.

2. Problem setting

2.1. Background: n-player games

Let us first recall the classical n-player games in a discrete time setting. There are n agents in a game with a finite state space $\mathcal{S}$ and a finite action space $\mathcal{A}$, in an infinite time horizon. At each step t, $t = 0, 1, \ldots, \infty$, decisions are made. At step t, each agent i has her own state $s_i^t \in \mathcal{S} \subseteq \mathbb{R}^d$ and needs to take an action $a_i^t \in \mathcal{A} \subseteq \mathbb{R}^p$; moreover, given the current state profile $\mathbf{s}^t = (s_1^t, \ldots, s_n^t) \in \mathcal{S}^n$ and an action profile $\mathbf{a}^t = (a_1^t, \ldots, a_n^t) \in \mathcal{A}^n$, an agent i will receive a reward $r_i^t \sim R_i(\mathbf{s}^t, \mathbf{a}^t)$, and her state will change to $s_i^{t+1} \sim P_i(\mathbf{s}^t, \mathbf{a}^t)$, where R_i is the reward distribution for agent i and P_i is the transition probability for agent i. The admissible policy/control for agent i is of a Markovian form $\pi_i : \mathcal{S}^n \to \Delta^{|\mathcal{A}|}$, which maps each state profile $\mathbf{s} \in \mathcal{S}^n$ to a randomized action. $\Delta^k := \{(x^1, \cdots, x^k) : x^j \geq 0, \sum_{j=1}^k x^j = 1\}$, which is the probability simplex.

In this infinite time setting, a discount factor $\gamma \in (0, 1)$ is used to define the accumulated reward (a.k.a. the value function) for agent i, with the initial state profile $\mathbf{s}$ and the policy profile $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$:

$$V_i(\mathbf{s}, \boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \bar{r}_i(\mathbf{s}^t, \mathbf{a}^t) \Big| \mathbf{s}^0 = \mathbf{s}\right], \tag{1}$$

where $i = 1, \ldots, n$, $\bar{r}_i(\mathbf{s}^t, \mathbf{a}^t) = \mathbb{E}[r_i \sim R_i(\mathbf{s}^t, \mathbf{a}^t) | \mathbf{s}^t, \mathbf{a}^t]$, $a_i^t \sim \pi_i(\mathbf{s}^t)$, and $s_i^{t+1} \sim P_i(\mathbf{s}^t, \mathbf{a}^t)$. The goal of each agent is to maximize her reward.

There are different performance criteria for solving an n-player game. The most well-known one is the Nash equilibrium (NE), a policy profile under which no agent can improve her value if other agents fix their policies.

Definition 2.1 (NE for n-player games)**.** *$\boldsymbol{\pi}^\star$ is a Nash equilibrium policy profile for the n-player game (1) if for all $i = 1, \ldots, n$ and the state profile $\mathbf{s}$,*

$$V_i(\mathbf{s}, \boldsymbol{\pi}^\star) \geq V_i(\mathbf{s}, (\pi_1^\star, \ldots, \pi_i, \ldots, \pi_n^\star)) \tag{2}$$

holds for any $\pi_i : \mathcal{S}^n \to \Delta^{|\mathcal{A}|}$.

2.2. Generalized MFGs.

A stochastic nonzero-sum n-player game is notoriously hard to analyze. When n is large, the complexity of the problem grows exponentially with respect to n. Mean field game (MFG), pioneered by (Huang et al., 2006) and (Lasry & Lions, 2007), considers the case when $n \to \infty$ and provides an ingenious and tractable aggregation approach to the otherwise challenging n-player stochastic games. By the functional strong law of large numbers, it was shown that the NE of an MFG is an ϵ -NE to the n-player game. (See (Cardaliaguet et al., 2015) for regular controls and (Guo & Lee, 2017) for singular controls.)

Now we formulate the simultaneous learning and decision-making in an generalized MFG framework. As in the classical MFG framework where there are infinite number of agents, due to the homogeneity of the agents, one can focus on a single (representative) agent. However, in contrast to the classical MFG framework where the dynamics of each agent are coupled through the population states with complete information, the dynamics of each agent in this generalized MFG framework are coupled through both the *population states* and *actions* with possibly incomplete information.

More specifically, consider an arbitrary agent in the population, whose state $s_t \in \mathcal{S}$ evolves under the following *controlled stochastic dynamics* of mean-field type. At each step t,

$$s_{t+1} \sim P(\cdot | s_t, a_t, L_t), \quad s_0 \sim \mu_0, \tag{3}$$

where $a_t \in \mathcal{A}$ is the action of the agent, μ_0 is the initial distribution, $L_t = (L_t^1, \ldots, L_t^{|\mathcal{S}||\mathcal{A}|}) = \mathbb{P}_{s_t, a_t} \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ is the joint distribution of the state and the action (*i.e.*, the population state-action pair), and $P(\cdot | s, a, L)$ is the transition probability (possibly unknown) over $\mathcal{S}$ given the action a, the state s, and population state-action pair L. Finally, r_t, the *reward* received at step t, follows a (possibly unknown) reward distribution $R(s_t, a_t, L_t)$.

For notational simplicity, denote the marginal distributions of s and a with joint distribution L as $\mu \in \Delta^{|\mathcal{S}|}$ and $\alpha \in \Delta^{|\mathcal{A}|}$. μ_t and α_t are the counterparts of $s_t^{-i} = (s_t^1, \ldots, s_t^{i-1}, s_t^{i+1}, \ldots, s_t^n)$ and $a_t^{-i} = (a_t^1, \ldots, a_t^{i-1}, a_t^{i+1}, \ldots, a_t^n)$ in an n-player game. Corresponding to the n-player game, the admissible (mean-field type) policy is of the form $\pi : \mathcal{S} \times \Delta^{|\mathcal{S}|} \to \Delta^{|\mathcal{A}|}$, which maps each state $s \in \mathcal{S}$ and distribution $\mu \in \Delta^{|\mathcal{S}|}$ to a randomized action $a \in \mathcal{A}$ distributed according to $\pi(s, \mu)$.

Accordingly, given a discount factor $\gamma \in (0,1)$, the value function for the MFG with an initial state s, an initial distribution μ_0, an agent policy π, and a population state-action pair sequence $\mathcal{L} := \{L_t\}_{t=0}^{\infty}$ is defined as

$$V(s,\pi,\mathcal{L}) := \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \bar{r}(s_t,a_t,L_t)\Big| s_0 = s, s_0 \sim \mu_0\right], \quad \text{(GMFG)}$$

where $\bar{r}(s,a,L) = \mathbb{E}[r|r \sim R(s,a,L)|s,a,L]$, $a_t \sim \pi(s_t,\mu_t)$, and $s_{t+1} \sim P(\cdot|s_t,a_t,L_t)$.

Definition 2.2 (NE for GMFGs)**.** *In (GMFG), an agent-population profile $(\pi^\star, L^\star)$ is called a stationary NE if*

1. *(Single agent side) For any policy π and any initial state $s \in \mathcal{S}$, we have*

$$V\left(s,\pi^\star,\{L^\star\}_{t=0}^{\infty}\right) \geq V\left(s,\pi,\{L^\star\}_{t=0}^{\infty}\right). \quad (4)$$

2. *(Population side) $\mathbb{P}_{s_t,a_t} = L^\star$ for all $t \geq 0$, where $\{s_t,a_t\}_{t=0}^{\infty}$ is the dynamics under control $\pi^\star$ starting from $s_0 \sim \mu^\star$, with $a_t \sim \pi^\star(s_t,\mu^\star)$, $s_{t+1} \sim P(\cdot|s_t,a_t,L^\star)$, and $\mu^\star$ being the population state marginal of $L^\star$.*

These two conditions correspond to the standard three-step fixed-point solution approach for classical MFGs. The first step is the time-backward step for solving a Bellman equation, the second step is the time-forward step for the underlying dynamics to ensure the consistency of the solution from the first step, and the last step is to find the fixed-point of these back-and forward iterations. Indeed, the first condition on the single agent side is exactly the counterpart of Eqn. (2). It is worth noting that the second condition is sometimes missing in the existing learning literature for MFGs. This is a crucial miss as this step ensures the consistency and the eventual correctness of the solution.

It is worth commenting that classical MFGs only involve the state-distribution μ without considering the action-distribution α, thus players taking *pure strategies*. In contrast, in the GMFG framework, players could take *mixed strategies* according to π.

Example. Take the example of the repeated auction with a budget constraint in Section 1. Denote $s_t \in [C_{\max}]$ as the budget of a representative advertiser at time t, where $[n] = \{0,1,\cdots,n\}$ and $C_{\max} \in \mathbb{N}^+$ is the maximum budget allowed on the Ad exchange with a unit bidding price, and a_t the bid price submitted by this advertiser.

At each round of the auction, assume a random number, say K bidders, will be randomly selected from the population to compete for the auction. Here K reflects the intensity of the bidding game. (See (Iyer et al., 2011) for the discussion on K and Figure 6 in this paper for related numerical analysis.) Denote α_t as the bidding distribution of the population. The maximum bid D_t from the $K-1$ opponents has a cumulative mass function $F_{D_t}(d) := \left(\sum_{a=0}^{d}\alpha_t(a)\right)^{K-1}$. Finally, the reward for the representative advertiser, say player 1, with bid a_t^1 and budget s_t at round t in which she is selected can be written as

$$r_t = \mathbf{I}_{w_t=1}\left[(v_t - a_t^{\text{second}}) - (1+\delta)\mathbf{I}_{s_t<a_t^{\text{second}}}(a_t^{\text{second}} - s_t)\right]. \quad (5)$$

Here v_t the conversion of clicks at time t follows an unknown distribution, $a_t^j \overset{\text{i.i.d}}{\sim} \alpha_t$ for $2 \leq j \leq K$, $a_t^{\text{second}} = \max\{\{a_t^1,\cdots,a_t^n\}\backslash\{a_t^{\max}\}\}$ and w_t is the winner. $w_t = 1$, meaning the first advertiser wins the bid, is with probability $\frac{1}{\#\arg\max_{[K]}\{a_t^1,a_t^2,\cdots,a_t^K\}}$.

The dynamics of the budget is

$$s_{t+1} = \begin{cases} s_t, & w_t \neq 1, \\ s_t - a_t^{\text{second}}, & w_t = 1 \text{ and } a_t^{\text{second}} \leq s_t, \\ 0, & w_t = 1 \text{ and } a_t^{\text{second}} > s_t. \end{cases}$$

The updated action distribution is $\alpha_{t+1} = \langle \mu_t, \pi_{t+1}(\cdot,\mu_t)\rangle$, where $\langle\cdot,\cdot\rangle$ stands for inner product. In this example, both the reward distribution r_t and the dynamics s_t are unknown.

2.3. (Unique) solution for GMFGs

Parallel to the classical MFG framework, we now establish the existence and uniqueness of the NE solution to (GMFG). This result is a generalization of Theorem 6 in (Huang et al., 2006).

To start, take any fixed $L \in \Delta^{|\mathcal{S}||\mathcal{A}|}$, define a mapping

$$\Gamma_1 : \Delta^{|\mathcal{S}||\mathcal{A}|} \to \Pi := \{\pi \mid \pi : \mathcal{S} \times \Delta^{|\mathcal{S}|} \to \Delta^{|\mathcal{A}|}\},$$

such that $\pi = \Gamma_1(L)$ with $\pi(s) = \textbf{argmax-e}(Q_L^\star(s,\cdot))$. Here the **argmax-e** operator is a generalized **argmax** operator, defined to map the set of **argmax** components to the randomized actions, such that actions with equal maximum Q-values would have equal probabilities to be selected. Note that, this is different from the usual set-valued argmax operator. Mathematically, **argmax-e** : $\mathbb{R}^n \to \mathbb{R}^n$ is a mapping with **argmax-e**$(x)_i = 1/N_x$ for $i \in \text{argmax}_i\, x_i$ and **argmax-e**$(x)_i = 0$ for $i \notin \text{argmax}_i\, x_i$, where $N_x = \#\text{argmax}_i\, x_i$. By definition, $\Gamma_1(L)$ is the optimal policy for an MDP with dynamics P_L and R_L (to be introduced in Section 3), and therefore satisfies the single agent side condition in Definition 2.2.

Assumption 1. *There exists a constant $d_1 \geq 0$, such that for any $L_1, L_2 \in \Delta^{|\mathcal{S}||\mathcal{A}|}$,*

$$\|\Gamma_1(L_1) - \Gamma_1(L_2)\|_2 \leq d_1\|L_1 - L_2\|_2. \quad (6)$$

This Assumption is closely related to the feedback regularity (FR) condition in the classical MFG literature (Huang et al., 2006).

Next, for any admissible policy $\pi \in \Pi$ and a joint population state-action pair $L \in \Delta^{|\mathcal{S}||\mathcal{A}|}$, define a mapping $\Gamma_2 : \Pi \times \Delta^{|\mathcal{S}||\mathcal{A}|} \to \Delta^{|\mathcal{S}||\mathcal{A}|}$ as follows:

$$\Gamma_2(\pi, L) := \hat{L} = \mathbb{P}_{s_1,a_1}, \tag{7}$$

where $a_1 \sim \pi(s_1, \mu_1)$, $s_1 \sim \mu P(\cdot|\cdot, a_0, L)$, $a_0 \sim \pi(s_0, \mu)$, $s_0 \sim \mu$, and μ is the population state marginal of L.

Assumption 2. *There exist constants d_2, $d_3 \geq 0$, such that for any admissible $\pi, \pi_1, \pi_2 \in \mathcal{S} \times \Delta^{|\mathcal{S}|}$ and joint distributions L, L_1, L_2,*

$$\|\Gamma_2(\pi_1, L) - \Gamma_2(\pi_2, L)\|_2 \leq d_2\|\pi_1 - \pi_2\|_2, \tag{8}$$

$$\|\Gamma_2(\pi, L_1) - \Gamma_2(\pi, L_2)\|_2 \leq d_3\|L_1 - L_2\|_2. \tag{9}$$

Theorem 1 (Existence and Uniqueness of Stationary MFG solution)**.** *Given Assumptions 1 and 2, and assume $d_1d_2 + d_3 < 1$. Then there exists a unique stationary MFG solution to* (GMFG) *under dynamics* (3)*.*

Proof. First by Definition 2.2 and the definitions of Γ_i $(i = 1, 2)$, (π, L) is a stationary Nash equilibrium solution iff $L = \Gamma(L) = \Gamma_2(\Gamma_1(L), L)$ and $\pi = \Gamma_1(L)$, where $\Gamma(L) := \Gamma_2(\Gamma_1(L), L)$. This indicates that for any $L_1, L_2 \in \Delta^{|\mathcal{S}||\mathcal{A}|}$,

$$\begin{aligned}
&\|\Gamma(L_1) - \Gamma(L_2)\|_2 \\
=&\|\Gamma_2(\Gamma_1(L_1), L_1) - \Gamma_2(\Gamma_1(L_2), L_2)\|_2 \\
\leq&\|\Gamma_2(\Gamma_1(L_1), L_1) - \Gamma_2(\Gamma_1(L_2), L_1)\|_2 \\
&+ \|\Gamma_2(\Gamma_1(L_2), L_1) - \Gamma_2(\Gamma_1(L_2), L_2)\|_2 \\
\leq&(d_1d_2 + d_3)\|L_1 - L_2\|_2.
\end{aligned} \tag{10}$$

And since $d_1d_2 + d_3 \in [0, 1)$, by the Banach fixed-point theorem, we conclude that there exists a unique fixed-point of Γ, or equivalently, a unique stationary MFG solution to (GMFG). □

3. RL Algorithms for GMFGs

Given the unknown reward and transition distributions in the generalized MFG setting, one needs to simultaneously learn P and R while solving the MFG in an online manner, where the data is collected on the run for both the learning and controlling purposes. This is related to the classical reinforcement learning problem, with the key differences being MDPs replaced with MFGs and global optimum replaced by NEs.

The learning aspect is best reflected in the single agent side in Definition 2.2. Indeed, given $L^\star$, one can retrieve $\pi^\star$ by solving an MDP with transition dynamics $P_{L^\star}(s'|s, a) := P(s'|s, a, L^\star)$ and reward distributions $R_{L^\star}(s, a) := R(s, a, L^\star)$. Extending this to a general population state-action pair L, the Q-function of an MDP $\mathcal{M}_L$ with dynamics P_L and R_L is written as

$$Q^\star_L(s, a) := \bar{r}_L(s, a) + \sum_{s' \in \mathcal{S}} P_L(s'|s, a)V^\star_L(s'), \tag{11}$$

where $\bar{r}_L(s, a) = \mathbb{E}[r|r \sim R_L(s, a)]$, $V^\star_L$ is the optimal value function of the MDP, and $V^*_L(s) = \max_a Q^\star_L(s, a)$.

Note that when reward R_L and transition dynamics $P_L(s'|s, a)$ are known, a value iteration method can be applied until convergence to estimate the Q-function in (11). However, when the reward R_L and transition dynamics $P_L(s'|s, a)$ are unknown, one can only update the Q-function based on historical observations. For example, take $r(s, a, L)$ as an approximation of $\mathbb{E}[r|r \sim R_L(s, a)]$ and $\max_{a'} Q_L(s', a')$ as an approximation of $V^*_L(s')$, then one can update Q_L as

$$\begin{aligned}
Q_L(s, a) \leftarrow\ & Q_L(s, a) + \beta\,[r(s, a, L) \\
&+ \gamma \max_{a'} Q(s', a') - Q_L(s, a)]\,,
\end{aligned} \tag{12}$$

where s' is the next state after taking action a at state s, $r(s, a, L)$ is the observed reward by taking action a at state s, β is the learning rate, and γ is the discounted rate. This is referred to as Q-learning in the literature.

3.1. Q-learning for GMFGs

In this section, we propose a Q-learning algorithm for the (GMFG). We begin by noticing the following lower bounds of *action gaps*.

For any $\epsilon > 0$, there exist a positive function $\phi(\epsilon)$ and an ϵ-net $S_\epsilon := \{L^{(1)}, \ldots, L^{(N_\epsilon)}\}$ of $\Delta^{|\mathcal{S}||\mathcal{A}|}$, with the properties that $\min_{i=1,\ldots,N_\epsilon} \|L - L^{(i)}\|_2 \leq \epsilon$ for any $L \in \Delta^{|\mathcal{S}||\mathcal{A}|}$, and that $\max_{a' \in \mathcal{A}} Q^\star_{L^{(i)}}(s, a') - Q^\star_{L^{(i)}}(s, a) \geq \phi(\epsilon)$ for any $i = 1, \ldots, N_\epsilon$, $s \in \mathcal{S}$, and any $a \notin$ argmax$_{a \in \mathcal{A}} Q^\star_{L^{(i)}}(s, a)$.

Here the existence of ϵ-nets is trivial due to the compactness of the probability simplex $\Delta^{|\mathcal{S}||\mathcal{A}|}$, and the existence of $\phi(\epsilon)$ comes from the finiteness of the action set $\mathcal{A}$.

These lower bounds characterize the extent to which actions are distinguishable in terms of the corresponding Q-values. They are crucial for approximation algorithms (Bellemare et al., 2016), and are closely related to the problem-dependent bounds for regret analysis in reinforcement learning, multi-armed bandits, and advantage learning algorithms including A2C (Minh et al., 2016).

In the following, we assume that for any $\epsilon > 0$, S_ϵ and $\phi(\epsilon)$ are known to the algorithm. In practice, one can choose a special form $D\epsilon^\alpha$ with $D > 0$ for $\phi(\epsilon)$. The exponent $\alpha > 0$ characterizes the decay rate of the action gaps on the fineness of the ϵ-nets.

In addition, to enable Q-learning, we assume that one has access to a population simulator (See (Pérolat et al., 2016; Wai et al.)). That is, for any policy $\pi \in \Pi$, given the current state $s \in \mathcal{S}$, for any joint law L (with population state marginal μ), we can obtain the next agent state $s' \sim P(\cdot|s, \pi(s, \mu), L)$, a reward $r \sim R(s, \pi(s, \mu), L)$, and the next population state-action pair $L' = \mathbb{P}_{s', \pi(s', \mu)}$. For brevity, we denote the simulator as $(s', r, L') = \mathcal{G}(s, \pi, L)$.

3.2. Alternating Q-learning

Observing the similarity between the fixed-point equation $L = \Gamma_2(\Gamma_1(L), L)$ and the Bellman equation in classical reinforcement learning, and noticing that the evaluation of $\Gamma_1(L)$ is equivalent to solving an MDP with transition dynamics $P_L(s'|s, a) := P(s'|s, a, L)$ and reward distributions $R_L(s, a) := R(s, a, L)$, it is natural to consider the following (naive) iterative algorithm (Algorithm 1) following the three-step fixed-point solution approach of MFGs. That is, $L_k \to Q_k^\star \to \pi_k \to L_{k+1} \to Q_{k+1}^\star \to ....$

Algorithm 1 Alternating Q-learning for GMFGs (Naive)

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 $Q_k^\star(s, a) = Q_{L_k}^\star(s, a)$ 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**

However, the update of $Q_k^\star$ requires fully solving an MDP with infinite steps. This issue can be resolved by using an approximate Q-learning updates. The second modification is to use Boltzmann policies to replace the argmax operator, whose discontinuity and sensitivity could be the culprit for the error and fluctuation shown in Figure 1. Finally, in each step L_k is projected to the ϵ-net to utilize the action gaps, which will be shown essential to control the sub-optimality of Boltzmann policies. The resulting algorithm is summarized in Algorithm 2.

Here $\textbf{softmax}_c : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$\textbf{softmax}_c(x)_i = \frac{\exp(cx_i)}{\sum_{j=1}^n \exp(cx_j)}, \tag{13}$$

and $\textbf{Proj}_{S_\epsilon}(L) = \text{argmin}_{L^{(1)}, \ldots, L^{(N_\epsilon)}} \|L^{(i)} - L\|_2$. The details about the choices of hyper-parameters c and T_k will be discussed in details in Lemma 5 and Theorem 2.

In the special case when the reward R_L and transition dynamics $P(\cdot|s, a, L)$ are known, Algorithm 2 can be reduced to a value-iteration based algorithm, as follows.

Algorithm 2 Alternating Approximate Q-learning for GMFGs (MF-AQ)

1: **Input**: Initial L_0, tolerance $\epsilon > 0$.
2: **for** $k = 0, 1, \cdots$ **do**
3: Perform Q-learning for T_k iterations to find the approximate Q-function $\hat{Q}_k^\star(s, a) = \hat{Q}_{L_k}^\star(s, a)$ of an MDP with dynamics $P_{L_k}(s'|s, a)$ and reward distributions $R_{L_k}(s, a)$.
4: Compute $\pi_k \in \Pi$ with $\pi_k(s) = \textbf{softmax}_c(\hat{Q}_k^\star(s, \cdot))$.
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**

Algorithm 3 Alternating Approximate Value Iteration for GMFGs (MF-VI)

1: **Input**: Initial L_0, tolerance $\epsilon > 0$.
2: **for** $k = 0, 1, \cdots$ **do**
3: Perform value iteration for T_k iterations to find the approximate Q-function and value function
4: **for** t=$1, 2, \cdots, T_k$ **do**
5: **for** all $s \in \mathcal{S}$ and $s \in \mathcal{A}$ **do**
6: $Q_{L_k}(s, a) \leftarrow \mathbb{E}[r|s, a, L_k] + \gamma \sum_{s'} P(s'|s, a, L_k) V_{L_k}(s')$
7: $V_{L_k}(s) \leftarrow_a Q_{L_k}(s, a)$
8: **end for**
9: **end for**
10: Compute a policy $\pi_k \in \Pi$: $\pi_k(s) = \textbf{softmax}_c(Q_{L_k}(s, \cdot))$.
11: Sample $s \sim \mu_k$, where μ_k is the population state marginal of L_k, and obtain $\tilde{L}_{k+1}$ from $\mathcal{G}(s, \pi_k, L_k)$.
12: Find $L_{k+1} = \textbf{Proj}_{S_\epsilon}(\tilde{L}_{k+1})$
13: **end for**

3.3. Convergence

We now show that for any given tolerance $\epsilon > 0$, the MF-AQ algorithm (Algorithm 2) converges to an ϵ-Nash solution of (GMFG).

Theorem 2 (Convergence of MF-AQ). *Given the same assumptions in Theorem 1 and the same conditions specified below in Lemma 5, with $T_k = T^{\mathcal{M}_L}(\delta_k, \epsilon_k)$, $\sum_{k=0}^\infty \delta_k = \delta < \infty$, and $\sum_{k=0}^\infty \epsilon_k < \infty$, and $c = \frac{\log(1/\epsilon)}{\phi(\epsilon)}$. Then for any specified tolerance $\epsilon > 0$, $\limsup_{k \to \infty} \|L_k - L^\star\|_2 = O(\epsilon)$ with probability at least $1 - 2\delta$.*

The proof of Theorem 2 depends on the following Lemmas 3, 4, and 5 regarding the softmax mapping and the Q-learning convergence of the MDP subproblem with a fixed L_k.

Lemma 3 ((Gao & Pavel, 2017)). *The softmax function is c-Lipschitz, i.e., $\|\textbf{softmax}(x) - \textbf{softmax}(y)\|_2 \le c\|x - y\|_2$*

for any $x,\ y \in \mathbb{R}^n$.

Lemma 4. *The distance between the softmax and the argmax mapping is bounded by*

$$\|\textbf{\textit{softmax}}_c(x) - \textbf{\textit{argmax-e}}(x)\|_2 \le 2n\exp(-c\delta),$$

where $\delta = \max_{i=1,\dots,n} x_i - \max_{x_j < \max_{i=1,\dots,n} x_i} x_j$, *with* $\delta := \infty$ *when all* x_j *are equal.*

Proof. Without loss of generality, let us assume that $x_1 = x_2 = \cdots = x_m = \max_{i=1,\dots,n} x_i = x^\star > x_j$ for all $m < j \le n$. Then

$$\textbf{argmax-e}(x)_i = \begin{cases} \frac{1}{m}, & i \le m, \\ 0, & otherwise. \end{cases}$$

$$\textbf{softmax}_c(x)_i = \begin{cases} \frac{e^{cx^\star}}{me^{cx^\star} + \sum_{j=m+1}^n e^{cx_j}}, & i \le m, \\ \frac{e^{cx_i}}{me^{cx^\star} + \sum_{j=m+1}^n e^{cx_j}}, & otherwise. \end{cases}$$

Therefore

$$\begin{aligned}
&\|\textbf{softmax}_c(x) - \textbf{argmax-e}(x)\|_2 \\
&\le \|\textbf{softmax}_c(x) - \textbf{argmax-e}(x)\|_1 \\
&= m\left(\frac{1}{m} - \frac{e^{cx^\star}}{me^{cx^\star} + \sum_{j=m+1}^n e^{cx_j}}\right) + \frac{\sum_{i=m+1}^n e^{cx_i}}{me^{cx^\star} + \sum_{j=m+1}^n e^{cx_j}} \\
&= \frac{2\sum_{i=m+1}^n e^{cx_i}}{me^{cx^\star} + \sum_{i=m+1}^n e^{cx_i}} = \frac{2\sum_{i=m+1}^n e^{-c\delta_i}}{m + \sum_{i=m+1}^n e^{-c\delta_i}} \\
&\le \frac{2}{m}\sum_{i=m+1}^n e^{-c\delta_i} \le \frac{2(n-m)}{m}e^{-c\delta} \le 2ne^{-c\delta},
\end{aligned}$$

with $\delta_i = x_i - x^\star$. □

Lemma 5 ((Even-Dar & Mansour, 2003))**.** *For an MDP, say* $\mathcal{M}$, *suppose that the Q-learning algorithm takes step-sizes*

$$\alpha_t(s,a) = \begin{cases} |\#(s,a,t)+1|^{-\omega}, & (s,a) = (s_t,a_t), \\ 0, & otherwise. \end{cases}$$

with $\omega \in (1/2, 1)$. *Here* $\#(s,a,t)$ *is the number of times up to time* t *that one visits the state-action pair* (s,a). *Also suppose that the covering time of the state-action pairs is bounded by* L *with probability at least* $p \in (0,1)$. *Then* $\|Q_{T^{\mathcal{M}}(\delta,\epsilon)} - Q^\star\|_2 \le \epsilon$ *with probability at least* $1 - 2\delta$. *Here* Q_T *is the* T*-th update in Q-learning, and* $Q^\star$ *is the (optimal) Q-function, given that*

$$T^{\mathcal{M}}(\delta,\epsilon) = \Omega\left(\left(\frac{L\log_p(\delta)}{\beta}\log\frac{V_{\max}}{\epsilon}\right)^{\frac{1}{1-\omega}} + \left(\frac{\left(L\log_p(\delta)\right)^{1+3\omega}V_{\max}^2\log\left(\frac{|\mathcal{S}||\mathcal{A}|V_{\max}}{\delta\beta\epsilon}\right)}{\beta^2\epsilon^2}\right)^{\frac{1}{\omega}}\right),$$

where $\beta = (1-\gamma)/2$, $V_{\max} = R_{\max}/(1-\gamma)$, *and* $R_{\max}$ *is an upper bound on the extreme difference between the expected rewards, i.e.,* $\max_{s,a,L}\bar{r}(s,a,L) - \min_{s,a,L}\bar{r}_{s,a,L} \le R_{\max}$.

Here the covering time of a state-action pair sequence is defined to be the number of steps needed to visit all state-action pairs starting from any arbitrary state-action pair.

Proof of Theorem 2. Define $\hat{\Gamma}_1^k(L) := \textbf{softmax}_c\left(\hat{Q}^\star_{L_k}\right)$. Let $L^\star$ be the population state-action pair in a stationary NE solution of (GMFG). Then we have $\pi_k = \hat{\Gamma}_1^k(L_k)$, and by denoting $d := d_1d_2 + d_3$,

$$\begin{aligned}
\|\tilde{L}_{k+1} - L^\star\|_2 &= \|\Gamma_2(\pi_k, L_k) - \Gamma_2(\Gamma_1(L^\star), L^\star)\|_2 \\
&\le \|\Gamma_2(\Gamma_1(L_k), L_k) - \Gamma_2(\Gamma_1(L^\star), L^\star)\|_2 \\
&\quad + \|\Gamma_2(\Gamma_1(L_k), L_k) - \Gamma_2(\hat{\Gamma}_1^k(L_k), L_k)\|_2 \\
&\le \|\Gamma(L_k) - \Gamma(L^\star)\|_2 + d_2\|\Gamma_1(L_k) - \hat{\Gamma}_1^k(L_k)\|_2 \\
&\le (d_1d_2 + d_3)\|L_k - L^\star\|_2 \\
&\quad + d_2\|\textbf{argmax-e}(Q^\star_{L_k}) - \textbf{softmax}_c(\hat{Q}^\star_{L_k})\|_2 \\
&\le d\|L_k - L^\star\|_2 \\
&\quad + d_2\|\textbf{softmax}_c(\hat{Q}^\star_{L_k}) - \textbf{softmax}_c(Q^\star_{L_k})\|_2 \\
&\quad + d_2\|\textbf{argmax-e}(Q^\star_{L_k}) - \textbf{softmax}_c(Q^\star_{L_k})\|_2 \\
&\le d\|L_k - L^\star\|_2 + cd_2\|\hat{Q}^\star_{L_k} - Q^\star_{L_k}\|_2 \\
&\quad + d_2\|\textbf{argmax-e}(Q^\star_{L_k}) - \textbf{softmax}_c(Q^\star_{L_k})\|_2.
\end{aligned}$$

Then since $L_k \in S_\epsilon$ by the projection step, by Lemma 4, Lemma 5 (and the choice of $T_k = T^{\mathcal{M}_L}(\delta_k, \epsilon_k)$), we have that with probability at least $1 - \delta_k$, $\|\tilde{L}_{k+1} - L^\star\|_2 \le d\|L_k - L^\star\|_2 + cd_2\epsilon_k + 2d_2|\mathcal{A}|e^{-c\phi(\epsilon)}$.

Finally, it is clear that with probability at least $1 - \delta_k$,

$$\begin{aligned}
\|L_{k+1} - L^\star\|_2 &\le \|\tilde{L}_{k+1} - L^\star\|_2 + \|\tilde{L}_{k+1} - \textbf{Proj}(\tilde{L}_{k+1})\|_2 \\
&\le d\|L_k - L^\star\|_2 + cd_2\epsilon_k + 2d_2|\mathcal{A}|e^{-cD\epsilon^\alpha} + \epsilon.
\end{aligned}$$

By telescoping, this implies that with probability at least $1 - \sum_{k=0}^K \delta_k$, $\|L_K - L^\star\|_2 \le d^K\|L_0 - L^\star\|_2 + cd_2\sum_{k=0}^{K-1} d^{K-k}\epsilon_k + \frac{(2d_2|\mathcal{A}|e^{-c\phi(\epsilon)} + \epsilon)(1-d^{K+1})}{1-d}$.

Since ϵ_k is summable and $\sup_{k\ge0}\epsilon_k < \infty$, we have $\sum_{k=0}^{K-1} d^{K-k}\epsilon_k \le \frac{\sup_{k\ge0}\epsilon_k}{1-d}d^{\lfloor(K-1)/2\rfloor} + \sum_{k=\lceil(K-1)/2\rceil}^{\infty}\epsilon_k \to 0$ as $K \to \infty$.

Taking the limit $K \to \infty$, and by $d \in [0,1)$ and $c = \frac{\log(1/\epsilon)}{\phi(\epsilon)}$, we have that with probability at least $1 - \delta$, $\limsup_{k\to\infty}\|L_k - L^\star\|_2 \le \frac{2d_2|\mathcal{A}| + 1}{1-d}\epsilon = O(\epsilon)$. □

4. Experiment: repeated auction game with budget constraint

In this section, we test the proposed MF-AQ Algorithm against Naive, MF-VI, and a couple of popular RL algorithms.

Recall that in the repeated auction game, the goal is for each advertiser to learn to bid in an online sequential repeated auction, when there are budget constraints and a large population of bidders. Here we assume that the distributions of the conversion of clicks v and the reward function R are unknown as in Section 2.2.

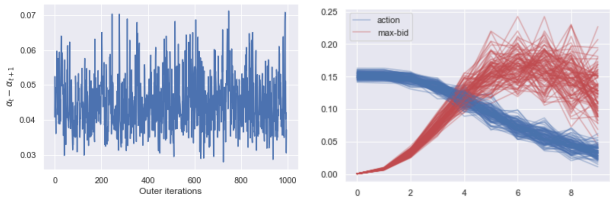
To emphasize the impact of α_t, here we fix the budget distribution as independent of time and known *a prior*. That is, $\mu_t = \mu_C$ and given. This can be realized by considering new players joining the competition according to a Poisson process.

Here are the parameters used in the experiments: the temperature parameter $c = 4.0$, the discount factor $\gamma = 0.2$, $\omega = 0.87$ in Lemma 5, and the overbidding penalty $\delta = 0.2$. Assume $|\mathcal{S}| = 10$, $|\mathcal{A}| = 10$, and the budget distribution $\mu_C = \text{uniform}[9]$.

4.1. Comparing MF-AQ with Naive algorithm

Naive algorithm Assume $K = 5$, v is uniform$[4]$, and α_0 is uniform$[9]$. Moreover, set 10000 inner iterations step between two consecutive updates of action distributions.

Figure 1 shows that the Naive algorithm does not converge in 1000 outer iterations. In particular, α_t keeps fluctuating.



(a) $\|\alpha_{t+1} - \alpha_t\|_\infty$ (b) μ_{D_t} and α_t.

Figure 1. Performance of Naive Algorithm.

MF-AQ Assume the same conversion of clicks v as uniform$[4]$.

In comparison to the Naive algorithm, MF-AQ Algorithm is computationally more efficient: it converges after about 10 outer iterations; as the number of inner iterations increases, the error decreases; and finally, MF-AQ is responsive to the conversion of the clicks even though it is unknown to the advertisers, as shown in Figure 3.

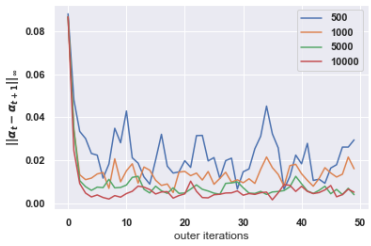
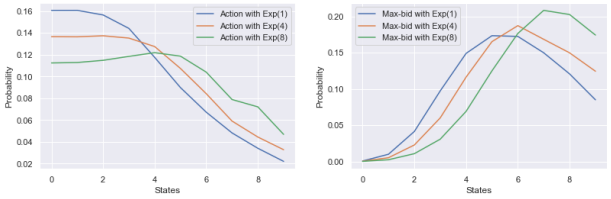


Figure 2. Different number of inner iterations within each outer iteration.



(a) Action distribution (b) Max-bid distribution

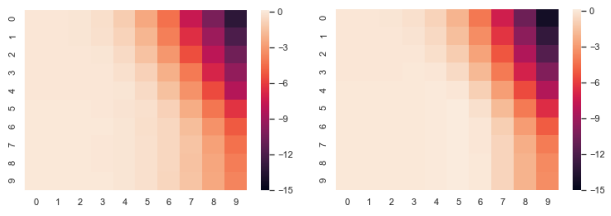
Figure 3. Equilibrium under different distributions of v.

4.2. Comparing MF-AQ with MF-VI

In addition to Theorem 2 that provides the convergence guarantee of the MF-AQ algorithm when reward and transition dynamics are unknown, the experiment shows that MF-AQ performs well against MF-VI, which assumes known reward and transition dynamics.

Set-up. $K = 5$, $\alpha_0 = \text{uniform}[9]$, and v is uniform$[4]$.

Result. The heatmap in Figure 4(a) is the Q-table for MF-AQ Algorithm with 20 outer iterations. Within each outer iteration, there are $T_k^{\text{MF-AQ}} = 10000$ inner iterations. The heatmap in Figure 4(b) is the Q-table for MF-AQ Algorithm after 20 outer iterations. Within each outer iteration, there are $T_k^{\text{MF-VI}} = 5000$ inner iterations. MF-VI Algorithm converges faster than MF-AQ Algorithm, as the latter needs to make decisions with simultaneous learning whereas the former is performed with given dynamics and rewards. The relative L_2 distance between the Q-tables of these two Al-



(a) $Q_{\text{MF-AQ}}$ (b) $Q_{\text{MF-VI}}$.

Figure 4. Q-tables: MF-AQ vs. MF-VI.

Table 1. Q-table comparison given $T_k^{\text{MF-VI}} = 5000$.

$T_k^{\text{MF-AQ}}$	1000	3000	5000	10000
ΔQ	0.21263	0.1294	0.10258	0.0989

gorithms is $\Delta Q := \frac{\|Q_{\text{MF-VI}} - Q_{\text{MF-AQ}}\|_2}{\|Q_{\text{MF-VI}}\|_2} = 0.098879$. This implies that MF-AQ Algorithm learns well: it learns the true MFG solution with 90-percent accuracy with 10000 inner iterations.

4.3. Comparing MF-AQ with existing learning algorithms.

We next compare the MF-AQ algorithm with some popular multi-agent reinforcement learning algorithms: (1) IL algorithm and (2) MF-Q algorithm. IL algorithm (Tan, 1993) considers n independent learners and each player solves a decentralized reinforcement learning problem without considering other players in the system. The MF-Q algorithm (Yang et al., 2018) is an extension of the NASH-Q Learning algorithm for general n-player game introduced in (Hu & Wellman, 2003). It considers the the aggregate actions ($\bar{\boldsymbol{a}}_{-i} = \frac{\sum_{j \neq i} a_j}{n-1}$) from the opponent of each player. This model reduces the computational complexity of the game but works for a restricted class of models where the interactions are only through the mean of the actions.

Performance Criterion: For a given policy π, the following metric (Pérolat et al., 2018) is adopted to measure how close the policy is to an NE policy:

$$C(\boldsymbol{\pi}) = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} \left(\max_{\pi^i} V_i(\boldsymbol{s}, (\boldsymbol{\pi}^{-i}, \pi^i)) - V_i(\boldsymbol{s}, \boldsymbol{\pi}) \right).$$

If $\boldsymbol{\pi}^*$ is an NE, by definition $C(\boldsymbol{\pi}^*) = 0$ and it is easy to check that $C(\boldsymbol{\pi}) \geq 0$. Policy $\arg\max_{\pi_i} V_i(\boldsymbol{s}, (\boldsymbol{\pi}^{-i}, \pi_i)))$ is called the best response to $\boldsymbol{\pi}^{-i}$.

Note that there are other performance criteria in the literature such as the total discounted reward. However, in a game setting, these may not be appropriate as high rewards for one player may not be beneficial for the others.

Here we assume v is uniform$[4]$. IL algorithm converges the fastest around 10000 iterations with the largest error 0.1913. This is because IL Algorithm does not incorporate information from other players. MF-Q algorithm converges around 250000 iterations with smaller error 0.1131. This error is still bigger than the error from MF-AQ. This is because MF-Q can only incorporate the first order information from the population instead of the empirical distribution from the whole population, which is not enough for the auction game. The performance of MF-AQ improves as the number of total iterations increases, and converges to the lowest error 0.0691 around 20000 total iterations.

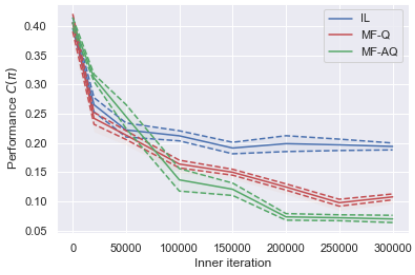
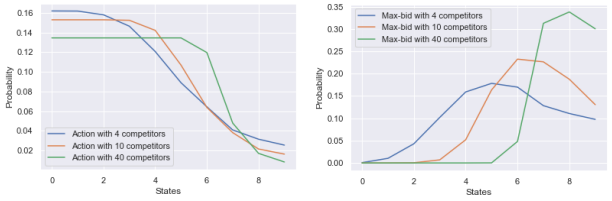
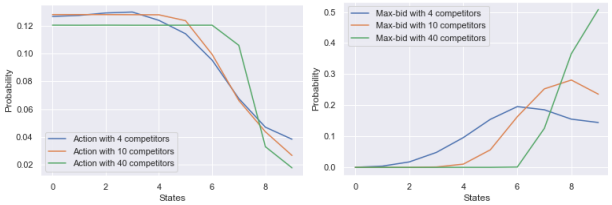


Figure 5. Performance comparison



(a) Action distribution with $v \sim$ uniform[3]

(b) Max-bid distribution with $v \sim$ uniform[3]



(c) Action distribution with $v \sim \exp(4)$

(d) Max-bid distribution with $v \sim \exp(4)$

Figure 6. Equilibrium under different number of competitors.

4.4. Additional insight of MF-AQ.

Figure 6 compares the action and max-bid distributions under different values of K with two distributions for v: uniform$[3]$ and $\exp(1)$. When v follows uniform$[3]$, it is not beneficial to bid over price 3 by the definition of reward function (5). In Figure 6(a), the center of the distribution is no bigger than 3. When K increases, players start to act more aggressively and bid over 3 more frequently. Similar observations are obtained when $v \sim \exp(4)$, as shown in Figure 6(c).

5. Conclusion

This paper builds a generalized mean-field games framework for simultaneous learning and decision-making, establishes the existence and uniqueness of appropriate Nash equilibrium solutions, and proposes a Q-learning algorithm with proven convergence. Numerical experiments demonstrate superior performance compared to existing RL algorithms.

References

Acciaio, B., Backhoff, J., and Carmona, R. Extended mean field control problems: stochastic maximum principle and transport perspective. *Arxiv Preprint:1802.05754*, 2018.

Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. Increasing the action gap: new operators for reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pp. 1476–1483, 2016.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., and Guo, D. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 661–670. ACM, 2017.

Cardaliaguet, P., Delarue, F., Lasry, J.-M., and Lions, P.-L. The master equation and the convergence problem in mean field games. *Arxiv Preprint:1509.02505*, 2015.

Even-Dar, E. and Mansour, Y. Learning rates for Q-learning. *Journal of Machine Learning Research*, 5(Dec):1–25, 2003.

Gao, B. and Pavel, L. On the properties of the softmax function with application in game theory and reinforcement learning. *Arxiv Preprint:1704.00805*, 2017.

Guo, X. and Lee, J. S. Mean field games with singular controls of bounded velocity. *Arxiv Preprint:1703.04437*, 2017.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. *Arxiv Preprint:1702.08165*, 2017.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. Is multiagent deep reinforcement learning the answer or the question? A brief survey. *Arxiv Preprint:1810.05587*, 2018.

Hu, J. and Wellman, M. P. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4(Nov):1039–1069, 2003.

Huang, M., Malhamé, R. P., Caines, P. E., et al. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, 6(3): 221–252, 2006.

Iyer, K., Johari, R., and Sundararajan, M. Mean field equilibria of dynamic auctions with learning. *ACM SIGecom Exchanges*, 10(3):10–14, 2011.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. In *Journal of Machine Learning Research*, 2010.

Jin, J., Song, C., Li, H., Gai, K., Wang, J., and Zhang, W. Real-time bidding with multi-agent reinforcement learning in display advertising. *Arxiv Preprint:1802.09756*, 2018.

Kapoor, S. Multi-agent reinforcement learning: A report on challenges and approaches. *Arxiv Preprint:1807.09427*, 2018.

Kizilkale, A. C. and Caines, P. E. Mean field stochastic adaptive control. *IEEE Transactions on Automatic Control*, 58(4):905–920, 2013.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, 2000.

Lasry, J.-M. and Lions, P.-L. Mean field games. *Japanese Journal of Mathematics*, 2(1):229–260, 2007.

Minh, V. M., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., and Petersen, S. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Osband, I., Russo, D., and Roy, B. V. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Pérolat, J., Strub, F., Piot, B., and Pietquin, O. Learning Nash equilibrium for general-sum Markov games from batch data. *Arxiv Preprint:1606.08718*, 2016.

Pérolat, J., Piot, B., and Pietquin, O. Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Tan, M. Multi-agent reinforcement learning: independent vs. cooperative agents. In *International Conference on Machine Learning*, pp. 330–337, 1993.

Wai, H. T., Yang, Z., Wang, Z., and Hong, M. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Arxiv Preprint:1806.00877*.

Yang, J., Ye, X., Trivedi, R., Xu, H., and Zha, H. Deep mean field games for learning optimal behavior policy of large populations. *Arxiv Preprint:1711.03156*, 2017.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. Mean field multi-agent reinforcement learning. *Arxiv Preprint:1802.05438*, 2018.

Yin, H., Mehta, P. G., Meyn, S. P., and Shanbhag, U. V. Learning in mean-field games. *IEEE Transactions on Automatic Control*, 59(3):629–644, 2014.