Reinforcement Learning for Mean-field Teams

Jayakumar Subramanian
McGill University
Montreal, Quebec, Canada
jayakumar.subramanian@mail.mcgill.ca

Raihan Seraj
McGill University
Montreal, Quebec, Canada
raihan.seraj@mail.mcgill.ca

Aditya Mahajan
McGill University
Montreal, Quebec, Canada
aditya.mahajan@mcgill.ca

ABSTRACT

We develop reinforcement learning (RL) algorithms for a class of multi-agent systems called mean-field teams (MFT). Teams are multi-agent systems where agents have a common goal and receive a common reward at each time step. The team objective is to maximize the expected cumulative discounted reward over an infinite horizon. MFTs are teams with homogeneous, anonymous agents such that the agents are coupled in their dynamics and rewards through the mean-field (i.e., empirical distribution of the agents' state). In our work, we consider MFTs with a mean-field sharing information structure, i.e., each agent knows its local state and the empirical mean-field at each time step. We obtain a dynamic programming (DP) decomposition for MFTs using a decomposition approach from literature called the common information approach, which splits the decision making process into a centralized coordination rule that yields prescriptions to be followed by each agent based on their local information. We develop an RL approach for MFTs under the assumption of parametrized prescriptions. We consider the parameters as actions and use conventional RL algorithms to solve the DP. We illustrate the use of these algorithms through two examples based on stylized models of the demand response problem in smart grids and malware spread in networks.

KEYWORDS

Reinforcement learning; mean-field teams; multi-agent reinforcement learning

1 INTRODUCTION

In this paper, we look at reinforcement learning in cooperative multi-agent systems. Several algorithms for multi-agent reinforcement learning have been proposed in the literature [2–4, 10–13, 21–24]. These algorithms perform well on certain benchmark domains but there is little theoretical analysis on whether these algorithms converge to a team optimal solution.

In this paper, we present a different view on multi-agent reinforcement learning. Our central thesis is that multi-agent systems for which the team optimal planning solution can be obtained by dynamic programming [14–16], it should be straightforward o translate these dynamic programs to reinforcement learning algorithms.

2 MODEL

Consider a multi-agent team with n agents, indexed by the set $N = \{1, \dots, n\}$. The team operates in discrete time for an infinite horizon. Let $X^i_t \in \mathcal{X}$ and $U^i_t \in \mathcal{U}$ denote the state and action of agent $i \in N$ at time t. Note that the state space $\mathcal{X}$ and action space $\mathcal{U}$ are the same for all agents. For ease of exposition, we assume that $\mathcal{X}$ and $\mathcal{U}$ are finite sets.

Given a vector $x = (x^1 \dots x^n) \in \mathcal{X}^n$ of length n, let $\xi(x)$ denote the mean-field (or empirical distribution) of x, i.e.,

$$\xi(x) = \frac{1}{n} \sum_{i \in N} \delta_{x^i}.$$

Let $Z_t = \xi(X_t)$ denote the mean-field of the team at time t and $\mathcal{Z}$ denote the space of space of realizations of Z_t. Note that $\mathcal{Z}$ has at most $(n+1)^{|\mathcal{X}|}$ elements.

Let $(\{x_t\}_{t\geq 0}, \{u_t\}_{t \geq 0})$ denote a realization of $(\{X_t\}_{t\geq 0}, \{U_t\}_{t\geq 0})$ and $z_t = \xi(x_t)$. We assume that the initial states of all agents are independent, i.e.,

$$\mathbb{P}(X_0 = x_0) = \prod_{i \in N} \mathbb{P}(X^i_0 = x^i_0) =: \prod_{i \in N} P_0(x^i_0),$$

where P_0 denotes the initial state distribution of agents. We assume that the global state of the system evolves in a controlled Markov manner, i.e.,

$$\begin{aligned}&\mathbb{P}(X_{t+1} = x_{t+1} \mid X_{0:t} = x_{1:t}, U_{0:t} = u_{0:t}) \\ &\quad = \mathbb{P}(X_{t+1} = x_{t+1} \mid X_t = x_t, U_t = u_t).\end{aligned}$$

All agents are partially exchangeable, so the state evolution of a generic agent depends on the states and actions of other agents only through the mean-fields of the states, i.e., for agent i:

$$\begin{aligned}\mathbb{P}(X_{t+1} = x_{t+1} \mid X_t = x_t, U_t = u_t) &= \prod_{i \in N} \mathbb{P}(X^i_t = x^i_t, U^i_t = u^i_t, Z_t = z_t) \\ &=: \prod_{i \in N} P(x^i_{t+1} \mid x^i_t, u^i_t, z_t),\end{aligned}$$

where P denotes the control transition matrix. Combining all of the above, we have

$$\mathbb{P}(X_{t+1} = x_{t+1} \mid X_{0:t} = x_{0:t}, U_{0:t} = u_{0:t}) = \prod_{i \in N} P(x^i_{t+1} \mid x^i_t, u^i_t, z_t). \tag{1}$$

The system has mean-field sharing information-structure, i.e., the information available to agent i is given by:

$$I^i_t = \{X^i_t, Z_t\}. \tag{2}$$

We assume that all agents use identical (stochastic) control law: $\mu_t : \mathcal{X} \times \mathcal{Z} \to \Delta(\mathcal{U})$ to choose the control action at time t, i.e.,

$$U^i_t \sim \mu_t(X^i_t, Z_t). \tag{3}$$

Let $\mu = (\mu_1, \mu_2, \dots)$ denote the team policy for all times. Note that, in general, restricting attention to identical policies may lead to a loss of optimality. See [1] for an example. Nonetheless, identical policies are attractive for reasons of fairness, simplicity, and robustness.

The team receives a per-step reward given by:

$$R_t \sim r(\boldsymbol{X}_t, \boldsymbol{U}_t). \tag{4}$$

Given strategy $\mu = (\mu_1, \mu_2, \dots)$ the expected total reward incurred by the team is given by:

$$J(\mu) = \mathbb{E}^{\mu}\bigg[\sum_{t=0}^{\infty} \gamma^t R_t\bigg], \tag{5}$$

where $\gamma \in (0, 1)$ is the discount factor. The objective is to choose a policy μ to maximize the performance $J(\mu)$ given by (5).

3 SOLUTION APPROACH

The mean-field team model formulated above is a multi-agent team problem with non classical information structure. A planning solution of this model was presented in [1], which we summarize below for completeness. We then present a framework for using reinforcement learning in such models.

3.1 Planning solution for mean-field teams

Given any policy $\mu = (\mu_1, \mu_2, \dots)$ and any realization, $z = (z_1, z_2, \dots)$ of the mean-field, define *prescriptions* $h_t : \mathcal{X} \to \Delta(A)$ given by

$$h_t(x) = \mu_t(x, z_t), \quad \forall x \in \mathcal{X}.$$

Let $\mathcal{H}$ denote the space of all such prescritions. When the mean field trajectory is a random process, the prescriptions $h_t(x)$ is a random vector which we denote H_t. The results of [1] relies on the following two key properties. Let $(z_{1:t+1}, h_{1:t})$ denote any realization of $(Z_{1:t+1}, H_{1:t})$. We have:

(1) $\{Z_t\}_{t \geq 1}$ is a controlled Markov process with control action h_t, i.e.,

$$\begin{aligned}&\mathbb{P}^{\mu}(Z_{t+1} = z_{t+1} \mid Z_{1:t} = z_{1:t}, H_{1:t} = h_{1:t})\\ &\quad = \mathbb{P}(Z_{t+1} = z_{t+1} \mid Z_t = z_t, H_t = h_t).\end{aligned}$$

Note that the right hand side does not depend on the choice of decision rule μ. Furthermore, the right hand side can be simplified as:

$$\begin{aligned}&\mathbb{P}(Z_{t+1} = z_t \mid Z_t = z_t, H_t = h_t)\\ &\quad = \sum_{x_{t+1}:\xi(x_{t+1}) = z_{t+1}} \prod_{i \in N} P(x^i_{t+1} \mid x^i_t, h_t(x^i_t), z_t),\end{aligned}$$

where x_t is any state such that $\xi(x_t) = z_t$.

(2) The expected per-step reward simplifies as follows.

$$\mathbb{E}[r(\boldsymbol{X}_t, \boldsymbol{U}_t)|Z_{1:t}, H_{1:t}] = \mathbb{E}[r(\boldsymbol{X}_t, \boldsymbol{U}_t)|Z_t, H_t] =: \tilde{r}(Z_t, H_t). \tag{6}$$

It is shown in [1] that these two properties imply that the optimal policy μ can be identified as follows.

Theorem 3.1. *Let $V : \mathcal{Z} \to \mathbb{R}$ be the unique bounded fixed point of the following equation:*

$$V(z) := \max_{h \in \mathcal{H}} \mathbb{E}[\tilde{r}(z, h) + \gamma V(Z_{t+1}) | Z_t = z, H_t = h]. \tag{7}$$

Let $\psi(z)$ be an arg max *of the right hand side of* (7). *Then the policy,*

$$\mu(x, z) = \psi(z)(x), \tag{8}$$

is an optimal policy for Problem (5).

The action space $\mathcal{H}$ of the above dynamic program is all functions functions from $\mathcal{X}$ to $\Delta(\mathcal{U})$. We assume that $\mathcal{H}$ is approximated by some family of parametrized functions $\mathcal{H}_\Phi = \{h_\phi\}_{\phi \in \Phi}$ (where Φ is a compact and convex set) such as Gibbs/Boltzmann functions or neural networks. With such a parametrization, the dynamic program of (7) may be approximated as:

$$V(z) = \max_{\phi \in \Phi} \mathbb{E}[\tilde{r}(z, h_\phi) + \gamma V(Z_{t+1}) \mid Z_t = z, H_t = h_\phi] \tag{9}$$

Let $\hat{\psi}(z)$ be an arg max of the right hand side of (9). Then the policy,

$$\mu(x, z) = h_{\hat{\psi}(z)}(x), \tag{10}$$

is the best policy for Problem (5) when $\mu_t(\cdot, z_t)$ is restricted to belong to $\mathcal{H}_\Phi$.

3.2 Reinforcement learning for mean-field teams (MFT-RL)

In this section, we present a reinforcement learning algorithm for the special case where the reward is a cumulative reward, i.e.,

$$R_t = \frac{1}{n} \sum_{i \in N} R^i_t, \tag{11}$$

where $R^i_t \sim \hat{r}(X^i_t, U^i_t, Z_t)$. We assume that we have access to a simulator for $P(\cdot \mid x^i_t, u^i_t, z_t)$ and $\hat{r}(x^i_t, u^i_t, z_t)$. This simulator is for a generic agent and takes the current local state, current local action and current mean-field as input and generates a sample of the local next state and the total reward as output. Using n copies of this simulator, we create a simulator for the mean-field dynamics. We start with n agents with initial local state sampled according to P_0. We assume that all these agents use a common stochastic policy $\hat{\psi} : \mathcal{Z} \to \Phi$ to generate prescription parameters $\phi_t \sim \hat{\psi}(z_t)$.

Given this sampled value of ϕ_t, each agent independently samples a control action $u^i_t \sim h_{\phi_t}(x^i_t)$. The actions u^i_t of agent i and the current mean-field z_t are given as input to the i^{th} simulator and the sampled output (X^i_{t+1}, R^i_t) are averaged to obtain (Z_{t+1}, R_t). Thus, we have a simulator with internal state z_t. This simulator takes ϕ_t as an input and gives (Z_{t+1}, R_t) as sampled next mean-field state and reward. Thus, this is a simulator for $P(z_{t+1} \mid z_t, h_{\phi_t})$ and $\tilde{r}(z_t, h_{\phi_t})$. We can use this simulator with any standard RL algorithm to find the optimal policy for the dynamic program (9). In our experiments below, we use TRPO [17], PPO [18] and NAFDQN [5].

4 NUMERICAL EXPERIMENTS

4.1 Benchmark domains

We consider the following domains to illustrate different decentralized reinforcement learning algorithms.

4.1.1 Demand response in smart grids. This is a stylized model for demand response in smart grids [1]. The system consists of n agents, where $\mathcal{X} = \{0, 1\}$, $\mathcal{U} = \{\emptyset, 0, 1\}$,

$$P(\cdot \mid \cdot, \emptyset, z) = M \tag{12}$$

$$P(\cdot \mid \cdot, 0, z) = (1 - \varepsilon_1) \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} + \varepsilon_1 M \tag{13}$$

$$P(\cdot \mid \cdot, 1, z) = (1 - \varepsilon_2) \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} + \varepsilon_2 M, \tag{14}$$

where M denotes the “natural” dynamics of the systems and ε_1 and ε_2 are small positive constants.

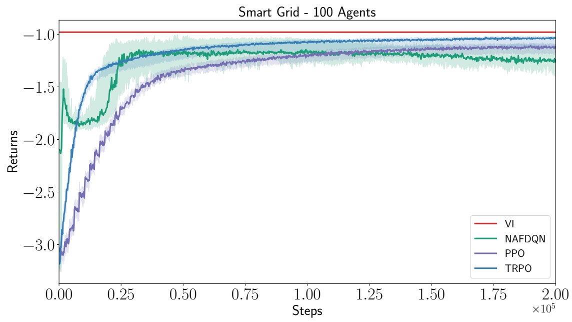

Figure 1: Performance of different variants of MFT-RL for demand response domain (25 independent runs).

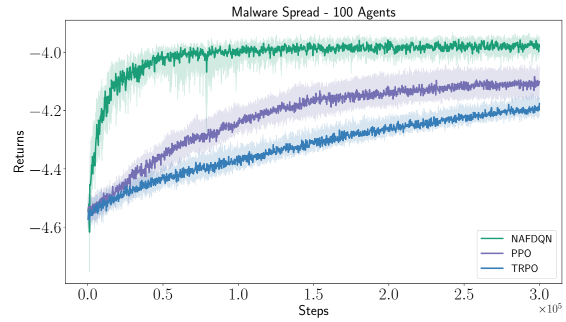

Figure 2: Performance of different variants of MFT-RL for malware spread domain (15 independent runs).

The per-step reward is given by:

$$R_t = -\left(\frac{1}{n}\sum_{i\in N}\left(c_0 \mathbb{1}_{\{U_t^i=0\}} + c_1 \mathbb{1}_{\{U_t^i=1\}}\right) + KL(Z_t \| \zeta)\right), \qquad (15)$$

where c_0 and c_1 are costs for taking actions 0 and 1 respectively, ζ is a given target distribution and $KL(Z_t\|\zeta)$ denotes the Kullback-Leibler divergence between Z_t and ζ.

In our experiments, we consider we consider a system with $n = 100$ agents, initial state distribution $P_0 = [1/3, 2/3]$, $M = \begin{bmatrix} 0.25 & 0.75 \\ 0.375 & 0.625 \end{bmatrix}$, $c_0 = 0.1$, $c_1 = 0.2$, $\zeta = [0.7, 0.3]$, $\varepsilon_1 = \varepsilon_2 = 0.2$ and discount factor $\gamma = 0.9$.

4.1.2 Malware spread in networks. This is a stylized model for malware spread in networks [6–8]. The system consists of n agents where $\mathcal{X} = [0, 1]$, $\mathcal{U} = \{0, 1\}$. The dynamics are given by:

$$X_{t+1}^i = \begin{cases} X_t^i + (1 - X_t^i)\omega_t, & \text{for } U_t = 0, \\ 0 & \text{for } U_t = 1, \end{cases}$$

where $\omega_t \sim \texttt{Uniform}[0, 1]$. The per-step reward is given by:

$$R_t = -\left(\frac{1}{n}\sum_{i\in N}(k + \langle Z_t\rangle)X_t^i + \lambda U_t^i\right),$$

where $\langle Z_t\rangle$ denotes the average of Z_t, and λ is the cost of taking action 1.

In our experiments, we consider $k = 0.2$, initial state distribution $P_0 = \texttt{Uniform}(\mathcal{X})$, $\lambda = 0.5$ and discount factor $\gamma = 0.9$. For the simulation, we discretize the state space into 11 bins—$0, 0.1, \ldots, 1$.

4.2 Simulation results

We consider three variants of MFT-RL algorithms, which use different RL algorithms for the mean-field system—TRPO, PPO and NAFDQN. Figure 1 shows the result for the demand response domain and Figure 2 shows the result for the malware spread domain. For each of the MFT-RL algorithms, the dark line shows the median performance and the shaded region shows the region between the first and third quartiles across multiple independent runs. For the demand response domain we also show the optimal performance obtained using the value iteration algorithm presented in [1].

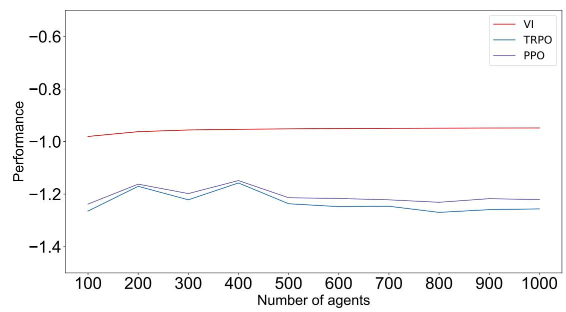

Figure 3: Performance of policy obtained in 100 agent system in systems with larger number of agents.

4.3 Mean-field approximations

Mean-field approximations are a common approach to simplify the planning solution of mean-field coupled systems. The main idea is to approximate a large population system with an infinite population system, find the optimal policy for the infinite population system and use that policy in the finite population system. Under appropriate regularity conditions, it can be shown that such an approximate policy is ε-optimal where ε is $O(1/n)$ or $O(1/\sqrt{n})$. Such approximations rely on the system model and are not appropriate in the learning setup.

However, the mean-field approximation results suggest some form of continuity in the optimal policy as the number of agents becomes large. This motivates us to investigate the reverse question. Can we find an approximate policy for a $n-$agent mean-field team by running MFT-RL on m agents, where $m < n$.

We investigate this idea in the demand response domain. We use MFT-RL for $m = 100$ agents using TRPO and PPO, and use the resultant policy in the systems with $n > 100$ agents. We compare this performance with optimal planning solution obtained using value iteration. The results are shown in Figure 3. This shows that the policy obtained for the 100 agent RL environment performs reasonably well in environments with larger number of agents as well.

5 CONCLUSION

There are many results in the Dec-POMDP/decentralized control literature where a team optimal solution can be obtained using dynamic programming. Our central thesis is that for such models one can easily translate the dynamic program to a reinforcement learning algorithm. We illustrate this point by using mean-field teams as an example. This allows us to use standard off-the-shelf RL algorithms to obtain solutions for some MARL setups.

REFERENCES

[1] Jalal Arabneydi and Aditya Mahajan. 2014. Team optimal control of coupled subsystems with mean-field sharing. In *IEEE Conference on Decision and Control*. IEEE, 1669–1674.

[2] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*. Springer, 183–221.

[3] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Philip Torr, Pushmeet Kohli, Shimon Whiteson, et al. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1702.08887* (2017).

[4] Jakob N. Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. 2018. Bayesian Action Decoder for Deep Multi-Agent Reinforcement Learning. *CoRR* (2018). arXiv:1811.01458 http://arxiv.org/abs/1811.01458

[5] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous Deep Q-learning with model-based acceleration. In *ICML*.

[6] M. Huang and Y. Ma. 2016. Mean field stochastic games: Monotone costs and threshold policies. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. 7105–7110.

[7] M. Huang and Y. Ma. 2017. Mean Field Stochastic Games with Binary Action Spaces and Monotone Costs. *ArXiv e-prints* (Jan. 2017). arXiv:math.OC/1701.06661

[8] M. Huang and Y. Ma. 2017. Mean field stochastic games with binary actions: Stationary threshold policies. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 27–32.

[9] Libin Jiang, Venkat Anantharam, and Jean Walrand. 2011. How bad are selfish investments in network security? *IEEE/ACM Transactions on Networking (TON)* 19, 2 (2011), 549–560.

[10] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Conference on Autonomous Agents and MultiAgent Systems*.

[11] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning., In ICML. *Proceedings of the eleventh ICML.*.

[12] Michael L. Littman. 2001. Friend-or-foe Q-learning in general-sum games.. In *ICML*.

[13] Michael L Littman. 2001. Value-function reinforcement learning in Markov games. *Cognitive Systems Research* 2, 1 (2001), 55–66.

[14] Aditya Mahajan and Mehnaz Mannan. 2016. Decentralized Stochastic Control. *Annals of Operations Research,* 241 (June 2016), 109–126.

[15] Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. 2013. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Trans. Automat. Control* 58, 7 (2013), 1644–1658.

[16] Frans A Oliehoek and Christopher Amato. 2015. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer.

[17] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *ICML*.

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[19] Jayakumar Subramanian and Aditya Mahajan. 2019. Reinforcement learning in stationary mean-field games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[20] Gabriel Y. Weintraub, C. Lanier Benkard, and Benjamin Van Roy. 2005. Oblivious Equilibrium: A Mean Field Approximation for Large-Scale Dynamic Games. In *Advances in Neural Information Processing Systems*. 1489–1496.

[21] Jiachen Yang, Xiaojing Ye, Rakshit Trivedi, Huan Xu, and Hongyuan Zha. 2018. Deep Mean Field Games for Learning Optimal Behavior Policy of Large Populations. In *ICLR*.

[22] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. In *ICML*.

[23] H. Yin, P. G. Mehta, S. P. Meyn, and U. V. Shanbhag. 2014. Learning in Mean-Field Games. *IEEE TAC* 59, 3 (Mar 2014).

[24] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. 2018. Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. In *ICML*.