Offline Multi-Agent Reinforcement Learning with Coupled Value Factorization

Extended Abstract

Xiangsen Wang
Beijing Jiaotong University
Beijing, China
wangxiangsen@bjtu.edu.cn

Xianyuan Zhan*
Tsinghua University
Beijing, China
zhanxianyuan@air.tsinghua.edu.cn

ABSTRACT

In offline multi-agent reinforcement learning (RL), most existing methods directly apply offline RL ingredients in the multi-agent setting without fully leveraging the decomposable problem structure, leading to less satisfactory performance in complex tasks. We present OMAC, a new offline multi-agent RL algorithm with coupled value factorization. OMAC adopts a coupled value factorization scheme that decomposes the global value function into local and shared components, and also maintains the credit assignment consistency between the state-value and action-value functions. Moreover, OMAC performs in-sample learning on the decomposed local state-value functions, which implicitly conducts max-Q operation at the local level while avoiding distributional shift caused by evaluating out-of-distribution actions. Based on the comprehensive evaluations of the offline multi-agent StarCraft II micro-management tasks, we demonstrate the superior performance of OMAC over existing offline multi-agent RL methods.

KEYWORDS

Multi-Agent Reinforcement Learning; Offline Reinforcement Learning; Multi-Agent Cooperation

ACM Reference Format:
Xiangsen Wang and Xianyuan Zhan. 2023. Offline Multi-Agent Reinforcement Learning with Coupled Value Factorization: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

In most real-world scenarios, reliable simulators are not available and it can be costly for online interaction with the real system. The recently emerged offline RL methods provide another promising direction by training the RL agent with pre-collected offline dataset without system interaction [1, 3, 4, 13]. Compared with offline single-agent RL, offline multi-agent RL (MARL) is a relatively underexplored area and considerably more complex [8, 11]. Under the offline setting, evaluating value function outside data coverage areas can produce falsely optimistic values, causing the issue of *distributional shift* [3]. When adding the multi-agent consideration, the joint action space grows exponentially with the number of agents, we need to consider regularizing multi-agent policy optimization with respect to the data distribution. This can potentially lead to over-conservative multi-agent policies due to extremely limited feasible state-action space under data-related regularization.

Consequently, an effective offline MARL algorithm needs to not only fully leverage the underlying multi-agent decomposable problem structure, but also organically incorporate offline data-related regularization. Ideally, the data-related regularization should be performed at the individual agent level to avoid the negative impact of sparse data distribution at the joint space and enable producing a more relaxed yet still valid regularization to prevent distributional shift. Under this rationale, a natural choice is to decompose the global value function as the combination of local value functions. However, existing offline MARL algorithms that naively combine the value decomposition framework with local-level offline RL [8, 11] still suffer from several drawbacks. First, the value decomposition scheme is not specifically designed for the offline setting. Second, they may still suffer from instability issues caused by the bootstrapping error accumulation. The instability of the local value function will further propagate and negatively impact the learning of the global value function.

To tackle above issues, we propose OMAC, a new offline multi-agent RL algorithm with coupled value factorization. OMAC organically marries offline RL with a specially designed coupled multi-agent value decomposition strategy. In addition to decomposing global action-value function Q_{tot}, OMAC also decomposes V_{tot} into local state-value functions V_i and a shared component V_{share}. Moreover, OMAC poses an extra coupled credit assignment scheme between state-value and action-value functions to enforce consistency and a more regularized global-local relationship. Under this factorization strategy, we can learn an upper expectile local state-value function V_i in a completely in-sample manner. It enables separated learning of the local action-value function Q_i and the policy π_i, which improves the learning stability of both the local and global action-value functions. We benchmark our method using offline datasets of StarCraft Multi-Agent Challenge (SMAC) tasks [10]. The results show that OMAC achieves state-of-the-art (SOTA) performance compared with the competing baseline methods.

2 METHOD

For each agent, we define the local state-value function V_i as the optimal value of the local action-value function Q_i. In particular, we decompose the global state-value function into a linear combination of local state-value functions $V_i(o_i)$ with weight function $w_i^v(\boldsymbol{o})$, as well as the shared component based on the full observation $V_{share}(\boldsymbol{o})$. The global action-value function is further decomposed

*Corresponding author.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

as the state-value function plus a linear combination of local advantages $Q_i(o_i, a_i) - V_i(o_i, a_i)$ with weight function $w_i^q(\boldsymbol{o}, \boldsymbol{a})$:

$$
\begin{aligned}
V_{tot}(\boldsymbol{o}) &= \sum_{i=1}^{n} w_i^v(\boldsymbol{o}) V_i\left(o_i\right) + V_{share}(\boldsymbol{o}) \\
Q_{tot}(\boldsymbol{o}, \boldsymbol{a}) &= V_{tot}(\boldsymbol{o}) + \sum_{i=1}^{n} w_i^q(\boldsymbol{o}, \boldsymbol{a})\left(Q_i\left(o_i, a_i\right) - V_i\left(o_i\right)\right) \\
V_i(o_i) &= \max_{a_i} Q_i(o_i, a_i), \;\; w_i^v, w_i^q \geq 0, \;\; \forall i = 1, \cdots, n
\end{aligned} \tag{1}
$$

In Eq. (1), the globally shared information is partly captured in the shared component of the state-value function $V_{share}(\boldsymbol{o})$, which is free of the joint actions and not affected by the OOD actions under offline learning. The information sharing across agents and credit assignment is captured in weight functions $w_i^v(\boldsymbol{o})$ and $w_i^q(\boldsymbol{o}, \boldsymbol{a})$. The local value functions $V_i(o_i)$ and $Q_i(o_i, a_i)$ are now only responsible for local observation and action information. The shared and the local information are separated, and agents can make decisions by using local V_i and Q_i at an individual level.

Ideally, the credit assignment on global state and action values should be coupled and correlated. Thus, we further design a coupled credit assignment scheme implemented with neural networks to enforce such consistency, which also leads to a more regularized relationship between $w^v(\boldsymbol{o})$ and $w^q(\boldsymbol{o}, \boldsymbol{a})$:

$$
\begin{aligned}
h_v(\boldsymbol{o}) = f_v^{(1)}(\boldsymbol{o}), \quad h_q(\boldsymbol{o}) &= f_q^{(1)}(\boldsymbol{o}, \boldsymbol{a}) \\
w_i^v(\boldsymbol{o}) &= |f_v^{(2)}(h_v(\boldsymbol{o}))| \\
w_i^q(\boldsymbol{o}, \boldsymbol{a}) &= |f_q^{(2)}(\text{concat}(h_v(\boldsymbol{o}), h_q(\boldsymbol{o}, \boldsymbol{a}))|
\end{aligned} \tag{2}
$$

where $f_v^{(1)}$, $f_v^{(2)}$, $f_q^{(1)}$, and $f_q^{(2)}$ are hidden neural network layers. We take absolute values on the network outputs to ensure the positivity condition of $w^v(\boldsymbol{o})$ and $w^q(\boldsymbol{o}, \boldsymbol{a})$. It enforces a coupled relationship between $w^v(\boldsymbol{o})$ and $w^q(\boldsymbol{o}, \boldsymbol{a})$ by sharing the same observation encoding structure, which makes training on $w^q(\boldsymbol{o}, \boldsymbol{a})$ can also update the parameters of $w^v(\boldsymbol{o})$. This coupling relationship allows more stable credit assignment between state and action value functions on the same observation $\boldsymbol{o}$. It can also improve data efficiency during training, which is particularly important for the offline setting since the size of the real-world dataset can be limited.

In the proposed coupled value factorization, the condition of $V_i(o_i) = \max_{a_i} Q_i(o_i, a_i)$ needs to be forced. Directly implementing this condition can be problematic under the offline setting, as it could lead to queries on OOD actions, causing distributional shift and overestimated value functions. To avoid this issue, one need to instead consider the following condition:

$$
V_i(o_i) = \max_{a_i \in \mathcal{A},\, \text{s.t. } \pi_\beta(a_i|o_i) > 0} Q_i(o_i, a_i), \tag{3}
$$

where π_β is the behavior policy of the offline dataset. Drawing inspiration from offline RL algorithm IQL [2], we can implicitly perform the above max-Q operation by leveraging the decomposed state-value functions V_i, while also avoiding explicitly learning the behavior policy π_β. This can be achieved by learning the local state-value function $V_i(o_i)$ as the upper expectile of target local action-values $\bar{Q}_i(o_i, a_i)$ based on (o_i, a_i) samples from dataset $\mathcal{D}$. For each agent, its local state-value function $V_i(o_i)$ is updated by minimizing the following objective:

$$
L_{V_i} = \mathbb{E}_{(o_i, a_i) \sim \mathcal{D}}\left[L_2^\tau\left(\bar{Q}_i(o_i, a_i) - V_i(o_i)\right)\right], \tag{4}
$$

where $L_2^\tau(u) = |\tau - 1(u < 0)|u^2$ denotes the expectile regression and $\tau \in (0, 1)$.

With the estimated local state-value function $V_i(o_i)$, we can then use it to update the global value functions V_{tot} and Q_{tot}, which are essentially parameterized by the shared state-value function $V_{share}(\boldsymbol{o})$, local action-value function $Q_i(o_i, a_i)$, as well as the credit assignment weight functions $w_i^v(\boldsymbol{o})$ and $w_i^q(\boldsymbol{o}, \boldsymbol{a})$ as in Eq. (1). These terms can be jointly learned by minimizing the following objective:

$$
L_Q = \mathbb{E}_{(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o}') \sim \mathcal{D}}\left[\left(r(\boldsymbol{o}, \boldsymbol{a}) + \gamma V_{tot}\left(\boldsymbol{o}'\right) - Q_{tot}(\boldsymbol{o}, \boldsymbol{a})\right)^2\right]. \tag{5}
$$

With the learned local state and action value functions Q_i and V_i, we can extract the local policies by maximizing the local advantage values with KL-divergence constraints to regularize the policy to stay close to the behavior policy. It can be shown equivalent to minimizing the following advantage-weighted regression objective [7, 9] by enforcing the KKT condition:

$$
L_{\pi_i} = \mathbb{E}_{(o_i, a_i) \sim \mathcal{D}}\left[\exp\left(\beta\left(Q_i(o_i, a_i) - V_i(o_i)\right)\right) \log \pi_i(a_i|o_i)\right], \tag{6}
$$

where β is a temperature parameter.

Dataset	OMAC	ICQ	OMAR	BCQ-MA	CQL-MA
5m_vs_6m (good)	**8.25±0.12**	7.94±0.32	7.17±0.42	8.03±0.31	8.17±0.20
5m_vs_6m (medium)	**8.04±0.42**	7.77±0.30	7.08±0.51	7.58±0.10	7.78±0.10
5m_vs_6m (poor)	7.44±0.16	7.47±0.13	7.13±0.30	**7.53±0.15**	7.38±0.06
6h_vs_8z (good)	**12.57±0.47**	11.81±0.12	9.85±0.28	12.19±0.23	10.44±0.20
6h_vs_8z (medium)	**12.17±0.52**	11.56±0.34	10.81±0.21	11.77±0.36	11.59±0.35
6h_vs_8z (poor)	**11.08±0.36**	10.34±0.23	10.64±0.20	10.67±0.19	10.76±0.11
3s5z_vs_3s6z (good)	16.81±0.46	16.95±0.39	8.71±2.84	**17.43±0.46**	9.27±2.53
3s5z_vs_3s6z (medium)	**14.47±1.11**	12.55±0.53	5.58±1.77	13.99±0.62	5.08±1.45
3s5z_vs_3s6z (poor)	**8.82±0.95**	7.43±0.42	2.12±1.07	8.36±0.45	3.22±0.87
corridor (good)	15.21±1.06	**15.55±1.13**	6.74±0.69	15.24±1.21	5.22±0.81
corridor (medium)	**12.37±0.51**	11.30±1.57	7.26±0.71	10.82±0.92	7.04±0.66
corridor (poor)	**5.68±0.65**	4.25±0.17	4.05±0.86	4.37±0.57	3.89±0.89

Table 1: Average scores and standard deviations over 5 random seeds on the offline SMAC tasks

3 EXPERIMENTS

We choose the StarCraft Multi-Agent Challenge (SMAC) benchmark [10] as our testing environment. The offline SMAC dataset used in this study is provided by [6]. The dataset is collected from the trained MAPPO agent [12], and includes three quality levels: good, medium, and poor. We consider 4 representative SMAC maps, including 1 hard map (5m_vs_6m), and 3 super hard maps (6h_vs_8z, 3s5z_vs_3s6z, corridor).

We compare OMAC against four recent offline MARL algorithms: ICQ [11], OMAR [8], multi-agent version of BCQ [1] and CQL [5], namely BCQ-MA and CQL-MA. BCQ-MA and CQL-MA use a linear weighted value decomposition structure for the multi-agent setting. We report the mean and standard deviation of average returns for the offline SMAC tasks in Table 1. The results show that OMAC consistently outperforms all baselines and achieves state-of-the-art performance in most maps. For the super hard SMAC map such as 6h_vs_8z or corridor, the cooperative relationship of agents is very complex and it is difficult to learn an accurate global value function. Due to the couple value factorization, the global Q_{tot} of OMAC has a stronger expressive capability, which makes OMAC have better performance than other baseline algorithms. Moreover, both the local and global value functions in OMAC are completely performed in an in-sample manner without the involvement of the agent policies π_i, which also leads to better offline performance.

ACKNOWLEDGMENTS

This work is supported by funding from Global Data Solutions Limited.

REFERENCES

[1] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*. PMLR, 2052–2062.

[2] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).

[3] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems* 32 (2019).

[4] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[5] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[6] Linghui Meng, Muning Wen, Yaodong Yang, Chenyang Le, Xiyun Li, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, and Bo Xu. 2021. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *arXiv preprint arXiv:2112.02845* (2021).

[7] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).

[8] Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. 2022. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International Conference on Machine Learning*. PMLR, 17221–17237.

[9] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).

[10] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 2186–2188.

[11] Yiqin Yang, Xiaoteng Ma, Li Chenghao, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. 2021. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 10299–10312.

[12] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. 2021. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games. *arXiv: Learning* (2021).

[13] Xianyuan Zhan, Haoran Xu, Yue Zhang, Xiangyu Zhu, Honglei Yin, and Yu Zheng. 2022. DeepThermal: Combustion Optimization for Thermal Power Generating Units Using Offline Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.