Learning Efficient Communication in Cooperative Multi-Agent Environment

Extended Abstract

Yuhang Zhao
Peking University
Beijing, Beijing
zhaoyuhang@pku.edu.cn

Xiujun Ma
Peking University
Beijing, Beijing
maxiujun@pku.edu.cn

ABSTRACT

Reinforcement learning in cooperate multi-agent scenarios is important for real-world applications. While several attempts before tried to resolve it without explicit communication, we present a communication-filtering actor-critic algorithm that trains decentralized policies which could exchange filtered information in multi-agent settings, using centrally computed critics. Communication could potentially be an effective way for multi-agent cooperation. We supposed that, when in execution phase without central critics, high-quality communication between agents could help agents have better performance in cooperative situations. However, information sharing among all agents or in predefined communication architectures that existing methods adopt can be problematic. Therefore, we use a neural network to filter information between agents. Empirically, we show the strength of our model in two general cooperative settings and vehicle lane changing scenarios. Our approach outperforms several state-of-the-art models solving multi-agent problems.

KEYWORDS

Collective intelligence; Multiagent learning

ACM Reference Format:
Yuhang Zhao and Xiujun Ma. 2019. Learning Efficient Communication in Cooperative Multi-Agent Environment. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Learning to cooperate between several interacting agents has been well studied [1, 11, 13]. And cooperative learning has been studied in multiple domains [2, 6, 8, 9]. In multi-agent reinforcement learning (MARL) collaboration, communication is critical, especially for scenarios where a large number of agents work collaboratively, such as autonomous vehicles planning [3], smart grid control [12], and multi-robot control [10].

We propose a communication-filtering actor-critic framework, called CFAC, to enable agents to learn effective and efficient communication under partially observable distributed environment. We train decentralized policies which could exchange filtered information in multi-agent settings, using centrally computed critics. The intuition behind our idea is that communication is crucial to get a best result in cooperative settings. If we made our agents use all kinds of information from other agents, it would be very difficult to study something really useful during training. So we need to extract relative information from it, and we use a neural network to do the filtering job.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

2 METHODS

2.1 Background

In this work, we consider multi-agent domains that are fully cooperative and partially observable. All agents are attempting to maximize the discounted sum of joint rewards. No single agent can observe the state of the environment. Instead, each agent receives a private observation that is correlated with that state.

2.2 Communication-Filtering Actor-Critic (CFAC)

Our CFAC network structure is shown in Figure 1. Our network consists of three parts: the critic network, the policy network, and the information filtering network. Among them, the critic network is centralized, the policy network is decentralized, and the information filtering network is semi-centralized. Each policy network obtains the partial observations that can be observed in the current global state from the environment, and obtains the filtered and efficient information from the information filtering network, and outputs the current time decision. The training process and pseudo code 1 are as follows. Note that b is defined in [4].

The training of our method is an extension of actor-critic. More concretely, consider a game with N agents, and the critic Q, actor π, and information-filtering network F is parameterized by θ, ψ, and κ, respectively. The experience replay buffer R contains the tuples (C^0, O, A, C, R, O'), recording the experiences of all agents, where $C^0 = (c_1^0, c_2^0, ..., c_N^0)$ is output information of state before for each agent, $O = (o_1, o_2, ..., o_N)$ is observation for each agent, $A = (a_1, a_2, ..., a_N)$ is action for each agent, $C = (c_1, c_2, ..., c_N)$ is output information for each agent, $R = (r_1, r_2, ..., r_N)$ is reward for each agent, and $O' = (o'_1, o'_2, ..., o'_N)$ is observation of next state for each agent.

2.3 Experiments

2.3.1 Setup and Baselines. We focus on experimental scenarios where observation space is continuous but action space is discrete. We evaluate our method on three experiments, a cooperative navigation, a cooperative treasure collection, and a cooperative lane

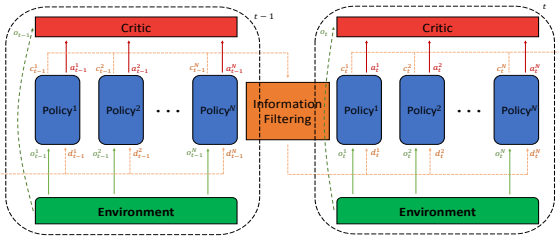

Figure 1: Communication-Filtering Actor-Critic architecture with a centralized critic, a semi-centralized information-filtering network, and decentralized policies.

Algorithm 1 Communication-Filtering Actor-Critic

Input: Policy parameters θ, Critic parameters ψ, Information-filtering network parameters κ

Randomly initialize policy network π, critic network Q, and information-filtering network F with parameters θ, ψ, and κ
Initialize target networks with parameters $\bar{\theta} \leftarrow \theta$, $\bar{\psi} \leftarrow \psi$, and $\bar{\kappa} \leftarrow \kappa$
Initialize replay buffer R
for episode = 1, M **do**
  Receive initial observation state o_1 and initial information state c_1^0
  for t = 1, T **do**
    Select action $a_t \sim \pi_\theta(o_t, d_t)$, where $d_t = F_\kappa(c_t^0)$
    Execute action a_t and observe reward r_t, new information state c_t, and new state o_t'
    Store transition $(c_t^0, o_t, a_t, c_t, r_t, o_t')$ in R
    Sample a random minibatch of M transitions $(c_i^0, o_i, a_i, c_i, r_i, o_i')$ from R
    Set $y_i = r_i + \gamma Q_{\bar{\psi}}(\pi_{\bar{\theta}}, F_{\bar{\kappa}})$
    Update critic by minimizing the loss:

$$\mathbb{L} = \frac{1}{M}\sum_i (y_i - Q_\psi)^2$$

    Update the actor policy and information-filtering network using the sampled policy gradient:

$$\nabla_{\theta,\kappa}\mathbb{J} = \frac{1}{M}\sum_i \nabla_{\theta,\kappa}\log \pi_\theta(F_\kappa)(Q_\psi - b)$$

    Update the target networks:

$$\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$$
$$\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$$
$$\bar{\kappa} \leftarrow \tau\kappa + (1-\tau)\bar{\kappa}$$

  end for
end for

changing that is important in the field of autonomous driving. Regarding our method and our experiments, we compare to four of the state-of-the-art approaches recently proposed for centralized training of decentralized policies: CommNet [13], COMA [4], MAAC [5], ATOC [7]. Besides that, in order to know how important our communication mechanism is, we use global state instead of partial observation as input to our method labeled ours with global observation.

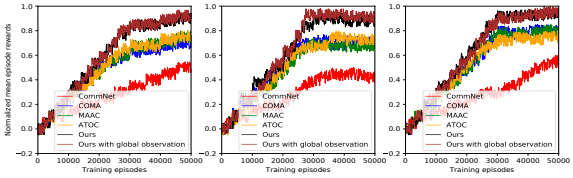

Figure 2: Results of our three experiments: cooperative navigation (left), cooperative treasure collection (middle), cooperative lane changing (right).

Cooperative Navigation N agents cooperatively reach L landmarks, while avoiding collisions. Each agent is rewarded based on the proximity to the nearest landmark, while it is penalized when colliding with other agents.

Cooperative Treasure Collection This cooperative environment involves N total agents, which are treasure hunters, M total treasures, which are colored purple or blue, and 2 banks, each of which is painted purple or blue. The role of the hunters is to collect the treasure of any color, which respawn randomly upon being collected, and then deposit the treasure into the correctly colored bank.

Cooperative Lane Changing We evaluate our model and other state-of-the-art approaches on the problem of learning cooperative policies for negotiating lane changes among multiple autonomous vehicles in the highway environment. We extend this environment so that it could meet our need for the experiment.

2.3.2 Results and Analysis. Our model and the models to be compared are suitable for experiments with continuous observations but discrete actions. So our experiments are meaningful. Models like DDPG that are well-known for their good effects but suitable for continuous actions are not used for comparison. Our experimental indicator is called normalized mean episode rewards, as shown in Figure 2. We only consider the total rewards of all agents, regardless of the reward of a single agent. We test models every 100 episodes, testing 10 episodes each time, and taking the average reward as an indicator.

3 CONCLUSIONS

We presented a general framework called communication-filtering actor-critic in cooperative multi-agent environment. Communication is indispensable in the context of cooperation, and we have adopted a semi-centralized approach to achieve effective communication. By comparing with state-of-the-art models, our model's experimental performance is still satisfactory. It can be seen from experiments that our communication model of partial observation can even be comparable to the complete observation model.

REFERENCES

[1] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary dynamics of multi-agent learning: a survey. *Journal of Artificial Intelligence Research* 53 (2015), 659–697.

[2] Romain François Cailliere, Samir Aknine, and Antoine Nongaillard. 2016. Multi-Agent Mechanism for Efficient Cooperative Use of Energy. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1365–1366.

[3] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2013. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2013), 427–438.

[4] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[5] Shariq Iqbal and Fei Sha. 2018. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:1810.02912* (2018).

[6] Alireza Janani, Lyuba Alboul, and Jacques Penders. 2016. Multi-agent cooperative area coverage: case study ploughing. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1397–1398.

[7] Jiechuan Jiang and Zongqing Lu. 2018. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*. 7265–7275.

[8] Philipp Kulms, Nikita Mattar, and Stefan Kopp. 2016. Can't do or won't do?: Social attributions in human–agent cooperation. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1341–1342.

[9] Gustavo Malkomes, Kefu Lu, Blakeley Hoffman, Roman Garnett, Benjamin Moseley, and Richard Mann. 2017. Cooperative set function optimization without communication or coordination. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1109–1118.

[10] Laëtitia Matignon, Laurent Jeanpierre, and Abdel-Illah Mouaddib. 2012. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*.

[11] Liviu Panait and Sean Luke. 2005. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems* 11, 3 (2005), 387–434.

[12] Manisa Pipattanasomporn, Hassan Feroze, and Saifur Rahman. 2009. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*. IEEE, 1–8.

[13] Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*. 2244–2252.