An Analysis of Connections Between Regret Minimization and Actor Critic Methods in Cooperative Settings

Extended Abstract

Chirag Chhablani
University of Illinois at Chicago
Chicago, Illinois, USA
cchhab2@uic.edu

Ian A. Kash
University of Illinois at Chicago
Chicago, Illinois, USA
iankash@uic.edu

ABSTRACT

Counterfactual Multi-agent Policy Gradients (COMA) is a popular algorithm for learning in cooperative multi-agent reinforcement learning settings. COMA computes difference rewards to solve the multiagent credit assignment problem by providing a local learning signal for each agent. Similar to other popular Cooperative multiagent RL (MARL) algorithms, there is a lack of theoretical justification for COMA's empirical success and specific way of doing credit assignment using difference rewards. We provide such a justification by connecting COMA's update rule to regret minimization. We then use this connection to improve COMA's performance by replacing usual softmax update with Neural Replicator Dynamics update from regret minimization literature. Experimental results on Starcraft II maps show the relevance of these theoretical insights for the performance of COMA in practice.

KEYWORDS

Cooperative stochastic game; Multiplicative weights update; COMA

ACM Reference Format:
Chirag Chhablani and Ian A. Kash. 2023. An Analysis of Connections Between Regret Minimization and Actor Critic Methods in Cooperative Settings : 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

Cooperative multi-agent reinforcement learning (Coop-MARL) is a framework for many complex real-world reinforcement learning problems such as the coordination of autonomous vehicles [2], network packet delivery [16], etc. *Counterfactual Multi-Agent Policy Gradients (COMA)* is a recent technique for learning cooperation among agents [7] which showed early empirical success in popular cooperative multiagent learning benchmarks like StarCraft II where each agent has to cooperate with the other agents to maximize the single shared reward when each agent only has partial access to the state of game. Key to COMA's success is efficient multiagent credit assignment through the implementation of *difference rewards* which were proposed by Wolpert and Tumer [15] and Tumer and Agogino [14]. COMA uses difference rewards to evaluate the contribution of each agent's actions by comparing with the expected value of actions based on its current policy. For each agent, the difference reward signal represents the advantage of including the agent in the system compared to the counterfactual case when it is excluded from the system. This individual advantage signal is called the *counterfactual advantage baseline*. Despite good performance, there is lack of theoretical justification for *this particular* choice of baseline. We argue that this baseline works well because it is similar to minimizing regret in a cooperative setting. This allows us to connect a variant of COMA's update rule to the classic regret minimization algorithm Hedge.

Beyond the conceptual contribution of this high-level justification for COMA, we use this connection to argue that the current version of gradient update rule of COMA can lead to slow learning. This problem, and a solution to it known as Neural Replicator Dynamics (NeuRD) update. The NeuRD update has previously been empirically examined in stateful competitive settings [8] and has good convergence properties in stateless settings. In our experimental results, we show that such an update rule leads to accelerated learning and higher performance in popular benchmark game environments StarCraft II. In the full version of the paper we also justify COMA's use of bounded softmax using the previous known properties of Hedge Algorithm in bandit settings.

2 NEW INTERPRETATION AND ANALYSIS OF COMA

To calculate its difference rewards, COMA uses a centralized critic which calculates the counterfactual advantage baseline A^a. A^a compares the value of $Q(s, \mathbf{u})$ and the baseline $\sum \pi(u'^a|\tau_a)Q(s, (u'^a, \mathbf{u}^{-a}))$ for agent a and it's action u^a. However, the correctness proof for COMA is independent of the choice of baseline, *leaving no theoretical justification* for this particular choice other than by analogy to prior successes of difference reward approaches. We show that, in the special case of cooperative stochastic games, the credit assigned to each agent through the advantage value is equivalent to the regret of the agent. Based on this connection, we provide an interpretation of and justification for COMA's difference reward implementation in terms of regret minimization in cooperative settings as Theorem 1. To be able to state the theorem, we begin by introducing a Tabular COMA, and it is given in Algorithm 1.

Tabular COMA has two changes. First, to aid in making the connections to regret minimization explicit we use an all-actions update rule rather than solely updating the action taken as COMA does. Second, we assume a softmax policy parameterized by a tabular representation where each parameter gives the logit of the policy for that action and agent. This leads to the given update rule for policies [8, equation (6)]. Tabular COMA is the natural all-actions implementation of COMA in the setting of stateful identical interest game. Hennes et al. [8] derive essentially the same algorithm in

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.

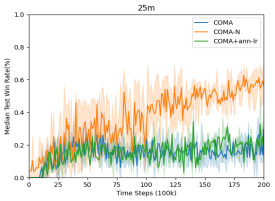

Figure 1: 25m

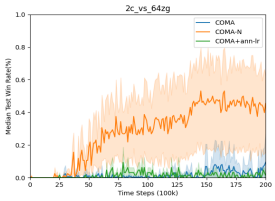

Figure 2: 2c-vs-64zg

stateless general-sum games from Softmax Policy Gradient. As they point out, algorithms like Tabular COMA do not quite match up with regret minimization. The issue is the inclusion of the $\pi^a(u_k)$ term in the update for $A^a_{sum}(u_k)$ in Algorithm 1. To exactly match up with the Hedge algorithm, it should instead be omitted yielding

$$A^a_{sum}(s, u_k) \leftarrow A^a_{sum}(s, u_k) + \eta A^a(s, u_k) \tag{1}$$

We refer to Algorithm 1 with the update according to Equation (1) as Tabular COMA-N, with the "N" representing the inclusion of this "NeuRD fix."With this variant, we can make a precise connection between COMA's difference rewards and regret minimization.

Algorithm 1: Tabular COMA update for an agent **a** having **K** actions

Result: Update policy for agent a given by π^a at state s
Sample joint action $\mathbf{u}^{-a}$ from other agents' policy $\pi(\mathbf{u}^{-a})$;
for k = *1 to* K **do**
  $A^a(s, u_k) \leftarrow Q(s, u^a_k, \mathbf{u}^{-a}) - \sum_{u'} \pi^a(u') Q(s, u', \mathbf{u}^{-a})$
  $A^a_{sum}(s, u_k) \leftarrow A^a_{sum}(s, u_k) + \eta \pi^a(u_k) A^a(s, u_k)$
  $\pi^a(s) \propto \exp(A^a_{sum}(s))$

Theorem 1. *Given a joint action* $(u, \boldsymbol{u}^{-a})$*, Tabular COMA-N is equivalent of running to a copy of Hedge at every state* s *with* $Q(s, u, \boldsymbol{u}^{-a})$ *as the reward for agent* a*.*

Proof Sketch (See full paper for definition of Hedge).

$$\begin{aligned}
\pi^a_t(s) \propto exp(R^t_{sum}(u, a)) &= exp\left(\sum_{\tau=1}^{t} \eta_\tau Regret^t(a, u)\right) \\
&= exp\left(\sum_{\tau=1}^{t} \eta_\tau \left(Q(s, u, \mathbf{u}^{-a}) - \sum_{u'} \pi^{a,\tau}(u') Q(s, u', \mathbf{u}^{-a})\right)\right) \\
&= exp\left(\sum_{\tau=1}^{t} \eta_\tau A^a(s, u)\right) = exp(A^a_{sum}(s, u)) \propto \pi^a(s)
\end{aligned}$$

□

Theorem 1 makes a conceptual contribution by establishing the equivalence of two concepts which have previously been explored separately in games with identical interests: difference rewards and regret minimization. In doing so it also connects to the rapidly growing literature on algorithms that learn in stateful settings via a collection of regret minimizers [1, 4–6, 9]. In particular, Tabular COMA-N can be viewed as a variant of LONR [10], so Theorem 1 combined with the general convergence guarantees of COMA provides a novel extension of convergence guarantees for LONR-style algorithms from MDPs to a stateful multi-agent setting. In the full version of this paper, we also show a richer connection between COMA and regret minimization literature by connecting bandit COMA and ϵ-Hedge[3] which also explains the empirical success of COMA.

3 EVALUATION

In this section, we analyze the importance of the NeuRD fix in stateful settings. We note here that, our goal is to demonstrate the relevance of our theoretical analysis to COMA, not attain state-of-the-art results and so similar to other works on improving COMA[11, 12], our experiments only include COMA and its variants. To implement the NeuRD fix, we use the following update rule for actors.

$$\Delta\theta^a = \Delta\theta^a + [1/\pi^a(u|h^a_t)]\hat{\nabla}_{\theta^a} v^a(\theta^a) A^a(s_t, \mathbf{u}) \tag{2}$$

For the implementation of COMA, we use the repository provided by Foerster et al. [7].[1] For COMA-N, we threshold the range of allowable logits using the same implementation as the OpenSpiel implementation of NeuRD[2]. This thresholding is described by Hennes et al. [8] as a way to prevent infinite gradients and our testing confirms that performance without it is poor. We present the results on *25m* and *2c-vs-64g* maps from the StarCraft Multi-Agent Challenge (SMAC)[13]. In the full version of the paper we present results on broader range of maps and environments.

3.1 StarCraft Multiagent Challenge

SMAC is built on the popular real-time strategy game StarCraft II. It introduces challenges like partial observability, decentralized execution, credit assignment and value assignment for joint actions. COMA-N outperforms COMA when both algorithms are run for 5 independent runs and improves COMA's performance on hard RL environments. To stabilize the NeuRD policy gradient, we linearly annealed the value of the actor's learning rate to 1/5 or 1/10 of its initial value over the first $150K$ iterations to stabilize training. The training curves in Figures 1-2 show that the resulting algorithm generally improves performance, most notably on the harder map 2c-vs-64zg where COMA is known to perform poorly. To confirm that our results are due to the COMA-N and not the annealing of the learning rate, each plot also includes a version of COMA with this feature added (COMA+ann-lr); we found it had no significant effect.

4 CONCLUSION

We provided a new justification for COMA's update rule by connecting it to regret minimization in identical interest games. Based on this we showed that COMA should apply the NeuRD fix and provided a justification for COMA's use of a bounded softmax policy. We demonstrated the efficacy of COMA-N on variety of environments including StarCraft where it consistently outperformed COMA and was able to learn on the harder map 2c-vs-64zg where COMA fails.

[1] https://github.com/oxwhirl/pymarl
[2] open_spiel/open_spiel/python/algorithms/neurd.py

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2217023

REFERENCES

[1] Yu Bai, Chi Jin, Song Mei, Ziang Song, and Tiancheng Yu. 2022. Efficient Φ-Regret Minimization in Extensive-Form Games via Online Mirror Descent. *arXiv preprint arXiv:2205.15294* (2022).

[2] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2012), 427–438.

[3] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. 2017. Learning with bandit feedback in potential games. In *Proceedings of the 31th International Conference on Neural Information Processing Systems*.

[4] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2009. Online Markov decision processes. *Mathematics of Operations Research* 34, 3 (2009), 726–736.

[5] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Regret circuits: Composability of regret minimizers. In *International conference on machine learning*. PMLR, 1863–1872.

[6] Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. 2022. Kernelized Multiplicative Weights for 0/1-Polyhedral Games: Bridging the Gap Between Learning in Extensive-Form and Normal-Form Games. *arXiv preprint arXiv:2202.00237* (2022).

[7] 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*.

[8] Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duéñez-Guzmán, et al. 2020. Neural Replicator Dynamics: Multiagent Learning via Hedging Policy Gradients. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 492–501.

[9] Ian A Kash, Lev Reyzin, and Zishun Yu. 2022. Slowly Changing Adversarial Bandit Algorithms are Provably Efficient for Discounted MDPs. *arXiv preprint arXiv:2205.09056* (2022).

[10] Ian A Kash, Michael Sullins, and Katja Hofmann. 2019. Combining No-regret and Q-learning. *arXiv preprint arXiv:1910.03094* (2019).

[11] Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Haifeng Zhang, David Mguni, Jun Wang, Yaodong Yang, et al. 2021. Settling the variance of multi-agent policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 13458–13470.

[12] Yueheng Li, Guangming Xie, and Zongqing Lu. 2022. Difference advantage estimation for multi-agent policy gradients. In *International Conference on Machine Learning*. PMLR, 13066–13085.

[13] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[14] Kagan Tumer and Adrian Agogino. 2007. Distributed agent-based air traffic flow management. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 1–8.

[15] David H Wolpert and Kagan Tumer. 2002. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*. World Scientific, 355–369.

[16] Dayong Ye, Minjie Zhang, and Yun Yang. 2015. A multi-agent framework for packet routing in wireless sensor networks. *sensors* 15, 5 (2015), 10026–10047.