Difference Rewards Policy Gradients

Extended Abstract

Jacopo Castellini
Dept. of Computer Science
University of Liverpool
Liverpool, United Kingdom
J.Castellini@liverpool.ac.uk

Sam Devlin
Microsoft Research Cambridge
Cambridge, United Kingdom
Sam.Devlin@microsoft.com

Frans A. Oliehoek
Dept. of Intelligent Systems
Delft University of Technology
Delft, The Netherlands
F.A.Oliehoek@tudelft.nl

Rahul Savani
Dept. of Computer Science
University of Liverpool
Liverpool, United Kingdom
rahul.savani@liverpool.ac.uk

ABSTRACT

Policy gradient methods have become one of the most popular classes of algorithms for multi-agent reinforcement learning. A key challenge, however, that is not addressed by many of these methods is multi-agent credit assignment: assessing an agent's contribution to the overall performance, which is crucial for learning good policies. We propose a novel algorithm called Dr.Reinforce that explicitly tackles this by combining difference rewards with policy gradients to allow for learning decentralized policies when the reward function is known. By differencing the reward function directly, Dr.Reinforce avoids difficulties associated with learning the Q-function as done by Counterfactual Multiagent Policy Gradients (COMA), a state-of-the-art difference rewards method. For applications where the reward function is unknown, we show the effectiveness of a version of Dr.Reinforce that learns a reward network that is used to estimate the difference rewards.

KEYWORDS

Multi-Agent Reinforcement Learning; Policy Gradients; Difference Rewards; Multi-Agent Credit Assignment; Reward Learning

ACM Reference Format:
Jacopo Castellini, Sam Devlin, Frans A. Oliehoek, and Rahul Savani. 2021. Difference Rewards Policy Gradients: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Many real-world problems [23, 24, 27], can be naturally modelled as *cooperative multi-agent systems* [4]. Such problems have commonly been approached with *(deep) multi-agent reinforcement learning* (MARL) [3, 12, 13, 15]. In the paradigm of *centralized training with decentralized execution* (CTDE) [13, 19] agents use global information during training, but then only rely on local sensing during execution, thus avoiding the prohibitive complexity of a centralized solution [2, 18]. Multi-agent policy gradients (MAPG) [20] methods have become one of the most popular CTDE approaches [11, 14].

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

However, one key problem that agents face with CDTE that is not directly tackled by many MAPG methods is *multi-agent credit assignment* [7, 17, 25, 28]. With a shared reward signal, an agent cannot readily tell how its own actions affect the overall performance. *Difference rewards* [9, 10, 21, 26] were proposed to tackle this problem: agents learn from a shaped reward that allows them to infer how their actions contributed to the shared reward.

Counterfactual Multiagent Policy Gradients (COMA) [11] is a state-of-the-art algorithm that does this differencing with a learned action-value function $Q(s, a)$. However, there are potential disadvantages to this approach: learning the Q-function is a difficult problem due to compounding factors of bootstrapping, the moving target problem [16], and Q's dependence on the joint actions [6, 8].

To overcome these potential difficulties, we propose *difference rewards REINFORCE (Dr.Reinforce)*, a new MARL algorithm that combines decentralized policy gradients directly with differencing of the reward function. Additionally, we provide a practical variant, called Dr.ReinforceR, that learns a centralized reward network during training for settings where the reward function is not known upfront. Although the dimensionality of the reward function is the same as the Q-function, and similarly depends on joint actions, learning the reward function is a simple regression problem and it does not suffer from the moving target problem.

2 DIFFERENCE REWARDS POLICY GRADIENTS

If the reward function $R(s, a)$ is known, we can directly use difference rewards with policy gradients. We define the *difference return* ΔG_t^i for agent i as the discounted sum of the difference rewards $\Delta R^i(a_t^i|s_t, a_t^{-i})$ from time step t onward:

$$\Delta G_t^i(a_{t:t+T}^i|s_{t:t+T}, a_{t:t+T}^{-i}) \triangleq \sum_{l=0}^{T} \gamma^l \Delta R^i(a_{t+l}^i|s_{t+l}, a_{t+l}^{-i}), \quad (1)$$

where $\Delta R^i(a_t^i|s_t, a_t^{-i})$ is the difference rewards for agent i, computed using the aristocrat utility [26]:

$$\Delta R^i(a^i|s, a^{-i}) = R(s, a) - \sum_{b^i \in A^i} \pi_{\theta^i}(b^i|s_t) R(s, \langle a^{-i}, b^i \rangle). \quad (2)$$

To learn the decentralized policies π_θ, we follow a modified version of the distributed policy gradients [20] that uses our difference return, optimizing each policy by using the update target:

$$\theta^i \leftarrow \theta^i + \alpha \gamma^t \Delta G_t^i(a^i_{t:t+T}|s_{t:t+T}, a^{-i}_{t:t+T}) \nabla_{\theta^i} \log \pi_{\theta^i}(a^i_t|s_t). \quad (3)$$

This way, each policy is guided by an update that takes into account its individual contribution to the shared reward, and an agent thus takes into account the real value of its own actions [1].

In many settings however, complete access to the reward function to compute the difference rewards is not available. Thus, we propose Dr.ReinforceR, which is similar to Dr.Reinforce but additionally learns a *centralized reward network* R_ψ, parametrized by a vector ψ, that is used only during training to estimate the value $R(s, \langle a^i, a^{-i} \rangle)$ for every local action $a^i \in A^i$. The reward network is trained to reproduce the sampled reward value $r_t \sim R(s_t, a_t)$ by minimizing a standard MSE regression loss:

$$\mathcal{L}_t(\psi) = \frac{1}{2} \Big(r_t - R_\psi(s_t, a_t) \Big)^2 . \quad (4)$$

Although the dimensionality of the function $R(s, a)$ that we are learning with the reward network is the same as that of $Q(s, a)$ learned by the COMA critic, learning R_ψ is a regression problem that does not involve bootstrapping or moving targets, thus avoiding many of the problems faced with an action-value function critic.

We can now compute an estimated ΔR^i_ψ to use in (1) as:

$$\Delta R^i_\psi(a^i_t|s_t, a^{-i}_t) \triangleq r_t - \sum_{b^i \in A^i} \pi_{\theta^i}(b^i|s_t) R_\psi(s_t, \langle b^i, a^{-i}_t \rangle). \quad (5)$$

Lemma 1. *Using difference return ΔG^i_t as the learning signal for policy gradients in* (3) *is equivalent to subtracting an unbiased baseline $b^i(s_{t:t+T}, a^{-i}_{t:t+T})$ from the distributed policy gradients in [20].*

Proofs of Lemma 1 and convergence are available in [5].

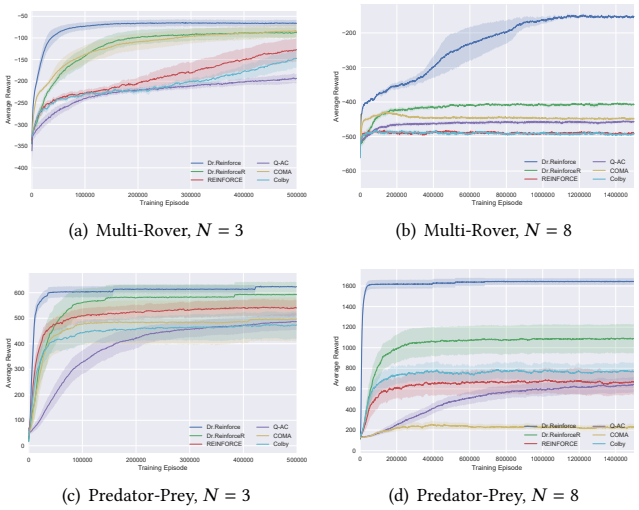
3 EXPERIMENTS

We are interested in investigating the following questions:

(1) How does Dr.Reinforce compare to existing approaches?
(2) How does the use of a learned reward network R_ψ instead of a known reward function affect performance?
(3) Is learning the Q-function (as in COMA) more difficult than learning the reward function $R(s, a)$ (as in Dr.ReinforceR)?

To investigate these questions, we tested our methods on the multi-rover domain [10], in which agents have to spread across a series of landmarks, and a variant of the classical predator-prey problem with a randomly moving prey [22]. We compare to a range of other policy gradients methods, including COMA [11] and an adaptation of the algorithm proposed by Colby et al. [9]. Additional results and analysis of the reward network are available in [5].

Multi-Rover Domain. While both COMA and Dr.ReinforceR easily learn good policies with three agents, when the system gets larger both begin to struggle, achieving sub-optimal performance. On the other hand, Dr.Reinforce learns high return policies. Given that this represents an upper bound to Dr.ReinforceR performance in case the reward network R_ψ learns a correct approximation, we can hypothesize that the gap in performance are due to a reward network: computing the difference rewards requires very accurate reward estimates, but the reward network may end up overfitting to the training reward samples and not exhibit appropriate generalization capabilities.



(a) Multi-Rover, $N = 3$ (b) Multi-Rover, $N = 8$

(c) Predator-Prey, $N = 3$ (d) Predator-Prey, $N = 8$

Figure 1: Training curves on the two problems, showing the mean reward and 90% confidence interval across 10 seeds.

Predator-Prey. In this environment, Dr.ReinforceR outperforms all the other methods, achieving performance that is equal or close to these of the Dr.Reinforce upper bound. In contrast, COMA struggles in learning something useful when more agents are introduced. This points out how accurately learning an optimal Q-function may be problematic in many settings: to compute the counterfactual baseline, estimates of Q-values need to be accurate even on state-action pairs that the policies do not visit often, rendering the learning problem more difficult. From this side, learning the reward function is an easier regression problem not involving bootstrapped estimates or moving target problems.

4 CONCLUSIONS

In cooperative multi-agent systems agents face the problem of figuring out how they are contributing to the overall performance of the team in which only a shared reward signal is available. We proposed Dr.Reinforce, a novel algorithm that tackles multi-agent credit assignment by combining policy gradients and differencing of the reward function. When the true reward function is known, our method outperforms all compared baselines and scales much better with the number of agents. For settings in which such reward function is not known, we proposed Dr.ReinforceR, that learns a centralized reward network used for estimating the difference rewards, which scales significantly better than COMA in the predator-prey benchmark.

ACKNOWLEDGEMENTS

This work was supported by an Azure for Research computing grant. This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

REFERENCES

[1] Adrian K. Agogino and Kagan Tumer. 2008. Analyzing and Visualizing Multiagent Rewards in Dynamic and stochastic Domains. *Autonomous Agents and Multi-Agent Systems* 17 (2008), 320–338.

[2] Craig Boutilier. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK '96)*. Morgan Kaufmann Publishers Inc., 195—-210.

[3] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (2008), 156–172.

[4] 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.

[5] Jacopo Castellini, Sam Devlin, Frans A. Oliehoek, and Rahul Savani. 2020. Difference Rewards Policy Gradients. *arXiv* abs/2012.11258 (2020), pp. 16.

[6] Jacopo Castellini, Frans A. Oliehoek, Rahul Savani, and Shimon Whiteson. 2019. The Representational Capacity of Action-Value Networks for Multi-Agent Reinforcement Learning. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'19)*. International Foundation for Autonomous Agents and Multiagent Systems, 1862–1864.

[7] Yu-Han Chang, Tracey Ho, and Leslie P. Kaelbling. 2003. All Learning is Local: Multi-Agent Learning in Global Reward Games. In *Advances in Neural Information Processing Systems 16*. MIT Press, 807–814.

[8] Caroline Claus and Craig Boutilier. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the 15th/10th AAAI Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence (AAAI'98/IAAI'98)*. American Association for Artificial Intelligence, 746–752.

[9] Mitchell K. Colby, William Curran, Carrie Rebhuhn, and Kagan Tumer. 2014. Approximating Difference Evaluations with Local Knowledge. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*. International Foundation for Autonomous Agents and Multiagent Systems, 1577–1578.

[10] Mitchell K. Colby, William Curran, and Kagan Tumer. 2015. Approximating Difference Evaluations with Local Information. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*. International Foundation for Autonomous Agents and Multiagent Systems, 1659–1660.

[11] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence (AAAI'18)*. AAAI Press, 2974–2982.

[12] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative Multi-agent Control Using Deep Reinforcement Learning. *Autonomous Agents and Multi-Agent Systems* (2017), 66–83.

[13] Landon Kraemer and Bikramjit Banerjee. 2016. Multi-Agent Reinforcement Learning as a Rehearsal for Decentralized Planning. *Neurocomputing* 190 (2016), 82–94.

[14] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 6379–6390.

[15] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. 2012. Independent Reinforcement Learners in Cooperative Markov Games: a Survey Regarding Coordination Problems. *Knowledge Engineering Review* 27, 1 (2012), 1–31.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-Level Control Through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533.

[17] Duc T. Nguyen, Akshat Kumar, and Hoong C. Lau. 2018. Credit Assignment for Collective Multiagent RL with Global Rewards. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8113–8124.

[18] Frans A. Oliehoek and Christoper Amato. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated.

[19] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. 2019. Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning. *arXiv* abs/1906.04737 (2019).

[20] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. 2000. Learning to Cooperate via Policy Search. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*. Morgan Kaufmann Publishers Inc., 489—-496.

[21] Scott Proper and Kagan Tumer. 2012. Modeling Difference Rewards for Multiagent Learning. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*. International Foundation for Autonomous Agents and Multiagent Systems, 1397–1398.

[22] Ming Tan. 1993. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Proceedings of the 10th International Conference on Machine Learning (ICML'93)*. Morgan Kaufmann Publishers Inc., 330–337.

[23] Kagan Tumer and Adrian Agogino. 2007. Distributed Agent-based Air Traffic Flow Management. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*. Association for Computing Machinery, pp. 8.

[24] Elise Van der Pol and Frans A. Oliehoek. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. In *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*. MIT Press.

[25] David H. Wolpert and Kagan Tumer. 1999. *An Introduction to COllective INtelligence*. Technical Report. NASA-ARC-IC-99-63, Nasa Ames Research Center.

[26] David H. Wolpert and Kagan Tumer. 2001. Optimal Payoff Functions for Members of Collectives. *Advances in Complex Systems* 4 (2001), 265–280.

[27] Dayon Ye, Minji Zhang, and Yu Yang. 2015. A Multi-Agent Framework for Packet Routing in Wireless Sensor Networks. *Sensors* 15, 5 (2015), 10026–10047.

[28] Logan Yliniemi and Kagan Tumer. 2014. Multi-Objective Multiagent Credit Assignment Through Difference Rewards in Reinforcement Learning. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer International Publishing, 407–418.