The challenge of redundancy on multi-agent value factorisation

Extended Abstract

Siddarth Singh
Instadeep Ltd
Johannesburg, South Africa
siddarthsingh2@gmail.com

Benjamin Rosman
The University of the Witwatersrand
Johannesburg, South Africa
benjamin.rosman1@wits.ac.za

ABSTRACT

In the field of cooperative multi-agent reinforcement learning (MARL), the standard paradigm is the use of centralised training and decentralised execution where a central critic conditions the policies of the cooperative agents based on a central state. It has been shown, that in cases with large numbers of redundant agents these methods become less effective. In a more general case, there is likely to be a larger number of agents in an environment than is required to solve the task. These redundant agents reduce performance by enlarging the dimensionality of both the state space and and increasing the size of the joint policy used to solve the environment. We propose leveraging layerwise relevance propagation (LRP) to instead separate the learning of the joint value function and generation of local reward signals and create a new MARL algorithm: relevance decomposition network (RDN). We find that although the performance of both baselines VDN and Qmix degrades with the number of redundant agents, RDN is unaffected.

KEYWORDS

Machine Learning, Reinforcement Learning, Multi-Agent Reinforcement Learning, Multi-Agent Systems

ACM Reference Format:
Siddarth Singh and Benjamin Rosman. 2023. The challenge of redundancy on multi-agent value factorisation: 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

For most multi-agent tasks in a practical setting, we would not know the precise number of agents required to optimally solve the problem. In general, there is likely to be a larger number of agents in an environment than is required to solve the task. As the number of independent agents increases so would the size of the state space that most problems require.

In complex environments constructing an accurate ground truth representation becomes difficult and in practical applications, the state representation data collected is often noisy or incomplete. Given the limitations of this space, it is not reliable to assume that the state space is reliable. When there is a large number of agents, many of them may be redundant for achieving an optimal policy. These redundant agents exacerbate the issue of the state space growth in the multi-agent setting. Therefore it is important to develop algorithms that can effectively separate agents that are essential to solving a task from the agents that are redundant to allow MARL algorithms to be deployed into more realistic and eventually real-world challenges.

We consider the idea of a collaborative task with a small margin of error like the traditional Piano Movers problem [1]. However, we consider the case where only n of m total agents in the environment are required to effectively complete the task. In the Piano Movers problem we therefore consider the case where an arbitrarily large number of agents are required to re-position the piano with each agent occupying a fixed amount of space in the environment along with the piano. Realistically with a large number of agents only a small subgroup n of m will be required to solve the task under the joint optimal policy. Ultimately during training, the policy will update to minimise the reliance on certain agents on the overall outcome as the joint optimal policy only require that they do not act in a manner that is destructive to the actions of the smaller group of required agents. In the Qatten [2] paper the idea is put forward that both VDN and Qmix exhibit poor performance in environments with a high number of redundant agents as it is difficult to assign credit for task completion accurately when a disproportionately high credit is assigned to only a small number of the agents.

In this paper, we propose a method to resolve this problem, relevance decomposition network (RDN) which makes use of layerwise relevance propagation (LRP) [3] as an alternative to learned value decomposition only using local agent observations. By not using learned decomposition we can separate the learning of the joint value function from the training of the independent agents and make full use of the relationships between the local observations of the agents, their identities and their actions at each timestep to decompose the relationship between the global and local rewards.

2 RELEVANCE DECOMPOSITION NETWORK (RDN)

We propose RDN to perform credit assignment between ad-hoc agents in the cooperative multi-agent setting under the assumption of a linear relationship between the individual local rewards and the shared global reward. However, unlike most central training decentralised execution (CTDE) methods which use learned decomposition as an end-to-end system, RDN separates the learning of the global reward function from the local Q-values of the agents [4].

The main motivation for RDN is that in difficult settings, where agents are not all similarly responsible for the total reward, Qmix and VDN see decreased performance [5][6]. In these cases to achieve accurate reward assignment, some agents must be weighted much higher than others when performing value decomposition [2]. In the

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.

case of many redundant agents, Qmix and VDN have difficulties decomposing the relationship between the high and low-value agents [7]. Due to this, the decomposed rewards have high variance during training which makes discovering stable policies difficult. LRP is more effective at determining this relationship, which stabilises the training process.

For RDN our independent agents are modelled as Deep Q Networks (DQNs) [8]. These DQNs gather data where $Q^i(o_{t,i,.})$ is the Q value of independent agent i at timestep t, h_t^i is the hidden state of agent i at timestep t, $o_{t,i,.}$ is the local observation of agent i at timestep t,$a_{t,i}$ is the action taken by agent i at timestep t and $\pi_i(Q^i(o_{t,i,.}), \epsilon(e))$ is the policy π of agent i dictated by a Q value function and an $\epsilon - greedy$ exploration strategy. We use the critic network to calculate the expected total Q value at each timestep parameterised by θ^c and a target Q value using a target critic parameterised by $\tilde{\theta^c}$ whose parameters are copied over from the critic at fixed intervals. The critic network is updated using loss $L(\theta^c) = E_{\vec{o},\vec{a},r,\vec{o}'}[(Q_{tot}^{\theta^c}(o_1, ..., o_n, a_1, ..., a_n) - y)^2]$ where $y = r + \gamma(Q_{tot}^{\theta'^c}(o_1^{'}, ..., o_n^{'}, a_1^{'}, ..., a_n^{'}))$ and θ^c is the critic's parameters and $\theta^{'c}$ is the target critic parameters, which are reset every C training epochs. To train the agents, we calculate the Q value at the current timestep for each independent agent. This is done in the same manner as standard independent Q-learning where we calculate $Q(o, a_i)$ for each agent based on their local observations and actions. We calculate the total target Q-value for the current timestep t using a separate critic network. This critic takes in the concatenation of local agent observations and actions to predict the target total Q-value of the global state at the current timestep then we use LRP to decompose the global reward calculated into local rewards for each agent. Essentially we assign relevance scores to all data points in the concatenated observation space and then separate the scores for each index based on the agent they were collected or generated from. The concatenated observation space is a concatenation of the local agent observations and the actions of each agent at each timestep. The relevance scores for each agent are then summed per agent and are used as the target Q-values to train the independent learners as if performing standard independent Q learning. Finally, the total expected Q value for each timestep is decomposed into independent target Q-values $\tilde{\theta^i}$ and the DRQNs [9] which act as the agent networks are trained using the loss $L(\theta^i) = E_{\vec{o},\vec{a},r,\vec{o}'}[(Q^{i,\theta^i}(o_i, a_i) - \tilde{Q}^i)^2]$ [10] and $\tilde{Q}^i = \sum_i R_{in}$ where $\tilde{Q}^i$ is the Q target for agent i and $\sum_i R_{in}$ is the sum of all relevance values associated with agent i.

A characteristic of LRP is it maintains a conservative calculation between layers of the neural network [3]. As such the relevance values from later layers are included completely in the calculation of the relevance values of the layers closer to the input. Essentially we can assume that the total sum of the relevance values is approximately the same as the output of the NN when LRP is used. Therefore we can equate Q_{tot} to the sum of relevance scores as $Q_{tot} \approx R_{in}$.

3 RESULTS

The pre-existing map from *the Starcraft Multi-Agent Challenge* (SMAC) we make use of is the *bane_vs_bane* map. In this map, each side has 20 zerglings and 4 banelings. The most optimal policy for this environment has the zerglings move out of the way to not obstruct the banelings' movement. We make use of 3 additional variants of the original map. *bane_med* which only has 15 zerglings, *bane_small* with has 10 zerglings and *bane_no_z* with has no zerglings. We vary the number of redundant agents (the zerglings) to show the effect of redundancy on performance.

From figure 1 we can see that across all maps RDN outperforms all baselines although the performance of VDN improves as the number of redundant agents is reduced. Interestingly QMIX is outperformed by QMIX_NS which does not use the global state in *bane_med* and in *bane _no_z* indicating that in cases where there are an intermediate number of redundant agents the central state already begins to become uninformative making accurate multi-agent credit assignment difficult.

In the case where all redundant agents are removed VDN performs similarly to RDN however, we also find that when no redundant agents are present at all in the *bane _no_z* map performance decreased across all algorithms when compared to *bane_small*. We suspect that having a small number of extra agents may aid in exploration early in the training regime.

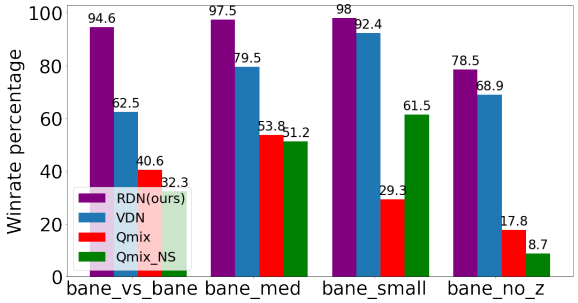

Figure 1: Percentage winrates from highest number of redundant agents (left) to least redundant agents (right) for all tested algorithms

4 CONCLUSION

We show how increasing numbers of redundant agents makes reaching stable convergence to an optimal joint policy difficult for both monotonic factorisation methods like QMIX and linear factorisation methods like VDN where only n of m total agents are required for an environment to be solved. We then propose RDN as a method that uses LRP to perform more optimal credit assignments in environments with high numbers of redundant agents using only local agent observation. RDN can reach near-optimal convergence on all environments used with similar overall winrates without the use of the ground truth state information. With VDN and QMIX we see a gradual decay in performance and, an increase in variance as the number of redundant agents increases.

REFERENCES

[1] J. T. Schwartz and M. Sharir, "On the "piano movers'" problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Communications on pure and applied mathematics*, vol. 36, no. 3, pp. 345–398, 1983.

[2] Y. Yang, J. Hao, B. Liao, *et al.*, "Qatten: A general framework for cooperative multiagent reinforcement learning," *arXiv preprint arXiv:2002.03939*, 2020.
[3] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.
[4] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.
[5] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *ICML*, 2018.
[6] P. Sunehag, G. Lever, A. Gruslys, *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18, Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, 2085–2087.
[7] M. Samvelyan, T. Rashid, C. S. de Witt, *et al.*, "The StarCraft Multi-Agent Challenge," *CoRR*, vol. abs/1902.04043, 2019.
[8] S. M. Schulze C., "Vizdoom: Drqn with prioritized experience replay, double-q learning and snapshot ensembling," in *Intelligent Systems Conference 2018 (IntelliSys 2018)*, London: Springer, Cham, 2018.
[9] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *arXiv preprint arXiv:1507.06527*, 2015.
[10] T. W. Sandholm and R. H. Crites, "On multiagent q-learning in a semi-competitive domain," in *International Joint Conference on Artificial Intelligence*, Springer, 1995, pp. 191–205.