Learning to Model Opponent Learning (Student Abstract)

Ian Davies,[1] **Zheng Tian,**[2] **Jun Wang**[3]
University College London
Gower Street
London, United Kingdom
WC1E 6BT
[1]ian.davies.12@ucl.ac.uk, [2]zheng.tian.11@ucl.ac.uk, [3]jun.wang@cs.ucl.ac.uk

Abstract

Multi-Agent Reinforcement Learning (MARL) considers settings in which a set of coexisting agents interact with one another and their environment. The adaptation and learning of other agents induces non-stationarity in the environment dynamics. This poses a great challenge for value function-based algorithms whose convergence usually relies on the assumption of a stationary environment. Policy search algorithms also struggle in multi-agent settings as the partial observability resulting from an opponent's actions not being known introduces high variance to policy training. Modelling an agent's opponent(s) is often pursued as a means of resolving the issues arising from the coexistence of learning opponents. An opponent model provides an agent with some ability to reason about other agents to aid its own decision making. Most prior works learn an opponent model by assuming the opponent is employing a stationary policy or switching between a set of stationary policies. Such an approach can reduce the variance of training signals for policy search algorithms. However, in the multi-agent setting, agents have an incentive to continually adapt and learn. This means that the assumptions concerning opponent stationarity are unrealistic. In this work, we develop a novel approach to modelling an opponent's learning dynamics which we term Learning to Model Opponent Learning (LeMOL). We show our structured opponent model is more accurate and stable than naive behaviour cloning baselines. We further show that opponent modelling can improve the performance of algorithmic agents in multi-agent settings.

In the context of multi-agent reinforcement learning, modelling an opponent can take many forms including inferring an opponent's motivation, representing an opponent through underlying characteristics and learning to predict an opponent's actions. Our work is concerned with action prediction, drawing from works on agent representation and meta-learning to model an agent's evolution throughout learning.

Previous works have considered adapting to a non-stationary opponent by learning a new policy once the opponent is perceived to have changed (Zheng et al. 2018). Such a setting requires the opponent to play a stationary policy while an effective response is learned. These prior approaches to playing with a non-stationary opponent do not consider the structure of the non-stationarity of an opponent. Our work aims to exploit the structure of an opponent's learning process to continuously adapt to a learning opponent. This is a fundamental and challenging issue in multi-agent reinforcement learning (Hernandez-Leal et al. 2017).

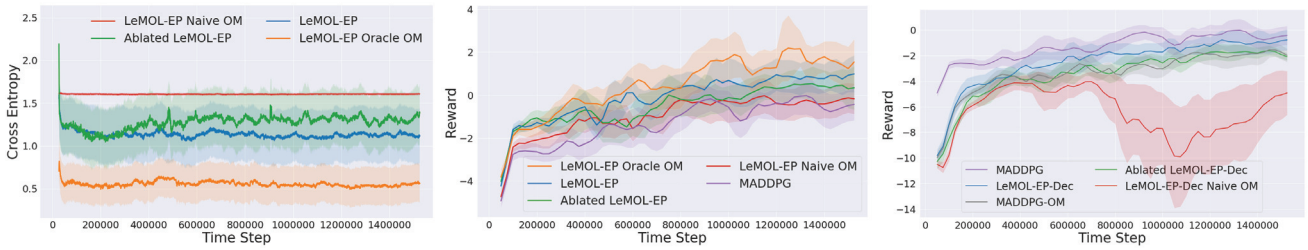
Conditioning an agent's policy upon the (predicted) action of an opponent stabilises policy updates. This follows from the update being specific to the gradient of the loss at a particular observation-opponent action pair. Accounting for the opponent's action means that, for different opponent actions the policy acts and is updated precisely.

In the decentralised setting, where the opponent's policy cannot be freely accessed to attain true actions, a sufficiently performant opponent model has the potential to overcome the loss of information from decentralisation and therefore enable decentralised training. Decentralisation through action prediction would be a key advancement in multi-agent reinforcement learning.

Methodology

We augment the centralised actor-critic architecture of multi-agent deep deterministic policy gradients (MADDPG) (Lowe et al. 2017) with a novel opponent model based on the meta-learning algorithm RL^2 (Duan et al. 2016). RL^2 is based on an LSTM network which stores the state of a task-specific agent in its activations. The role of the LSTM is to adapt the task-specific agent to a new task. The core LSTM is trained to learn a generalisable update rule for its hidden state which can replace closed-form gradient descent techniques for training on new tasks. The state update rule of the LSTM therefore becomes an optimisation algorithm trained on the performance of the agents it generates in varied environments.

We aim to emulate an opponent's learning rather than learn a generalisable optimisation technique. In light of this, our recurrent module stores and updates a representation of the opponent. The state update function is therefore trained to emulate the opponent's learning. This training is treated as a regression problem for predicting opponent actions from the observed history of the game. We utilise a method we term Episode Processing (EP) whereby each episode of experience is summarised by a bidirectional LSTM and is then used to update our agent's representation of its opponent.



(a) Prediction Cross-Entropy During Play (b) Centralised LeMOL-EP Results (c) Decentralised LeMOL-EP Results

Figure 1: Results from experiments with LeMOL-EP in the centralised and decentralised setting. Solid lines are averages (mean) from 15 runs. Shaded regions denote one standard deviation.

Experiments

We use the Open AI particle environments (Lowe et al. 2017) for experiments. Specifically, we focus on the two-player adversarial game Keep-Away.

Our LeMOL agents are endowed with an in-episode LSTM network to aid with the issue of partial observability. Our agents take on the role of the defender trying to keep the attacker away from the goal. The goal is one of two landmarks and the defender does not know which. The attacker is trained using MADDPG.

Our experiments compare our MADDPG baseline, our full LeMOL-EP model, LeMOL-EP where modelling of the opponent's learning process is removed (Ablated LeMOL-EP), LeMOL-EP where the opponent model has perfect prediction accuracy (LeMOL-EP Oracle OM) and LeMOL-EP where the opponent model is untrained (LeMOL-EP Naive OM). In the decentralised setting we also include a version of MADDPG with opponent modelling to make it amenable to the decentralised setting (MADDPG-OM).

Results

Comparison of opponent model performance for the full and ablated LeMOL-EP models in Figure 1(a) demonstrates the benefit, in terms of action prediction accuracy, of modelling the opponent's learning. Having a continuously updated opponent model improves and stabilises opponent model performance. Figure 1(b) shows the impact of improved opponent modelling on agent performance. We find that the reduction in the variance of policy updates resulting from conditioning an agent's policy on predictions of the opponent's actions improves overall agent performance.

In the decentralised setting (Figure 1(c)), we find that using the opponent model can enable effective decentralised training, as the opponent model compensates for the inability to access to others' policies in the decentralised setting. Note that the architecture of the opponent models is consistent across centralised and decentralised settings. Our decentralised model attains a similar level of performance to the centralised MADDPG agent. The opponent model is the only means of accounting for non-stationarity under decentralised training. Results are therefore highly sensitive to the accuracy of the opponent model. This is demonstrated by the instability and poor performance of the model with a naive (untrained) opponent model. This model collapses back to a single agent approach ignoring the opponent's presence.

We find that the more accurate an opponent model, the greater the improvement in agent performance. This is particularly pronounced in the decentralised setting where the increased opponent model accuracy and stability provided by modelling the opponent's learning process is essential to attain similar performance to centralised MADDPG.

Directions for Future Work

This work provides initial evidence for the efficacy of modelling opponent learning as a solution to the issue of non-stationarity in multi-agent systems. Furthermore, we have shown that such modelling improves agent performance over the strong MADDPG baseline in the centralised setting. When our approach is applied to decentralised training it achieves comparable performance to the popular centralised MADDPG algorithm.

Despite these promising results there is significant work to be done to extend and enhance the framework we develop for handling non-stationarity through opponent modelling. We hope to pursue a formal Bayesian approach to opponent learning process modelling in future. We hope such an approach will enable a theoretical framework to emerge which can be validated through further experiments.

References

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and de Cote, E. M. 2017. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Zheng, Y.; Meng, Z.; Hao, J.; Zhang, Z.; Yang, T.; and Fan, C. 2018. A deep bayesian policy reuse approach against non-stationary agents. In *Advances in Neural Information Processing Systems*, 954–964.