A Bayesian Approach to Learning Command Hierarchies for Coordination in Zero and Few-Shot Multi-Agent Scenarios

Timothy Flavin and Sandip Sen*

[1]The University of Tulsa
800 S Tucker Dr, Tulsa, OK 74104, USA
timmy-flavin@utulsa.edu, sandip-sen.utulsa.edu

Abstract

The challenge of developing Multi-Agent reinforcement learning (MARL) algorithms which can leverage very limited amounts of experience to coordinate with new teammates is a part of the more general problem of nonstationarity due to the dependence of environment dynamics' on agent's emerging policies. Recent works have focused on generating diverse training partners and using either a large single model which can adapt to all training partners, or a mixture of experts to choose from pretrained policies based on the observed actions of other agents. We propose a modular solution that can be added on to existing agents to learn a command hierarchy within a single episode so that a group of agents may approach the competency of the best agent in the group. We view learning to communicate as a set of non-stationary multi-armed bandit (MAB) problems where each agent has dedicated incoming and outgoing command MAB samplers that adjusts their policies. When giving commands, each agent's goal is to choose the subject which is most likely to follow their command. When receiving commands, each agent uses it's own estimate of advantage after having followed a command to decide whether to listen to this commander again in the future. We show that competent agents are able to quickly adapt to incompetent teammates by instructing while ignoring them whereas incompetent agents learn to defer to more skilled teammates. If pretrained agents are capable of sending or receiving commands before adding our communication structure, the agent's desired actions may be used as a prior distribution which will influence the MAB samplers to mitigate early exploration regret.

Introduction

Traditionally, MARL algorithms use centralized training / parameter sharing (Lowe et al. 2017; Terry et al. 2020), low learning rates, learned gradient-based communication (Zhu, Dastani, and Wang 2022), or a diverse set of training partners to address the problem of non-stationary environment dynamics (Terry et al. 2020) Training with diverse partners has shown success in few shot scenarios through opponent modeling (Albrecht and Stone 2018), and other play (Hu et al. 2020). Our approach serves as an augmentation to MARL algorithms like the ones above, or a replacement to them if diverse training partners are not feasible. Our algorithm grants agents the ability to adapt to novel policies using a human-like mental model of command-based communication. It uses the past experiences of trained models as a Bayesian prior combined with limited recent experience with specific teammates to create a model which is capable of quickly adapting to new individual teammates without risking policy collapse by quickly retraining large models.

We propose a communication structure between agents comprising of incoming and outgoing commands. Both incoming and outgoing command selections are treated as non-stationary multi-armed bandit problems (Kuleshov and Precup 2014). From the perspective of some agent $a_i \in \mathcal{A}$ where $\mathcal{A}$ is the set of agents in the environment, an outgoing command is a suggested action given by a_i to some other agent $a_j \in \mathcal{A}$. An incoming command to a_i is a suggested action from agent a_j to agent a_i. Selection of outgoing commands are modeled as a multi-armed bandit problem where the reward for instructing $a_j \in \mathcal{A}$ is 1 if a_j follows the command, and 0 if it does not. Note that an agent may instruct itself, if $i = j$. This allows an agent to command itself in the case that the other agents stop listening. This behavior is desirable if a_i is sending poor instructions. The outgoing MAB problem is necessarily non-stationary because teammates may change their probability of listening over time. The reward for incoming commands is based on the agent's estimation of advantage after following a command. While listening, agent's must estimate either a value function $V(s)$ or Q function $Q(s, a)$ or a way to query whether they think the command helped. Advantage, A, is calculated as either 1 or 2. Here, s_t refers to the state at time t and a_t refers to the action taken at time t. The discount factor γ accounts for uncertainty about future rewards, and r_t is the reward received at time t.

$$A = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{1}$$

$$A = r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \tag{2}$$

If an agent chooses to follow a command, the advantage will serve as an estimate of whether the command was better or worse than the agent expected to do on it's own. Agents will learn to listen to instructions coming from agents resulting in a positive advantage and they will learn to instruct agent's which follow them most often. A natural "command" hierarchy can emerge where agents listen to more competent teammates and instruct less competent ones.

[*] Representing The University of Tulsa

L \S	P1	P2	P3	P4
P1	0.97, **4%**	79, **66%**	201, **90%**	262, **97%**
P2	3.9, **5%**	0.26, **0.1%**	-18, **-5%**	6.4, **1%**
P3	106, **64%**	12, **3%**	7.1, **1%**	1.2, **0.2%**
P4	115, **60%**	-16, **-3%**	-1.6, **-0.3%**	-4.6, **-1%**

Table 1: Each cell represents both the absolute and percent difference when the agents used our algorithm to listen to another agent compared to listening randomly.

MAB Samplers and prior information

We used three families of MAB samplers to solve the incoming and outgoing communication problem where the hyperparameters prior strength and experience strength are used by a sampler to determine how quickly an agent will adjust from it's own policy to new information about a teammate.

$$\theta_t = \theta_0 + w_0 \hat{a}_t + w_1 r_t \tag{3}$$

The sampler estimate, θ_t, is a linear combination of an initial value, θ_0, a desired action generated by the pretrained agent $\hat{a}_t$ multiplied by the prior strength, w_0, and recent rewards for communication, r_t, multiplied by the experience strength, w_1. The first sampler used is a Thompson Sampler (Thompson 1936) based on the Dirichlet distribution(Riou and Honda 2020). For ϵ-greedy and UCB sampling, we used a constant learning rate to serve the non-stationary nature of the problem where the reward for pulling a given arm is the observed advantage. θ_t represents the set of alphas for the Dirichlet distribution in Thompson sampling, and the estimated mean rewards for each arm in ϵ-greedy and UCB.

Benchmark Environments

Cart pole Listener

We use the OpenAI Gym cartpole (Brockman et al. 2016) environment where one agent, the listener, is playing the game and another agent, the speaker, is giving instructions. For this experiment, we used four policies with mean scores of 22, 203, 484, and 498 respectively when playing with no communication. For each experimental run, we chose one policy to listen and another to command. We ran the cart pole problem for 100 episodes where the listener's prior experience with it's speaker is reset before each episode to measure zero-shot performance. We compared our algorithms performance to an algorithm which chooses randomly to listen to or ignore another agent with a probability of 0.5. The mean score of the four policies without no commands is 301. With random mixing, the score increases 355, and with our learning algorithm it increases to 447. In table 1 we present the actual and percent differences in average score between our algorithm and that with random mixing. For cases where policies are similar, our algorithm performs comparable to randomly listening, but when agent skills are different, e.g., with player 1 and 3, our algorithm significantly improves performance.

MARL Grid world

The second environment used here is an 8x8 grid world with four agents and consisting of paths and pits. Each agent receives a small negative reward for moving along a path and a large positive reward for reaching the exit. Additionally, agents are given a large negative reward and are forced to move randomly if they enter a pit. The game ends when all agents exit. Due to individual rewards, the value function for an optimal agent in this environment can be solved for using either policy or value iteration. We used 4 policies varying in quality from random to optimal: choosing optimal actions 0%, 30%, 80%, and 100% of the time and choosing randomly otherwise. Each agent gives a command to an agent of it's choice. If an agent is given multiple commands, it must choose one. For this experiment, we ran 5,000 trials with no communication, random communication, and learned communication. The mean reward for the team as a whole was -9.3 for no communication, -7.7 for random communication, and -4.3 for learned communication.

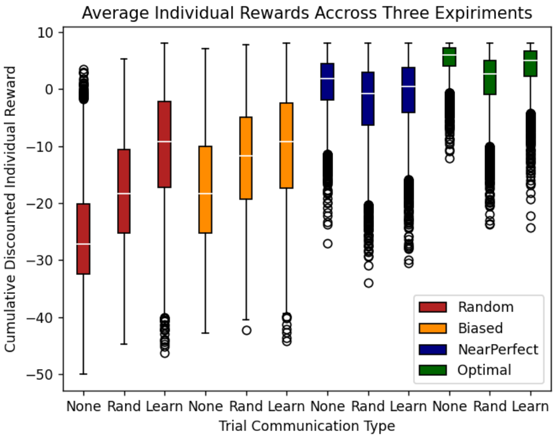

Figure 1: Policy rewards over experiments in Grid World.

Conclusion

The ability to send commands between agents has the potential to enable better coordination and team performance. Our command based sampling coordination framework allows for agents to learn desirable command protocols within a single episode of play with the aid of only a value function estimate. It works either with no prior knowledge, or as an extension of existing architectures that are capable of communication with tun-able parameters for how quickly an agent should adapt its prior beliefs. Our method shows the greatest team performance increases when the difference in competence between players is large and our algorithm protects competent players from listening to commands from ineffective agents. These initial results reinforce our belief that our learning to effectively use commands approach may be widely applicable to MARL environments with zero shot coordination requirements between teammates.

References

Albrecht, S. V.; and Stone, P. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258: 66–95.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. .

Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. 2020. “other-play” for zero-shot coordination. In *International Conference on Machine Learning*, 4399–4410. PMLR.

Kuleshov, V.; and Precup, D. 2014. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Riou, C.; and Honda, J. 2020. Bandit algorithms based on thompson sampling for bounded reward distributions. In *Algorithmic Learning Theory*, 777–826. PMLR.

Terry, J. K.; Grammel, N.; Hari, A.; Santos, L.; and Black, B. 2020. Revisiting parameter sharing in multi-agent deep reinforcement learning.

Thompson, W. R. 1936. On confidence ranges for the median and other expectation distributions for populations of unknown distribution form. *The Annals of Mathematical Statistics*, 7(3): 122–128.

Zhu, C.; Dastani, M.; and Wang, S. 2022. A survey of multi-agent reinforcement learning with communication. *arXiv preprint arXiv:2203.08975*.