COMP0036 – Literature Review:

Multiagent Reinforcement Learning for Noised Communication in Fully Cooperative MPEs

**Jeffrey Li**, supervised by **Professor Mirco Musolesi**

BSc Computer Science, UCL

November 17, 2022

# Project Background

* TBD
* Centralised Training, decentralized execution MARL algorithm for agents to learn communication protocols to effectively communicate over noisy channels, such that agents can achieve coordination tasks in selected Multi-Particle Environments.
* Assumptions and constraints …

# Background Knowledge

## 2.1 Markov Decision Processes

Markov Decision Processes (MDPs)[[1]](#footnote-1) are classical formulations of sequential decision-making, which seek to abstract the problem of goal-directed learning from interactions. This set of frameworks consists of two components: the agent being the decision maker and the environment in which the agent interacts with.

At each timestep , the agent would receive a representation of the environment current state and the agent selects an action where the action space depends on the current state . In response to the action, at the next time step , the agent would receive a real number reward and would be in a new state . There is a probability of those values occurring at time , given values of current state and action:

Which characterize the environments dynamics and also satisfies the law of total probability for each choice of and :

From this, we could also observe that the probability of each value of and depend entirely on the state and the action of the agent one timestep before. In other words, Markov Property states that the future depends completely on the present and not on the past, and any scenario which follows this property are defined as Markovian.

## 2.2 Reinforcement Learning

### 2.2.1 Problem Setting

Reinforcement Learning (RL) sits within the MDP framework, and the goal is the maximize the expected value of the return (cumulative sum of rewards). More precisely, it is maximizing the expected value of the discounted return with discount factor to take into account of continuous tasks which may have infinite .

This expected return is represented as value functions, which defines the expected return of an agent at a particular state, while following certain agent behaviours that are governed by what is known as policies. Policies are functions that maps from state to the probability of selecting each action , denoted as . Thus the value function at state while following policy has the following definition:

Value of a state can also be considered in conjunction with a specific action instead of an expectation over all possible actions, called Q functions, which is more commonly used in many RL algorithms and can also be used to express value functions:

Since the goal is to maximize the expected return , a RL problem can be formulated as estimating value functions given a certain policy, and search for the optimal policy that leads to the expected return (defined by its value function ) greater or equal to any other policies.

### 2.2.2 Bellman’s Equations

### 2.2.3 Well-known algorithms

Value iteration, Policy Iteration

* Known environment dynamics

Q-learning

* Model Free
* Tabular approach

Policy gradient

* Actor Critic
  + Rather similar to GAN
  + Temporal Difference version of policy gradient
  + Contains an actor and critic
    - Actor decides which action should be taken
    - Critic inform the actor how good was the action and how it should adjust
  + Actor learns through policy gradient approach; critics evaluate the action produced by actor by computing the value function

## 2.2 Deep Reinforcement Learning

As shown in the Q-Learning method, the

To account for larger state spaces, RL algorithms using …

* Successes in DRL
  + Mastering the game of Go with deep neural networks and tree search.
* Deep Q Networks
  + Use of neural network
  + Experience Replay
  + V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

## Multiagent Reinforcement Learning

From the surface, it seems that Multiagent Reinforcement Learning is solving similar MDPs as RL but in multi-agent setting[[2]](#footnote-2)

More on explaining the multi-agent MDP settings…

Unlike single-agent reinforcement learning, which can only interact with its environment, in the multi-agent setting, agents would in addition be able to interact with other agents. This leads to a … namely competing with each other where all agents seek to accomplish a goal over the other agents like …, cooperation where agents would work together to achieve a common goal … or mixed cooperative competitive settings like … .

Cooperative AI[[3]](#footnote-3) using MARL has gained increasing attention over the years and radiates through a wide range of impactful fields, such as the AI Economist[[4]](#footnote-4) which aims to improve equality and productivity with AI-Driven Tax policies, the study of sequential social dilemmas which require multiple agents to learn policies that implement their strategic intentions[[5]](#footnote-5) and the study of reputation in cooperative systems[[6]](#footnote-6) to name a few, all of which can be integrated well into the study of strategic games and society protocols.

OpenAI agents playing hide and seek[[7]](#footnote-7)

* Multiagent bidirectionally coordinated nets for learning to play starcraft combat games.
  + Multiagent game playing

This is a very interesting problem to tackle that can be useful in practical applications such as setting up protocols for multiple robots to do security patrolling[[8]](#footnote-8),

An initial attempt to such problem is known as the Independent Q-Learning algorithm

Independent Q Learning variants

* Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems
  + Independent Q-Learning agents, shown in paper that they don’t perform well in multi-agent settings.
  + Policy of each agent changes during training, causing non-stationarity in the perspective of a single agent, preventing naïve approach of experience replay
    - <http://proceedings.mlr.press/v70/foerster17b/foerster17b.pdf>
      * introduces non-stationarity that makes it incompatible with the experience replay memory on which deep Q-learning relies
  + Violates Markov assumption for convergence of Q Learning
* <https://proceedings.neurips.cc/paper/2003/hash/e71e5cd119bbc5797164fb0cd7fd94a4-Abstract.html>
  + Attempts in inputting other agents’ policy parameters to Q function to overcome such non-stationarity
* <https://arxiv.org/abs/1702.08887>
  + using a multi-agent variant of importance sampling to naturally decay obsolete data
  + and conditioning each agent's value function on a fingerprint that disambiguates the age of the data sampled from the replay memory
    - Essentially indexing the experiences
* <https://ieeexplore.ieee.org/abstract/document/4399095>
  + Hysteric Q Learning
* VDN
* QMIX
* MAPPO

Policy Gradient

* General Policy Gradient
  + <https://proceedings.neurips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html>
  + Known to exhibit high variance gradient estimations
    - More so in multiagent context as agent’s reward depends on action of many agents, therefore reward conditioned only on agent’s own actions exhibits higher variance
* Extend policy gradient framework to Deterministic Policy Gradient (DPG) algorithms
  + Deep DPG is a variant of DPG where the policy and critic are approximated with deep neural networks
    - <https://arxiv.org/pdf/1509.02971.pdf?source=post_page--------------------------->
    - Also make use of experience replay as in DQN to stabilize the neural network
* <https://ojs.aaai.org/index.php/AAAI/article/view/11794> (COMA)
  + Counterfactual multi-agent policy gradients, uses centralised critic to estimate Q function and decentralised actors to optimise agents’ policies
  + Address the challenges of multi-agent credit assignment, uses a counterfactual baseline that marginalised out a single agent’s action while keeping the other agents’ actions fixed
  + Learns a single centralized critic for all agents
* <https://arxiv.org/abs/1706.02275> (MADDPG)
  + Actor critic policy gradient where the critic is augmented with extra information about the policies of other agents, while the actor only has access to local information. After training is completed, only the local actors are used at execution phase
  + Since the centralized critic function explicitly uses the decision-making policies of other agents, we additionally show that agents can learn approximate models of other agents online and effectively use them in their own policy learning procedure.
  + acting in a decentralized manner and equally applicable in cooperative and competitive settings.
  + learn a centralized critic for each agent, allowing for agents with differing reward functions including competitive scenarios
  + consider environments with explicit communication between agents
  + learns continuous policies

# Related work

This project focuses on communication in MARL context to establish efficient cooperation.

Some earliest attempts involve genetic algorithm to learn the languages for the predator agents in the predator-prey problem[[9]](#footnote-9), however such algorithm would not be scalable for larger problems.

Learning grounded cooperative communication protocols between agents

* <https://proceedings.neurips.cc/paper/2016/file/55b1927fdafef39c48e5b73b5d61ea60-Paper.pdf>
* <https://proceedings.neurips.cc/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf>
* usually only applicable when the communication between agents is carried out over a dedicated, differentiable communication channel.

Deciding when to communicate

* <https://arxiv.org/abs/1812.09755>
  + Extended from CommNet
  + Individualized Controlled Continuous Communication Model (IC3Net)
  + Controls communication with gating mechanism to decide when to communicate and uses individualized rewards for each agent to gain better performance and scalability
  + Applied to cooperative, semi-cooperative and competitive settings
* <https://arxiv.org/pdf/1902.01554.pdf>
  + SchedNet
  + Limited bandwidth
  + the agents share the communication medium so that only a restricted number of agents are able to simultaneously use the medium
    - to simulate state-of-the-art wireless network architectures
  + Learns to decide which agents should be entitled to  
    broadcasting their (encoded) messages, by learning the importance of each agent’s  
    partially observed information.

My project aims to tackle the communication protocol learning dimension within the bigger picture of cooperative MARL communication. As identified in the survey[[10]](#footnote-10) , it is commonly accepted by the research community in this field that there are two paradigms to learning a communication protocol, one that assumes continuous property of the communication channel and applies backpropagation through gradient descent for optimizing, and the other treating the channel as an extension to action space, and the communication protocol is learned through standard RL algorithms (Reinforced Communication Learning).

For the backpropagation approach, it has been discovered that it tends to converge quicker to better policies compared to the other framework, due to not needing any actual interactions with the message channel space. The state-of-the-art algorithm DIAL[[11]](#footnote-11) is a centralised training and decentralised execution algorithm which allows real valued messages to pass between agents at training, …

* <https://proceedings.neurips.cc/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf>
  + Centralised training, decentralised execution
  + DIAL approach, allows real valued messages to pass between agents during centralised learning allowing gradients to be pushed through the communication channel, and allows real-valued messages to pass between agents during centralised learning, thereby treating communication actions as bottleneck connections between agents. As a result, gradients can be pushed through the communication channel, yielding a system that is end-to-end trainable even across agents. During decentralised execution, real-valued messages are discretised and mapped to the discrete set of communication actions allowed by the task. Because DIAL passes gradients from agent to agent, it is an inherently deep learning approach.

Another well-known algorithm CommNet[[12]](#footnote-12) …

* <https://arxiv.org/pdf/1605.07736.pdf>
  + CommNet
  + Learns a shared Deep Neural Net that is shared across agents
  + Shared reward

On the other hand, for the other approach, it is more commonly applied to problems with the assumption that communication channels are discrete, meaning it would not be differentiable for backpropagation techniques to be applied for optimization. Thus, optimization of communication protocol is achieved via applying RL algorithms. The assumption is closer to real life …

Earliest algorithm developed … tabular Q Learning … and similarly in …

* <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5045926>
  + Make use of tabular Q Learning to solve the predator-prey task with communication
  + Similarly, in <https://scholar.google.co.uk/scholar?q=Efficient+Distributed+Reinforcement+Learning+through+Agreement,&hl=zh-CN&as_sdt=0&as_vis=1&oi=scholart>

However, as explained in previous sections, … not scalable… This led to the RIAL algorithm which is introduced in the same paper as DIAL. RIAL …

* <https://proceedings.neurips.cc/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf>
  + RIAL method makes use of deep Q Learning for better scalability to learn content of the message

Another algorithm that is worth mentioning is the DDRQN …

* <https://arxiv.org/pdf/1602.02672.pdf>
  + DDRQN
  + Discover communication protocols to solve multiagent learning problems based on well-known riddles (partially observable tasks)
  + Not really solving coordination problems
* parameter sharing is very popular in recent works, however, it assumes homogeneous learning models

However, the algorithm that is most similar to my approach is the DiffDiscrete algorithm[[13]](#footnote-13). The algorithm seeks to learn a communication protocol on a discrete communication channel with additive noise that is unknown to the agents. However, instead of using the Reinforced Communication Learning approach mentioned before, the algorithm uses a stochastic message encoding/decoding procedure that makes a discrete communication channel mathematically equivalent to an analogue channel with additive noise, then which gradients can be backpropagated for optimization.

What separates my approach from DiffDiscrete is that …

1. Howard, R. A. (1960). Dynamic programming and Markov processes. John Wiley. [↑](#footnote-ref-1)
2. <https://courses.cs.duke.edu/spring07/cps296.3/littman94markov.pdf> [↑](#footnote-ref-2)
3. A. Dafoe, Y. Bachrach, G. Hadfield, E. Horvitz, K. Larson, and T. Graepel, “Cooperative AI: Machines must learn to find common ground,” *Nature News*, 04-May-2021. [Online]. Available: https://www.nature.com/articles/d41586-021-01170-0. [Accessed: 27-Mar-2022]. [↑](#footnote-ref-3)
4. S. Zheng, “The AI economist: Improving equality and productivity with AI-driven tax policies,” *Salesforce Research*, 30-Jun-2020. [Online]. Available: https://blog.salesforceairesearch.com/the-ai-economist/. [Accessed: 30-Mar-2022]. [↑](#footnote-ref-4)
5. “Understanding agent cooperation,” *RSS*. [Online]. Available: https://www.deepmind.com/blog/understanding-agent-cooperation. [Accessed: 30-Mar-2022]. [↑](#footnote-ref-5)
6. Nicolas Anastassacos, Julian García, Stephen Hailes, Mirco Musolesi: “Cooperation and Reputation Dynamics with Reinforcement Learning”, 2021; [http://arxiv.org/abs/2102.07523 arXiv:2102.07523]. [Accessed: 30-Mar-2022]. [↑](#footnote-ref-6)
7. https://arxiv.org/abs/1909.07528 [↑](#footnote-ref-7)
8. “Collaborative security robots use multi-agent Reinforcement Learning,” *YouTube*, 15-Apr-2021. [Online]. Available: https://www.youtube.com/watch?v=7QDr\_zxQitU. [Accessed: 30-Mar-2022]. [↑](#footnote-ref-8)
9. <https://link.springer.com/chapter/10.1007/978-3-540-45173-0_29> [↑](#footnote-ref-9)
10. https://arxiv.org/pdf/2203.08975.pdf [↑](#footnote-ref-10)
11. <https://proceedings.neurips.cc/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf> [↑](#footnote-ref-11)
12. <https://arxiv.org/pdf/1605.07736.pdf> [↑](#footnote-ref-12)
13. <https://ojs.aaai.org/index.php/AAAI/article/view/6205> [↑](#footnote-ref-13)