Université libre de Bruxelles

Analyzing MARL algorithms in dynamic environments: Evaluating performance with an additional unknown element

Preparatory work for the master thesis -- MEMO-F-403

Promoter:
Yannick MOLINGHEN

Author:
Kevin VANDERVAEREN

Supervisor:
Prof. Tom LENAERTS

Abstract

#todo{Abstract}

Table of Contents

I Introduction

I.1 Background and Objectives

A Multi-Agent Reinforcement Learning (MARL) is a subfield of the Reinforcement Learning domain which focuses on the interaction between multiple agents in a shared environment. Through the recent years, an increasly amount of research has been conducted in this field to resolve issue that has arisen in the reel world [1], [2]. However, most of the research are done through simulations on environments which does not involve unknown elements in existing envirement. This thesis aims to evaluate the learning performance of MARL algorithms from a know environment with with proven working result, to an slightly modified modified envirement by adding an unknown elements. Under the suppervision of Prof. Tom Lenaerts, and advisor Yannick Molinghen, from the Machine Learning Group (MLG) of the Université Libre de Bruxelles (ULB).

Currently, the research is focused on the the environment of LLE (Laser Learning Reinforcement) which is a environment created by Yannick Molinghen based on the original game. The environment is a 2D world also known as grid world where a single or multiple agents will be interacting in a Cooperative manner. The goal of each individual agent is to reach an exit point while aquiring rewards (under the form of Gems) and avoiding obstacles.

The objective of the Master thesis is to develop a new feature in the LLE environment that was also includes in the original game of Oxen. Morever this feature has also a other objective which is to add a new element in the environment which are not included in the agents learning process and thus reevaluate the performance of a already fine tuned algorithms that is trained on the original environment and observe of any possible bottleneck that may arise from the addition of additional elements.

I.2 Notations and Definitions

Notations	Description
mathematical base notation:	
$f : X \times Y \rightarrow Z$	a function f that has a domain of $X \times Y$ and return value in Z
$\Pr(s' \mid s, a)$	the probability of transitioning to state s' from state s given action a
Markov Decision Process:	
s,s'	state
a	an action
r	a reward
S	the state space
A	the action space
t	discrete time step

Notations	Description
s_t	the state at time t
a_t	the action at time t
r_t	the reward at time t
$A(s)$	the action space available in state s
$T(s, a, s')$	the transition function from state s to state s' given action a
$T(s, a)$	the transition function from state s given action a
$R(s, a, s')$	the reward function from state s to state s' given action a
$R(s, a)$	the reward function from state s given action a
ρ_0	the initial state distribution
π	the policy
$\pi(s)$	the policy at state s
Multi-Agent Markov Decision Process:	
$\mathcal{A}$	the joint action space
$\mathcal{a}$	the joint action
$\mathcal{T}(s, \mathcal{a}, s')$	the transition function from state s to state s' given joint action $\mathcal{a}$
$\mathcal{T}(s, \mathcal{a})$	the transition function from state s given joint action $\mathcal{a}$
$\mathcal{R}(s, \mathcal{a}, s')$	the reward function from state s to state s' given joint action $\mathcal{a}$
$\mathcal{R}(s, \mathcal{a})$	the reward function from state s given joint action $\mathcal{a}$
A^i	the action space of agent i
a^i	the action of agent i
τ	a transition defined as $\tau = \langle s, \mathcal{a}, r, s' \rangle$

(end temporary place)

II State of the Art

II.1 Distributed artificial intelligence

!!(this section has content form the article “Cooperative Multi-Agent Learning The State of the Art” by [ref to article])!!

Distributed artificial intelligence (DAI) is the a field of study which is rising in the last two decades. which is mainly focused on the domain of distributed systems. A distributed system by the definition of [3] is “where a number of entities work together to cooperatively solve problems” . this kind of study is not new, it has been studied for a long time. But what is new is the rise of the internet and the multiple electronic devices that we have today. Which bring the need of a new field of study which is the DAI that simply is the study of the interaction between multiple artificial intelligence (AI) or agents in a distributed system.

II.1.1 Multi-Agent Systems vs. Distributed problems Solving

In the field of DAI, we can find two main subfields a more traditional one which is the Distributed Problem Solving (DPS) which us the paradigm of a divide and conquer. The DPS is a field which is focused on distributing the problem to independent slaves which are solving the problem independently. On the other hand, the Multi-Agent Systems (MAS) emphasizes on the interaction between the agents.

II.1.2 Multi-Agent Systems

In MAS there are few constraints that are imposed on the agents. such as even though the agents are working together to solve a problem in a same environment they are not able to share their knowledge of the envirement with each other they may only acces to the information that they have, in RL we often refer this as a local obsevation. This is a important point because if they were able to share their knowledge this would be able to simply syncronize their knowledge and solve this problem as a DPS problem if the problem need no interaction between the agents (`todo` may be more).

II.2 Multi-Agent Learning

The Multi-Agent Learning (MAL) (todo):

- use article that explain different MAS article to explain what is MARL
- new why MARL is intresting
- get the mollinghen article to explain the LLE environment
- explain why adding a new element in the environment is intresting
- explain LLE agent standard

II.3 Machine Learning

(todo)

- is this section needed for explaining base of ML and split between SL unsupervised and RL

II.3.1 Supervised Learning

Supervised Learning (SL) is a subfield of Machine Learning (ML) which focuses on the learning of a model from a set of labeled data. The goal of SL is to learn a function that maps as much as possible the entry data (something e.g image) to a outgoing data (or label e.g. a class of the image). The SL is often used in the field of computer vision or natural language processing. Where the goal is to get a model that is able to classify the data into a certain class based on the data that it has initially learned from training.

II.3.2 Reinforcement Learning

The domain of Reinforcement Learning (RL) is a subfield of Machine Learning (ML) which focuses on the interaction between an agent and its environment. Compared to supervised learning, no initial data is required for it to be able to learn. It mainly focuses on the idea of trial and error, agent by interacting with its environment will be aquiring or losing point set on predetermined rules. Thus the agents will be trying to maximize the number of points given that initial rules.

II.4 Single Agent Reinforcement Learning

II.4.1 Markov Decision Process

In a Single agent Reinforcement Learning (RL) the methodology used to model the environment is the Markov Decision Process (MDP). The MDP is a mathematical framework that is used to model the interaction between an agent and its environment. It is often used to represent the decision-making process of an agent in a stochastic environment. The MDP is a powerful tool that allows us to model the environment in a way that is easy to understand and analyze.

The Markov Decision Process (MDP) is often represented as a 5-tuple $\langle S, A, T, R, \rho_0 \rangle$ where the elements are:

- S is the state space
- A is the action space
- T is the transition function
- R is the reward function
- ρ_0 is the initial state distribution

One of the key properties of the MDP is that it based on the Markov property, which states that the future state of a system only depends on the current state and not on the previous states. In mathematical term this is often represented as:

$$\Pr(s_{t+1} \mid s_t, a_t) = \Pr(s_{t+1} \mid s_t, a_{t-1}, ..., s_0, a_0)$$

A another strenght is that by doing the reduction to a MDP we can abstract all sensory, memory and control aspects(ref rl: an introduction sutton and barto) to a simply 3 signal between the agent and the environment:

- the state s
- the action a
- the reward r

but also introduce key functions such as the Bellman equation which is used the markov properties to represent the relationship between the value of a state and the value of its successor states.

II.4.1.1 State

A ways to represent the environment is to use a state. A state is an abstract ways to decribe the joint information of all elements in the environment. we can use as exemple the game of tick-tac-toe where the representation of the board at a given time such as this image Figure 1 is a state. But a state is not only the representation of the board but also the information of the player turn. So a state is a representation of the environment at a given time. In the mathematical notation we usually use the notation s to represent a state, and S to represent the state space. The state space is the set of all possible states imagineable for a given environment.

- S is the state space of the environment
- s is a state in the state space given that $s \in S$ (s' may be used for a new state)
- s_t is the state at time t

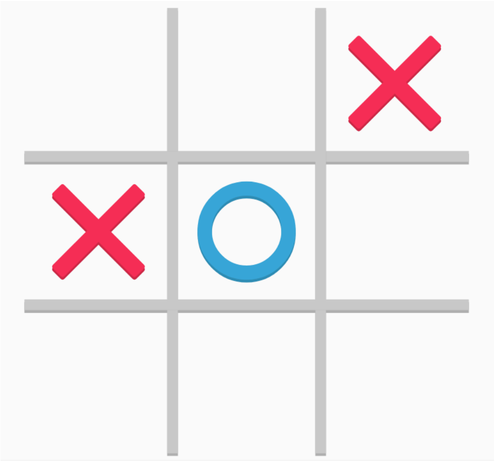

Figure 1: A state in the game of tick-tac-toe

II.4.2 Action

A action reffers to the possible movement doable by the agent in the environment. In the case of the game of tick-tac-toe, the possible actions is to put a mark in one of the available cell out the 9 cells, for exemple in the previous example Figure 1 the "O" player has the following possible action to choose from [top left, top center, middle right, bottom left, bottom center, bottom right]. In the mathematical notation we usually use the notation a to represent an action (e.g. 'top left'), and A to represent the action space (e.g list of all actions listed above).

- A is the action space of the environment
- a is an action in the action space given that $a \in A$

II.4.3 Transition

The transition is the function that is used to represent the change of a given state, given a action. The transition is a probability function that is used to represent the stochasticity of a given environment. A more real life example, for those who have done sport, you may have experience the case where you where about to do a certain action like a squat or a sprint but you got a cramp or a mucle tear which put you in a state where you were not expecting to be. This is a good example of the stochasticity of a given environment. If we use that example we can put it this way:

- s or s' is the state of my body which is "healthy"
- c is the state of my body which is "cramped" or "unhealthy"
- a is the action that I am about to do

and then the transition function T is the function that is used to represent the change of state of my body given a action. and in this case we can simply use this notation:

- $T(s, a, s')$ is the probability of having nothing happen to my body given a action a.
- $T(s, a, c)$ is the probability of having a cramp or a muscle tear given a action a.

they also posses certain properties such as:

- the function $T : S \times A \times S \to [0, 1]$
- $\sum_{s' \in S} T(s, a, s') = 1$

note that mathematically the transition function is a re-writing of the conditional probability function often represented as Pr(s'|s, a)

II.4.4 Reward

The reward function, which takes a initial state, an action and a final state. Unlike the transition is a function which return a probability the reward function return a scalar which can be interpreted as a score. Instead of representing the change of a state, the reward function is to give a purpose or goal to the agent. Going back to the example of the sport, the score can be seen as the motivation to perform the action based on a certain goal, such as on the treadmill when you are aiming to lose certain amount of calories, the reward function is the calories burned. While running faster put you in a state where you are burning more calories but also put your body in a state that is more likely to have a cramp. The reward function is often represented as: R(s, a, s') where s is the initial state, a is the action and s' is the final state. And mathematically the reward function is:

- $R : S \times A \times S \to \mathbb{R}$

II.5 Multi-Agent Reinforcement Learning

II.5.1 Stationary vs. Non-stationary

Originaly we can say that multiple independent agents may not increase dramaticaly in complexity from the RL with single agant but proven The MARL can be naively seen as adding more than one agent to the RL environment. This leads to new challenges such as non-stationarity, as the presence of multiple agents can change the dynamics of the environment.

II.5.2 Search space

III LLE Environment

III.1 Overview

The Laser Learning Environment (LLE) is a 2D grid world with discrete times and multiple cooperative agents. The game is based on the original game of Oxen, where the goal of each agent is to reach an exit point while acquiring gems (bonus points). All agents are cooperating to reach they respective exit point while avoiding obstacles. The envirement is designed to be simple and esay to understand, while still being challenging enough to test the performance of MARL algorithms.

III.2 Enviroment challenges

The envirement is aimed at testing the performance of MARL algorithms tailored for decentralized cooperative scenarios and possess some challenges that are not pressent in other envirement such as StarCraft Multi-Agent Challenge or SMAC [4] or the Hanabi environment [5]. Instead this envirement is designed to take into account other cooperating factors such as the perfect coordination, interdependence and the zero incentive dynamics[6].

III.3 multiagent Markov Decision Process

The model of the environment is based on the multiagent Markov decision process (MMDPs)[7] is a generalization of the Markov decision process (MDP) to multiple agents. The MMDP is a tuple $\langle n, S, \mathcal{A}, \mathcal{T}, \mathcal{R}, s_0, s_f \rangle$ where:

- n is the number of agents
- S is the set of states
- $\mathcal{A} \equiv A^1 \times A^2 \times \ldots \times A^n$ is the joint action space and A^i is the set of actions available to agent i[*])
- $\boldsymbol{a} \equiv (a^1, a^2, \ldots, a^n) \in \mathcal{A}$ is the joint action of all agents a^i is an action of agent i
- $\mathcal{T} : S \times \mathcal{A} \to \Delta_S$ is a function that gives the probability of transitionning from state s to state s' given a joint action $\boldsymbol{a}$
- $\mathcal{R} : S \times \mathcal{A} \times S \to \mathbb{R}$ is the function return the reward obtained by the transitioning from state s to state s' given a joint action $\boldsymbol{a}$
- $s_0 \in S$ is the initial state
- $s_f \in S$ is the final state

A transition is defined as $\tau = \langle s, \boldsymbol{a}, r, s' \rangle$ with $r \in \mathbb{R}$

III.4 Algorithm

Bases on the current state of the LLE environment, only a few algorithms where tested on the envirement [6].

[*] A^i has be modified from the original notation A_i to avoid confusion with the action space at a given time t

III.4.1 Value Decomposition Networks

The Value Decomposition Networks (VDN) [8] is a MARL algorithm that is

IV Objectives

The objective of this thesis is to develop a new feature in the LLE environment which consists of adding an lift which allow agents to have more possibilities of action. with this new feature, we aim to evaluate the performance of previously trained MARL algorithms on the original environment and observe if potential bottlenecks arise from the addition of this new element. The lift is designed to be used in conjunction with the lever, which is used to activate the lift.

IV.1 Lift and Lever

IV.1.1 Lift

The lift will be a terrain type that allows agents to reach higher levels in the environment. It is designed to be used in conjunction with the lever, which is used to activate the lift. The lift can be used to reach new areas of the environment, allowing agents to explore and find new paths to their goals.

IV.1.2 Lever

The lever is a terrain type which will be intercatible for the agents are on it. The lever is used to activate the lift, allowing agents on the lift to switch floors. The lever is designed to be used in conjunction with the lift, allowing agents to reach new areas of the environment.

IV.1.3 Plane extension

The plane extension is the addition of a new dimension which will allow the lift to move vertically…

IV.2 Evaluation

…

Bibliography

[1] G. Weiss, Ed., *Multiagent systems: a modern approach to distributed artificial intelligence*, 3. print. Cambridge, Mass.: MIT Press, 2001.

[2] P. Stone and M. Veloso, “Multiagent Systems: A Survey from a Machine Learning Perspective:,” Fort Belvoir, VA, Dec. 1997. doi: 10.21236/ADA333248.

[3] L. Panait and S. Luke, “Cooperative Multi-Agent Learning: The State of the Art,” *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, Nov. 2005, doi: 10.1007/s10458-005-2631-2.

[4] M. Samvelyan *et al.*, “The StarCraft Multi-Agent Challenge.” Accessed: Jan. 31, 2025. [Online]. Available: http://arxiv.org/abs/1902.04043

[5] N. Bard *et al.*, “The Hanabi challenge: A new frontier for AI research,” *Artificial Intelligence*, vol. 280, p. 103216, Mar. 2020, doi: 10.1016/j.artint.2019.103216.

[6] Y. Molinghen, R. Avalos, M. V. Achter, A. Nowé, and T. Lenaerts, “Laser Learning Environment: A new environment for coordination-critical multi-agent tasks.” Accessed: Jan. 31, 2025. [Online]. Available: http://arxiv.org/abs/2404.03596

[7] C. Boutilier, “Planning, Learning and Coordination in Multiagent Decision Processes.”

[8] P. Sunehag *et al.*, “Value-Decomposition Networks For Cooperative Multi-Agent Learning.” Accessed: Jan. 31, 2025. [Online]. Available: http://arxiv.org/abs/1706.05296