Reducing the Global Carbon Footprint based on MARL

Report for Mid Phase Evaluation of Minor-I Project

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Name of Supervisor:

Dr. Gagandeep Kaur

Submitted By:

Mansi Agarwal (17103042)
Parth Agarwal (17103060)
Rajat Kumar Garg (17103062)

This Project based on the research paper:
"Reducing the Global Carbon Footprint based on MARL" ~ *by Valentin Kahn*

Snippets from the Research Paper:

Abstract

This research paper intends to model the investment of groups of countries into carbon emission reductions based on a Mixed Markov Game setting, applying the principles of off-policy single-agent Reinforcement Learning to a multi-agent setting with a Markov Decision Process (MDP). The study shows that countries which are choosing their carbon emission reduction actions under the constraint of optimizing their and their partners' mid-term economic benefit, achieve both higher cumulative rewards and higher reductions in their per capita CO2 consumption, than their counterparts. It also shows that action choices considering immediate and future rewards for the individual agents, as well as the cumulative reward of all agents, converge to large reductions in the carbon emission level per capita of these countries. Multi-agent reinforcement learning (MARL) bears important problem-solving potential by modelling economic and political decision makers in simulated environments.

https://github.com/Valdini/Carbon-Footprint-Multi-Agent-Reinforcement-Learning/blob/master/Reducing the Global Carbon Footprint based on Multi-Agent Reinforcement Learning (MARL)_Valentin Kahn_School of AI.pdf

Keywords: reinforcement learning, game theory, climate change, global warming, global carbon emissions, global carbon footprint, multi-agent reinforcement learning, mixed markov games, markov decision process, q-learning, correlated equilibrium, temporal-difference learning

Problem Statement:

We are not discussing the impact and consequences of global warming in any detail and assume that undertaking measures to preventing global warming is reasonable and when talking about the Global Carbon Footprint, it is referring to CO_2.

Given the benefits of CO_2 reductions outweighing the cost and given the increased pressure on policy makers and carbon emitters – why is progress in CO_2 reductions not taking place in an accelerated manner? Arguably, because the magnitude of benefits many times is unclear and the benefits cannot be distributed to one party but the measures in general tend to benefit all actors, also passive ones, called "free-riders".

Entering the field of game theory might lead to a better understanding of the dynamics between the actors that are responsible for reducing CO_2 emissions and thus should support in drawing conclusions on what results these actors might achieve depending on the reward maximization scheme they are applying. **In game theory, a game in which individuals collectively deplete a common-pool resource is called the *"tragedy of the commons"*, here common-pool being the earth's capacity to absorb CO_2**

Overview of proposed solution:

This study models the behaviour of multiple agents representing groups of countries (categorised on the basis of being affected by the effects of global warming) in reducing their CO_2 consumption while taking actions selfishly but not competitively, in a mixed Markov game setting with a global state and correlated reward functions and policies, modelled based on Coordination Equilibrium Q-Learning (CE-Q) reaching a utilitarian equilibrium, and reinforcement learning. The goal of the Markov game is to overcome the "Tragedy of the commons", where multiple agents deplete a natural resource by purely maximizing their individual benefit.

Details:

Agents:

- Long-term impact agent (LT): Only affected in the longer term.
- Mid-term impact agent (MT): Slightly affected in the shorter term and equally affected in the longer term
- Small-term impact agent (ST): Already affected in the shorter term and equally affected in the longer term

There is one **global state** which is initialized to 4.97 //4.97 tons CO_2 per capita as of 2014

- Actions are defined as the per capita reduction in the CO2 consumption of the agents.
- **The action space is defined as** (-0.2, -0.16, -0.12, -0.08, -0.04, 0, 0.04, 0.08, 0.12, 0.16, 0.2).
- Agents can take actions in parallel.
- Agents have epsilon ε values for regulating exploitative-explorative trade-off and epsilon decay over time

The state transition function for a given period is defined as the global state before that period, minus the average per capita CO2 consumption reduction achieved by all agents in that period.

Three different policies emerge from training the agents in the simulation – selfish, greedy and utilitarian policies. These policies are compared regarding the cumulative reward they achieve, as well as the per capita CO2 consumption reduction they achieve, expressed by the final global state in the end period. In the selfish policy, the agents choose those actions which they have learned to maximize the sum of their respective individual immediate and future rewards. Under greedy policies, the agents choose those actions that maximize their respective individual immediate rewards in the respective period. Finally, under utilitarian policies, the agents choose those actions which maximize the cumulative reward over all periods.

Background Study - Literature Survey
1. Summary of Resources

Resource 1

TITLE OF PAPER	Can game theory help solve the problem of climate change?
AUTHORS	James Dyke
YEAR OF PUBLICATION	2016
PUBLISHING DETAILS	The Guardian blog Wed 13 Apr 2016 18.07 BST.Last modified on Wed 14 Feb 2018 17.16 GMT
SUMMARY	In game theory speak, man-made climate change can be cast as an iterated game over a common-pool resource that no one owns and everyone has access to. If it will manage properly than it can provide us but it can collapse also and these collapse are known as "**tragedy of the commons**". The **tragedy of the commons** is a situation in a shared-resource system where individual users, acting independently according to their own self-interest, behave contrary to the common good of all users, by depleting or spoiling that resource through their collective action. It is possible we can avoid it if incentives are right. For eg: Free rides but government doesn't make any rule and do not show any seriousness. Climate change seems to require a game to facilitate cooperation, but there is nothing to stop any individual nation picking up the board and scattering all the pieces if things aren't going their way. One approach is a bootstrapping process. It has been shown that cooperation can be enhanced if people repeatedly play the same game with the people. This is a simple and robust strategy. Finally a tipping point is reached in which limiting emissions becomes a social norm that is widely recognized.
Web Link	https://www.theguardian.com/science/blog/2016/apr/13/can-game-theory-help-solve-the-problem-of-climate-change

Resource 2

TITLE:	Economics and climate applications: exploring the frontier
AUTHORS	1.Debra J. Rubas, 2.Harvey S. J. Hill ,3. James W. Mjelde
YEAR OF PUBLICATION	2006
PUBLISHING DETAILS	1.United States Department of Agriculture, Foreign Agricultural Service, 1400 Independence Avenue SW, Stop 1034, Washington, DC 20250, USA 2. Agriculture and Agri-Food Canada, Prairie Farm Rehabilitation Administration, Room 1101, 11 Innovation Blvd., Saskatoon, Saskatchewan S7N3H5, Canada 3. Department of Agricultural Economics, Texas A&M University, College Station, Texas 77843-2124, USA
SUMMARY	Climate forecast issues are generally treated by economists as applied problems. Though applied studies are extremely important, we believe climate forecast issues have the potential for more innovative and rigorous treatment that could lead to theoretical advances in the economics of information. Several examples of researchable issues that we believe could lead to such advances (e.g. the use of climate forecasts in pollution trading; natural disaster mitigation) are then discussed. Maximum benefits can be attained from such a research process that involves policy-makers and end-users, as well as other scientist. the distinctions between decision theory, equilibrium modeling, and game theory are fuzzy. In decision theory, the decision maker's choice has no effect on anybody but him/herself. This is not the case in game theory. 'Game theory is concerned with the actions of individuals who are conscious that their actions affect each other' Game theory has not been widely used in applications of seasonal climate forecasts largely because of the increase in information requirements and increased methodological knowledge necessary to develop and solve games
Web Link	https://naldc.nal.usda.gov/download/28648/PDF

Resource 3

TITLE:	Self Learning AI-Agents: Markov Decision Processes
AUTHORS	Artem Oppermann
YEAR OF PUBLICATION	2018
PUBLISHING DETAILS	The Medium Article 14 October 2018
SUMMARY	A Markov Process is a stochastic process. It means that the transition from the current state ***s*** to the next state ***s'*** can only happen with a certain probability. *It* can be considered as an entry in a state transition matrix *P* that defines transition

	probabilities from all states ***s*** to all successor states ***s'***. Agent observes the current **State** of the **Environment** and decides which **Action** to take on basis of the current **State** and the past experiences. Based on the taken **Action** the AI Agent receives a **Reward.** The amount of the **Reward** determines the quality of the taken **Action** with regards to solving the given problem. The objective of an **Agent** is to learn taking **Actions** in any given circumstances that maximize the accumulated **Reward** over time.
Web-Link	https://towardsdatascience.com/self-learning-ai-agents-part-i-markov-decision-processes-baf6b8fc4c5f

Resource 4

TITLE:	Reinforcement Learning - Introducing Goal Oriented Intelligence
AUTHORS	Deeplizards
YEAR OF PUBLICATION	2018
PUBLISHING DETAILS	Published on DeepLizard.com on September 15, 2018
SUMMARY	The process of receiving a reward is an arbitrary function f that maps state-action pairs to rewards. At each time t, we have f(St, At) = Rt+1. For all s′ ∈ S, s ∈ S, r ∈ R, and a ∈ A(s), the probability of the transition to state s′ with reward r from taking action a in state s is p(s′, r| s, a)=Pr{St = s′, Rt =r | St−1 = s , At−1 = a} Rather than the agent's goal being to maximize the expected return of rewards, it will instead be to maximize the expected discounted return of rewards.The discount rate γ(0 to 1) discounts the future rewards and determine the present value of future rewards. $$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.$$ A policy is a function that maps a given state to probabilities of selecting each possible action from that state. , at time t, under policy π, the probability of taking action ***a*** in state ***s*** is π(a|s). Value functions are functions of states, or of state-action pairs, that estimate how good it is for an agent to be in a given state. The *state-value function* for policy π, denoted as vπ, tells us how good any given state is for an agent following policy π. $$v_\pi(s) = E_\pi[G_t \mid S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$$

Web-Link	https://deeplizard.com/learn/video/my207WNoeyA

Resource 5

TITLE	Navigating a Virtual World Using Dynamic Programming
AUTHORS	Siraj Raval
YEAR OF PUBLICATION	2017
PUBLISHING DETAILS	Committed, Add files via Upload on Nov 23,2018
SUMMARY	In Deterministic Environment the next state is completely determined by the current state but in Stochastic Environment, random elements decide how the state changes. Two fundamental methods for solving MDPs are policy-iteration and value-iteration algorithms. Value iteration computes the optimal state value function by iteratively improving the estimate of V(s). The algorithm initializes V(s) to arbitrary random values. It repeatedly updates the Q(s, a) and V(s) values until they converges. Value iteration is guaranteed to converge to the optimal values. Policy-iteration instead of repeated improving the value-function estimate, it will re-define the policy at each step and compute the value according to this new policy until the policy converges. Policy Iteration is generally faster than value iteration as policy converges more quickly than value function.
Web-Link	https://github.com/llSourcell/navigating_a_virtual_world_with_dynamic_programming/blob/master/Navigating%20a%20Virtual%20World%20Using%20Dynamic%20Programming.ipynb

Resource 6

TITLE:	Solving an MDP with Q-Learning from scratch — Deep Reinforcement Learning for Hacker
AUTHORS	Venelin Valkov
YEAR OF PUBLICATION	2017
PUBLISHING DETAILS	Originally published at curiousily.com on December 8, 2017
SUMMARY	Since our our environment is stochastic, The discount factor allows us to value short-term reward more than long-term ones, Our agent would perform great if he chooses the action that maximizes the (discounted) future reward at every step. Q function represents the “quality” of a certain action given a state. More formally, the function Q^π(s, a) gives the expected

	return when starting in s, performing action a and following π. Q-Learning is a model-free form of machine learning, in the sense that the AI "agent" does not need to know or have a model of the environment that it will be in. The same algorithm can be used across a variety of environments. In Q-learning equation have both previous and current value and α is the learning rate that controls how much the difference between previous and new Q value is considered.
Web Link	https://naldc.nal.usda.gov/download/28648/PDF

Resource 7

TITLE:	Reinforcement learning (RL) 101 with Python
AUTHORS	Gerard Martínez
YEAR OF PUBLICATION	2008
PUBLISHING DETAILS	The Medium Article 20 Dec 2018 1.01 GMT.
SUMMARY	we quantify how good states are if we calculate a function V(S*t*) Dynamic programming is one iterative alternative to a hard-to-get analytical solution. However, it suffers from the major flaw, which is the necessity of knowing the transition matrix as well as the rewards. This is not always possible in a real life problem. Monte Carlo method does not have to know the transition probabilities and the reward system beforehand but there is no guarantee that we will visit all the possible states, another weakness of this method is that we need to wait until the game ends to be able to update our V(s) and Q(s) Temporal Difference is better than Dynamic Programming method because it does not require a model of the environment, nor the rewards and probability distributions. TD has also advantage over Monte Carlo methods since no need to wait until the end of the episode to know the return, only one time step is required.
Web Link	https://towardsdatascience.com/reinforcement-learning-rl-101-with-python-e1aa0d37d43b

2. Integrated Summary of Literature Studied:

While a developing country in the midst of the sea rather tends to face a strong negative impact in a mid-term scenario of further rising sea levels, a landlocked and developed country with a moderate climate and a strongly service and industrial sector-oriented economic output might not face significant negative impact by climate change in the near future. Additionally, the economic operations of these developed and industrialised countries tend to have the largest impact on the climate, with these countries remaining least affected in the mid-term. In game theory, a game in which individuals collectively

deplete a common-pool resource is called the “tragedy of the commons”. The existence of this sort of game in climate change is currently fostered by the absence of a multinational regulatory system that would reward cooperation and punish free riding caused by a reward maximization scheme that is purely based on short-term self-interest, hence agents should be categorised on this basis. Now every time a certain action is selected for reduction the Climatic conditions come under a new state influenced by the previous one therefore the stochastic process.
Now choosing the method for processing, since we know about our environment and about the next state for each action taken, we can simply use one-step Temporal Difference, we learn at each and every step we take. This particularly powerful because: on one hand, the nature of learning is truly “online” and on the other hand we can deal with tasks which do not have a clear terminal state, learning and approximating value functions ad infinitum (suitable for non-deterministic non-episodic or time-varying value functions).

Q-Table is just a fancy name for a simple lookup table where we calculate the maximum expected future rewards for action at each state and is updated by

Multi-agent Q-Learning (MultiQ) is the concept of extending the Q-function by not just mapping all possible states with all possible actions of one agent, but with the potential combination of actions of all agents in the game.

Analysis, Design and Modelling

We have come to a conclusion that group of countries should be categorised solely on the basis of short-term self-interest, so we define 3 agents **Long-term**, **Mid-term**, **Short-term** (in increasing order of their short-term impact)

There is Global State Variable called “CO2 consumption of the world” which needs to be minimized.

Action Space (as given in paper): (-0.2, -0.16, …… ,0.16, 0.2) each unit refers to how much a group of countries reduces or increases CO2 consumption, for example (-0.2) means they have contributed in decrease per capita consumption.

```
LT_immediate_reward = -1 * LT_action * LT_reward_factor
```

Let there be constants for each agent called “Reward Factor”. Reward Factor when multiplied with reward of that action will yield reward for that action of that agent. Reward Factor of Long-Term will be less as compared to others as lesser RF means immediate actions do not matter much to LT.

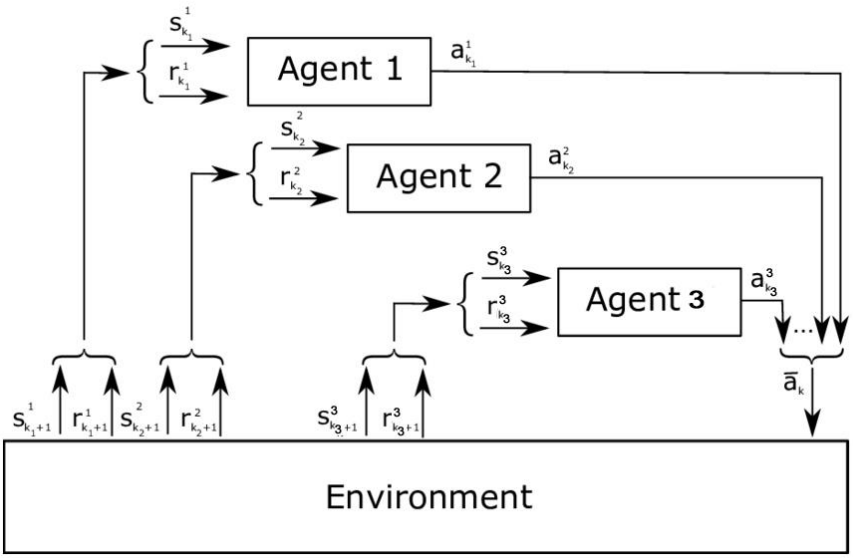

Fig1. Interaction between agents and environment

Action → Period ↓	0	.....	9	10
0		.....		
. . .	Each cell corresponds to reward given by performing that action in that period; This table fill-up over epochs.			. . .
p-1				
p		.....		

Fig2. Q-Table for each Agents

Q-function for each given state-action pair is denoted as:

$$Q*(s,a)=r(s,a)+\gamma * Q*(s',a)$$

It calculates Q-values, representing the reward r of an action a in a given state s, plus the maximum reward achievable by the best action a' in the next state s', discounted by the factor gamma γ (temporal-difference learning). Q* denotes the highest Q-value for a given row, thus defines the action in a given state that achieves the highest Q-value. The Q-table is updated after each iteration by:

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha * (r + \gamma * Q*(s',a))$$

Some of data that can be extracted during p periods over e epochs:

Number of epoch ↓	Cumulative Reward at that Epoch ↓
1	CR(1)
. . .	. . .
e-1	CR(e-1)
e	CR(e)

Fig3. List of Cumulative reward per epoch

```
cumulative_reward[epoch][1] += LT_immediate_reward + MT_immediate_rewar
d + ST_immediate_reward
```

Number of epoch ↓	Immediate Rewards at the end of that epoch ↓
1	IR(1)
. . .	. . .
e-1	IR(e-1)
e	IR(e)

Fig4. List of Immediate Rewards at the end of each epoch

Now from these tables three policies can be observed:

- Agent's Strategy to achieve Highest Cumulative Reward (Utilitarian)
  Find max of Fig3 and choose that epoch as Strategy

- Agent's Strategy to achieve Highest Immediate Reward (Greedy)
  Find max of Fig4 and choose that epoch as Strategy

- Strategy based on Agent's Final Q-Table (Selfish)
  Find max of Fig2 and choose that epoch as Strategy

Now use these Strategies to compute for 1 epoch and final comparison between these global state will yield results

Tools and Technologies:

Python: it is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

IPython (Interactive Python): it is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history.

Project Jupyter: Spun-off from IPython in 2014 by Fernando Pérez, It is a web-based interactive computational environment for creating Jupyter notebook documents. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media.

NumPy: it is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.