International Journal of Disaster Recovery and Business Continuity
Vol.11, No. 1, (2020), pp. 2378–2382

# Simulating Human-like Navigation in Urban Transportation Environments with Multi-agent Deep (Inverse) Reinforcement Learning

[1]Dr S. Praveen Kumar, ,[2]V.Sunil Kumar[3]G.V.S Narayana,[4]K.Sandeep Varma

*[1,2,3,4]Dept. Of CSE, GIT, GITAM (Deemed to be University),Vizag.*
*spkmtech@gmail.com, sunil.veee@gmail.com, gvsnarayana.mtech@gmail.com, varmagitamcse@gmail.com*

***Abstract:***
*This paper focuses on an approach that uses two methodologies: Maximum Entropy Inverse Reinforcement Learning and Multi-agent Deep reinforcement learning. It stress on each method in a sufficiently profound manner. The nature of data is also specified and it is also displayed how they prepared it to suit their purpose. The use of Keras as one of their tools to construct modest deep neural networks. For results, they found the relation between the outcomes of each method with proper explanation. They also stated the direction in which the approach can be led forward in the future.*
***Keywords:*** *keras, MaxEnt IRL, Non-Markov*

**Key Aspects:**
The objectives of this research are:
1. To find a reward function motivating RL agent characteristics in high-dimensional environments.
2. To utilize the inferred reward function to teach multiple agents adaptive navigation in urban traffic networks.

**Document Summary:**
Researchers have found complex, dynamic interactions in urban environments with random pattern games with the help of which numerous virtual agents learn to make decisions that are similar to a decision made by a human. Reinforced Learning (RL) had been applied in a successful manner to concepts which are derived from game theory

so as to model pragmatically, how groups of virtual agents have evolved strategically from their past experience have given affirmation that Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) is mainly a closely connected methodology, allowing improvements of a choice functions as the suboptimal policies displaying many a time irrational and irresponsible human behavior. Disappointingly, as with econometric methods, it is still held by the experts of its domain to define the useful characteristics of the reward function for the purpose of estimations. Even with the assistance of human experts, the over fitting behavior has high probability which might restrict the generalizability of policies that can be recuperated.

**Methodology:**
The methodology used is Maximum Entropy Inverse Reinforcement Learning and Multi-agent Deep reinforcement learning. Markov Decision Problems (MDPs) utilizes the reward function to be utilized in form of a guide to the agent the behavior and helps to persuade agents into policies that are best with the help of encoding suitable tunings. The act of learning of optimal policies of professional demonstrations is often acknowledged as the apprenticeship learning problem. The associated problem of developing the reward function of the behavior that is observed is identified as the inverse reinforcement learning problem (IRL) and has specified that their full details can be found in Markus Wulfmeier, Peter

ISSN: 2005-4289 IJDRBC

Ondruska, and Ingmar Posner," Maximum Entropy Deep Inverse Reinforcement Learning".For the Multi-agent Deep
reinforcement learning, this problem has been composed by them as a constant sum stochastic game. Its aim is to find ways to solve problems on equilibrium using multi-agent virtual systems which are necessary to precisely simulate routing in networks that are saturated. One of the methods is d Neural Fictitious Play (NFP).In this algorithm, there is a need to maintain the tracks of actions of other agents so as to enable the agent to react optimally to the strategies of the opponents. This algorithm is an extension of Q-learning with extra memory that traces the average behavior of the agent. They used a multi-agent NFP which is an extension of the original, to obtain a more stable performance, it is replaced with deterministic policy gradients. To do that, in Q-learning they replaced the single neural network of the algorithm with 2 networks namely the "critic" network and the "actor" network. The "actor" network to calculate the probability of actions by using a softmax function according to the state space input. The "critic" network to provide a score of the state space and action combination. It also makes the necessity of a third network to maintain the agent's average strategy, which is called as the "Tracking" Network. When an agent desires to make a movement, the "optimal" policy or the "average" policy must be first randomly chosen. if the "optimal" policy is chosen, the agent performs the movement as created by the "actor" and "critic" network. Else, the agent follows the movement as created by the "tracking" network. They use a strategy known as an epsilon-greedy strategy which includes a linearly decreasing epsilon for state space exploration for all the agents.

As major methods of Reinforcement Learning show a slow convergence when utilizing optimal solutions to real-time problems, this paper proposes stochastic shortest path-based Q-learning (SSPQL) which is a mixture of the stochastic shortest path-finding method with Q-learning. Their domain is on letting autonomous robots to perform a particular task with less resources and knowledge. They are determined to make the robot inherit: learning at fast speed and adaptability in a dynamic environment. The paper discusses an SSPQL algorithm that fuses both model-free and model-based RL methods. And then it compares the performance of this algorithm with other existing algorithms. The major idea is of producing an effective network routing which in turn selects optimal paths for communication. The adjustment of parameters for these depends on the performance of the network globally. Their approach is to update local policies and not having a central control or global knowledge of the network arrangement. They tested their algorithm on a domain which is a discrete-time simulator of communication networks with various different topologies and dynamic structures. The packets that are traveling in the network have certain cost parameters which are referred to as transition time and waiting time (both are set as unit cost).

$\mu(a, o, \theta) = \Pr a(t)=a \mid o(t)=o, \theta = P \exp(\theta oa/\Xi)\ a' \exp(\theta oa'/\Xi) > 0.$ -----(1)

Where '$\Xi$' is a temperature parameter.

$$V(\theta) = \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H_t} \Pr(h \mid \theta) r(t, h)$$

The result of following a policy $\mu$ with parameters $\theta$ is the anticipated cumulative reward

---(2)

RL has slow convergence and to overcome this some state-action pairs if the thought of as disturbance is discarded from the long-term memory (LTM), this has proved to enhance the learning speed. Also, it partly solves the Non-Markov problem, if a stimulus is recurring then it is classified as a sequential percept for a hidden Markov state. There are some shortcomings to RL when it's applied to real-time problems:

ISSN: 2005-4289 IJDRBC

- Rewards might not be given and generally are given after the goal is achieved.

- RL methodology might not work as needed for non-Markov environments.

They put forward a learning method that has a comparatively fast speed of convergence and can be applied to both non-Markov and Markov environments.

The markovian property is given by

$$\Pr(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \Pr(X_{n+1} = x \mid X_n = x_n)$$

----3

And the non-markovian property is given by

$$\Pr(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \neq \Pr(X_{n+1} = x \mid X_n = x_n)$$

----4

The system only learns when a reward signal is given by its environment, learning is progressed by checking what S-R (Situation-Response) behaviors should be added among the previous S-R behaviors for getting a reward.
In the proposed RL method, the process of internal reasoning is employed to decrease the probability of selecting the behaviors that are wrong. The RL is applied till a successful episode is completed and all the S-R values stored in STM (Short Term Memory) are transferred to be stored in LTM(Long Term Memory) where to find the path from initial S-R to goal, the shortest path algorithm is used. For exploration, they used Boltzman exploration and Q-learning which make the task of reason processing easy and can deal efficiently with non-Markov hidden states. A Finite State Machine (FSM) represents the LTM data and employs Dijkstra's Algorithm in order to find the shortest path.
Pseudo codes for reasoning, learning, generating Sequential Percept and overall learning are given.

To overcome the slow convergence, some of the states that were considered disturbances are removed from LTM to achieve speed in convergence. And in order to identify non-Markov hidden states, a sequential percept generation mechanism is used. Their findings were proved with experimental simulations of sequence learning. A framework is used to add an emotional model into the decision making process of a machine learning agent Which uses a hierarchical structure to fuse reinforcement learning with a dimensional emotional model. The dimensional model calculates two dimensions that represent the actual affective state of the autonomous agent. To calculate their combination model, they chose the grid-world example Dyna Maze. An agent can take up to four possible moves (up, down, right, left). A deterministic world is assumed for the proper execution of an action. For selection and learning of actions, they used the Dyna-Q algorithm.

The emotional model is designed as a three-dimensional PAD model (pleasure P, arousal A, dominance D).For making it simple and for a current scenario D is dropped making it 2-dimensional. A point in this two-dimensional space with a process that is time-varying is used for the effect of an agent. There are three different levels of the architecture: reaction level (Lowest level), routine level(Middle level), reflection level (Highest level).

The reaction level performs the state transitions of the agent within the maze, detects collisions with obstacles and walls and receives the reward from the environment.

ISSN: 2005-4289 IJDRBC

In routine level - There are two steps that the reinforcement model must perform :

1.Updating the action-value function (Update Q-learning).

2. Selecting the following action according to a routinized policy (e.g. epsilon- Greedy Action Selection).

In reflection level – does the calculations that require computationally intense steps.

Results show that their methodology for integrating affective states into reinforcement learning can be employed in autonomous effective systems. The generated core effects can be used as an alternative

**Results And Discussion:**

During the preliminary experimentation, they found out that the vanilla MaxEnt IRL formulation obligated a great deal of fine-tuning in accordance with the learning rate for parameter updates. Deep MaxEnt IRL, on the other hand, accomplished well under diverse learning rates and reaching strong convergence swiftly. Results from this confirmation experiment to evidence to the conclusion that the approximation of the reward function to a nonlinear model might be better. And for the Multi-agent Deep reinforcement learning, there were a total of two experiments conducted, one with congestion cost another without congestion cost. In the experiment without congestion cost, in order to get a faster convergence time, they used deterministic policy gradients. For the one with congestion cost, for achieving system equilibrium, they added an NFSP component. In the experiment with no congestion cost, since the agent only has to find the shortest path to the destination, the system reached convergence relatively faster. In the experiment with congestion cost, it took much time before convergence because the agents needed to explore extensively for the best routing path to avoid traffic.

**Conclusion:**

The system proposed to this research magnificently inferred approximate reward functions of features resulted from expert routes using maximum entropy inverse reinforcement learning. They compared rewards modeled as a weighted linear combination of feature vectors against rewards parameterized as a deep neural network. Deep MaxEnt IRL produced better results and also proved to be easier to train. Congested routed behavior was imitated both with and without the recovered reward function by deep multi-agent reinforcement learning. Convergence was achieved by both IRL and MARL. Their concluding simulation result was judicious.

**References:**

1. Simulating Human-like Navigation in Urban Transportation Environments with Multi-agent Deep(Inverse) Reinforcement Learning, Ziheng Lin, Sidney A. Feygin, December 14, 2016.
2. Praveen Kumar S, Srinivas Y, Bhargav K, " An n-gram analysis approach for sharing authentication of data using model based techniques", Test Engineering and Management, 2020.
3. Praveen Kumar S, Sahithi Choudary A, " An Innovative ModelBased Approach for CreditCard Fraud Detection Using Predictive Analysis and Logical Regression." , *International Journal of Innovative Technology and Exploring Engineering (IJITEE), Scopus, 2019, 8 , 1683-1688*
4. Praveen Kumar S, Srinivas Y, Vamsi Krishna M, "A Data Leakage Identification System Based on Truncated Skew Symmetric Gaussian Mixture Model.", *International Journal of Recent Technology and Engineering (IJRTE), Scopus, 2018, 7, 111-113*

ISSN: 2005-4289 IJDRBC

5. Praveen Kumar S, Srinivas Y, Vamsi Krishna M, "Latent Semantic Indexing based Approach for Identification of Frequent Itemset.",*Jour of Adv Research in Dynamical & Control Systems, Scopus, 2018, Vol. 10, 686-690.*
6. Praveen Kumar S, Srinivas Y, Vamsi Krishna M, "A mechanism for identifying the guilt agent in a network using vector quantization and skew Gaussian distribution."*International Journal of Engineering & Technology, Scopus, 2018, 7, 149-151*
7. Praveen Kumar S, Srinivas Y, Ranjansenapat M, Kumar A, "An Enhanced Model for Preventing Guilt Agents and Providing Data Security in Distributed Environment."*IEEE Conference International conference on Signal Processing, Communication, Power and Embedded System (SCOPES), others, 2016, 1, 337-339*
8. S Venkata Lakshmi, Valli Kumari Vatsavayi. Query optimization using clustering and Genetic Algorithm for Distributed Databases. International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2016.

9. S Venkata Lakshmi, Valli Kumari Vatsavayi. Query Plan Generation in DDS Using NonDominant Based Teacher-Learner Optimization (ND-TLBO) Algorithm. International Journal of Soft Computing. Medwell Journals, 2016, 11(3), pp.145-154.

10. Vytarani Mathane, P V Lakshmi. Adaptive Security Framework for the Blockchain on IOT.International Journal of Innovative Technology and Exploring Engineering (IJITEE), July 2019.

ISSN: 2005-4289 IJDRBC