CONSTRUCTING THE ENVIRONMENT FEEDBACK ON COMPETITIVE MULTI-AGENT GAMES USING IRL

Furkan Kuse
furkankuse99@gmail.com

Erdem Ağca
agcaerdem98@gmail.com

Büşra Ağdacı
agdacibusra@gmail.com

Supervisor

Assoc. Prof. M. Borahan Tümer

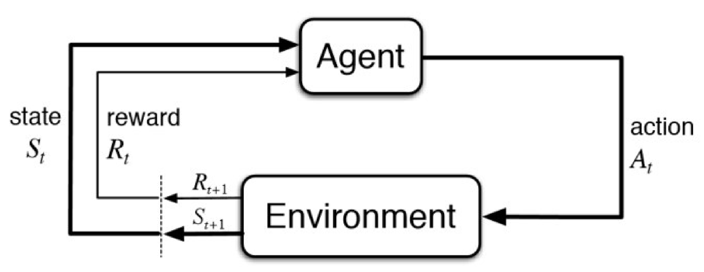
Introduction

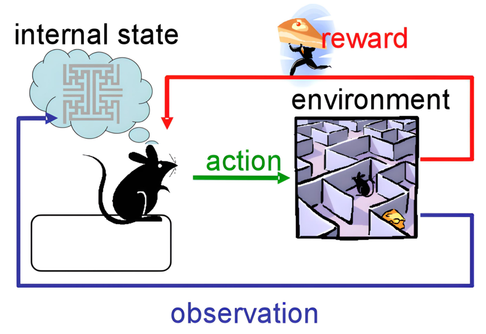
Problem: Generally, in IRL with competitive multi-agent problems, problem is decoupled into sub-problems and each agent is trained in its own sub-problem. Even though they are decoupled into sub-problems, their reward functions are correlated. We are trying to find a way to improve the learning speed by learning from other agents' reward functions.

Solution: We intend to improve the algorithm using the opponent's trajectories, instead of the generated policy. We show that this approach improves the accuracy of the algorithm by 30% and, wins 60% of the games against the agent that is trained with original approach.

Reinforcement Learning

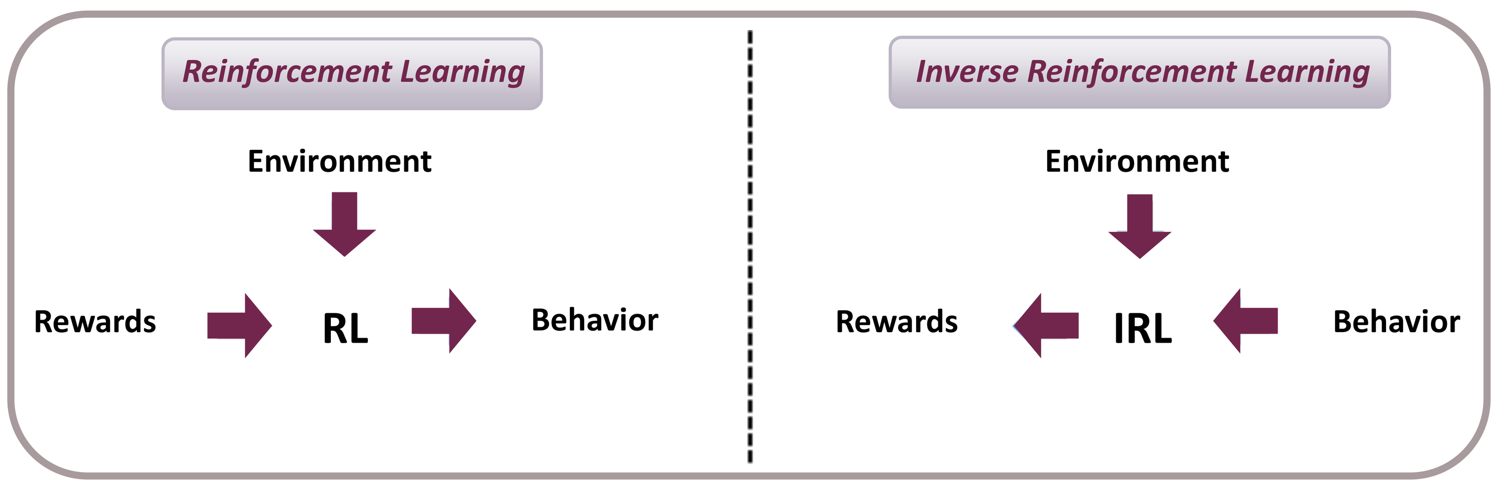
Reinforcement learning (RL) is an area of machine learning, which is used for solving sequential decision making problems.





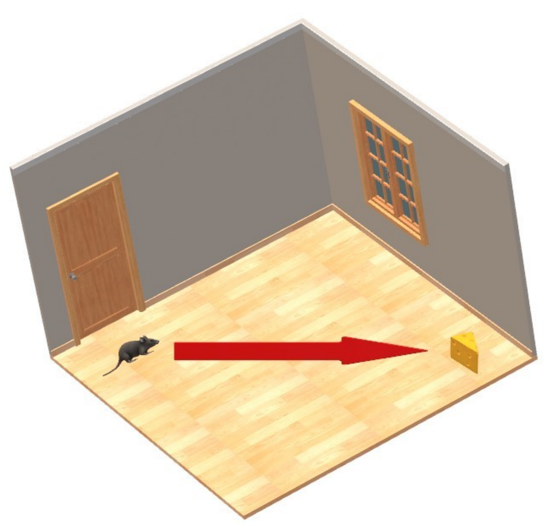
Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) is the method that we are looking for when we want to generate a reward function, by using the observations of an expert. By using IRL we can generate the reward function by using that reward function we can train agents that mimic the things which the expert does.



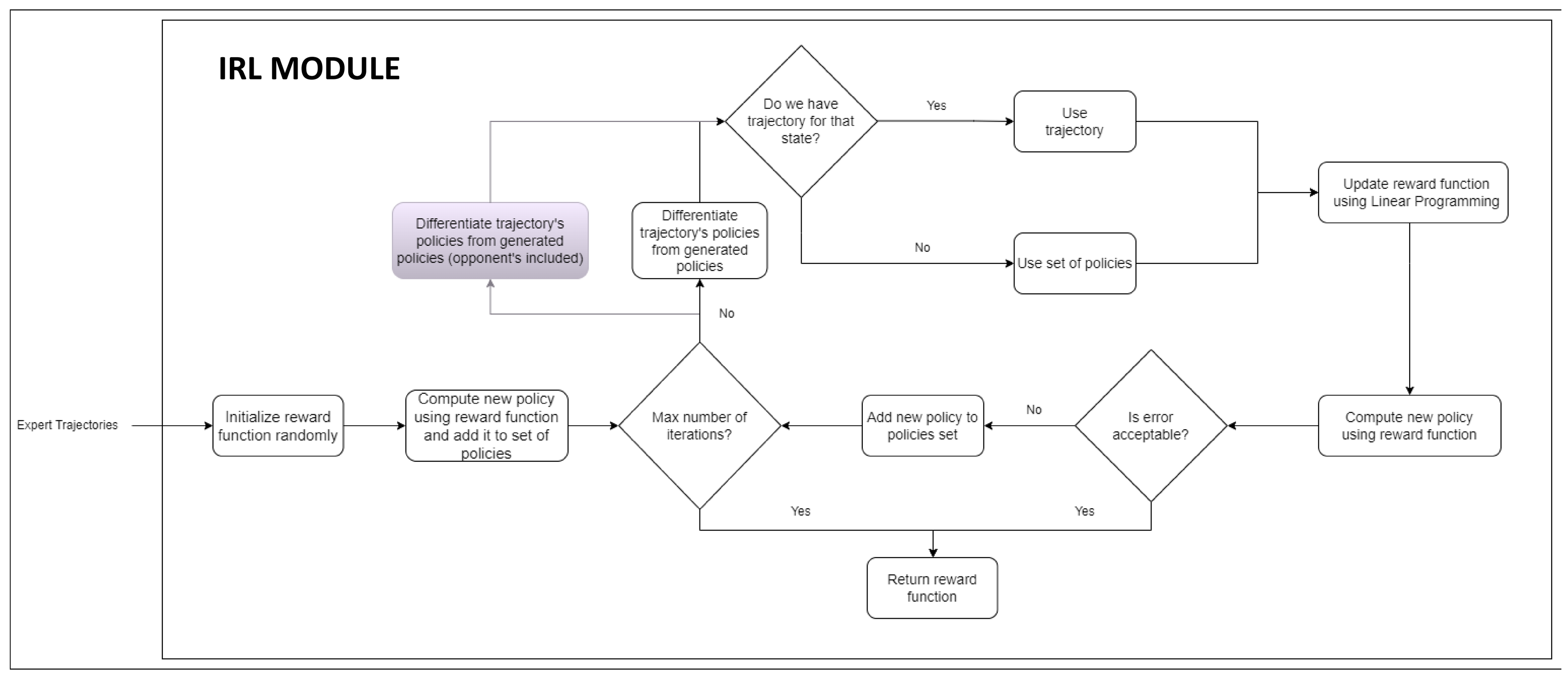
Nash-Q Learning

In Nash-Q Learning, the agent attempts to learn its equilibrium Q-values, starting from an arbitrary guess. The Nash Q-learning agent maintains a model of other agents' Q-values and uses that information to update its own Q values.



Our goal is to find the best strategy for our agent, relative to how other agents play in the game. In order to do this, our agents have to learn about other agents' strategies, and construct a best response.

Algorithms

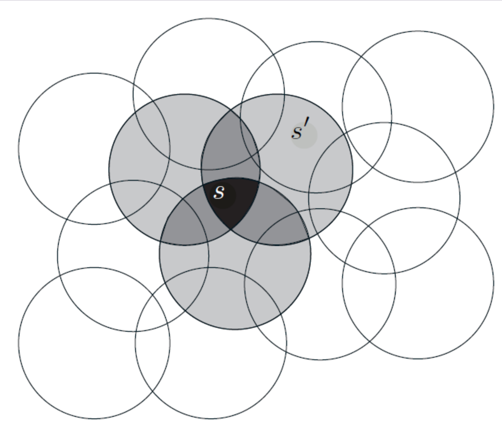


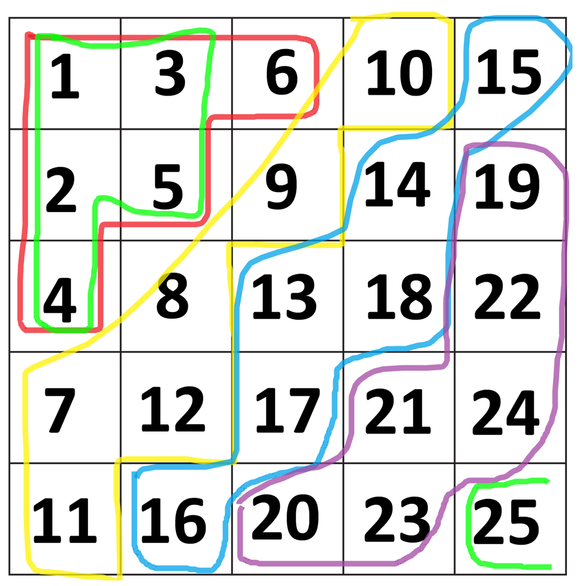
Flow Diagram of the Algorithm

Normally, when using Inverse Reinforcement Learning with multi-agent systems, agents can be decoupled and trained seperately. We think for zero-sum stochastic games with homogenous agents instead of decoupling we can gain advantages using opponent's trajectories.

Coarse Coding

In a task where the natural representation of the state set is a continuous two-dimensional space, if the state is in a circle, the corresponding feature has the value 1 and is said to be present; otherwise the feature is 0 and is said to be absent. Representing a state with features that overlap in this way is known as coarse coding.



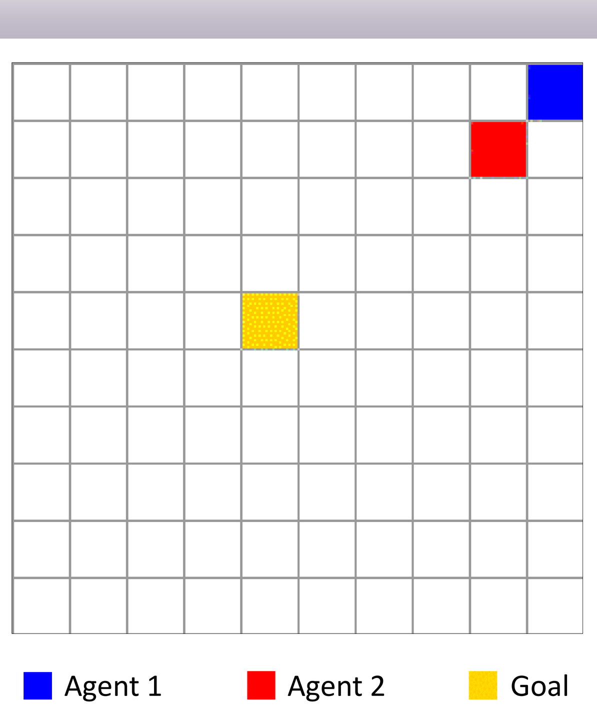


We start enumerating grids by their distance to the upper left most corner. If the grids have same distance to the upper left most corner grid that is closer to the left wall will have smaller number.

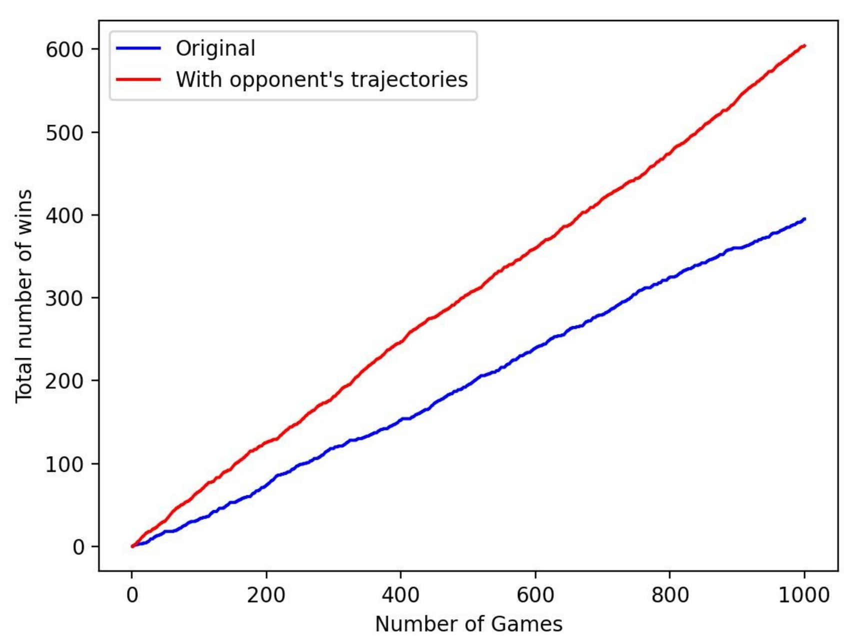
We created group of six grids and used those groups as Φ functions. We cover our grid world with thirty Φ functions and we have four actions in total we have one hundred twenty Φ functions.

$$R(s) = \alpha_1\phi_1(s) + \alpha_2\phi_2(s) + \cdots + \alpha_d\phi_d(s)$$

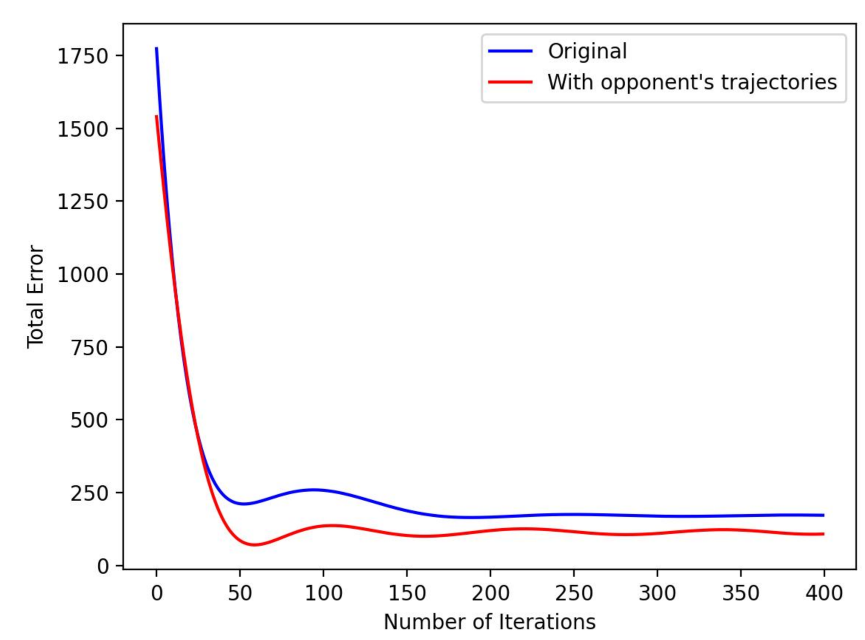
Experimental Results



Experimental Setup



Cumulative win graph



Error of latest created policy graph

Conclusion

- We acquired a reward function with less error and obtain a more effective agent using this reward function.
- Even though our approach gives better results difference is not high, that is because of state aggregation.
- We compete two different agents which one of them is trained with original method and the other with ours. The agent where trained with our method win sixty percent of the games.
- We expected the error rate to decrease more rapidly with our approach but it did not happen as we expected.

References

[1] Ng, Andrew Y., and Stuart J. Russell. "Algorithms for inverse reinforcement learning." Icml. 2000. p. 2

[2] WANG, Xingyu; KLABJAN, Diego. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations. In: International Conference on Machine Learning. PMLR, pages 5143-5151, 2018

[3]Tummalapalli Reddy, Vamsikrishna Gopikrishna, Gergely Zaruba, and Manfred Huber. Inverse reinforcement learning for decentralized non-cooperative multiagent systems. In Systems, Man, and Cybernetics, pages 1930–1935, 2012

[4]Web.archive.org. 2021, Inverse Reinforcement Learning Tutorial | part I | thinking wires

Used Technologies

python™ PC NumPy