CS6284 Project Proposal: CollabPoker

Aishik Pyne
A0250592E
e0945774@u.nus.edu
aipyne@comp.nus.edu.sg

Harshavardhan Abichandani
A0250610X
e0945792@u.nus.edu
harsh@comp.nus.edu.sg

Niharika Shrivastava
A0254355A
e0954756@u.nus.edu
niharikacomp.nus.edu.sg

1 Introduction

In this project, we aim to achieve the following 3 goals

1. Implement a single-agent poker-playing AI where the policy is learned in distributed adversarial setup using common RL methods, such as DQNs, SAC, ReBeL (Brown et al., 2020).

2. Train a multi-agent collaborative poker playing algorithm, with the objective of - Can two agents learn to play collaboratively so that they jointly can beat n other agents playing as single agents. We shall consider two styles of agents for this task

   (a) Full observable: In a simplistic case the multi-agent will be given access to the other player's hand.

   (b) Partial observable: For simulation of a more practical case, the collaborative agents would know the position of the other agents but will have no access to the hands.

3. If these two policies are learned optimally, can a discriminator be learned to discriminate the play styles of the policies by observing the actions taken by players and the overall game state? If so this discriminator can be used to detect cheating at casinos.

These tasks are very challenging because the agents only received partial information about the environment and has to learn to operate optimally which dealing with uncertainty.

2 Environment

The RLCard Simulator is proposed to be used for simulation. It is capable of simulating a lot of card-based game environments but we are interested in Limit-Holdem-Poker. The RL Card Library also has an extension RLCardShowdown to visualize the simulation of RL-Card

3 Experiments

3.1 Single-Agent Poker AI

We set up a poker table of 5 or 7 agents each agent or player shares the same policy. A single-agent poker AI policy π_s^* would have the objective to maximize its own reward, that is, the money it earns from playing a whole round. If trained successfully, it should have super-human performance. However, when policy plays with itself it should converge into a nash-equilibrium.

3.2 Multi-Agent Poker AI

Once single-agent poker AI policy has reached optimality π_s^* it can be treated as a baseline. We set up a new table where we have all but 2 seats being played according to π_s^*. Now in order to learn fully observable multi-agent poker AI, we allow a new pair of agents to sit at the remaining seats at the poker table and see each other hand. As they play with fixed π_s^* they jointly learn π_m^* with the objective of maximizing their joint reward. This means one agent can learn to sacrifice its reward as long as their total rewards are maximized.

We can repeat this experiment where we inform the pair of agents about the other existence but do not allow them to see the other's cards. This is a much more practical situation and would be hard for the agents to learn in an imperfect information environment.

3.3 Behaviour Discriminator

Assuming we have learned the policies optimally π_s^* & π_m^* we can observe the shift in behaviour. We set up a table with 2 collaborative agents and the rest single agents. We would also have control with all agents playing like single agents as control. Then we would observe the difference in trajectories of the setup vs the control. If the divergence of entire trajectories is significant, we wish to train a model to identify which policy a particular trajectory came from. This discriminator would have practical use cases of being a poker cheating detector but also a generic way to observe divergence in policies.

References

[Brown et al., 2020] Brown, N., Bakhtin, A., Lerer, A., and Gong, Q. (2020). Combining deep reinforcement learning and search for imperfect-information games. *CoRR*, abs/2007.13544.

Keywords: Collaborative Multi-Agent RL, Poker AI