AI Project Report

Phase 1

Group: Robo Squad

Swapnil Bhosale (swapnil.bhosale@umbc.edu)
Himanshu Londhe (hlondhe1@umbc.edu)
Pratik Jogdand (pratikj1@umbc.edu)

INTRODUCTION

We have been given a problem to design an agent which will play spades card game. The goal of our agent is to win the game with the maximum amount of tricks. Given the hand, our agent will be choosing the appropriate card to play using Reinforcement Learning which will target winning the trick.

Our Approach:

Our agent will take the current observation, corresponding values of that observation. (We will calculate corresponding values of respective observations using feature extraction technique and input to this technique will be given by transformation from the given state as followed). P1 = Number club cards that have been played in the current game or held by our agent. P2 = Number diamond cards that have been played in the current game or held by our agent. P3 = Number Spades cards that have been played in the current game or held by our agent. P4 = Number Hearts cards that have been played in the current game or held by our agent. P5, P6, P7, P8 is a bit sequence that will represent the leading player in the current trick.

The reason for our prescribed approach is because these types of card games are imperfect information games, i.e for each game player there are unobservable state variables.

> For example: Cards held by other people or undealt cards.

So the information in the environment of such card playing games is partially observable. And with a little bit of research, we came to the conclusion that we will be able to solve this problem using POMDP - RL(Partially Observable Markov Decision

Process - Reinforcement Learning) and it will return us the best results. The POMDP consists of state transition probability.

State transition probability = P(New state | Old state, action)

Observation probability = P(Current observation | current state)

Reward Function for our agent = R(Current state, current action).

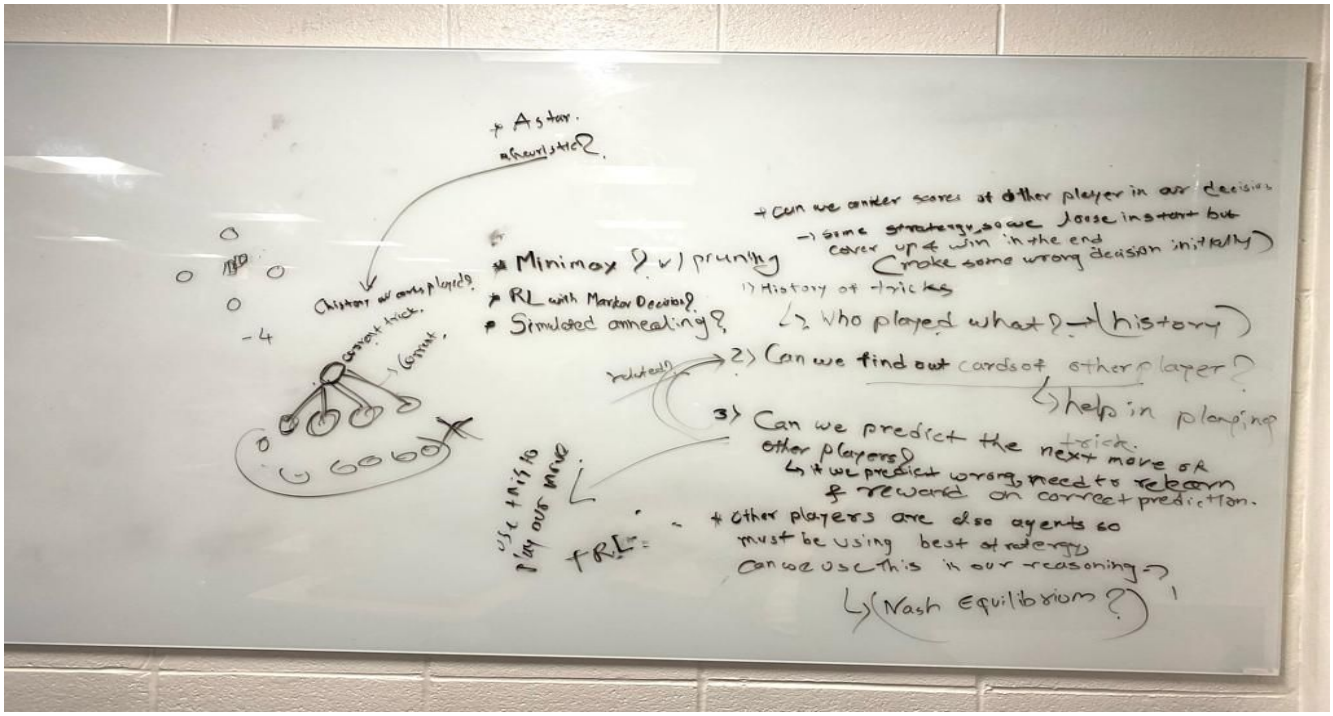
Functions and their return values:

- def get_name() : This will return the name of our group, i.e.'robo squad'.

Added functions for our program:

- def RL():
  input: current hand, cards put on the table so far
  - Calls def track_history().
  - Calls state_eval_module().
  - Output: select one card from the hand to play the round using Partially Observable Markov Decision Process (POMDP).
- def update_reward():
  - Input: outcome of previous trick
  - Output: if we win the trick, add reward or if we lose the trick add negative reward.
- def track_history():
  Input: This will keep track of all the cards which were drawn out for each trick.
  Returns the history of all the used cards by players so far.
- def state_eval_module():
  Input: current observation, corresponding values of that observation.
  Output: Approximates the value function.

Brainstorming we did in class:



The brainstorming session in class resulted in choosing POMDP-RL as the most relevant technique to design a card-playing agent. Other techniques that we considered were Minimax and simulated annealing for search techniques. The picture depicts the questions which we asked ourselves to agree upon design considerations for the agent.

REFERENCES

- Ishii, S., Fujita, H., Mitsutake, M. et al. Mach Learn (2005) 59: 31.A Reinforcement Learning Scheme for a Partially-Observable Multi-Agent Game https://doi.org/10.1007/s10994-005-0461-8
- Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* Human-level control through deep reinforcement learning. *Nature* 518, 529–533 (2015) doi:10.1038/nature14236
- http://papers.nips.cc/paper/3306-regret-minimization-in-games-with-incomplete-information.pdf
- https://arxiv.org/abs/1603.01121