Intelligent Texas Hold'Em No Limit PokerBot
CS281 - Advanced Machine Learning Project Proposal

Srivatsan Srinivasan, Sebastien Baur, Donghun Lee

I. Introduction - Problem Statement

POKER is a quintessential game of imperfect information. The challenges involved in poker include decision sequences, large state-action space, imperfect information and adversarial behavior, presenting a perfect template for the application of probabilistic modeling and reinforcement learning that employ neural networks to interpret complicated non-linearities.

Heads-up no-limit Texas holdem (HUNL) is a two-player version of poker in which two cards are initially dealt face-down to each player, and additional cards are dealt face-up in three subsequent rounds. No limit is placed on the size of the bets although there is an overall limit to the total amount wagered in each game.Imperfect information games require more complex reasoning than similarly sized perfect information games. The correct decision at a particular moment depends upon the probability distribution over private information that the opponent holds, which is revealed through their past actions. However, how our opponents actions reveal that information depends upon their knowledge of our private information and how our actions reveal it. This kind of recursive reasoning is why one cannot easily reason about game situations in isolation, which is at the heart of heuristic search methods for perfect information games.

Given the state space is large and we have limited resources at our disposal, it would be unwise to run brute force simulations and hence we propose to use predictive and inference models in approximating several components of the problem. Instead of a random policy, an initial policy built upon probabilistic models studied by game theory in imperfect information settings could be a starter. Inferring the ”goodness” value function of a given state(that involves the dealt hand, the current pot etc.) is another case of probabilistic inference with smart priors. Having an approximate model for the adversary enables better convergence and robustness of learning algorithms and there has been well-studied literature on using Bayesian nets for this use case. On the RL segment, Fitted-Q iteration is a off-policy learning framework that employs ensemble supervised learning methods to build Q-values from a set of samples. With such a huge modeling scope available, we intend to borrow several vital inference concepts learned throughout the course in order to produce satisfactory models.

II. Proposed Approach

We propose to start with the definition of our state and action spaces. We will also look at options for efficient state space abstractions of the game. This might involve finding similarity metric between states and learning the action-value of the network based on it. Many states in the game could involve similar reasoning and hence such state space information could be abstracted into bins(for example, handling any lower valued card when the hand is dominated by face-cards). Simplification of action space would involve integrating game-theoretic insights with incomplete information in order to prune the domain of actions, given a particular state space. Again, there is a rich availability of resources in algorithmic game theory aspects of such imperfect information games.

Following good state space abstraction, we then go about setting up the reinforcement learning problem and the MDP we intend to solve. We propose to perform off-line learning and evaluation in this case and hence, need to run several simulations with a few relaxed constraints to collect training data. Fitted-Q iteration approach helps improve learned policies by iteratively solving supervised learning problems(we can work on a boosted version of the same). Inferring intermediate rewards for the MDP is a complex probabilistic model that needs to understand abstract state features and assign scores for the current state based on priors and posteriors on conditional probability of earnings given the current state and history(if needed). Besides, an important aspect of creating a poker bot is to give it a 'personality'. The bot must carry its strategy and behave in a non-deterministic way so that it cannot be reverse-engineered and beaten easily. Our algorithm needs to be able to combine different levels of aggression/strategies and also by teaching it to bluff, slow-play etc through appropriate reward settings.

Apart from the hand strength modeling described earlier, the models that would assist in guiding the RL solution would be opponent modeling and short vs long-term decision making and risk management. Adversarial modeling is the vital feature of poker that differentiates it from several other well-modeled games that have been extensively studied in the past. This behavior could be modeled as a Bayesian net that tries to understand the parameters of the playing style of the opponent based on key metrics learned over tons of historical iterations. The decision risk management score is another ML model that constructs utility functions, lists and rates strategies from prior collection of data. While we harbor noble intentions to incorporate every one of these modeling components, several of these approaches could change based on time and implementation constraints.

Once we deliver an initial model, we can focus on thinking about alternate angles to the problem. There has been recent advances around NFSP(Neural Fictitious Self Play) and recursive CFR to solve such imperfect information games to the levels of approximately close to Nash equilibrium. For instance, FSP chooses the best move based on a model of the opponent's expected average behavior.Again, this circles back to constructing a probabilistic model of opponent behavior. At this point, this project can take several directions and we intend to prune our domain once we setup our problem space and understand the computational complexities.

III. Contributions

The project involves several different components that are to be modeled. Once the models are developed, we also need a strong simulator to test the model and allow it to learn. We might also need to setup framework for allowing the model to play against its own clone with different parameters to learn different adversary behavior. Given that intelligent poker bots is still a nascent area of research, we foresee a lot of unknown deterrents and some serious structural changes to our approach. We adopt a model where each teammate will have a major hand in few tasks and would be ably assisted by others, while he reciprocates the same in other majors, thus setting up a meaningful learning curve in several dimensions for all the collaborators.Keeping these constraints in mind, we propose the following assignment of tasks.

Sebastien will be primarily focusing on state-action space definition and abstraction, followed by setting up the Reinforcement Learning framework involved for the problem. Donghun will focus on integrating the model with simulators and also contribute significantly to probabilistic models. Srivatsan will focus on providing game-theoretic approaches for optimal policies and on constructing probabilistic inference models along with handling documentation and presentation of results. The assignment is largely fluid and we expect major overhauls along the course.

IV. Evaluation Methods

There are several off-the-shelf simulators that we could run our algorithms to understand performance and figure out incremental differences added by each models. Starting from a simple Q-learner, we propose to build several models incrementally and run them on the simulator to understand marginal improvements contributed by different models. Since the model representations are extremely abstract, it is tricky to evaluate the models on a standalone basis for their accuracy. The incremental evaluation approach also helps us understand which models significantly contribute to better learning attributes and attribution of the performance improvement to the model characteristics will be an integral part of our final study of this problem. In a nutshell, the success of different probabilistic models could be inferred from the success-rate of the agent against diverse adversaries and convergence attributes(robustness and time taken) of training phase among other salient attributes. Recently, Prof.Rush pointed us to the MIT-PokerBot simulator and we have got in touch with the team at MIT to learn more about the mechanics of their system.

References

[1] J. Heinrich, D. Silver, and L. Papke. Deep Reinforcement Learning from Self-Play in Imperfect-Information Games.
[2] M. Moravcic et.al. DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker.
[3] N. Yakovenkov, L. Cao, J. Fan Poker-CNN: A Pattern Learning Strategy for Making Draws and Bets in Poker Games
[4] A. Heiberg Using Bayesian Nets to model a Poker player.
[5] O. Eckmecki. Learning Strategies for Opponent Modeling in Poker
[6] MIT PokerBots - Simulator and Tutorials. http://mitpokerbots.com/
[7] University of Alberta - Computer Poker Research Group. http://poker.cs.ualberta.ca/resources.html