Machine Learning Project

Application of Machine Learning on an Imperfect Information Game

(Poker)

Project by

Jaishree Dhage (140905350)

Tuhin Khare (140905056)

**Abstract**

Game playing is a favourite application area of Artificial Intelligence, specially when it comes to imperfect information games where the optimal gameplay strategy is to be chosen without the complete information parameters. The game of poker offers a clean and well defined domain to investigate some of the truly fundamental issues in computer science, such as how to handle deliberate information and how to make intelligent guesses based on partial knowledge. Success in the game of poker not only requires a good understanding of the ground rules but also requires a strategy of informed guesswork, risk management and opponent modelling. Computer Poker Bots is a system of intelligent programs that attempts to model a human by learning these qualities with the goal of winning the game. The agents are adversarial and not necessarily equally smart, but provide a wide array of opportunities for research in machine intelligence. This paper mainly depicts the idea of developing a poker bot using concepts of reinforcement learning and bayesian decision theory.

**Introduction**

The game of poker is currently the newest and one of the most challenging areas for the researchers in the field of Artificial Intelligence and Machine Learning, being an imperfect information game in which each player has some hidden information which is central to the game in the form of private cards, developing a computer program which could play any variant of the game is giving the researchers a hard time. Many approaches have been used in order to develop agents which can mimic playing strategy to that of humans along with improving their gameplay strategies with every game they play.

Game Description

A hand of poker begins with the pre-flop, where each player is dealt two hole cards, face down, followed by the first round of betting. Then three community cards are dealt face up on the table, called the flop, and the second round of betting occurs. On the turn, a fourth community card is dealt face up and another round of betting ensues. Finally, on the river, a fifth community card is dealt face up and the fourth (final) round of betting occurs. All players still in the game turn over their two hidden cards for the showdown. The best five card poker hand formed from the two hole cards and the five community cards wins the pot. If a tie occurs, the pot is split. Typically the game is played with 8 to 10 players.

The order and amount of betting is strictly controlled on each betting round. We use 4 betting denominations: 1, 2, 5 and 10. When it is a player's turn to bet, one of five options is available: fold (withdraw from the hand, leaving all previously wagered money n the pot), call (match the current outstanding bet; if there is no current bet, one is said to check), or raise the bet (put something more than the current bet; if there is no current bet, one is said to bet). There is usually a maximum of three raises allowed per betting round. The betting option rotates clockwise until each player that has not folded has put the same amount of money into the pot for the current round, or until there is only one player remaining. In the latter case, this player is the winner and is awarded the pot without having to reveal their cards. There is a strategic advantage to being the last bettor in any given round; so to maintain fairness, the order of betting is rotated clockwise after each hand.

Our Approach to the Problem

This report describes a connectionist approach to the development of such an agent, through techniques like reinforcement learning and bayesian decision theory. In the approach used in this paper the work is divided i.e calculating the hand strength of the hand and choosing a strategy which would maximise the agents chances of winning the game. The calculation of the hand strength is done without considering the dynamics of a multi-player game. This subpart of the agent’s gameplay is called the *Learning Model*. The *Playing Model* makes use of the hand-strength and the likelihood probabilities calculated in order to make a decision. This separation into learning and playing models enables us to manage a complex task better. The separation is also justified on the grounds that evaluation of hand strength is dependent only on the cards a player has – we are not determining relative hand strength in the learning model.

Scope

The major scope of the project are mainly to enable the bot to make decision which would maximise its chances of winning with every gameplay alongside the bot should also be able to memoize the sequence of actions which leads to the bots win and loss respectively, through a reward function which mainly provides a positive reward value for every action sequence vector which leads to the bots win and a respective negetive reward value