Learning to Play Poker using Counterfactual Regret Minimization

James Ni and **Chris Lamb**
{jani,chlamb}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

In our investigation into Counterfactual Regret Minimization (CFRM), we attempt to determine Nash equilibria for Kuhn poker and Leduc poker. Both of these games are relatively simple examples of two player, sequential, imperfect information, zero-sum games. Thus, they provide a good test bed for assessing CFRMs ability to determine optimal bluffing strategies in similar, more complicated games like No Limit Texas Holdem.

1 Introduction

Poker is an example of an imperfect information game. Until recently, computers had not found much success against expert human players in imperfect information games. An optimal strategy in an imperfect information game involves not only determining which player has the statistical advantage in a particular situation but also finding the appropriate mixed strategy for betting when you have an advantage and bluffing when you have a disadvantage. One algorithm that has been successful in doing just that is counterfactual regret minimization (CFRM).

Our experiment implements CFRM to determine the Nash equilibria for two simplified games, Kuhn poker and Leduc poker. We are exploring these simplified examples as even with only two players, No Limit Texas Holdem has 10^{160} possible game states, which is beyond our computational resources. (Waugh et al. 2015) These simplified games share the same general characteristics and are therefore good test cases. We compare our derived expected utilities and strategies with those published by Kuhn and Hoehn et al. for Kuhn poker and Waugh et al. for Leduc poker.

This paper is organized as follows: in Section 2, we will discuss the rules of each game and the CFRM algorithm. In Section 3, we will explain our experimental implementation of counterfactual regret minimization. In Section 4, we will present our experimental expected utilities found by CFRM for each game. In section 5, we will conclude by presenting our analysis of our results and suggesting further work. Finally, in section 6 we detail the individual contributions to this research.

Kuhn Poker Utilities			
Sequential Actions			**Utility**
Player 1	Player 2	Player 1	
Check	Check		+1 to winning player
Bet	Fold		+1 to player 1
Bet	Call		+2 to winning player
Check	Bet	Fold	+1 to player 2
Check	Bet	Call	+2 to winning player

Figure 1: Kuhn poker terminal action sequences and associated payouts.

2 Background

We used counterfactual regret minimization to solve two simplified poker variants Kuhn poker, named after its inventor mathematician and game theorist Harold Kuhn (Kuhn 1950), and Leduc poker. Below we describe the rules of the two games and the counterfactual regret minimization algorithm.

Kuhn Poker

Kuhn poker is a two player, zero sum, imperfect information, sequential game. It is a simplified poker variant played with a three card deck. The cards are Jack, Queen, and King, valued in that order. Each player antes a single chip and is dealt a single card. There is one round of betting, where bets are a single chip. If a player folds to a bet, they lose the pot. The potential payouts are shown in figure 1.

The expected utility for player 1 in a Nash equilibrium is $-1/18$ chips per hand. For Kuhn poker, there are an infinite number of Nash equilibria mixed strategies. The optimal strategy is to select a value α such that $0 < \alpha \leq 1/3$. The player should bet with probability α when they are dealt a Jack. They should bet with a probability 3α when they are dealt a King. They should also bet with probability $\alpha + 1/3$ when the are dealt a Queen check and their opponent bets. (Hoehn et al. 2005) The full description of the theoretical Kuhn poker equilibrium strategy can be seen in figure 2.

Average Utility: $-1/18$		
Info State	**Pass %**	**Bet %**
J	$1-\alpha$	α
JB	1	0
JP	2/3	1/3
JPB	1	0
Q	1	0
QB	2/3	1/3
QP	1	0
QPB	$2/3-\alpha$	$1/3+\alpha$
K	$1-3\alpha$	3α
KB	0	1
KP	0	1
KPB	0	1
$0 < \alpha \leq 1/3$		

Figure 2: Theoretical optimal Kuhn poker strategy. In info state J, Q, and K represent Jack, Queen, and King respectively, P represents check and fold, and B represents bet and call.

Leduc Poker

Leduc poker is also a two player, zero sum, imperfect information, sequential game. It is also a simplified poker variant. It is played with a six card deck. The deck includes two Jacks, two Queens, and two Kings. Each player antes a single chip and is dealt a single card. There is a round of betting followed by the flop, when a community card which is included in both players hands is dealt. After the flop there is a second round of betting. A pair beats a single card, and if neither player has a pair, the player with the higher ranked card wins. A tie results in the pot being split evenly between the two players. The potential payouts are shown in figure 3. We have not found a published Nash equilibrium expected utility or strategy for this Leduc poker rule set.

We also implemented a secondary rule set to match the rules described by Waugh et al. which allows a single reraise per betting round. The bet sizing is also changed. Bets and raises in the first round of betting are two chips, and bets and raises in the second round of betting are four chips. In the interest of space, the additional action sequence and associated payouts for this rule set is omitted.

Leduc Poker Utilities	
Sequential Actions	**Utility**
Check Bet Fold	+1 to player 2
Bet Fold	+1 to player 1
Check Check, Check Bet Fold	+1 to player 2
Check Check, Bet Fold	+1 to player 1
Check Check, Check Check	+1 to winning player
Check Bet Call, Check Bet Fold	+2 to player 2
Bet Call, Check Bet Fold	+2 to player 2
Check Bet Call, Bet Fold	+2 to player 1
Bet Call, Bet Fold	+2 to player 1
Check Bet Call, Check Check	+2 to winning player
Bet Call, Check Check	+2 to winning player
Check Check, Check Bet Call	+2 to winning player
Check Check, Bet Call	+2 to winning player
Check Bet Call, Check bet Call	+4 to winning player
Check Bet Call, Bet Call	+4 to winning player
Bet Call, Check Bet Call	+4 to winning player
Bet Call, Bet Call	+4 to winning player

Figure 3: Terminal action sequences and associated payouts for Leduc poker without re-raises.

Counterfactual Regret Minimization

Counterfactual regret minimization builds a game tree through simulated self play. Each node of the tree is identified by an information state. In the case of poker, the information state consists of the cards known to the current player and the history of actions taken during the hand. Each node also stores the cumulative regret associated with not choosing each action in this state in previous iterations. Regret associated with an action is the difference in utility u_i resulting from choosing that action versus the action actually chosen. Finally each node stores the cumulative probability assigned to each action choice in previous iterations. The process of self play goes as follows. The cards are dealt and the algorithm selects a policy σ for the given player i, training iteration T, information state I and action a from the set of available actions A, using non-negative counterfactual regret matching as shown in equation 1.

$$\sigma_i^{T+1}(I,a) = \begin{cases} \dfrac{R_i^T(I,a)}{\sum\limits_{\alpha \in A(I)} R_i^T(I,\alpha)} & \text{if } \sum\limits_{\alpha \in A(I)} R_i^T(I,\alpha) > 0 \\ \dfrac{1}{\mid A(I) \mid} & \text{otherwise.} \end{cases} \tag{1}$$

This results in each action being chosen with a weight such that the expected regret associated with not selecting each action is equal. The chosen strategy is added to the cumulative strategy. If the game state is non-terminal each action probability is passed recursively into the algorithm with the updated information state to include the selected action. If a terminal game state is reached the utility for the hand is calculated and weighted according to the probability of reaching that particular terminal state. The regret $R(h,a)$ associated with all actions a not chosen in history h is calculated using equation 2 and the cumulative regret is updated.

$$R(h,a) = u_i(\sigma_{I\rightarrow a}, h) - u_i(\sigma, h) \tag{2}$$

The cumulative regret and cumulative strategies chosen for each information state are updated through multiple iterations of self play. The weighted average strategy converges to the Nash equilibrium strategy over time and can be extracted using equation 3. (Neller and Lanctot 2013)

Average utility: -0.0555		
Info State	**Pass %**	**Bet %**
J	0.829	0.171
JB	1.000	0.000
JP	0.666	0.334
JPB	1.000	0.000
Q	1.000	0.000
QB	0.666	0.334
QP	1.000	0.000
QPB	0.495	0.505
K	0.486	0.514
KB	0.000	1.000
KP	0.000	1.000
KPB	1.000	1.000
$\alpha = .171$		

Figure 4: Experimental Kuhn poker strategy after five million training iterations. In info state J, Q, and K represent Jack, Queen, and King respectively, P represents check and fold, and B represents bet and call.

$$\sigma_i^*(I,a) = \begin{cases} \lim\limits_{T\to\infty} \dfrac{\sum\limits_{t\in T} \sigma_i^t(I,a)}{\sum\limits_{\alpha\in A(I)} \sum\limits_{t\in T} \sigma_i^t(I,\alpha)} & \text{if } \sum\limits_{\alpha\in A(I)} \sum\limits_{t\in T} \sigma_i^t(I,\alpha) > 0 \\ \dfrac{1}{\mid A(I) \mid} & \text{otherwise.} \end{cases} \tag{3}$$

3 Experiments

In our experiment, we implemented CFRM and used self play to learn strategies for Kuhn poker and Leduc poker. For Leduc poker, we utilized both rule sets described in the background section. We ran the CFRM algorithm on each game for training lengths ranging from one iteration to five million iterations in increments of ten thousand iterations to view the oscillatory convergence of the expected utilities over time.

4 Results

Our experimental results for Kuhn poker provided good agreement with the theoretical predicted strategy. A sample strategy after five million iterations of CFRM gives the strategy shown in figure 4. Here, we see an $\alpha = .171$. The rest of the figure follows the values agree with the ratios theoretically expected. The average utility for player 1 was -0.055514, equivalent to the $-1/18$ predicted value to four decimal places. Addition training iterations also continued CFRM's convergence to the predicted value to higher precision.

We recorded the average utility after training runs of varying length from one iteration to five million iterations in increments of ten thousand iterations to confirm the convergence to the $-1/18$ expected value. We predicted that there would be some level of oscillation early on due to overcompensation of strategies from the initial strategy.

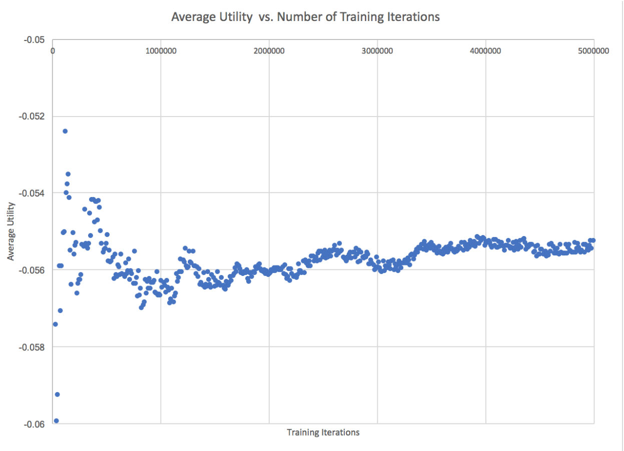

Figure 5: The convergence of the average utility for player 1 in Kuhn poker from one training iteration to five million training iterations.

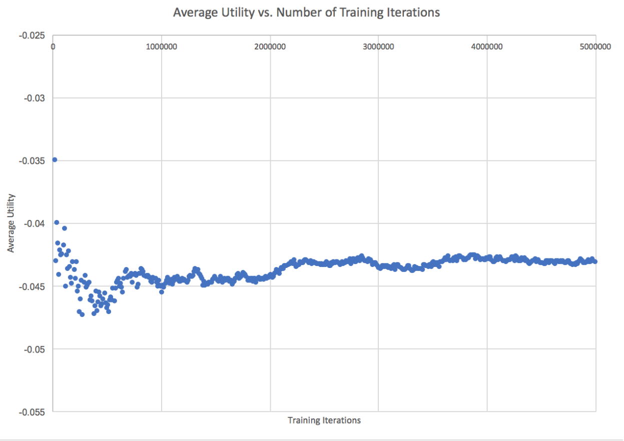

Figure 6: The convergence of the average utility for player 1 in Leduc poker with no re-raises from one training iteration to five million training iterations.

Figure 5 shows the expected oscillation and convergence our upon the Nash equilibrium utility of $-1/18$ over time.

We also performed the same analysis on Leduc poker with both rule sets discussed in the background section. Due to the size of the strategy table for Leduc poker, we are not including it here. The average utility convergence plot seen in figure 6 exhibits a similar damped oscillation converging upon an expected utility. The average utility for player 1 in Leduc poker without re-raising converges to a value of approximately -0.043549 after five million iterations.

Finally, for Leduc poker with re-raising and round-specific bet sizing, the resulting average utility after five million runs was -0.033084 chips per hand for player 1. This does not match with the value of -0.08 chips per hand given

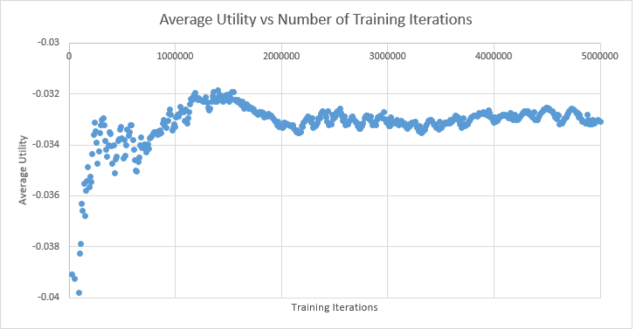

Figure 7: The convergence of the average utility for player 1 in Leduc poker with re-raises from one training iteration to five million training iterations.

by Waugh et al. despite matching the rule set and bet-sizing. The convergence of Leduc poker with re-raising is shown in figure 7.

5 Conclusions

Our implementation of CFRM is able to successfully converge upon the analytically derived Nash equilibrium for Kuhn poker very quickly. It also converges upon an expected utility value for both versions of Leduc poker. For the variant without re-raises the converged value appears to be consistent with publish values for similar games although we did not find any published results for our exact rule set. For the variant with re-raising the converged value does not agree with the value published by Waugh et al.. We are not entirely confident that our rule set and payout matched theirs entirely as the description was somewhat ambiguous. We attempted various interpretation of their rule set and were not able to exactly match their average utility value. Still, these results demonstrate the ability of CFRM to solve simple two-player, zero-sum, imperfect information, sequential games.

There are however considerable challenges that must be overcome to apply this algorithm to more complicated games with larger game trees. First the entire game tree must be maintained as the algorithm updates the entire tree on every iteration. This would be prohibitive for large game trees like that of No Limit Texas Hold'em which contains 10^{160} nodes. This problem can be somewhat mitigated by combining logically equivalent game tree nodes but the memory requirements remain large, on the order of 10^{18} nodes. The time complexity of applying CFRM to more complicated games is also an issue, but this can also be somewhat alleviated by implementing the algorithm in parallel.

6 Contributions

CL implemented the CFRM algorithm, Kuhn poker rules, Leduc poker rules, and ran the Kuhn and Leduc poker trials. JN implemented the rules for Leduc poker with re-raising, and ran the Leduc poker with re-raising trials. JN implemented the demonstration allowing for human play against the Nash equilibrium strategy. Both CL and JN contributed to every section of this paper.

7 Acknowledgements

We would like to thank Dr. Ramanujan for his guidance on this project and all semester.

References

Hoehn, B.; Southey, F.; Holte, R. C.; and Bulitko, V. 2005. Effective short-term opponent exploitation in simplified poker. In *AAAI*, volume 5, 783–788.

Kuhn, H. W. 1950. A simplified two-person poker. *Contributions to the Theory of Games* 1:97–103.

Neller, T. W., and Lanctot, M. 2013. An introduction to counterfactual regret minimization.

Waugh, K.; Morrill, D.; Bagnell, J. A.; and Bowling, M. 2015. Solving games with functional regret estimation. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.