Estimating Card Fitness for Discard in Gin Rummy

Jacob Gallucci,[1] Richard Bowser,[2] Sarah Kettell,[3] Christian Overton[4]
The Pennsylvania State University
jag6241@psu.edu,[1] rjb5973@psu.edu,[2] sxk1552@psu.edu,[3] cto5068@psu.edu[4]

Abstract

Due to the computation time and resources required, there is no known optimal strategy for the game of Gin Rummy. Previous work in extensive games, such as Texas Hold'em Poker, has found that hand fitness and information sets about the state of the game can be used to determine an improved strategy. These information sets, combined with algorithms for Counterfactual Regret Minimization, can arrive at a Nash Equilibrium strategy for smaller abstractions of extensive games. This paper builds on previous research by extending the premise of hand fitness to card fitness in the discard decision point of Gin Rummy. We argue that a card can be ranked based on whether it meets four specific characteristics at that stage in the game. These characteristics include its effect on deadwood points after one more turn, its utility to the opponent, and if it can contribute to a meld. An optimal discard choice can then be picked from the highest-ranked card by using a simplified Counterfactual regret minimization strategy that can be trained in less time due to its limited information set. While this does not look at every potential characteristic of card fitness, it outperformed other bots when evaluated in a large number of games. These bots did not consider card fitness as a whole, but rather considered characteristics separately. We argue that the characteristics defined are a part of the total information set that can determine the discard fitness of a card within a hand in the game of Gin Rummy.

Introduction

Our purpose was to provide a Gin Rummy bot that was more advanced than a purely heuristic-based one. We decided to do this because implementing a bot that can train itself is an area not too thoroughly researched.

Since Gin Rummy is an imperfect information game, or a game in which the bot does not know everything that is going on (such as what is in the opponent's hand), our bot uses a combination of heuristics and counterfactual regret minimization to make decisions when drawing, discarding, knocking, and melding cards. We combine all of these strategies together to create an agent that can be trained in a reasonable amount of time without requiring a large amount of storage.

Our preliminary results show that a bot that used heuristics and CFR performed on average about 4% better than a bot that just used heuristics.

Game of Gin Rummy

History

Gin Rummy is yet another card game in the realm of poker. With the score relying on similar though different mechanics such as straights, runs, sets to create melds. A more closely related game however of course is Rummy.

"The earliest true Rummy, a kind of proto-Gin, was first described briefly under the name Coon Can in The Standard Hoyle (New York, 1887), and in more detail under the name Conquian by R. F. Foster in Foster's Complete Hoyle of 1897." (Parlett, n.d.)

The basic premise of Rummy is the idea of drawing a card and discarding on the same turn. This style of card games was very popular in Mexico and also found in various Chinese card games with more focus on melding. The popularity of Rummy and Gin Rummy in the United States could come from either Chinese immigrants or from a similarly popular Spanish game called "Chinchón" which plays nearly the same as Gin Rummy but with a smaller deck of cards. It seems like with any new game or idea it is a culmination of various card games through the years, with each new culture having their own take on how to play and the rules creating new versions such as another notable game "Conquian".

The current version of Gin Rummy as it stands is believed to be created by Elwood T. Baker in 1909. As the story goes it got its name from Baker's son who liked "Rum-my" and "Gin". It grew in popularity when showcased in both radio and television. Furthermore, during the depression when people did not have more money to spend on other entertainment, it became a popular game to play as it was easier to understand and faster to play than other versions.

Jumping to the present Gin Rummy is among the classic games listed along with Bicycle Cards. It can be found as playable games online cementing its history forever.

Rules

The rules of Gin Rummy, described by Bicycle Cards, are as follows. Gin Rummy is played in rounds with two players, with each round points are scored, with the main objective to reach a total of 100 points before the other player.

At the start of the game each player is dealt 10 cards, the rest of the deck is placed face down and a single card from it is overturned. The first player is given the option to either draw the faceup card or pass to allow the other player to draw.

When a player is able to draw a card, they can pick either the faceup card or draw from the top of the deck. Once a card is drawn, they must then discard one of their now 10 cards, not including the card they just drew. This is how the rounds are played, with one player drawing then discarding and switching turns to the opponent to do the same.

A round ends when one player knocks. In order to knock a player must have a deadwood of 10 or less. Deadwood is part of the scoring system and is based upon how many cards a player has that are not melded. So, what is a meld? A meld is a group of cards that is created from either a set or a run, with a set being three or more cards of the same rank (i.e. 8 of hearts, 8 of clubs, 8 of diamonds) and a run being three or more cards of the same suit in an incremental order (i.e. 2 of hearts, 3 of hearts, 4 of hearts).

The deadwood as stated before comes from the cards not currently in a meld. So, of the ten cards a player is holding, any cards that are in a meld count as 0 towards the deadwood score, whereas anything else counts with its respective rank. For example, in a hand of

Ace of Hearts, 5 of Hearts, 5 of Clubs, 5 of Diamonds, 9 of Hearts, 10 of clubs, Jack of Clubs, Queen of Clubs, King of Clubs, King of Diamonds

The deadwood would be 20 from the cards not currently in melds (shown by underline)

The knocking phase begins when a player accomplishes a deadwood of 10 or fewer. At this point the player now has the option to knock or can continue to play out the game. It is important to note that a deadwood of 0 is the most profitable form of knocking called "Gin". This phase of the game is where score comes into play, and the point system is based on both the player's deadwood and the opponents. When a player or opponent decides to knock the points, they gain are determined by the difference in deadwood they currently have and their opponent has. This said if the opponent has a lower deadwood than the opponent wins the points for the round. Additionally, when a player knocks as a Gin there are bonus points award, typically 20 or in our case 25. As well, failing to undercut the opponent by losing when knocking can result in a bonus for the opponent with points typically of 10 (though subject to change).

After a knock the round is over and the next begins with the first player to reach a score of 100 wins the game (Bicycle Cards 2020).

Counterfactual Regret Minimization in Gin Rummy

Finding an optimal strategy in Gin Rummy is challenging due to the large number of states that the game may reach, based on what cards are in play and what actions both players take. Adding to this difficulty is the inability to know what the opponent's strategy is or the cards in their hand. In games with a smaller set of states and perfect knowledge, it is possible to find a Nash Equilibrium strategy that guarantees a player will choose an action that, at minimum, ties with the utility of the opponent's utility. Computing this for an extensive game such as Gin Rummy would be intractable. However, by utilizing Counterfactual Regret Minimization, a Nash Equilibrium strategy can be obtained within a smaller abstraction of the entire game.

Counterfactual regret minimization utilizes the concept of regret, a numerical value that represents how much utility a player would have gained or lost if they had chosen a different action at a point in the game (Zinkevich et al. 2008). Because the players have incomplete information about the state of the game at any specific timestamp during it, we group together game states that are indistinguishable to a player. These game states are held in sets called information sets. Using these information sets, the regret that is measured is considered counterfactual because it is weighted only by the provability that the opponent would play to a specific node. One approach to calculating counterfactual regret exhaustively explores the game tree, exploring every option that a player could take. This process is done over many iterations of the game or its abstraction, each time accumulating regret to show which actions were more beneficial than others. This counterfactual regret is then used to update the player's strategy for each node. The average strategy generated from training in this manner has been shown to converge to a Nash Equilibrium strategy profile for zero-sum games.

The algorithm for counterfactual regret minimization begins by assigning an arbitrary strategy to the players, then continues through the possible nodes based on their information sets. The information set is dependent on the game abstraction and what information is being tracked, but is used to guide the strategy across identical game states. In a knock strategy for Gin Rummy, for example, this may include factors such as the number of deadwood in a hand or the number of turns taken in the current game.

Once an information set, I, is reached, the counterfactual utility for a player i using strategy σ at node h is calculated by

$$u_i(h) = \frac{\sum_{h\in I, h' \in Z} \pi^{\sigma}_{-i}(h)\pi^{\sigma}_{i}(h,h')u_i(h')}{\pi^{\sigma}_{-i}(I)}$$

where h' is a node in the set of terminal nodes Z, $\pi^{\sigma}_{-i}(h)$ is the probability that player $-i$ will reach node h based on their strategy, $\pi^{\sigma}_{i}(h,h')$ is the probability that player i will go from node h to node h', and $u_i(h')$ is the actual utility value of that terminal node. The sum of this weighted utility is then divided by the probability that player $-i$ would reach the information set I.

The sum of counterfactual regret, R_i^{sum}, for an action, a_j, taken by player i at node h is then calculated by

$$R_i^{sum}(a_j,h) = \sum_{h\in I}(u_i(h_j) - u_i(h))$$

where $u_i(h_j)$ is the counterfactual utility for player i if they reach node h_j by taking action a_j from node h, and $u_i(h)$ is the counterfactual utility for player i if they reach node h based on their current strategy, which may involve choosing actions other than a_j with nonzero probability.

The strategy, $\sigma_i(I)(a)$, for player i taking that action from that information set can be updated by

$$\sigma_i(I)(a) = \begin{cases} \dfrac{max(R_i^{sum}(I,a),0)}{\sum_{a\in A(I)} max(R_i^{sum}(I,a),0)} & if \sum_{a\in A(I)} max(R_i^{sum}(I,a),0) > 0 \\ \dfrac{1}{|A(I)|} & otherwise \end{cases}$$

where the max value between 0 and $R_i^{sum}(I,a)$, the sum of counterfactual regret for player i taking action a from information set I, is divided by the sum of counterfactual regret across all actions available in set I, $A(I)$, if that sum is greater than 0. Otherwise, the strategy $\sigma_i(I)(a)$ is assigned by equal probability across all possible actions in the information set, $A(I)$.

Finally, based on the above calculations, the average, or equilibrium, strategy can then be calculated and normalized by

$$\overline{\sigma}_i^t(I)(a) = \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\sigma^t(I)(a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}$$

where $\pi_i^{\sigma^t}(I)$ is the probability the current player played to the node in the information set, $\sigma^t(I)(a)$ is the strategy for the action, and t is the timestamp of the action across the entire training time T.

The difference between CFR and MCCFR

Counterfactual regret minimization (CFR) comes in many forms. One such form is that of the Monte Carlo Counterfactual regret minimization (MCCFR). This variant of CFR aims to create a more optimal solution by decreasing the overall run time of the algorithm. CFR works by traversing the game state tree and looking for all possible decisions to see which path it can choose to better perform, and doing so by calculating regret when other actions result in a better outcome. MCCFR improves on the training time of CFR by only reviewing necessary parts of the game state tree. By looking at these abstracted samples it can lower the overall amount of run time on an iteration by iteration basis.

Further Reducing Computation Time with CFR+

There are several ways to reduce the computation time required for a counterfactual regret minimization algorithm to converge to a Nash Equilibrium. The simplest way to drastically reduce the time required for a strategy to converge is CFR+. The formula to adapt CFR to CFR+ a simple modification to the original CFR equation.

$$\boldsymbol{R_i^{+,T}(I,a) = \begin{cases} max(v_i(\sigma^T_{I->a}, I) - v_i(\sigma^T, I), 0) & T = 1 \\ max(R_i^{+,T-1}(I,a) + v_i(\sigma^T_{I->a}, I) - v_i(\sigma^T, I), 0) & T > 1 \end{cases}}$$

$$\boldsymbol{\sigma^{T+1} = \begin{cases} \dfrac{R_i^{+,T}(I,a)}{\sum_{a'\in A(I)} R_i^{+,T}(I,a')} & if\ denominator\ is\ positive \\ \dfrac{1}{|A(I)|} & otherwise \end{cases}}$$

The idea behind CFR+ is that if the regret is not permitted to go below 0 and thus the strategy will converge significantly faster as the step to average the accumulated strategy is no longer required (Tammelin, 2014). For this reason, all CFR in the agents developed for this experiment used CFR+.

Related Works

We looked into various papers that provided research into multiple areas of study with the intent of detailing how and why our method for discard makes sense. What exactly lends a hand in determining card fitness, why is it important, and how can we use it to better our agent.

The first paper we looked into for this matter was “Shapely Effects for Global Sensitivity Analysis: Theory and Computation”. In this paper, Song, Nelson, and Staum discuss Shapely Value Regression as a tool to quantify how sensitive a model or function is to certain inputs. They assess how it compares to using more traditional methods of quantification and discuss how output can be determined based on uncertain input variables, such as the way wind speed and direction can influence a forest fire. In addition, they utilize the Monte Carlo method to reduce the computational burden on calculating the Shapely Value of inputs to a given function (Song, Nelson, and Staum 2016).

In a similar way to the example of how wind variables can affect the output of a forest fire (Song, Nelson, and Staum 2016), card fitness in Gin Rummy is dependent on uncertain variables related to the opponent’s hand and strategy. By allowing some uncertainty into our Monte Carlo computations, we can form a more robust strategy than if we stuck strictly to known information.

Our next area of research was looking into how the actions we take affect the opponent and vice versa. Importantly, using that information to see if we can determine types of opponents and styles of play to better effect our decisions overall. We found this in a paper titled “Opponent Modeling in Poker” In their paper, Opponent Modeling in Poker, Billings et al. discuss how a poker program named Loki adjusts its strategy to exploit potential weaknesses in the opponents based on hand assessment and opponent modeling. Hand assessment is determined by a combination of hand strength, a percentile based on ranking within all potential hands, and hand potential, which considers the probability that a hand will improve to win the round. These values are then weighted by opponent modeling values that consider the probability that an opponent player starts with a specific hand and adapts based on the opponent’s betting actions. To test this form of opponent modeling, they ran a set of simulation games against non-adaptive versions of conservative and liberal players, as well as adaptive versions that applied the opponent modeling generically across all players and specific modeling per player. In the end, their results showed that the adaptive opponent modeling players performed stronger (Billings et al. 1998). However, they were unable to sufficiently test the strategy against human players to see if this success held up in real-world games.

Although opponent modeling is beyond the scope of our research for this paper, the strategy that Billings et al. used to measure hand strength and potential is useful in calculating card or hand fitness within the game of Gin Rummy. In poker, a hand is considered stronger than another based on a combination of factors, such as the ranking of cards contained in a straight or set (Billings et al. 1998). However, in Gin Rummy, it is the difference in deadwood points at the end of a round that dictates the payoff for the player. Tracking information sets that contain the number of deadwood points, the number of un-melded cards, and the types and number of melds allow us to consider a similar ranking system for card fitness.

Lastly, we wanted to look at more than just the cards value to the player, its value depending on the opponent’s playstyle, and what the current state of the game is. We found this outlet by looking into the probabilities of cards to understand the rhythm of the game better. “Computer card probabilities in Texas Hold’em” shed some light on this topic. Teófilo, Reis, and Cardoso discuss using Monte Carlo to create a strong strategy in the imperfect knowledge extensive game of Magic: The Gathering. This strategy involves three decision points that are categorized as attack, block, and card play. The first two decision points, attack and block, use either randomization or rule-based strategies, while the third, card play had a Monte Carlo strategy introduced. The authors used all combinations for those three options and pitted the resulting AI bots against each other. It found that the Monte Carlo strategy gave an advantage of about 5-7% over the rule based or random players. In fact, the Monte Carlo and random strategy players can beat a rule-based player given enough training time (Teófilo, Reis, and Cardoso 2013).

Since Gin Rummy can be broken up into a similar three decision points of draw, discard, and knock, the results of this paper suggest that the use of Monte Carlo may produce an effective Gin Rummy strategy. This strategy can then, similarly, be compared against other bots that utilize rule or random based strategies to determine how strong the Monte Carlo strategy is.

Methods

Estimating Card Fitness in Gin Rummy

To estimate card fitness in Gin Rummy, we simplified the potential number of game states by focusing solely on the discard action. Within this action, the player can choose any of the eleven cards in their hand to discard. Even with a limited context, each card’s value is tied to many factors within the current game state, such as how many deadwood points it contributes to the player’s hand, whether it is likely to become part of a meld, and whether it will benefit the opponent if they are given the chance to draw it. Some of these variables are known to the player and some are not. Thus, being able to judge the fitness, or utility, of a card when selecting one to discard is both valuable and complex.

This can be accomplished in multiple ways. One is to utilize Monte Carlo sampling (Lanctot et al. 2009) and Counterfactual Regret+ (Tammelin 2014) to produce a Nash Equilibrium strategy that relates specifically to card choice. Another is to choose a set of variables that can be combined to create a ranking system for each card. In this paper, we discuss a simplified combination of a ranking system and CFR strategy that led to an improvement in overall play against other bots that relied purely on rules and randomization.

Ranking the Fitness of a Card in a Game State

To determine a card's ranking, we created a point system that assigned a numerical ranking value to each candidate discard card. First, candidate cards were determined from the player's hand by selecting only cards that, when discarded, resulted in the minimum deadwood points possible. Then, each candidate card was checked to determine if the average deadwood points decreased after an additional draw, if the card is unmeldable based on cards the player has seen, if the card cannot be used by the opponent for a meld, and if the opponent has discarded a card that would be melded with it. For each of these factors, the card would be assigned 1 point, with a maximum ranking value of 4. After each card was assessed, only the candidate card(s) with the highest-ranking value were kept for potential discard as seen in figure 1.

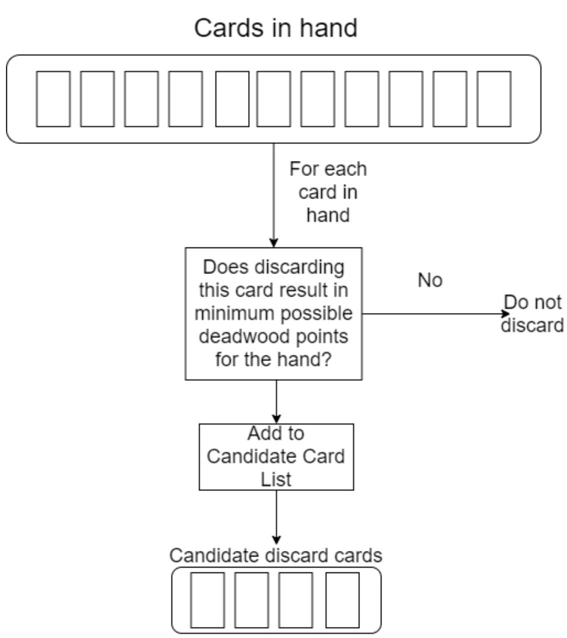

Figure 1

From this selection of candidate cards, Counterfactual Regret+ was used in conjunction with Monte Carlo sampling in order to achieve a Nash Equilibrium strategy. This strategy was used when the heuristics and card fitness evaluation result in a tie for the highest ranked potential discards. The information set for each card consisted of: Card ID (Suit and Value), Stage in the game (early, middle and late where early was classified as 20 to 30 cards being dealt, middle as 31 – 40 cards being dealt and 41 – 50 for late), along with the current deadwood of the player's hand. This generated a CFR strategy that complemented the ranking characteristics by looking at additional aspects of the game state, while minimizing the calculation time to determine the strategy. Training took place approximately 12 hours a day over the course of a week until the strategy appeared to converge. The generated table of over 24,000 values was converted into a lookup table where the agent could evaluate each card on whether they should discard that card, or that it held some value unforeseen by heuristics.

Due to the large game tree, there were some game states that did not arise during the Monte Carlo CFR+ sampling. In these cases, where no CFR strategy was found, a discard would be chosen at random from the original set of highest-ranked candidate cards. Additionally, if at any point only 1 candidate card is left in the set, that card is immediately chosen as the discard.

Testing Effectiveness Through Gameplay Against Other Agents

To test the effectiveness of this discard strategy, we inserted it into a Gin Rummy agent that contained draw and knock strategies utilizing pre-trained Monte Carlo CFR strategies and simple heuristics. This agent will be referred to as V4 and was played against a set of 3 other agents through several evaluation rounds.

To directly compare the ranked card fitness and CFR strategy against a strategy that looked at those characteristics of the card independently, without a combined ranking system, we developed an additional bot. This bot, V2, had identical draw and knock strategies to V4, but had a simpler discard strategy. Instead of increasing a point value for each evaluation point, V2 removed a candidate card as soon as it returned true on any one of the ranking considerations. If more than one candidate card remained after these evaluations, a discard was chosen at random instead of by CFR.

Other Agents Used in Evaluation

Two control group bots were also used, one simplistic agent, and one intermediate agent. The Simple agent's strategy for the draw decision point only considers whether the face up card will create a meld with its current hand. Its discard strategy considers the card that will decrease the deadwood of the hand the most and, if there are any ties, the discarded card is randomly chosen from the cards within the tie. Its knock strategy is to knock as soon as allowed by the game's rules.

The Medium bot's strategies at the same decision points are as follows. The draw strategy considers if the face up card would produce a minimum drop in deadwood after a card is discarded. The discard strategy determines which cards cannot be made into a meld with the known cards in the deck and creates a set of those cards. From that set, it considers if all cards can be made into a meld, then it chooses the cards with the greatest deadwood decrease by discarding them. From there it determines which cards are least likely to help the opponent by giving them a card for a meld they are creating and if any cards remain after, choosing the card with the highest deadwood value. Finally, the Medium bot's knock strategy is to knock when it has a total of seven or nine deadwood in its hand or has gin. The seven or nine deadwood rules relate to whether the bot is likely to undercut the opponent.

Evaluation

The two agents, Fitness + CFR and Independent Heuristics, were evaluated against each other and independently against the control group bots. This was to determine if there was a valid advantage to combining the heuristic rules into a ranking system that worked with the CFR strategy. Each agent under evaluation was played against the other bots in a series of games up to 10,000 games, recording the number of wins against the opponent agent.

Results and Analysis

Through the conducted experiment it was concluded that the agent which used the combined heuristics used to determine card fitness in a hand and a Nash Equilibrium strategy, referred to as "Fitness + CFR", outperformed the agent using independent heuristics and random selection as a tie breaker, referred to as "Independent Heuristics". Fitness + CFR won on average 7.8% of the games played against Independent Heuristics. Figure 2 shows the number of games won by each bot against the other over the course of the experiment.

When evaluated against the control group agents, the same pattern emerges where Fitness + CFR consistently outperforms Independent Heuristics. Against the simple bot, Fitness + CFR won on average 4% more games than Independent Heuristics as seen in figure 3.

Even further, against the middle intermediate agent, Fitness + CFR won on average 3% more games than Independent Heuristics shown in figure 4.

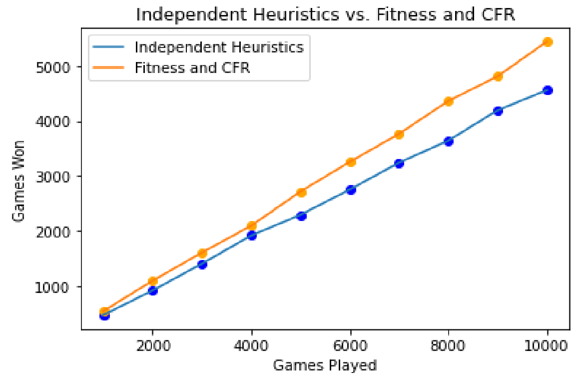

Figure 2

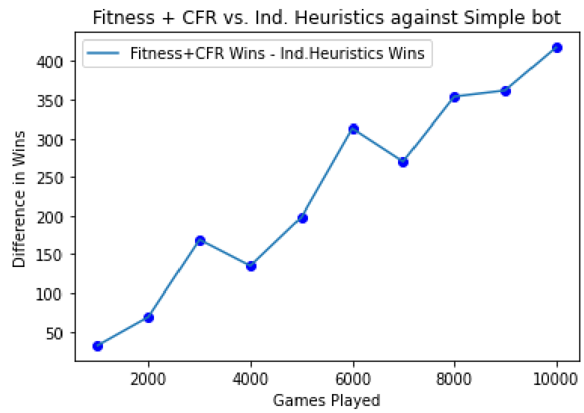

Figure 3

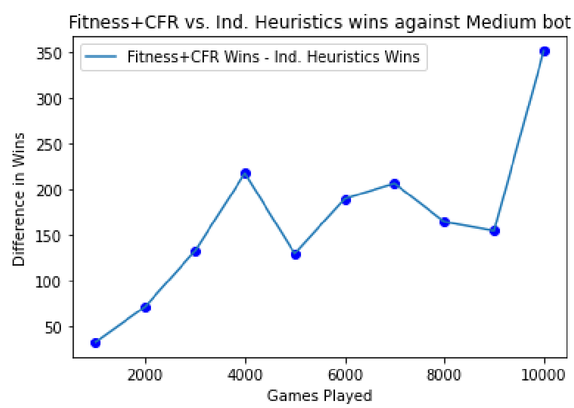

Figure 4

Conclusions and Future Work

From here a conclusion can be drawn that a strategy that combines multiple heuristics to determine which card to discard, followed by a Nash Equilibrium strategy will always outperform a strategy with independent heuristics and random choice to break ties. It should be noted that at this modification to the strategy impacts the performance against high level bots (such as Independent Heuristics) significantly more so than a lower, less sophisticated bot. This most likely is because the simpler agents' strategies are more easily exploitable, allowing for more advanced bots to consistently perform at a similar level against them. Whereas when two advanced agents play, the small details are important to avoiding mistakes that could be exploited by the opponent. Thus, it can be concluded that evaluating all characteristics of a hand and the known information set for discard can be the determining factor for a win over a loss in an imperfect information game such as Gin Rummy.

Due to the limited scope of our research time for this experiment, there are several ways that future work could build upon our findings. One addition could be to take this estimation of card fitness at discard and test it against a wider variety of bots or human players. Future work could also be put into longer training times to develop a more robust CFR strategy that truly does encounter every combination of cards at that decision point.

Another idea would be to test if there are more heuristical approaches that, when combined with CFR or left on its own, could perform better than pure CFR. We did attempt to add another heuristic to our bot that would cause it to knock immediately when a certain point threshold was reached, but the bot's ELO decreased as the threshold got lower, and never ended up performing better than the bot without this approach in it.

There is also the potential that a strategy based so strongly on game abstraction could be exploitable in certain circumstances. One of these circumstances involves game abstractions such as limiting the CFR strategy to early, mid, and late times in the game. This kind of weakness is explored by Sandholm in his paper Abstraction for Solving Large Incomplete-Information Games, where he describes a function that could be used to deal with cases that fall between known abstraction points (Sandholm 2015). Something similar could be explored within the discard card fitness abstractions.

Additionally, future works could include evaluating each card given an information set for their importance to the success of a given hand using Shapley Value Regression. Discussed by Song, Nelson, and Staum, the use of Shapley Value Regression can accurately determine important factors for a given outcome. Shapley Value can also be evaluated using Monte Carlo Sampling in a similar manner to Counterfactual Regret Minimization (Song, Nelson, and Staum 2016). This could be further examined by exploring use of Shapley Value on dependent variables; however, this greatly increases the computation time for evaluation (Song, Nelson, and Staum 2016).

References

Billings, D.; Papp, D.; Schaeffer, J.; and Szafron, D. 1998. Opponent Modeling in Poker. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*: 493–499. Menlo Park, Calif.: AAAI Press.

Lanctot, M.; Waugh, K; Zinkevich, M; and Bowling, M. 2009. Monte Carlo Sampling for Regret Minimization in Extensive Games. *Advances in Neural Information Processing System*s 22: 1078-1086.

Sandholm, T. 2015. Abstraction for Solving Large Incomplete-Information Games. In *Proceedings of the Twenty-Ninth National Conference on Artificial Intelligence*: 4127-4131. Palo Alto, Calif.: AAAI Press.

Song, E.; Nelson, B. L.; and Staum, J. 2016. Shapley Effects for Global Sensitivity Analysis: Theory and Computation. *Uncertainty Quantification* 4(1), 1060–1083.

Tammelin, O. 2014. Solving Large Imperfect Information Games Using CFR+. arXiv eprint. arXiv:1407.5042 [cs.GT]. Ithaca, NY: Cornell University Library.

Teófilo, L. F..; Reis, L. P.; and Cardoso, H. L. 2013. Computing Card Probabilities in Texas Hold'em. *8th Iberian Conference on Information Systems and Technologies* (CISTI): 1-6.

Zinkevich, M.; Johanson, M; Bowling, M; and Piccione, C. 2008. Regret Minimization in Games with Incomplete Information. *Advances in Neural Information Processing Systems* 20: 905-912.

Parlett, D. 2020. Gin Rummy History. *Historic Card Games by David Parlett*:

https://www.parlettgames.uk/histocs/ginrummy.html. Date Accessed: 10 August 2020.

Scarne; John; and Karger, J. 2004. Scarne on Card Games. H*ow to Play and Win at Poker, Pinochle, Blackjack, Gin and Other Popular Card Games*: Dover Publications.

Bicycle Cards 2020. Gin Rummy. A Modern Classic that is One of the Most Demanding of All Card Games. *How to Play: Gin Rummy*:

https://bicyclecards.com/how-to-play/gin-rummy/. Date Accessed: 10 August 2020.