Models of Strategic Deficiency and Poker

Gabe Chaddock, Marc Pickett, Tom Armstrong, and Tim Oates
University of Maryland, Baltimore County (UMBC)
Computer Science and Electrical Engineering Department
{gac1, marc}@coral-lab.org and {arm1, oates}@umbc.edu

Abstract

Since Emile Borel's study in 1938, the game of poker has resurfaced every decade as a test bed for research in mathematics, economics, game theory, and now a variety of computer science subfields. Poker is an excellent domain for AI research because it is a game of imperfect information and a game where opponent modeling can yield virtually unlimited complexity. Recent strides in poker research have produced computer programs that can outplay most intermediate players, but there is still a significant gap between computer programs and human experts due to the lack of accurate, purposeful opponent models. We present a method for constructing models of strategic deficiency, that is, an opponent model with an inherent roadmap for exploitation. In our model, a player using this method is able to outperform even the best static player when playing against a wide variety of opponents.

Introduction

The game of poker has been studied from a game theoretic perspective at least since Emile Borel's book in 1938 (Borel 1938), which examined simple 2-player, 0-sum poker models. Borel was followed shortly by von Neumann and Morgenstern (v. Neumann & Morgenstern 1944), and later by Kuhn (Kuhn 1950) with developing simplified models of poker for testing new theories about mathematics and game theory. While these models worked well and served as a catalyst for research in the emerging field of computer science, they are overly simple and less useful in today's research. Practical applications for research in full-scale adversarial games of imperfect information are pervasive today. Goods, services and commodities like electricity are traded and auctioned online by autonomous agents. Military and homeland security applications such as battlefield simulations and adversary modeling are endless and the entertainment and gaming industries have used these technologies for years.

In the real world, self interested agents are everywhere, and imperfect information is all that is available. It is domains such as these that require new solutions. Research in games such as poker and bridge are at the forefront of research in games of imperfect information.

Opponent modeling has been identified recently as a critical component in the development of an expert level poker player (Billings *et al.* 2003). Because the game tree in Texas Hold'em, a popular variant of poker, is so large, it is currently infeasible to enumerate the game states, let alone compute an optimal solution. By using various abstraction methods, the game state has been reduced but a suboptimal player is not good enough. In addition to hidden information, there is misinformation. Part of the advanced poker player's repertoire is the ability to bluff. Bluffing is the deceptive practice of playing a weak hand as if it were strong. Indeed there are some subtle bluffing practices where a strong hand is played as a weak one early in the hand to lure in unsuspecting players and with them more money in the pot. These are just a few expert-honed tricks of the trade used to maximize gain. In some games it is appropriate to measure performance in terms of small bets won. This is often applied to games that are played over and over again thousands of times. The more exciting games such as No Limit Texas Hold'em have much greater short term winning potential and often receive more attention. It is this variant of poker that is played for the Main Event at the World Series of Poker (WSOP) as of 2006 which establishes the world champion.

Rather than developing a model of an opponent's strategy, we seek to develop a model of strategic deficiencies. The goal of such a model is not to predict all behaviors but, instead, identify which behaviors lead to exploitable game states and bias an existing decision process to favor these states.

We developed a simulator for 2-player no-limit Texas Hold'em to demonstrate that models of weakness can have a clear benefit over non-modeling strategies. The agents were developed using generally accepted poker heuristics and parameterized attributes to make behavioral tweaks possible. The simulator was also used to identify emergent properties that might yield themselves to further investigation. In the second set of simulations, the agents are permitted to take into account a rudimentary model of an opponent's behavior. This model is comprised of tightness which is characterized by the frequency with which an agent will play hands. This taxonomy of players, including aggression (i.e., the amount of money a player is willing to risk on a particular hand), was first proposed by Barone and While (Barone & While

1999), (Barone & While 2000) and makes a simple yet motivating case for study.

We constructed a model for a poker player that is parameterized by these attributes such that the player's behavior can be qualitatively categorized according to Barone and While's taxonomy. Some players in this taxonomy can be considered to have strategic deficiencies that are exploitable. Using this model, we investigate how knowledge of a player's place in the taxonomy can be exploited.

The remainder of this paper is organized as follows. The next section describes related work on opponent modeling and prior attempts to solve poker. We then describe our approach to modeling opponent's deficiencies and discuss our experimental design to test the utility of our models. Next, we evaluate the results of our experiment and discuss the implications before concluding and pointing toward ongoing and future work.

Poker and Opponent Modeling

The basic premise of poker will be glossed over to present some of the terminology used in the rest of the paper. The interested reader should consult (Sklansky & Malmuth 1994) for a proper treatment of the rules of poker and widely accepted strategies for more advanced play.

Texas Hold'em is a 7-card poker game where each player (typically ten at a table) is dealt two *hole cards* which are private and then shares five public cards which are presented in three stages. The first stage is called the *flop* and consists of three cards. The next two stages each consisting of one card are the *turn* and the *river*.

A wide variety of AI techniques including Bayesian players, neural networks, evolutionary algorithms, and decision trees have been applied to the problem of poker with marginal success (Findler 1977). Because of the complexity of an opponent's behavior in a game such as poker, a viable alternative is to model opponent behavior and use these models to make predictions about the opponents cards or betting strategies.

There have been recent advancements in poker and opponent modeling. Most notable perhaps is the University of Alberta's poker playing bot named Poki, and later a pseudo optimal player PsiOpti (Davidson *et al.* 2000).

Many learning systems make the assumption that the opponent is a perfectly rational agent and that this opponent will be making optimal or near-optimal decisions. In poker, this is far from the case. The large state space in this game and the rich set of actions a player can choose from creates an environment with extraordinary complexity. The ability to bluff or counterbluff and other deceptive strategies create hidden information that is not easily inferred. In addition, the high stakes that usually accompany poker and the psychological aspect of the game lead to interesting and bizarre plays that sometimes seem random. In variants of Texas Hold'em such as No-Limit, a player can risk their entire stack at any time.

Models of opponent behaviors can provide useful advantages over agents that do not use this information (Billings *et al.* 2003). In games of imperfect information this becomes especially apparent since the hidden cards and misinformation provide richness in terms of game dynamics that does not exist in most games. A special case of an opponent model is a model of weakness. By determining when an opponent is in a weak state, one can alter their decision process to take advantage of this. Using a simpler model has advantages as well. For example, since the model is only used when these weak states are detected, there exists a decision process that is independent of the model. That is, the model only biases or influences the decision process rather than controlling it.

Markovitch has established a new paradigm for modeling weakness (Markovitch & Reger 2005) and has investigated the approach in the two-player zero-sum games checkers and Connect Four. Weakness is established by examining next board states from a set of proposed actions. An inductive classifier is used to determine whether or not a state is considered weak using a teacher function and this determination is used to bias the action selection policy that is being used.

Markovitch addresses the important concepts of risk and complexity are addressed. The risk involved is that the use of an opponent model would produce an incorrect action selection- perhaps one with disastrous consequences for the agent. Markovitch has established that complexity can be managed by modeling the weakness of an opponent rather than the strategy. By modeling weakness, an agent works with a subset of a complete behavioral model that indicates where exploitation can occur. Exploitation of weakness is critical to gain an advantage in certain games like poker. Risk is reduced in the implementation by designing an independent decision process that is biased by the weakness model. This way, an incomplete or incorrect model will not cause absurd or detrimental actions to be selected.

Their algorithm relies on a teacher function to decide which states are weak. The teacher is based on an agent which has an optimal or near optimal decision mechanism in that it maximizes some sort of global utility. The teacher must be better than a typical agent assuming that during ordinary play, mistakes will be made from time to time. The teacher is also allowed greater resources than a typical agent and the learning is done offline. It is also possible to implement the teacher as a deep search. Weakness is distilled to a binary concept which is then learned by an inductive classifier.

They evaluate their teacher-driven algorithm on 2-player, zero-sum, perfect information games (e.g., Connect Four, checkers). In this research, the concept of models of weakness is taken to a new level by applying it to the game of heads up no-limit Texas Hold'em and excluding the use of a teacher. Heads up poker is still a 2-player, zero-sum game but features imperfect information in the form of private cards. Imperfect information makes poker a difficult domain because there are hundreds of distinct hand rankings with equal probability and near continuous action sets leading to millions of game states. It is computationally difficult to evaluate all of these states in an online fashion. A shallow lookahead is simple in board games like checkers because the number of actions from each state is relatively small. The states can be enumerated and evaluated with a

trivial amount of computational power. The ability to enumerate board states and look ahead to evaluate them is a convenience that is not available in games such as poker.

Behavior Space Analysis and Strategic Deficiency

Since strategy can be quite complex, modeling it in a game like poker is computationally burdensome. Since most players follow certain established 'best practices' it makes sense to look for the places where players deviate from these generally accepted strategies. Being able to predict what cards an opponent holds is very important. In the showdown (when players who have made it this far compare cards) the best hand wins. Knowing what an opponent's hole cards are can make a huge difference in the way the hand plays out. Knowing you possess the best hand is reason to go all the way no matter the cost. At this point the only concern is how to extract more money from one's opponent. The second most important reason to have a model is being able to predict an opponent's betting behavior. This includes reacting to one's own bets. Being able to predict an opponent's reactions will allow a player to extract maximum winnings from them. Being able to maximize winnings means turning a mediocre pot into substantial earnings. It is also valuable to eliminate opponents in tournament settings. The idea behind modeling opponents' strategic deficiencies is that it is a simple but effective way to maximize winnings.

We now present the concept of a player behavior space using features proposed by Barone and While illustrated in figure 1. The purpose of this figure is to illustrate how behaviors can be mapped out in space and observing a player's trajectory through this space can reveal a lot about the player's skill level, playing style, and emotional state.

Using the four broad categories Loose Passive, Loose Aggressive, Tight Passive, Tight Aggressive, we can observe a player in the state space shown in figure 1. In this space, some strategies are clearly inferior to others. For example, an agent at the extreme bottom of the figure will have a overly strong tendency to fold (and will fold every hand if completely tight). On the other extreme, a completely loose agent will refuse to ever fold, and will be betting on cards that have a low probability of winning. A similar balance must also be made with an agent's aggression.

Because of the complexity of poker, it is necessary to use common heuristics for playing to identify exploitable conditions. These conditions come in a variety of formats and it is required to use different classifiers with different features to model them. For example, aggressive styles can be discovered by looking at how much an opponent bets.

In figure 2, we see how a player's strategy might change over the course of several hands. For example, the "tight, aggressive" player in the lower right corner might have a series of wins and follow trajectory (a), causing it to be less tight (and fold less often), or the player might lose a few high-stakes hands and follow trajectory (b), causing it to keep its bets lower. However, our model assumes a player remains consistent over multiple hands. Investigating the effects of game events on a player's strategy is left for future work.
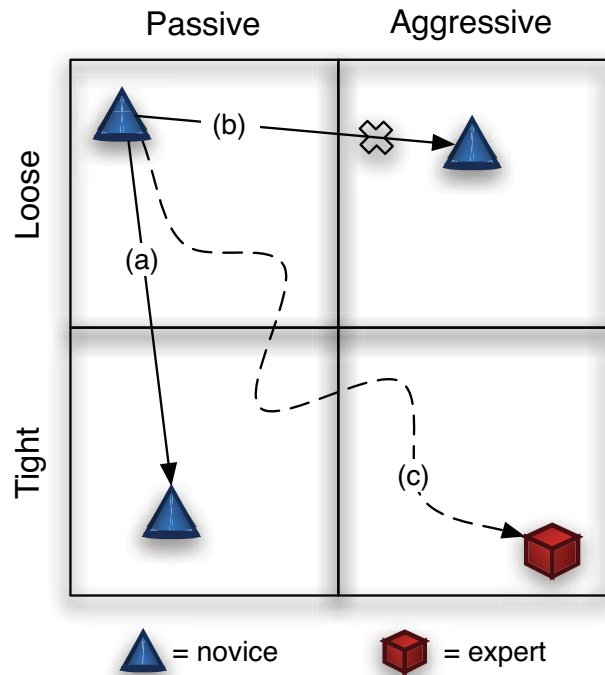

Figure 1: Player Behavior Space

A strategic deficiency in general is a lapse in a poker player's ability to make sound decisions consistently. This deficiency could be attributed to a lack of experience in a human player or perhaps a static action selection mechanism employed by a computer program. These strategic deficiencies are quasi permanent and can be exploited over and over again. There are also temporary deficiencies, such as emotional transformations, which are more difficult to capitalize on, but these are beyond the scope of this paper.

An example of a weakness is a player that should fold a weak hand when an opponent makes a large raise early on. Of course, staying in could lead to an effective bluff, but doing this too often is indicative of a beginners reluctance to fold when they have already committed some money to the pot. This can be exploited by making moderately aggressive bets.

Empirical Results

We created several experiments to evaluate the importance of modeling strategic deficiency. There are many such models with very different feature spaces, and we present one such models will be explored in this section: tightness (we empirically discovered that tightness, and not aggression, seemed to be the dominant factor in our models).

We build a poker simulation to perform an empirical evaluation of behaviors and game dynamics. The program simulated repeated games of heads up (2-player) no-limit Texas Hold'em. The two players were coded so that their tightness could be scaled between 0 and 1. A tightness of 0,
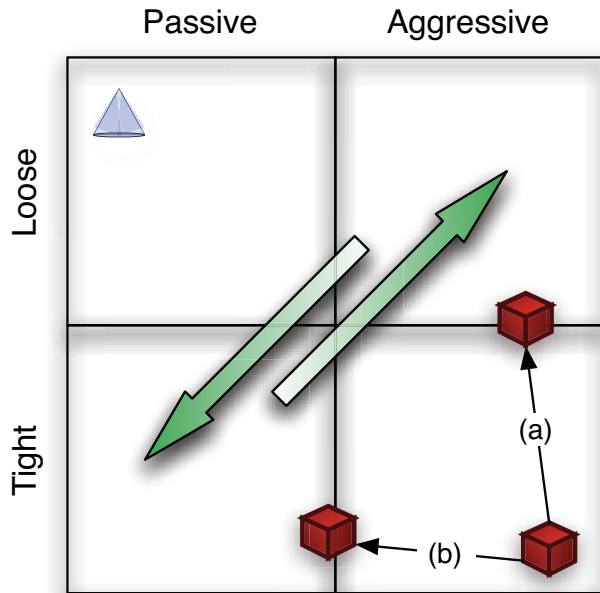
Figure 2: Behavioral Transformations

for example, would mean that an agent would never fold and a tightness of 1 meant that the agent would fold any time money was required to meet a bet or raise. The agents also took into account their hand strength by computing an approximation of their probability of winning. Each agent used Monte Carlo simulations to determine their probability of winning at each stage of betting using the cards that were visible to them. The agent would deal out some number of random hands and count how many times it would beat those hands, assuming an equal distribution. In order to generate realistic game play, the agents played against each other with random settings for these parameters.

A "hand" of poker consists of a single set of up to 9 cards (2 cards for each player, and 5 table cards). A "game" of 2-player poker consists of several hands. For our experiments, a game started with each player having 100 betting units, and the game ends when one of the players has taken all the betting units from the other player (thus ending with 200 units).

One of the primary goals was to determine if there was an optimal static strategy that would represent the strongest player and to make sure that these parameters were not on the boundaries of our behavior space. The second research goal was to examine the behavior of a dynamic player that is, an agent that can alter its parameters in response to an opponent's playing style. The way this was accomplished was to allow one of the agents to know the other agent's static parameters. This is realistic to assume since with repeated interaction one can generally infer an opponent's tightness and aggression whether playing against human opponents or computer agents.

The empirical study produced several results. It is im-

portant to note that even in the most simple of experiments, behavioral patterns emerged. These patterns were useful in determining when an opponent was playing in a strategically deficient manner and allowed an obvious exploit to be made. The plot in figure 3 shows wins and losses for games played where aggression was held at 0.5 and the tightness of both players was modified systematically. By observing the outcomes of this experiment with various constant values of aggression, it was easy to see how tightness affected the results. For example, setting player1's tightness to a very low level (e.g., 0.03) will usually cause him to lose to player2. This makes sense because a low tightness corresponds to a "loose" player that tends to play too many hands (by not folding when its cards are unfavorable).

We used our model to obtain Monte Carlo estimates for the probability of winning a heads up game given each player's tightness. To obtain these estimates, we discretized each player's tightness into 101 partitions from 0 to 1 (inclusive) in increments of .01. For each pair of tightness values, we generated 100 games. The players are symmetrical, and there are no ties, so the probability of Player 1 winning is 1 - the probability of Player 2 losing. Exploiting this, we effectively had 202 games for every pair of tightness values. The results are shown in figure 3, with the blackness of an x, y point corresponding to the probability that Player 1 will lose. Note that, due to symmetry, the probability of winning against an opponent with the same tightness value as a player is .5.

We used these estimated probabilities to determine the probability of winning a game for a particular tightness value against an unknown opponent (assuming a uniform distribution of opponent tightness values). These probabilities are shown in figure 4. Using this graph, we can see a strategy with a tightness value has about a .4 probability of winning against a randomly picked opponent. Note that the area under this curve should be exactly .5. This curve peaks for a tightness value of .47. Therefore, the best static strategy (i.e., a strategy that's not allowed to change its tightness value) is to set the tightness value to .47. Note that the worst static strategy is to set the tightness value to 1, which corresponds to folding whenever there's a raise. The only time this strategy might win is if the opponent also has a very high tightness.

We also used the data in figure 3 to generate a dynamic strategy. A dynamic strategy is allowed to view the opponent's tightness value, then change its own tightness value in response. To do this, we first smoothed the data using a Gaussian, to ameliorate the experimental error. Using this smoothed surface, for every tightness value of Player 2, we found the tightness value for Player 1 that resulted in the highest probability of winning. This graph is shown in figure 5, where the x axis is Player 2's tightness value, and the y axis is Player 1's response. For example, if Player 2's tightness is .2, then Player 1 should adapt by setting his/her own tightness value to .52. We've plotted the line $x = y$ to show where the Nash equilibria are for this strategy. These are the points of convergence if there are 2 dynamic agents. In this case, there is only 1 Nash equilibrium at .46. We suspect that it's merely coincidental that this is so close to the
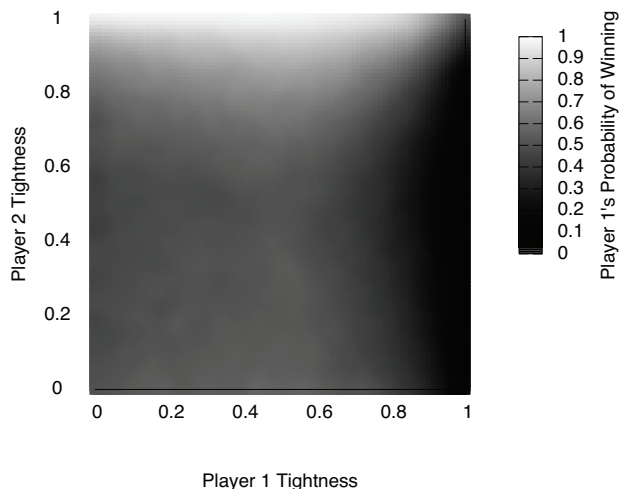
Figure 3: The effect of tightness on probability of winning. This graph shows Player 1's probability of winning depending his own tightness and on Player 2's tightness. Note both the (inverted) symmetry in the graph about $x = y$ and the white area near the top of the plot. This means that Player 1 is likely to win if Player 2 sets his tightness near 1 (so Player 2 is likely to fold), and symmetrically, Player 1 is likely to lose if he sets his tightness too high. Although not as obvious, the area at the extreme left of the graph is darker than the area in the middle. Thus, the optimal strategy is neither to be completely tight nor completely loose.

Figure 4: How tightness affects probability of winning against a randomly selected opponent. If players aren't allowed to adjust their tightness, the best tightness is .47, which gives a 61% probability of winning against a opponent whose tightness is randomly selected (uniformly from 0 to 1).

optimal static value (.47).

We compared performance of our best static strategy (fixing the tightness value to .47) against the dynamic strategy. To do this, we ran each 100 times against each of 101 opponents (distributed uniformly from 0 to 1, inclusive). This made a total of 20,200 games. The dynamic strategy had a marginal, but statistically significant advantage over the static strategy: the dynamic strategy won 6,257 (or 61.95%) of its games while the static strategy won 6,138 (or 60.77%) of its games. Using a t-test, we calculated that this is well over 95% significant ($t = 1.7138$, where the 95% confidence interval is when $t \geq 1.645$). Since a randomly chosen static strategy is expected to win half its games (50%), the static strategy is a 10.77% improvement over random, and the dynamic strategy is a 11.95% improvement over random. This means that the dynamic strategy is (11.95% - 10.77%)/10.77% = 10.96% improvement over the static strategy.

Conclusion

Our models demonstrate the utility of exploiting an opponent's strategic deficiency. Future work using these models will proceed in three directions. We will develop methods for autonomously discovering deficiency models using hybrid features and composite behavior spaces; these models may result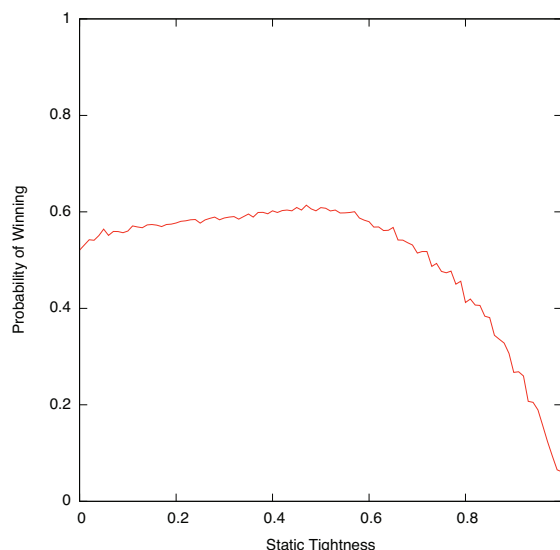 in unintuitive, yet powerful models. Second, we are interested in using the learned models to discover classes of opponent's weaknesses (e.g., temporal changing behavior, intimidation, etc.). Once weakness classes are discovered, we will evaluate our models' effectiveness against various weakness classes. Finally, we hope to extend our work to additional game domains where we can explore classes of games and transfer of learned models.

Poker remains an important domain for research in artificial intelligence. The real world applications that can benefit from this research are very complex and cannot benefit from overly-simplified games. Since poker is an example of a domain that mirrors the complexity of real world problems, it is the authors belief that beneficial research must come from full-scale versions.

The complexity of the domain can be partially handled by methods of abstraction that reduce the spaces to more tractable sets. Additional benefit is derived from choosing to model opponents only in terms of their strategic deficiencies. This approach offers the benefit of reduced complexity and managed risk. It is not intended as a replacement for an action selection mechanism but, instead, a supplemental source of information. This information is not always available and is is not always actionable when it is. When the model can be used, however, it provides enormous earning potential on hands that would otherwise slip by. Since the ultimate goal in poker is to win money, we use the model to help us do so in a focused manner.

References

Barone, L., and While, L. 1999. An adaptive learning model for simplified poker using evolutionary algorithms.
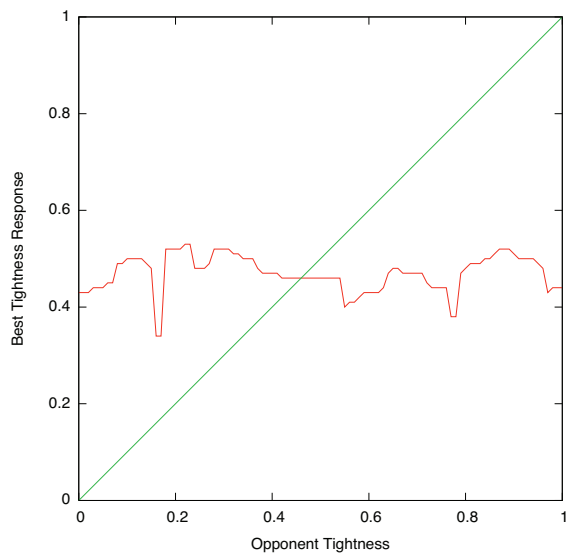
Figure 5: **The Dynamic Strategy** This plot shows a player's strategy for setting its own tightness in response to an opponent's tightness. The line $x = y$ is plotted to show where the Nash equilibrium is, at .46.

In *Proceedings of the Congress of Evolutionary Computation*, 153–160. GECCO.

Barone, L., and While, L. 2000. Adaptive learning for poker. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 566–573.

Billings, D.; Burch, N.; Davidson, A.; Holte, R.; Schaeffer, J.; Schauenberg, T.; and Szafron, D. 2003. Approximating game-theoretic optimal strategies for full-scale poker. In *In Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*.

Borel, E. 1938. *Applications aux Jeux des Hazard*. Gautier-Villars.

Davidson, A.; Billings, D.; Schaeffer, J.; and Szafron, D. 2000. Improved opponent modeling in Poker. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI'2000)*, 1467–1473.

Findler, N. 1977. Studies in machine cognition using the game of poker. In *CACM 20 (4)*, 230–245.

Kuhn, H. W. 1950. Simplified two-person poker. In Kuhn, H. W., and Tucker, A. W., eds., *Contributions to the Theory of Games*, volume 1. Princeton University Press. 97–103.

Markovitch, S., and Reger, R. 2005. Learning and exploiting relative weaknesses of opponent agents. *Autonomous Agents and Multi-agent Systems* 10(2):103–130.

Sklansky, D., and Malmuth, M. 1994. *Hold'em Poker for Advanced Players*. Two Plus Two Publishing, 2nd edition.

v. Neumann, J., and Morgenstern, O. 1944. *The Theory of Games and Economic Behavior*. Princeton University Press.