Fish or Shark – Data Mining Online Poker

Ulf Johansson and Cecilia Sönströd

Abstract— **In this paper, data mining techniques are used to analyze data gathered from online poker. The study focuses on short-handed Texas Hold'em, and the data sets used contain thousands of human players, each having played more than 1000 hands. The study has two, complementary, goals. First, building predictive models capable of categorizing players into good and bad players, i.e., winners and losers. Second, producing clear and accurate descriptions of what constitutes the difference between winning and losing in poker. In the experimentation, neural network ensembles are shown to be very accurate when categorizing player profiles into winners and losers. Furthermore, decision trees and decision lists used to acquire concept descriptions are shown to be quite comprehensible, and still fairly accurate. Finally, an analysis of obtained concept descriptions discovered several rather unexpected rules, indicating that the suggested approach is potentially valuable for the poker domain.**

Keywords: Concept description, Decision trees, Decision lists, Comprehensibility, Online poker, Texas Hold'Em

I. INTRODUCTION

In this application paper, we apply a bag of data mining tricks with the intention of studying datasets acquired from online poker. The overall goal is to analyze player profiles, trying to pin-point what separates successful, i.e. winning, players from players losing money. More specifically, the purpose is to combine explorative, descriptive and predictive techniques, in order to make it possible to not only predict if a certain player profile represents successful play, but also to obtain concept descriptions for winning and losing players. In addition, we compare strategies used at low-limit and middle-limit.

II. BACKGROUND

Texas Hold'em (often referred to as just "Hold'em") is the most popular poker game played online. The first subsection therefore presents the rules of the game. The second subsection discusses general basic strategy. The third subsection describes *short-handed play*, which is the focus of this study. Short-handed play is defined as six or fewer players at a table. The fourth subsection, finally, gives a brief description of a family of software products, used by online poker players for, among other things, profiling their opponents.

A. Texas Hold'em poker

When playing Hold'em, each player is dealt two private cards face down. These cards are referred to as *hole cards*. Now the initial betting round takes place. After that, three public cards (the *flop*), are placed face up in the middle of the table. The second betting round follows. When the betting round has finished, another public card (the *turn*), is placed alongside the flop. Next is the third betting round. After that, the final, fifth, public card (the *river*) is turned up, followed by the final betting round. Each player still remaining in the pot now combines the public cards with her hole cards to obtain a five card poker hand. When doing so, a player may use one, both or none of her hole cards. Naturally, the player now (at the *showdown*) having the best poker hand wins the pot.

The betting structure varies between games, but the games studied in this paper are *fixed limit*; i.e., betting is restricted to bets and raises of predefined amounts. As an example, a fixed limit game at the level $0.5-$1, would have the following betting structure: Before the deal the player to the left of the dealer must place a forced bet of $0.25 (the *small blind*) whereas the next player (the *big blind*) must make a similar bet of $0.50. In the first betting round, the player to the left of the big blind is first to act. Her options are to either concede the hand (*fold*), pay an amount equal to the big blind to remain in the hand (*call*) or make a bet twice the size of the big blind (*raise*). After a raise, all players must, of course, match the size of the raised bet to remain in the hand, or raise the bet again to $1.50 (*reraise*). Normally, only three raises and reraises are allowed, making the betting limit (called the *cap*) for the first round $2.00. In the second betting round, the first remaining player to the left of the dealer acts first. Now she has the option to remain in the hand without making a bet (*check*), but as soon as one player makes a bet ($0.50), all players must again either fold, call or raise. The third and fourth betting rounds are identical to the second, with the important difference that all bets are doubled; i.e. a bet or raise is now $1.00, and the cap is $4.00.

B. Basic strategy

As most beginners soon find out, Hold'em is a very sophisticated game, requiring mastery of many different skills. Some of these skills, most importantly the ability to read and interpret physical tells, are, however, of no value when playing online. On the other hand, the very short time available for each decision (compared to when playing "live") makes it extremely important to be able to quickly assess the situation and act accordingly. Furthermore, many players elect to play on multiple tables simultaneously, reducing the available time even more. From this, it is fair to say that most decisions online actually have to be made based on a very limited analysis of the specific situation. Or,

Ulf Johansson and Cecilia Sönströd are both with the CSL@BS at School of Business and Informatics, University of Borås, Sweden. e-mail: {ulf.johansson, cecilia.sonstrod}@hb.se

put in another way, players must rely on a *basic strategy* (almost an "auto-pilot"), which they normally will adhere to.

It must be noted that although the term basic strategy seems to suggest a simple mapping, from a situation to an action, this involves a number of subtleties. First of all, the situation is almost always both complex and only partially observable. In addition, in all betting rounds except the last, future, random events (new cards), will seriously change the situation. Finally, the actions taken by a player will of course also change the situation. Using terminology from agent theory (see e.g. [1]) the environment for poker is in fact *multi-agent*, *partially observable*, *strategic*, *sequential*, *static* and *discrete*. So, a basic strategy is actually a mapping from an *observed* situation to an action. Naturally, a perfect strategy would always maximize the expected profit, given the current situation, so a successful basic strategy must both be able to correctly identify the current situation, and to recommend the best action given this situation. Specifically, the strategy should, in contrast to game theory, not assume that all opponents play optimally. Based on this, some key abilities needed for estimating the expected values for the different actions, are listed below:

- The ability to guess an opponent's hand, or more correctly, to narrow down an opponent's range of hands.
- The ability to calculate current (pot) odds and to estimate future (implied) odds.
- The ability to calculate probabilities for making certain hands.
- The ability to predict the effects of a certain action.

For some players, their basic strategy is very deliberate, typically the result of seriously studying of poker literature, their own previous play and their opponents. For other players, their basic strategy is, at best, just a number of "rules of thumb". Although there exist both intentional (e.g. when a player "mixes up" her play in order to be more deceptive) and unintentional (e.g. when a player is upset after suffering a *bad beat*) deviations from the basic strategy, the quality of the basic strategy will ultimately determine the player's success.

Most poker literature, aimed at beginners and intermediate players, discuss basic strategy in terms of two dimensions; *loose-tight* and *passive-aggressive*; see e.g. [2][3]. The loose-tight dimension primarily captures how many hands a specific player decides to play, and how far she will take the hand. The passive-aggressive dimension describes how much a player, when playing a hand, bets and raises rather than checks and calls. The overall recommendation is to play tight and aggressive; i.e., to play very few hands, but when playing make a lot of bets and raises, as long as the expectation appears to be positive. The underlying assumption is that this way of playing will lead to a relatively high percentage of pots won when contending.

Based on this description, four player archetypes have emerged; the *calling station*, the *rock*, the *maniac* and the *solid player*. A calling station is the weakest opponent and is loose-passive, i.e., plays a lot of (poor) hands and constantly takes them too far. In addition, she prefers to always call and rarely raises. A calling station will almost surely lose money quickly; as a matter of fact this player is most often referred to as a *fish*, i.e., easy prey for the *sharks*. A rock is a tight-passive player who plays mainly premium hands. The conservative hand selection makes it rather hard to win any significant money from a rock, but the fact that she is quite predictable and does not play aggressively enough, makes her only mildly successful, typically winning a few small pots to approximately break-even. The maniac is loose-aggressive, playing a lot of hands, just like the calling station, but the difference is that the maniac constantly bets, raises and reraises. A maniac will inevitably lose money in the long run, but if lucky during a session, may leave it as a big winner. In addition, the maniac forces other players to adapt to her style, thus introducing giant fluctuations in everybody's bankroll. The solid player, finally, is tight-aggressive, mostly playing premium hands and a few speculative hands. The most basic trademark of a solid player is that she rarely calls, most often she either folds or raises.

Naturally, the level (i.e. how large the bets are) affects the quality of the poker played. Until recently, most small stakes Hold'em games were very loose-passive; the predominant player was often the calling station. One reason for this is that tight play requires a lot of discipline, it is much more fun to gamble by playing many more hands. Another reason is that most players playing at the small stakes tables were unaware of even the most basic principles for successful strategies. The last two years, however, online poker has changed fundamentally. Most importantly, the number of beginners has decreased significantly, while many players have improved their play, mainly from acquiring a lot of experience. This is especially true for fixed limit Hold'em, since nowadays most beginners will actually start playing No limit Hold'em right away. In addition, as the quality improves, poor players will start to lose consistently, and eventually either quit or try to somehow develop their play. Either way, the games, even at the lowest levels, now tend to be more competitive.

C. Short-handed play

In this paper, we study only short-handed play. Historically, Hold'em theory relates mainly to full tables; i.e., tables with nine or ten players. Even though the mechanics of the two games are identical, playing short-handed is profoundly different. First of all, since the blinds remain the same, a player now has to pay the same amount to play six hands instead of ten. Even more importantly, the probability that at least one opponent has a "good" hand is dramatically decreased. The position (i.e. if you act first or last during a betting round) is also even more important when playing short-handed. Specifically, a player will play a large proportion of all hands from the inherently hard blind positions. The overall (and extremely simplified) implication is that a player has to adapt to short-handed play, mainly by being significantly more aggressive, but also by playing more hands.

The poker boom saw hundreds of new books, all trying to convince the reader that she could indeed be a consistent winner, just by following the advice given in the book. Since most beginners started by playing fixed limit, small stakes Hold'em at full ring tables, a large majority of these books focused on exactly that form of poker. Furthermore, the strategies suggested were clearly biased towards fairly safe play, i.e., tailored to make a small but certain profit at the loose-passive tables dominant at the time; see e.g. [4][5].

In books focusing on full ring play, the advice for short-handed play tends to be either very vague, such as "play more hands more aggressively", or outright incorrect like "you can regard short-handed play as a full table where the first four players have folded". Furthermore, the number of books focusing on short-handed play, especially addressing the improved level of play, is surprisingly small. One possible exception is [6] but here the presentation must be considered quite impeding. This is clearly a book written by poker professionals for an audience consisting of quite sophisticated players with high ambitions. With this in mind, the best source for learning the essential skills for short-handed play is instead regarded to be discussion forums on the Internet, where very good players share their thoughts on all forms of poker. Most famous are probably the very active forums at Two Plus Two Publishing [7] which include specific discussion groups for, for instance, short-handed limit play, on different levels.

D. *Profiling software*

Many players use software tools like Poker Office[1] Poker Tracker[2] or Hold'em manager[3] to automatically collect information about both their own play and their opponents. These programs, which are legal to use at most but not all online poker sites, track a large number of attributes for each player and hand, including every action taken by the player, several attributes describing the situation and, of course, the outcome of the hand. Naturally, only information available to the player can be stored. Specifically, cards held by opponents remain unknown, unless they were involved in a showdown. With this in mind, some players use these tools mainly for analyzing their own play between sessions. Other players, however, use the tools primarily for assessing their opponents, both when choosing which tables to play, and during actual hands. Generally speaking, these programs produce profiles, consisting of several attributes, for each player. The exact attributes used, and how the information is aggregated and presented, differ slightly between the programs, but they are all able to describe a specific player using the two dimensions; *loose-tight* and *passive-aggressive*. Naturally, the exact overall profit achieved by a player is also stored, making it fairly straightforward to, based on this attribute alone, target weaker players while avoiding players who have been successful in the past.

[1] www.pokeroffice.com
[2] www.pokertracker.com
[3] www.holdemmanager.net

III. METHOD

As mentioned in the introduction, the overall purpose of this study is to analyze data collected from online poker. More specifically, the first goal is to obtain accurate predictive models, making it possible to classify players as good or bad players (actually winners or losers) based on their profiles. The second goal is to develop concept descriptions, in order to be able to understand and analyze what separates winning and losing strategies.

A. *Data preparation and explorative analysis*

While collecting data, all six-handed tables with the limits \$0.50-\$1.00 and \$5-\$10 at a major online casino were monitored around the clock, for approximately three months. Hands played with less than four players were not added to the database. All data was imported into Hold'em manager, where a number of attributes were calculated for each player. All in all, a player profile, in this study, consists of 21 attributes describing the strategy used by the player, and one attribute measuring the player's success. In the following analysis and experimentation, only players having played more than 1000 hands were used. This resulted in two datasets consisting of 3262 (\$0.5-\$1.0) and 2555 (\$5-\$10) instances, i.e., players. In addition, two smaller datasets were generated, each consisting of players having played more than 10000 hands. These datasets have 270 (\$0.5-\$1.0) and 288 (\$5-\$10) instances. All attributes used in player profiles are described below.

- **Voluntarily put money in put (VPIP):** The percentage of all hands that the player has added money to the pot, not counting the forced bets in the blinds. This attribute is a vital indicator for how tight-loose the player is.
- **Pre flop raise (PFR):** The percentage of all hands that the player raised preflop. This is a very important attribute indicating how aggressive the player is.
- **PFR/VPP:** This value, which aggregates the two previous attributes, shows how tight-aggressive the player is preflop.
- **3Bet PF:** Percentage of times a player reraises preflop, when facing a raise. Important for determining whether the player is passive or aggressive.
- **Won when saw flop (W\$WSF):** Percentage of hands where the player wins the pot when she sees the flop. This attribute also concerns the tight-loose dimension.
- **Aggression factor (AF):** An attribute trying to describe how aggressively a player plays. Since calls are considered passive plays, whereas bets and raises are aggressive, AF is defined as the number of all bets and raises divided by the number of all calls. As an example, a player having an AF value of 2.0, bets or raises twice as often as she calls.
- **Aggression percentage (AF%)**: Percentage of betting rounds where a player bets or raises.
- **Flop continuation bet (CBetF):** Percentage of hands where the player is the preflop aggressor (i.e. put in the last raise) and then bets the flop. When playing short-

handed, a majority of all flops are taken heads-up (i.e. by two players only) so often both players fail to improve on the flop. With this in mind, it is important for a preflop raiser to often try to take down the pot with a bet on the flop even when missing it. A high CBet value, consequently, is an indicator of aggressive play.

- **Turn continuation bet (CBetT):** Percentage of hands where the player is the preflop aggressor and then bets both on the flop and on the turn. This attribute is related to both the tight-loose and the passive-aggressive dimension.
- **Check-raise (CR):** Percentage of time a player check-raises after checking the flop, turn or river. A check-raise is a very aggressive move, often associated with good players.
- **Cold call preflop (CC):** Percentage of times a player calls a raise (more than one bet) as her first action preflop. A high cold-call rate is associated with poor players, since poker theory says that you should almost always either fold or reraise against a raise.
- **Steal:** Percentage of times a player opens with a raise from late position when no previous player has entered the pot. Good players will often open with a raise in order to try to steal the blinds, even with fairly weak hands. This attribute, consequently, is an indicator of aggressive play.
- **Reraise preflop raise in blind (BBRR):** When playing short-handed, the play from the blinds is very important. Specifically, the blinds have to act first in all betting rounds after the flop, so calling a raise in the blinds without a good hand is a common but costly mistake. Naturally, good players in late position try to exploit this by steal-raising in order to win the pot immediately. Consequently, the big blind must defend his blind by sometimes reraising. This attribute, which is defined as the percentage of times a player reraises a preflop raise from the big blind, therefore relates mainly to the passive-aggressive dimension.
- **3bet:** Percentage of times a player reraises a raise. Reraising is an important tool to try to get the pot heads-up, typically by forcing players with drawing hands to fold. In addition, 3-bets is a way of building a large pot for a player who thinks she is ahead at the moment. A 3bet is, of course, an aggressive play.
- **4bet:** Percentage of times a player reraises a reraise.
- **PFR_B, PFR_UTG, VPIP_B, VPIP_UTG:** Position is extremely important in Hold'em, so a player must use her positional advantage to play more hands, more aggressively from late positions. These attributes represent how often a player enters and raises a pot when acting first (UTG) and last (Button).
- **Raise_pos, Play_pos:** Two aggregated attributes trying to capture the players' positional awareness. Raise_pos, which is defined as (PFR_B-PFR_UTG)/PFR_UTG, measures how much more often a player raises from the button, compared to when acting first. Similarly, Play_pos, defined as (VPIP_B-VPIP_UTG)/VPIP_UTG, indicates how many more pots the player contests from the button, compared to when acting first. Naturally, the hypothesis is that a solid player should be more aware of how important the position is, and consequently have higher values for these two attributes.
- **Big bets won per 100 hands (BB/100):** This is the profit achieved by the player, measured as number of big bets won or lost, on average, over 100 hands. It must be noted that internet casinos take a fee (the *rake*) from all pots, so the winner does not get all the money put into the pot. In this study, player profits are actual profits; i.e., a player only wins the raked pot.

Table I below shows a summary of the $0.5-$1.0 dataset. The last column is the linear correlation between the specific attribute and the BB/100 attribute.

TABLE I
Summary of $0.5-$1.0 dataset (3262 players)

Attribute	Mean	Std. Dev.	Min	Max	Corr
VPIP	34.68	14.81	10.72	90.50	-0.60
PFR	13.47	6.40	0.00	49.15	0.31
PFR/VPP	47.20	26.63	0.00	91.38	0.50
3Bet PF	6.51	3.88	0.00	30.98	0.18
W$WSF	39.75	3.60	26.59	54.40	0.47
CCPF	16.05	16.55	0.00	88.89	-0.61
AF	1.69	0.74	0.11	5.02	0.43
AF%	47.49	10.04	7.47	76.17	0.44
CR	8.55	4.71	0.00	35.69	0.30
CBetF	11.85	6.00	0.00	35.08	0.35
CBetT	9.43	5.01	0.00	31.88	0.30
Steal	24.36	12.52	0.00	62.68	0.43
BBRR	9.57	6.64	0.00	50.00	0.15
3Bet	8.45	4.17	0.00	41.28	0.14
4Bet	12.21	5.56	0.00	44.10	-0.05
Raise_Pos	38.23	86.78	-100.00	2698.08	0.13
PFR_B	16.41	8.41	0.00	53.08	0.37
PFR_UTG	12.69	6.87	0.00	68.28	0.20
Play_Pos	18.97	21.80	-77.38	75.09	0.36
VPIP_B	34.27	14.58	10.31	95.37	-0.59
VPIP_UTG	29.17	17.47	4.49	93.42	-0.61
BB/100	-3.02	6.25	-56.81	17.3	

Table I presents several interesting observations. First of all, it can be noted that all players in the dataset lose, on average, 3.02 BB per 100 hands. This is of course due to the rake and, if nothing else, should be reassuring for the Casinos. The overall picture is that a few attributes have a clear negative correlation with BB/100, while most other attributes are positively correlated with BB/100. Unfortunately, no specific attribute shows a very strong correlation with BB/100. On the other hand, only five attributes (3BetPF, BBRR, 3Bet, 4Bet and Raise_Pos) obtain correlations with BB/100 lower than 0.2.

Turning to individual attributes, it is very obvious that playing to loosely is the easiest way to lose money. VPIP, CCPF, VPIP_B and VPIP_UTG are all clearly negatively correlated with BB/100. Aggressive play, on the other hand, seems to be the key to successful play. AF, AF%, and Steal are all clearly positively correlated with BB/100. Looking at attributes focusing on positional awareness, the fairly high positive correlation between Play_pos and BB/100, indicates that good players, at the very least should play significantly fewer hands from poor position. W$WSF is a somewhat delicate attribute. It is no surprise that it is a good thing to win a high percentage of pots contested. Naturally, this attribute is more affected than the others by luck in individual hands. This is especially true for players having played relatively few hands. Table II below summarizes the $5-$10 dataset.

TABLE II
Summary of $5-$10 dataset (2555 players)

Attribute	Mean	Std. Dev.	Min	Max	Corr
VPIP	35.78	12.51	10.57	85.63	-0.55
PFR	16.69	6.36	0.06	50.43	0.18
PFR/VPP	51.82	22.33	0.15	96.60	0.45
3Bet PF	8.12	4.28	0.00	39.69	0.17
W$WSF	41.90	3.44	28.38	54.55	0.48
CCPF	14.97	14.66	0.00	80.12	-0.57
AF	1.59	0.60	0.20	4.26	0.39
AF%	48.47	8.95	14.42	68.60	0.38
CR	12.38	5.68	0.00	33.74	0.36
CBetF	14.87	6.05	0.00	41.68	0.24
CBetT	11.48	5.05	0.00	40.74	0.16
Steal	30.89	11.88	0.00	80.80	0.30
BBRR	10.24	6.30	0.00	66.67	0.10
3Bet	9.72	4.45	0.35	35.92	0.15
4Bet	10.40	4.95	0.00	41.67	-0.02
Raise_Pos	41.07	131.60	-100.00	5311.11	0.10
PFR_B	20.75	8.35	0.00	58.98	0.27
PFR_UTG	16.58	8.19	0.00	73.60	0.02
Play_Pos	17.81	21.87	-59.57	72.77	0.34
VPIP_B	34.62	11.84	10.51	87.50	-0.53
VPIP_UTG	29.42	14.99	5.56	91.07	-0.56
BB/100	-2.92	5.70	-42.30	14.81	

Although the overall impressions are quite similar for this dataset, it is interesting to see that the mean values for attributes capturing aggressive play (e.g. PFR, PFR/VPIP, AF, AF%, CR, CBetF, CbetT and Steal) are all higher on this level. Despite this, most of these attributes are even now evidently positively correlated with BB/100; i.e., it is still beneficial to be more aggressive than the opponents.

In addition, it is obvious that, even on this level, a lot of players contest way too many pots, making all VPIP attributes clearly negatively correlated with BB/100. Finally, on this level, the loose-passive action CCPF is actually the most negatively correlated attribute.

B. Predictive modeling and concept description

When performing predictive classification, *accuracy*, i.e., the percentage correct predictions on novel data, is normally the prioritized criterion. Alternative metrics, especially for unbalanced datasets include *area under the ROC-curve (AUC)* and different information theoretic measures; e.g., the *F-measure*, which is the harmonic mean of precision and recall. While accuracy and the F-Measure are based only on the final classification, AUC measures the ability to rank instances according to how likely they are to belong to a certain class; see e.g. [8]. AUC can be interpreted as the probability of ranking a true positive instance ahead of a false positive; see [9].

Most high-accuracy techniques for predictive classification produce opaque models like artificial neural networks (ANNs), ensembles or support vector machines. Opaque predictive models make it impossible to follow and understand the logic behind a prediction, which often must be considered a serious drawback. In practice, the usability of systems utilizing black-box prediction machines is often reduced, simply because decision makers have a hard time accepting recommendations without accompanying explanation of the underlying reasoning. Furthermore, opaque predictive models also make it impossible to inspect the model, looking for interesting patterns.

When models need to be interpretable (or ultimately comprehensible) accuracy is often sacrificed by using simpler but transparent models; most typically decision trees. This tradeoff between predictive performance and interpretability is normally called the *accuracy vs. comprehensibility tradeoff*. It must be noted, however, that decision trees often are so complex that their comprehensibility is limited. While it is certainly possible to trace the reasoning behind a specific prediction even in a very complex tree, understanding the overall relationship or discovering hidden patterns becomes quite prohibiting. With this in mind, it makes sense to distinguish between transparent and comprehensible models, and to measure comprehensibility (for tree models and rule sets) based on *model size*. Some typical choices are (for trees) *number of nodes* and (for rule sets) *number of rules* or *number of tests*.

The data mining task *concept description*, as defined by the CRISP-DM consortium [10] aims to generate understandable descriptions of concepts or classes. Consequently, the purpose is not to generate predictive models, but to gain insights. The results from a concept description project would most often be new information, typically presented as verbal descriptions or rules. Models produced by classification algorithms can, however, be regarded as concept descriptions, as long as they are comprehensible. In terms of standard performance measures, this corresponds to a transparent and relatively small model with good generalization ability, measured as high performance on unseen data, i.e., a comprehensible and accurate model. For the poker domain, a concept description would consist of a small model accurately capturing the differences between winning and losing strategies, or describing typical winners or losers.

The overall purpose of the predictive modeling was to find what strategies winning players in general apply, and similarly, to investigate the strategies of players losing money. With this in mind, we decided to use two pairs of binary classification experiments, at each level. In the first experiment, WIN, the task was to predict and explain what makes a player successful. The classes for this experiment thus became successful (*Winner*) and not successful (*No_Winner*). The second experiment (LOSE) was aimed at explaining what makes a player lose money. The two classes were *Loser* and *No_Loser*.

In both experiments, approximately one fourth of all instances were assigned to the targeted class; i.e., *Winner* for experiment 1 and *Loser* for experiment 2. As expected, the 25% players having the highest BB/100 were put in the

Winner class, while the 25% players having the worst BB/100 were put in the *Loser* class. Naturally, this setup makes the classes unbalanced, which may be a problem since many techniques tend to obtain pretty high accuracy, just by focusing on the majority class. With this in mind, we considered using cost-sensitive classifiers, but settled for using *oversampling*. More specifically, we applied the *SMOTE* [11] oversampling technique to produce artificial instances (based on five nearest neighbors), making the datasets balanced. Since there are two experiments and two levels, each with two differently sized datasets, we have altogether eight experiments; see Table III below.

TABLE III
EXPERIMENTS. (#INSTANCES INCLUDES ARTIFICIAL)

Experiment	Min #hands	Level	Win/Lose	#instances
1000:w	1000	\$0.5-\$1.0	Win	4888
1000:l	1000	\$0.5-\$1.0	Lose	4894
10000:w	10000	\$0.5-\$1.0	Win	403
10000:l	10000	\$0.5-\$1.0	Lose	401
1000:W	1000	\$5-\$10	Win	3803
1000:L	1000	\$5-\$10	Lose	3821
10000:W	10000	\$5-\$10	Win	431
10000:L	10000	\$5-\$10	Lose	429

All experimentation in this study was performed using the Weka data mining tool [12]. As mentioned above, we wanted to employ several different data mining techniques, but with different goals. First of all, we decided to use one technique producing opaque models. The intended use of such a black-box prediction machine is, of course, to categorize new instances (player strategies) into winners or losers. With this in mind, high predictive performance becomes the only important criterion for this technique, so we decided to use ANN ensembles. More specifically, we used the Weka meta technique *Bagging*, combining ten multilayer perceptron networks into an ensemble.

For the predictive modeling producing interpretable models, we evaluated altogether four techniques; *J48*, *Cart*, *JRip* and *Chipper*. J48 is Weka's implementation of the tree inducer C4.5 [13]. The Cart implementation used is also from Weka, where it is called *SimpleCart*. This implementation follows the original Cart algorithm [14] closely. Both Cart and J48, produce decision trees. JRip, which is the Weka implementation of the RIPPER algorithm [15], on the other hand produces *ordered rule sets* or *decision lists*. For the standard algorithms, default settings were used in all experimentation.

Chipper is an algorithm built for concept description, introduced in [16] and since then implemented in Weka. Chipper produces decision lists by greedily formulating rules with high coverage, whilst maintaining acceptable accuracy. To allow the user to control the tradeoff between accuracy and comprehensibility, two parameters, called *ignore* and *stop*, are available. The *ignore* parameter sets the acceptable misclassification rate for each rule produced, given either as an absolute number of instances, or as a percentage of the remaining instances. The *stop* parameter controls how long the rule generation procedure should continue, by simple specifying the proportion of instances that should be classified before formulating the default rule.

In this study, Chipper was used together with the Weka meta-procedure *CVParameterSelection*, which uses internal cross-validation to optimize settings for one or more parameters. Two different Chipper setups were evaluated, one favoring accuracy (Chipper - A), where *ignore* ranges between 0.5% and 2% and *stop* is between 96% and 99%, and another setting favoring concept description (Chipper - C), with *ignore* at 4% or 5% and *stop* between 75% and 95%.

For all experimentation, standard, 10-fold stratified, cross-validation was used. When measuring size, *total number of nodes* was used for J48 and Cart, while *total number of tests* was used for Jrip and Chipper.

IV. RESULTS

Tables IV and V below show the results from the predictive modeling.

TABLE IV
RESULTS FOR ANN ENSEMBLE AND TREE MODELS

\$0.5-\$1	**ANN**			**J48**				**Cart**			
	Acc	**Auc**	**F**	**Acc**	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**
1000:w	.758	.840	.750	.720	.749	.718	405	.719	.753	.718	85
1000:l	.858	.930	.860	.825	.836	.828	439	.825	.868	.824	75
10000:w	.811	.890	.800	.705	.720	.703	55	.710	.705	.710	51
10000:l	.815	.871	.815	.713	.751	.713	41	.678	.708	.677	55
\$5-\$10	**ANN**			**J48**				**Cart**			
	Acc	**Auc**	**F**	**Acc**	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**
1000:W	.760	.840	.760	.705	.734	.704	419	.717	.739	.717	195
1000:L	.847	.920	.850	.809	.822	.809	291	.807	.853	.807	139
10000:W	.764	.830	.750	.708	.741	.700	57	.701	.712	.700	45
10000:L	.800	.890	.800	.718	.741	.717	41	.734	.744	.734	13
Mean:	**.802**	**.876**	**.798**	**.738**	**.762**	**.737**	**218**	**.736**	**.760**	**.736**	**82**

The results for the ANN ensemble clearly show that it is possible to obtain very accurate predictive models for the situation at hand. Over all eight datasets, the ANN ensemble obtained a mean accuracy just over 80%, and an AUC close to 0.9. The use of decision trees, as expected, resulted in significantly worse predictive performance. In addition, the induced trees are actually quite large, making them very hard to manually inspect or analyze. Comparing the different datasets, it is obvious that the easiest task is to find losers on the \$0.5-\$1.0 level (1000:l dataset). This was of course no surprise either since we would expect really poor players to play mainly low-limit.

TABLE V
RESULTS FOR RULE SET MODELS

\$0.5-\$1	**JRip**				**Chipper - A**				**Chipper - C**			
	Acc	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**
1000:w	.708	.728	.708	33	.705	.750	.690	47	.703	.740	.702	13
1000:l	.816	.842	.815	39	.808	.860	.810	36	.801	.855	.801	11
10000:w	.717	.737	.717	8	.767	.777	.766	33	.680	.726	.680	10
10000:l	.661	.696	.661	11	.711	.715	.710	31	.671	.685	.670	11
\$5-\$10	**Jrip**				**Chipper - A**				**Chipper - C**			
	Acc	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**	**Acc**	**Auc**	**F**	**Size**
1000:W	.693	.703	.693	67	.688	.732	.687	63	.686	.725	.685	16
1000:L	.806	.826	.806	18	.791	.842	.791	32	.795	.849	.795	13
10000:W	.664	.701	.664	8	.694	.719	.693	53	.640	.668	.639	13
10000:L	.709	.737	.708	6	.709	.723	.709	51	.713	.757	.713	11
Mean:	**.722**	**.746**	**.722**	**24**	**.734**	**.765**	**.720**	**50**	**.711**	**.751**	**.711**	**12**

When comparing the techniques producing rule sets, Chipper – A obtains the best predictive performance overall. As a matter of fact, both accuracy and AUC are comparable to J48 and Cart. Unfortunately, Chipper – A's decision lists tend to be quite lengthy, making their comprehensibility questionable. Both JRip and Chipper – C, on the other hand, generate mainly smaller models, even if there are a couple of exceptions for JRip. Naturally, the increased comprehensibility comes at a price of reduced accuracy, but in this study, the difference in predictive performance is actually quite small. Figures 1-3 below show some sample models from the different datasets.

```
IF CCPF       >= 28.7    THEN Loser      [1740/220]
IF W$WSF      >= 41.1    THEN No_Loser   [1201/141]
IF VPIP_UTG   <= 16.18   THEN No_Loser   [495/87]
IF CCPF       >= 24.62   THEN Loser      [254/65]
IF W$WSF      >= 40.06   THEN No_Loser   [205/54]
IF W$WSF      >= 39.11   THEN No_Loser   [173/44]
IF Steal      <= 6.62    THEN Loser      [120/37]
IF CBETT      >= 12.78   THEN Loser      [100/31]
IF W$WSF      >= 38.42   THEN No_Loser   [94/27]
IF 3Bet       <= 4.3     THEN No_Loser   [78/23]
DEFAULT:      No_Loser[434/211]
```

Figure 1: Chipper – C rule set for 1000:l

In this rule set from $0.5-$1.0, the first rule singles out a large number of *Losers* based on their tendency to cold-call raises preflop. The fact that, on this level, cold-call turned out to be even more differentiating than, for instance, VPIP and AF, was somewhat unexpected, and could be a valuable finding for the poker domain. The second rule classifies quite a few players as *No_Losers* based on a high W$WSF. Although this attribute is, as mentioned above, somewhat affected by luck in individual hands, it also shows the importance of only seeing the flop with hands that have a good chance of winning.

```
CCPF < 10.9
|  W$WSF < 44.6
|  |  PFR < 19.2
|  |  |  VPIP_B < 22.4: Loser(16/8)
|  |  |  VPIP_B >= 22.4: No_Loser(59/24)
|  |  PFR >= 19.1
|  |  |  AF < 2.3
|  |  |  |  PFR/VPP < 69.8: No_Loser(8/3)
|  |  |  |  PFR/VPP >= 69.8: Loser(46/13)
|  |  |  AF >= 2.3: No_Loser(9/1)
|  W$WSF >= 44.6: No_Loser(110/30)
CCPF >= 10.9: Loser(93/9)
```

Figure 2: Cart tree for 10000:L

This Cart tree is from the $5-$10 level and contains only players having played more than 10000 hands. Still, cold-call preflop is again used directly to accurately categorize a relatively large number of *Losers*. Here, however, cold-calling "only" 11% is found to be too much. W$WSF is again very important for finding *No_Losers*, even when the number of hands played is as large as 10000. The rest of the tree tends to favor aggressive play, but the number of instances classified in each leaf is rather small and the variables used in the splits clearly interact, making the interpretation somewhat awkward.

```
(Steal >= 34.9)=> Winner(121/33)
(W$WSF >= 42.1) and (4Bet >= 10.4) and
(AF <= 2.5) and (PFR/VPP <= 78.7)=> Winner(49/7)
(PFR/VPP <= 72.9) and (VPIP_UTG <= 12.9) and
(VPIP >= 19.8)=> Winner (31/3)
 => No_Winner (202/43)
```

Figure 3: JRip rule set for 10000:w

This JRip rule set from $0.5-$1.0 starts by picking a large number of *Winners* based on a high Steal value. Again, this must be considered a surprising first rule, making it an interesting observation. The second and third rules are also somewhat unexpected, mainly because they seem to warn against a too aggressive play. The last conjunct in the last rule, finally, indicate that it is possible to play too tightly.

V. Conclusions

We have in this paper combined different data mining techniques to analyze data from online poker. The results show that it is possible to create very accurate black-box prediction machines (here ANN ensembles), clearly able to categorize players into *Winners* and *Losers*, based on their profiles. Examining decision trees and decision lists produced to obtain concept descriptions, it is obvious that these models are quite comprehensible, and still fairly accurate. An analysis of some concept descriptions also identified several rather unexpected findings. Most notably, cold-calling too many raises preflop turned out to be the most significant indicator for identifying Losers.

References

[1] S. Russel and P. Norvig, *Artificial Intelligence - A Modern Approach*, 2nd edition, Prentice Hall, 2003.

[2] D. Sklansky and M. Malmuth, *Hold'em Poker for Advanced Players*, Two Plus Two Publishing, 1999.

[3] E. Miller, *Getting Started in Hold'Em*, Two Plus Two Publishing, 2005.

[4] E. Miller, D. Sklansky and M. Malmuth, *Small Stakes Hold'em – Winning big with expert play*, Two Plus Two Publishing, 2004.

[5] L. Jones, *Winning Low-Limit Hold'em*, ConJelCo, 2000.

[6] N. Grudzien and G. Herzog, *Winning in Tough Hold'em Games*, Two Plus Two Publishing, 2007.

[7] http://forumserver.twoplustwo.com/

[8] T. Fawcett, Using rule sets to maximize roc performance, *15th International Conference on Machine Learning*, pp. 445-453, 2001.

[9] A. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, *Pattern Recognition*, 30(6):1145-1159, 1997.

[10] The CRISP-DM Consortium, CRISP-DM 1.0, www.crisp-dm.org, 2000.

[11] N. V. Chawla, K. W. Bowyer and W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, Vol. 16:321-357, 2002.

[12] I. H. Witten and E. Frank, Data Mining: *Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.

[13] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[14] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

[15] W. Cohen, Fast Effective Rule Induction, *Proceeding of 12th International Conference on Machine Learning*, pp. 115-123, 1995.

[16] U. Johansson, C. Sönströd, T. Löfström, and H. Boström, Chipper – A Novel Algorithm for Concept Description, *Scandinavian Conference on Artificial Intelligence*, pp. 133-140, 2008.