University of Minnesota, Twin Cities

PokerBots – Smart Poker Agents

Final Project Report
Date: 12/13/05

Amit Bose (3323259)
Kalyan Beemanapalli (3330244)

CSci 5511 (Artificial Intelligence 1), Fall 2005 Course Project
University of Minnesota, Twin Cities

1. Abstract

Game playing is a favorite application area of AI techniques. Poker is an interesting and popular game of cards that requires skills very peculiar to intelligent creatures. Success in the game not only requires a good understanding of ground rules, but also informed guesswork, risk management and ability to "read the mind" of other players. PokerBots is a system of intelligent programs that attempts to model a human player by learning these qualities with the goal of winning the game. The agents are adversarial and are not necessarily equally "smart". PokerBots thus provides an ideal opportunity for research in machine intelligence and a test-bed for experimenting with AI techniques.

2. Problem definition

We intend to devise a set of programs that is able to play the game of poker in way that reflects intelligent behavior. The programs are expected to model the human way of learning to play the game: initially they are only aware of the rules that govern the game and the goal to achieve, but have no strategies. By playing more often, the agents must learn ways to improve their chances of winning. Much like human players, all agents are not equally smart and may not make identical decisions in identical situations. Agents have memory but with varying retentive ability. Each agent also has to put up a face that is visible to others, and can be used as an input for making decisions. Finally, the agents are competitive and not collaborative – they are expected to do everything legal that may be reduces the chances of others.

3. Poker

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

The order and amount of betting is strictly controlled on each betting round. We use 4 betting denominations: 1, 2, 5 and 10. When it is a player's turn to bet, one of five options is available: *fold* (withdraw from the hand, leaving all previously wagered money

in the *pot*), *call* (match the current outstanding bet; if there is no current bet, one is said to *check*), or *raise* the bet (put something more than the current bet; if there is no current bet, one is said to *bet*). There is usually a maximum of three raises allowed per betting round. The betting option rotates clockwise until each player that has not folded has put the same amount of money into the pot for the current round, or until there is only one player remaining. In the latter case, this player is the winner and is awarded the pot without having to reveal their cards. There is a strategic advantage to being the last bettor in any given round; so to maintain fairness, the order of betting is rotated clockwise after each hand.

4. Related Work

Mathematicians and economists have studied aspects of poker for reasons ranging from purely monetary to modeling the dynamics of the game. One of the ways in which the problem has been attempted to solve is by using a simplified variation of the game. The simplification adds more constrains than the full-fledged game and hence makes it possible to analyze the game better. For example, the number of players could be limited to two only, or one could restrict the betting rules. The other approach has been to mathematically analyze the full-blown game and combine it with simulations, experiments and expert knowledge of "ad-hoc experts". This approach has been usually taken by expert players of the game with an inclination towards mathematics.

Though simplification is a common and useful technique for solving difficult problems, it may remove the complex activities that we are interested in studying. For example, Findler tried to model human cognitive processes and build a program that could learn. But his simplified approach was not very useful in the real world. Theoretical work by Koller et al provides an algorithm for finding optimal randomized strategies in two-player imperfect information competitive games. Their system builds trees to find the optimal strategy. But owing to the size of decision trees built, only simplified versions of poker can be solved.

5. Our approach to the problem

Devising a poker agent is a difficult problem, and our approach to devising one has been to divide the task into independent sub-problems that can be tackled separately. Specifically, there are two major tasks that any poker player has to master: evaluating the strength of his cards, and making betting decisions that will maximize his/her chances of winning the pot (or minimizing the loss if the player cannot win the game). We tackle the first task without considering the dynamics of a multi-player game. This method is called the *Learning Model*. Making betting decisions is handled by a separate method, the *Playing Model*, and makes use of the output of the learning model, besides other playing information. This separation into learning and playing models enables us to manage a complex task better. The separation is also justified on the grounds that evaluation of hand strength is dependent only on the cards a player has – we are not determining relative hand strength in the learning model.

6. Learning model

The goal of the learning model is to assess the strength of a particular hand of cards. Strength is defined precisely later in the section. Intuitively, the strength of a hand determines the expected utility value of the hand - greater the utility value, better are the chances of winning the pot. The learning model is based on principles of Bayesian decision theory. There are ten winning hands in poker: Royal Flush, Straight Flush, Four of a Kind, Full House, Flush, Straight, Three of a Kind, Two Pair, One Pair and No Pair. Each winning hand is treated as a *class* of a classification problem. The inputs on which the classification is to be made is the hand of cards, consisting of two hole cards and 3 to 5 community cards, that a player has to his/her disposal. According to Bayesian theory, the classification can be made on the posterior probability of a class given a hand.

Let $\omega_1, \omega_2, \omega_3, \ldots, \omega_n$ denote the various classes of winning hands. Let $\boldsymbol{x}$ represent the set of cards in a hand; **x** is thus the input feature vector. By Bayes' rule, the posterior probabilities of winning hands is given by

$$p(\omega_i \mid x) = \frac{p(x \mid \omega_i)\ p(\omega_i)}{p(x)}$$

where ,

$$p(x) = \sum_{i=1}^{n} p(x \mid \omega_i)\ p(\omega_i)$$

Here, $p(\omega_i)$ is the prior probability of reaching a winning hand ω_i. Given particular hand we calculate the posterior probabilities of all winning hands which we use to calculate the strength of that hand. The prior probabilities are easily obtained by elementary combinatorics: the prior of each class is the total number of hands that are possible for that class divided by the total number of possible 5-card hands.

There is no straight forward way to determine the likelihood probabilities $p(\boldsymbol{x} \mid \omega_i)$. hence these need to be learnt. An important issue in determine the likelihood of hands is that the number of cards in the hand keeps increases from 5 in pre-flop to 7 before river. Hence we formulate the likelihood as,

$$p(x \mid \omega_i) = \prod_{c \in x} p(c \mid \omega_i)$$

where, c is a card in the hand **x**. Thus the task of obtaining hand likelihood is reduced to that of obtaining card likelihoods $p(c \mid \omega_i)$.

A simple procedure is used to determine the card likelihood: we play the game several thousand times with only one player. A counter is maintained for each card/class combination. In each game, seven cards are drawn at random, and we enumerate all possible winning hands that are possible using only 5 cards from the 7 drawn. If a card c is used in a making a winning hand (or class) ω_i then the counter for c/ω_i is incremented. Finally, for each class, the sum of counters of all cards, s_i is computed. This gives the total number of games where that class was achieved. Dividing the counter for each card for that class by s_i gives the desired card likelihood $p(c \mid \omega_i)$.

Next, we define the utility value of a winning hand (or a class) in terms of its rarity: rarer the class, the better are the chances of winning the pot with that hand, and hence the higher is its utility. Currently we use the reciprocal of prior probability of a class ω_i as its utility,

$$U(\omega_i) = \frac{1}{p(\omega_i)}$$

A better utility function would incorporate not only the winning hand, but also the rank of the cards in the winning hand – this is useful in resolving ties. Actually, we do so depending on the cards available in a hand. The subset of cards that form a goal has a rank within all hands that belong to that goal. This position is captured by adding a hand-specific constant to the above utility function. Finally, given a hand of cards $\boldsymbol{x}$, let ω_{max} be the highest order goal (winning hand) that can be formed using the hand. The strength of the hand $S(\boldsymbol{x})$ is then defined as the utility of the best goal plus the expected utility of better goals. That is,

$$S(x) = U(\omega_{\max}) + \sum_{\omega_i \in G} p(\omega_i \mid x)\, U(\omega_i)$$

where, G is the set of all goal states that are higher than ω_{max}. The strength value will be further used in the playing model.

7. Playing model

The playing model can be very complicated. The goals that the playing model is expected to achieve are:

- to guess what the unknown cards are,
- to interpret facial expressions of others,
- to guess when others are bluffing,
- to understand betting patterns and to remember the outcome of previous games,
- to make bets – whether to fold, call or raise and the amount to raise,
- to decide expressions to make for others to see etc

Owing to time constraints, we were able to address only a few of the goals described above.

In the least, the playing model must be able to make a betting decision that improves the chances of winning for an agent. We use a simple "inward"-looking model for betting. The betting decision is based on two factors:

- the estimated strength of the hand calculated earlier, and
- the agent personality

We use an agent's "personality" to model different kinds of players. Some players are optimistic and raise bets even when they don't have strong hands in the hope that their hand will improve or that their opponents don't have strong cards. Others may want to play it safe, and make bets only when they are very sure of their winning chances. We use four classes of agents based on personality: aggressive-loose, aggressive-tight, conservative-loose and conservative-tight. The optimism of these agents decreases in the given order so that the aggressive agents are more optimistic than the conservative agents.

We have manually decided thresholds on the values of hand strength that enable an agent to make a decision. These thresholds divide the number space spanned by hand strength into 5 regions. For each agent type and each strength region, a set of probabilities is defined. These are the probabilities of specific betting decisions. In this way, given a hand and an agent type, the betting decisions are made with the assigned

probability. The values assigned were done using intuition, taking into consideration the agent personality and strength zone.

Tables 1 to 4 give the probabilities of various decisions that can be made by an agent for different hand strength values.

Table 1: Decision probabilities of aggressive-loose agent

Strength *S*	Decision					
	Fold	**Check/Match**	**Raise $1**	**Raise $2**	**Raise $5**	**Raise $10**
0 < s < 200	0.05	0.4	0.3	0.15	0.05	0.05
200 <= s < 500	0.05	0.4	0.25	0.15	0.1	0.1
500 <= s < 2000	0	0.3	0.25	0.15	0.15	0.15
2000 <= s < 10000	0	0.2	0.25	0.2	0.2	0.15
s >= 10000	0	0.25	0.25	0.25	0.2	0.2

Table 2: Decision probabilities of aggressive-tight agent

Strength *S*	Decision					
	Fold	**Check/Match**	**Raise $1**	**Raise $2**	**Raise $5**	**Raise $10**
0 < s < 200	0.15	0.35	0.2	0.15	0.1	0.05
200 <= s < 500	0.15	0.3	0.25	0.15	0.1	0.05
500 <= s < 2000	0.1	0.25	0.2	0.2	0.15	0.1
2000 <= s < 10000	0.1	0.2	0.2	0.25	0.15	0.1
s >= 10000	0	0.25	0.25	0.25	0.15	0.1

Table 3: Decision probabilities of conservative-loose agent

Strength *S*	Decision					
	Fold	**Check/Match**	**Raise $1**	**Raise $2**	**Raise $5**	**Raise $10**
0 < s < 200	0.2	0.45	0.15	0.1	0.05	0.05
200 <= s < 500	0.1	0.55	0.15	0.1	0.05	0.05
500 <= s < 2000	0.1	0.55	0.15	0.15	0.05	0.05
2000 <= s < 10000	0.05	0.3	0.2	0.2	0.15	0.1
s >= 10000	0.05	0.45	0.2	0.25	0.15	0.1

Table 4: Decision probabilities of conservative-tight agent

Strength *S*	Decision					
	Fold	**Check/Match**	**Raise $1**	**Raise $2**	**Raise $5**	**Raise $10**
0 < s < 200	0.2	0.65	0.1	0.05	0	0
200 <= s < 500	0.15	0.65	0.1	0.1	0	0
500 <= s < 2000	0.1	0.7	0.1	0.1	0	0
2000 <= s < 10000	0.05	0.5	0.2	0.1	0.1	0.05
s >= 10000	0.05	0.45	0.2	0.15	0.1	0.05

8. Results

The prior probabilities of the winning hands are listed in Table 5.

Table 5: Prior probabilities of winning hands

Winning Hand	Probability
Royal Flush	1.50e-06
Straight Flush	1.38e-05
Four of Kind	0.00024
Full House	0.00144
Flush	0.00198
Straight	0.00355
Three of Kind	0.02353
Two-Pair	0.10372
One Pair	0.58823
No Pair	1

Table 6 lists the class likelihoods obtained by playing 100,000 single-player games.
Table 7 compares the performance of different agent personalities in playing against each other. Totally 200 games of poker were played with 4 players, one of each type.

Table 7: Performance of different agents

Agent personality	Number of wins	Average profit per win ($)
Conservative-Tight	59	10.45763
Aggressive-Loose	68	8.705882
Conservative-Loose	39	9.307692
Aggressive-Tight	34	6.617647

The comparison reveals some interesting things about the different player personalities:

- The aggressive-loose player (most optimistic) was the most successful, and won close to one-third of the games
- The conservative-tight player (most pessimistic) was not far behind in number of wins
- The other two types of players won comparatively much lesser games.
- The conservative players showed a better average profit per win than the loose player. This happened because the opponents of conservative players were loose, and therefore bet larger amounts. So if the conservative player won, he/she won it big. Exactly the reverse case applies to aggressive players, making their profits small
- Overall, in this game set-up it appears that being a conservative-tight player is the most beneficial. That way the player manages to win large number of games, and the same time win large sums due to opponents' mistakes.

Table 6: Likelihood probabilities of all cards for various winning hands

Rank	Royal Flush	Straight Flush	Four of Kind	Full House	Flush	Straight	Three of Kind	Two Pair	One Pair	High Card
Clubs										
2	0.00000	0.00000	0.01494	0.01651	0.04171	0.00185	0.01376	0.00757	0.01488	0.01215
3	0.00000	0.00000	0.02955	0.01680	0.05734	0.00567	0.01659	0.00936	0.01627	0.01194
4	0.00000	0.05351	0.03796	0.01503	0.04356	0.00873	0.01160	0.01053	0.01494	0.01245
5	0.00000	0.05351	0.01969	0.01735	0.04714	0.00741	0.01383	0.01019	0.01499	0.01224
6	0.00000	0.05351	0.01997	0.01393	0.04025	0.01400	0.01282	0.01059	0.01526	0.01286
7	0.00000	0.00000	0.03510	0.01175	0.03284	0.01022	0.01328	0.01013	0.01551	0.01281
8	0.00000	0.06933	0.01961	0.00893	0.03526	0.00930	0.01143	0.01038	0.01642	0.01236
9	0.00000	0.14649	0.01071	0.02052	0.03039	0.01309	0.01273	0.01189	0.01592	0.01275
10	0.00000	0.14649	0.02588	0.01088	0.03532	0.01148	0.01436	0.01004	0.01612	0.01238
J	0.00000	0.07716	0.03036	0.01577	0.02071	0.00739	0.01391	0.01021	0.01584	0.01226
Q	0.00000	0.00000	0.02287	0.01760	0.01725	0.00605	0.01074	0.01029	0.01657	0.01210
K	0.00000	0.00000	0.00856	0.01668	0.01605	0.00327	0.01099	0.00884	0.01642	0.01256
A	0.00000	0.00000	0.00902	0.00808	0.01525	0.00094	0.01385	0.00741	0.01668	0.01270
Diamonds										
2	0.00000	0.00000	0.01569	0.01311	0.00000	0.00250	0.01421	0.00697	0.01554	0.01286
3	0.00000	0.00000	0.01861	0.01408	0.00016	0.00249	0.01421	0.00771	0.01438	0.01237
4	0.00000	0.00000	0.02074	0.01551	0.00089	0.00644	0.01037	0.00969	0.01366	0.01315
5	0.00000	0.00000	0.01170	0.01870	0.00175	0.00722	0.01339	0.01045	0.01482	0.01240
6	0.00000	0.00000	0.01277	0.01286	0.00182	0.00963	0.01099	0.00975	0.01454	0.01318
7	0.00000	0.00000	0.01396	0.01191	0.00019	0.00898	0.01228	0.01062	0.01560	0.01250
8	0.00000	0.00000	0.03266	0.01182	0.00384	0.01325	0.01149	0.01141	0.01519	0.01228
9	0.00000	0.00000	0.02251	0.01414	0.00634	0.00907	0.01372	0.01083	0.01605	0.01243
10	0.00000	0.00000	0.01030	0.01480	0.00922	0.01100	0.01330	0.00874	0.01607	0.01305
J	0.00000	0.06933	0.01485	0.01301	0.00931	0.00679	0.01210	0.00892	0.01624	0.01274
Q	0.00000	0.06933	0.01420	0.02061	0.00825	0.00638	0.01187	0.00984	0.01573	0.01271
K	0.00000	0.00000	0.01544	0.01428	0.00814	0.00378	0.01015	0.00807	0.01603	0.01272
A	0.00000	0.00000	0.01967	0.01442	0.00871	0.00247	0.01386	0.00695	0.01638	0.01215
Hearts										
2	0.00000	0.00000	0.02362	0.02007	0.00000	0.00319	0.01400	0.00719	0.01461	0.01163
3	0.00000	0.00000	0.01647	0.01827	0.00000	0.00428	0.01220	0.00936	0.01459	0.01285
4	0.00000	0.00000	0.01427	0.01515	0.00000	0.00424	0.01153	0.00914	0.01424	0.01246
5	0.00000	0.00000	0.02715	0.01903	0.00076	0.01007	0.01590	0.01091	0.01725	0.01140
6	0.00000	0.00000	0.01605	0.00879	0.00135	0.01066	0.01033	0.01083	0.01434	0.01343
7	0.00000	0.05351	0.01639	0.01117	0.00186	0.01245	0.01192	0.01083	0.01652	0.01288
8	0.00000	0.05351	0.01720	0.01055	0.00261	0.00946	0.01150	0.01135	0.01651	0.01223
9	0.00000	0.00000	0.01241	0.01392	0.00444	0.01011	0.01436	0.00985	0.01656	0.01288
10	0.00000	0.00000	0.00934	0.01367	0.00568	0.01165	0.01077	0.01043	0.01575	0.01258
J	0.00000	0.00000	0.01801	0.01702	0.00802	0.00762	0.01327	0.01137	0.01553	0.01210
Q	0.00000	0.07716	0.01378	0.02242	0.00969	0.00661	0.01062	0.00969	0.01596	0.01264
K	0.00000	0.07716	0.00873	0.01451	0.01223	0.00428	0.00935	0.00729	0.01539	0.01305
A	0.00000	0.00000	0.01757	0.01775	0.00845	0.00187	0.01419	0.00705	0.01615	0.01232
Spades										
2	0.00000	0.00000	0.03369	0.01567	0.00000	0.00164	0.01233	0.00668	0.01368	0.01237
3	0.00000	0.00000	0.02090	0.01297	0.00000	0.00514	0.01149	0.00839	0.01403	0.01268
4	0.00000	0.00000	0.02182	0.01594	0.00000	0.00767	0.01156	0.00936	0.01469	0.01244
5	0.00000	0.00000	0.02130	0.01348	0.00000	0.00855	0.01302	0.01047	0.01528	0.01251
6	0.00000	0.00000	0.01260	0.01267	0.00254	0.01050	0.01339	0.00900	0.01332	0.01247
7	0.00000	0.00000	0.02488	0.01932	0.00139	0.01262	0.01266	0.01032	0.01420	0.01206
8	0.00000	0.00000	0.02380	0.01356	0.00348	0.01251	0.01265	0.01018	0.01475	0.01298
9	0.00000	0.00000	0.01470	0.01358	0.00182	0.01046	0.01018	0.01100	0.01538	0.01257
10	0.00000	0.00000	0.01259	0.01097	0.00686	0.00999	0.00882	0.01050	0.01566	0.01260
J	0.00000	0.00000	0.01198	0.01377	0.01169	0.01077	0.01069	0.01013	0.01669	0.01219
Q	0.00000	0.00000	0.01416	0.01536	0.00972	0.00487	0.01151	0.00936	0.01705	0.01174
K	0.00000	0.00000	0.01967	0.01515	0.01210	0.00392	0.01210	0.00866	0.01616	0.01267
A	0.00000	0.00000	0.01513	0.01329	0.00742	0.00225	0.01105	0.00658	0.01703	0.01207

9. Issues and Challenges

All the work of proposing a model to play poker in this project was done from scratch. Though people have been working on the game for years, there is no one strategy that always wins. Theoretical work of others is either too complex for a project at this level or too difficult to implement in practice. We therefore took a first-principles approach to the problem, guided by our desire to imitate human players.

Separation of the learning and playing models was a significant achievement of the project. However we found ourselves wanting with regard to the playing model. Many ideas and methods crossed our minds as we tried to tackle the problem; however given the time frame of the project, most were not fully implement-able. This was partly because we tried to consider as many factors as possible in the playing decision-making, instead of doing it in a factor-by-factor basis. Eventually we settled for a simplistic view of the game that helped us gain useful insights into the game. Finally, we didn't have enough time to make our agents play with human players – doing that would enable us to draw better conclusions about our approach.

10. Future Work

In the future, we would like to address other challenging issue in modeling the agent such as modeling guess-work, deception, playing patterns, strategies etc. One of the drawbacks of the present model is that there is no feedback learning. Intuitively, this is similar to learning from experience in human terminology. A better agent would be one, which learns better strategies as it plays the game. To do this we are thinking of modeling the decision process as Hidden Markov Model (HMM) where the state transition probabilities are learnt from experience. The decisions will make use of these probabilities and other factors of interest such as, opponent's playing style, and opponent's estimated hand strength, net profit/loss, deception etc in making a decision. We believe that a robust model would be one which takes into consideration all these factors in making decision. In a way, this approach of making decision makes sense because a human being also learns from experience and always tries to base his/her decisions by considering all the factors mentioned above. In general, the final model would be one which captures and utilizes the human cognitive process in entirety to make decisions.

11. Conclusions

In this project we tried to address one of the interesting and challenging tasks of designing an intelligent agent capable of playing poker. From our experience, we understood that designing a game such as poker which is characterized by uncertainty is a difficult task and there are many issues which haven't been addressed till date. Apart from modeling uncertainty in the world, the agent needs to be competitive, perform risk management and also consider the opponent's style of play. Due to the inherent disadvantages in the approaches taken by mathematicians, which are based on decision trees, for instance, in our approach we tried to take a probabilistic approach to model our agent. The task of modeling the agent has been divided into two independent parts: Learning Model and Playing Model.

The Learning Model is designed using the Bayesian decision theory. Here we calculate the strength of the cards possessed by the agent. The strength is calculated from likelihood probability of the hand and the prior probabilities of the goal state under consideration. The likelihood probability of the hand is a product of the likelihood probabilities of its individual cards. Hence, in essence by random trials we learn the likelihood probability of individual cards contributing to each of the goal states. The utility of the hand is defined as the rarity of the likelihood. Using this, hand strength is calculated which was used in the playing model. The playing model is a simple one which considers the expected hand strength and player style to make one of four decisions: Fold, Check, Match or Raise. Initially, the player styles have been randomly chosen by assigning a threshold value on strength to make one of the four decisions. We have thought of testing the model with four types of player styles which are name as Aggressive-Tight, Aggressive- Loose, Conservative-Tight and Conservative-Loose. The difference between these player styles is the way in which the cut-off threshold value is assigned for making decisions.

In summary, we have learnt that designing a game like poker which is characterized by issues from uncertainty to modeling human cognitive process is a challenging and interesting problem and a perfect solution doesn't seem to exist. Existing AI techniques cannot be readily applied as it is neither a search problem nor a logic problem. Hence, there is need for improving and devising new techniques which can address this problem. Though our model is a simple one, we wish that it will pave way for more sophisticated and robust poker agents in the future.

12. Distribution of tasks

Only two people worked on this project, and therefore an effort was made to share the responsibilities equally. It is very difficult to categorize the work in terms of design of the model. Since little work has been done in this area, we had to start from scratch and hence most of the ideas incorporated in the design of the model have been a combined effort. The specific tasks that we were involved in are as follows:

- Amit's tasks
  - Problem definition and choosing the scope of activities
  - Developing Bayesian decision theory based model for learning
  - Developing threshold based model for playing
  - Implementation of playing model
  - Implementation of overall framework for game playing *game-controller*
  - Testing
  - Report writing
- Kalyan's tasks
  - Scoping out activities
  - Proposing separation of learning and playing models
  - Working out probability values needed for Bayesian learning
  - Implementation of Bayesian learning model
  - Developing model for playing
  - Testing
  - Report writing

13. Timeline

Period	Activity
09/06 – 10/14	Problem selection Feasibility analysis Scoping out the problem Submission of initial report
10/17 – 10/21	Analysis of the problem, evaluation of play possibilities
10/24 – 11/11	Design of data structures Devise and implement learning model
11/14 – 11/25	Devise and implement playing model
12/28 – 12/16	Testing and validation User Interface development Final report submission

14. References

[1] Billings D., Papp D., Schaeffer J., Szafron D. *Poker as a Testbed for Machine Intelligence Research*
[2] Kimberg D. *PokerBot World Series 2004*, Card Player Magazine Volume 17, No. 11
[3] Bernstein O., Margulies J., Tsai C. *Intelligent Poker Player*, Technical report
[4] Gruman M., Edwards M., Nichols R. *No Limit Texas Hold'em Poker AI Agent*, Technical report
[5] Ankeny N., *Poker Strategy: Winning with Game Theory*, Basic Books, Inc., 1981.
[6] Carmel D. and Markovitch S., *Incorporating Opponent Models into Adversary Search*, AAAI, 1995, 120-125.
[7] Findler N., *Studies in Machine Cognition Using the Game of Poker*, Communications of the ACM 20, 4 (1977), 230-245.
[8] Koller D. and Pfeffer A., *Representations and Solutions for Game-Theoretic Problems*, Artificial Intelligence 94, 1-2(1997), 167-215.
[9] Sklansky D. and Malmuth M., *Hold'em Poker for Advanced Players*, Two Plus Two Publishing, 1994.