Risk Aversion in Texas Hold'em Poker

Karan Tibrewal

Jon Park, Williams College, F'16

17th December, 2016

ABSTRACT

This paper studies the effect of risk aversion on optimal policies in Heads Up Texas Hold'em Poker. We begin by developing a "risk-neutral" poker agent, and extend our insights to account for risk aversion in players. Due to the size of this game, complete analysis of the game tree is not computationally feasible; instead, we examine a *jam/fold* strategy. The game tree for this strategy is of manageable size, and enables us to use minimax search to obtain the optimal policy. Our analysis shows that a player's level of risk-aversion is a significant determinant of success in the heads-up tournament setting.

PROBLEM FORMULATION

Poker Rules and Terminology

We begin with a brief review of rules and terms for Heads Up Texas Hold'em Poker. The game comprises two players, each begin with a given chip stack of S_1 and S_2 respectively. They alternate paying small (small blind, or *SB*) or big (big blind, or *BB*) antes, followed by discretionary bets into the *POT*. Players can place discretionary bets at 4 instances for each hand: after the antes are collected (*PRE-FLOP*); or, after the first three cards on the board are opened (FLOP); or, after the fourth card on the board is opened (*TURN*); or, after the fifth card on the board is opened (*RIVER*).
For each round of betting, the current highest discretionary bet by any player is called the *TABLE BET*. For each discretionary bet placed by a player, the other player can:

(1) Surrender the hand (*FOLD*), in which case, the former player gets all the chips currently in the *POT*.

(2) Match the bet (*CALL*), in which case, the next card is opened;

(3) Increase the bet (*RAISE*), in which case, the former player has all the above

options.

(4) Go *ALL-IN*, in which case, the player puts in all of her chips into the *POT*. If the *ALL-IN* bet is lesser than the opponent's discretionary bet, or if the opponent *CALLs*, there is a *SHOW-DOWN*.

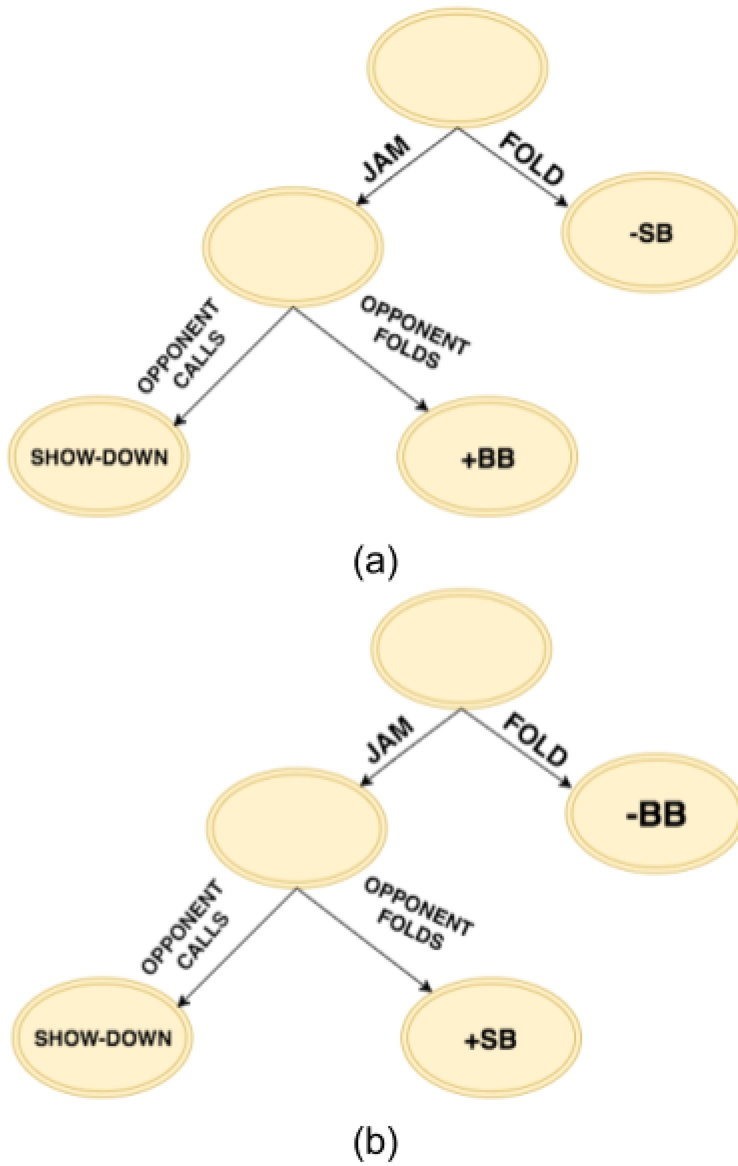
If no player *FOLDS* after the *RIVER*, or if there is an *ALL-IN* scenario, then there is a *SHOW-DOWN* in which the player with the higher ranking[1] hand takes all the chips currently in the *POT*. The game ends when one of the players is bankrupt. Notice, the total number of chips on the board remain fixed for the entire duration of the game at $S_1 + S_2 = S_{total}$, and therefore, the winner of the game walks away with all S_{total} of the chips.

A Jam/Fold Strategy

The full game tree for Heads Up Texas Hold'em Poker comprises $O(10^{18})$ nodes, which is far too large to analyze completely [2]. Instead, we consider a *jam/fold* strategy wherein, at each round of bets, our agent has two options: go all in (i.e., *"JAM"*), or surrender (i.e., *"FOLD"*). Observe that following this strategy, our agent is either all in, or has surrendered before the *FLOP*. In other words, a *JAM/FOLD* strategy renders analysis of the *FLOP, TURN,* and *RIVER* irrelevant; we only need to analyze the *PRE-FLOP*.

Let us begin by considering possible actions in the *preflop* scenario. The agent could either be posting *SB* or *BB*, and under each scenario, the agent can either *JAM* or *FOLD*. Then, given the hole cards of the agent and her adversary, we can construct deterministic game trees for each scenario. For example, consider the scenario when our agent posts big blind. Given her hole cards, our agent might choose to *FOLD* (in which case we lose the big blind), or *JAM*. If the opponent *CALLS* our *JAM*, then there is a *SHOW-DOWN*. Else, the opponent *FOLDS,* and we win the big blind. Figure 1 (b) shows a diagrammatic representation of such a game tree with relevant actions and rewards. The corresponding tree for when the agent posts small blind is sketched in Figure 1 (a).

[1] Please refer to Appendix (Related Works) for a complete list of hand rankings.
[2] Miltersen, Sorenson. "A Near Optimal Strategy for a Heads-Up Texas Hold'em Poker Tournament".



<u>Figure</u> *<u>1:</u>* (a) Game tree when agent posts small blind; (b) Game tree when agent posts big blind.

For a player's given hole cards, she can undertake two actions: *JAM* or *FOLD*. The rational agent should pick the action that maximizes her utility preference. Then, in order to find the optimal *JAM/FOLD* strategy, we consider agents who have "Mean-Variance" utility preferences.

Mean-Variance Utility Preferences

Consider the following proposition: Bob is an avid poker player who frequently visits the local casino for Heads Up Texas Hold'em tournaments. The tournaments each pay $100, and Bob wins 40% of the time, giving us an expected return of $40 on the tournament. Now, the new governor of Bob's town, Donald, being heavily invested in the casino business, reduces taxes on the casinos. The surge of business is profitable for the local gaming houses, and now, they can afford to pay $1000 on each tournament. In the new state of events, Bob's expected payoff for the tournament is $400 (10x his expected payoff from before!). Yet, is Bob a better poker player now?

The key insight of the above story is that expected value of payoffs alone is not enough to rank poker hands. Although Bob has a higher expected return after favorable election results, he presumably has to pay a higher fee to enter the tournament now: he, therefore, has to take additional risk in order to secure the higher expected payoff. In the 1950s, Harry Markowitz extended this insight to give us a very accurate prediction of how investors facing risky assets behave in equilibrium[3]. Markowitz maintained that investors prefer higher returns and lower risk (measured as the volatility of the action). Such investors are said to have "Mean-Variance" preferences given by:

$$U(action) = E[action] - \lambda^{h} \sigma[action]$$

Where:

- $U(action)$ is the utility obtained from a given action.
- $E[action]$ is the expected payoff of the action.
- λ^{h} is the agent specific parameter which measures aversion to risk.
- $\sigma[action]$ is the standard deviation of the payoffs to the action. It measures the "riskiness" of the action.

[3] In fact, Markowitz's theory became so popular that it finds application in modern day finance as "Modern Portfolio Theory". Please refer to the "Related Works" section in the Appendix for more details on Markowitz's theory.

Jam/Fold Utilities

Let us begin by calculating $U(fold)$. Notice, folding is a completely deterministic action: the agent loses the antes for the round (-*SB* or -*BB* as the case maybe). This is to say that the action has no variance. The utility can therefore be given as

$$U(fold) = -antes$$

Where *antes* is *BB* or *SB* as the case maybe.

We now calculate $U(jam)$. Upon jamming, the agent's opponent might respond by folding (in which case the agent wins the opponent's antes for the round). If the opponent calls, then there is a showdown, in which case the game may end in a tie (0 pay off), win (agent wins all of her opponent's chips in the pot = + $min\ \{K_1, K_2\}$), or a loss (agent loses all of her opponent's chips in the pot = - $min\ \{K_1, K_2\}$).

Then, let x_{jam} be a random variable which is equal to the payoff of jamming. We have:

$$x_{jam} = \begin{cases} +ante & opponent\ folds \\ +min\{K_1, K_2\} & opponent\ calls,\ win\ showdown \\ -min\{K_1, K_2\} & opponent\ calls,\ loses\ showdown \\ 0 & opponent\ calls,\ tied\ showdown \end{cases}$$

Then, by Linearity of Expectation, we have:

$$E[x_{jam}] = +ante \times Pr(opponent\ folds) +$$

$$min\{K_1, K_2\}(Pr(oppenent\ calls,\ win\ showdown) - Pr(opponent\ calls,\ loses\ showdown))$$

Then, again using Linearity of Expectation, we have:

$$E[x_{jam}^2] = +ante^2 \times Pr(opponent\,folds) +$$

$$min\{K_1, K_2\}^2(Pr(oppenent\ calls,\ win\ showdown) - Pr(opponent\ calls,\ loses\ showdown))$$

It follows that

$$\sigma(jam) = \sqrt{E[x_{jam}{}^2] - (E[x_{jam}])^2}$$

ALGORITHMS AND OPTIMAL POLICIES

Learning the Transition Model

Then, using the aforementioned formulation, we are able to calculate the utility estimate for each given state. However, we took the probabilities of different resultant states as given in the above section. Here, we discuss how the agent learns these probabilities.

The problem of learning the probabilities of resultant states in stochastic environments is much discussed in the field of Artificial Intelligence[4]. In this paper, we attempt to learn the probabilities using a simple Adaptive Dynamic Programming paradigm. We run several simulations of the agent facing jam/fold decisions, and record each result as a sample to calculate relevant probabilities.

More specifically, for given hole cards of the agent, we record the events where:

- the opponent folds;
- the opponent calls, and loses the showdown.
- the opponent calls, and wins the showdown.
- the opponent calls, and the showdown is tied.

[4] Russel and Norwig outline various relevant techniques such as TD learning, Q Learning, and so on.

The above provide us with estimates of *Pr(opponent folds), Pr(opponent calls, win showdown), Pr(opponent calls, loses showdown), Pr(opponent calls, tied showdown).* We provide results of these calculations below[5].

Pr(opponent folds) is the same for all given hole cards of the agent (that is, the player's hole cards do not affect the opponent's decision to fold because the opponent has no information about the player's cards!). We assume that the opponent is a risk-neutral (i.e. $\lambda^h = 0$) player. We calculate *Pr(opponent folds) = 0.49* for such an opponent.

Pr(opponent calls, win showdown), Pr(opponent calls, loses showdown), Pr(opponent calls, tied showdown) depend on the agent's hole cards (since the hole cards determine the results of the showdown!). We calculate these probabilities for each possible pairing of hole cards below:

CARDS	WIN%	TIE%	LOSS%	CARDS	WIN%	TIE%	LOSS%
A, 2 o	52.20	6.44	41.36	3, 9 o	34.35	13.31	52.35
A, 3 o	52.93	6.44	40.63	3, 10 o	36.10	12.74	51.15
A, 4 o	53.77	6.45	39.78	3, J o	38.77	11.77	49.46
A, 5 o	54.69	6.45	38.86	3, Q o	41.38	11.14	47.48
A, 6 o	54.82	6.20	38.98	3, K o	46.81	8.59	44.60
A, 7 o	55.92	6.05	38.03	4, 5 o	34.40	12.82	52.78
A, 8 o	56.89	5.81	37.30	4, 6 o	34.06	12.83	53.11
A, 9 o	57.73	5.51	36.76	4, 7 o	34.18	12.83	53.00
A, 10 o	59.39	5.39	35.22	4, 8 o	34.40	13.27	52.33
A, J o	60.35	5.11	34.55	4, 9 o	34.98	13.17	51.85
A, Q o	61.30	4.58	34.12	4, 10 o	36.87	12.78	50.36
A, K o	62.58	4.12	33.30	4, J o	39.41	11.87	48.72
2, 3 o	28.85	13.04	58.11	4, Q o	42.25	11.11	46.65
2, 4 o	29.61	12.86	57.53	4, K o	47.22	8.92	43.85
2, 5 o	30.66	12.93	56.42	5, 6 o	36.06	12.83	51.12
2, 6 o	30.33	12.99	56.68	5, 7 o	36.14	12.67	51.20
2, 7 o	30.45	12.91	56.64	5, 8 o	36.35	13.03	50.63
2, 8 o	31.85	13.23	54.93	5, 9 o	36.99	13.09	49.93

[5] For details of implementation and calculation, please refer to appendix.

2, 9 o	33.46	13.15	53.39	5, 10 o	37.65	12.67	49.68
2, 10 o	35.19	12.73	52.08	5, J o	40.33	11.83	47.84
2, J o	37.71	11.73	50.56	5, Q o	43.17	11.18	45.66
2, Q o	40.56	11.07	48.37	5, K o	48.07	8.92	43.01
2, K o	45.25	8.91	45.84	6, 7 o	37.95	12.48	49.57
3, 4 o	31.69	13.07	55.24	6, 8 o	38.20	12.81	48.98
3, 5 o	32.63	12.94	54.44	6, 9 o	38.76	12.76	48.49
3, 6 o	32.19	13.11	54.69	6, 10 o	39.45	12.42	48.13
3, 7 o	32.38	13.06	54.56	6, J o	41.00	11.55	47.44
3, 8 o	32.60	13.21	54.18	6, Q o	44.10	10.92	44.98

CARDS	WIN%	TIE%	LOSS%	CARDS	WIN%	TIE%	LOSS%
6, K o	49.16	8.65	42.19	A, 8 s	60.50	2.87	36.63
7, 8 o	40.06	12.40	47.54	A, 9 s	61.50	2.54	35.96
7, 9 o	40.63	12.41	46.97	A, 10 s	63.48	2.22	34.30
7, 10 o	41.33	12.08	46.59	A, J s	64.39	1.99	33.62
7, J o	42.74	11.33	45.93	A, Q s	65.31	1.79	32.90
7, Q o	44.70	10.68	44.62	A, K s	66.21	1.65	32.14
7, K o	50.06	8.47	41.47	2, 3 s	33.09	5.78	61.13
8, 9 o	42.51	12.15	45.34	2, 4 s	33.91	5.82	60.27
8, 10 o	43.20	11.63	45.17	2, 5 s	34.92	5.83	59.25
8, J o	44.62	11.11	44.27	2, 6 s	34.83	5.66	56.68
8, Q o	46.56	10.39	43.05	2, 7 s	35.43	5.43	56.64
8, K o	51.02	8.10	40.89	2, 8 s	37.67	5.18	54.93
9, 10 o	45.07	11.45	43.48	2, 9 s	39.97	4.88	53.39
9, J o	46.44	10.86	42.70	2, 10 s	42.54	4.59	52.08
9, Q o	48.88	9.71	41.41	2, J s	45.20	4.35	50.56
9, K o	52.61	8.08	39.31	2, Q s	48.10	4.13	48.37
10, J o	48.35	10.31	41.34	2, K s	51.23	3.94	45.84
10, Q o	50.12	9.58	40.31	3, 4 s	35.72	5.82	55.24
10, K o	54.44	7.57	37.99	3, 5 s	36.75	5.86	54.44
J, Q o	51.22	9.34	39.44	3, 6 s	36.68	5.69	54.69
J, K o	54.76	7.48	37.76	3, 7 s	37.30	5.46	54.56

Q, K o	55.84	7.20	36.95	3, 8 s	38.28	5.18	54.18
A, 2 s	55.50	3.74	40.76	3, 9 s	40.80	4.91	52.35
A, 3 s	56.33	3.77	39.90	3, 10 s	43.37	4.62	51.15
A, 4 s	57.13	3.79	39.08	3, J s	46.04	4.37	49.46
A, 5 s	58.06	3.71	38.23	3, Q s	48.93	4.16	47.48
A, 6 s	58.17	3.45	38.38	3, K s	52.07	3.96	44.60
A, 7 s	59.38	3.19	37.43	4, 5 s	38.53	5.84	52.78

CARDS	WIN%	TIE%	LOSS%	CARDS	WIN%	TIE%	LOSS%
4, 6 s	38.48	5.7	53.11	7, K s	55.84	3.38	41.47
4, 7 s	39.1	5.48	53.00	8, 9 s	48.85	3.88	45.34
4, 8 s	40.1	5.19	52.33	8, 10 s	50.5	3.65	45.17
4, 9 s	41.4	4.9	51.85	8, J s	52.31	3.4	44.27
4, 10 s	44.2	4.65	50.36	8, Q s	54.41	3.2	43.05
4, J s	46.86	4.4	48.72	8, K s	56.79	3.04	40.89
4, Q s	49.76	4.18	46.65	9, 10 s	52.37	3.3	43.48
4, K s	52.88	3.99	43.85	9, J s	54.11	3.1	42.70
5, 6 s	40.34	5.57	51.12	9, Q s	56.22	2.88	41.41
5, 7 s	40.97	5.39	51.20	9, K s	58.63	2.7	39.31
5, 8 s	41.99	5.1	50.63	10, J s	56.15	2.74	41.34
5, 9 s	43.31	4.81	49.93	10, Q s	58.17	2.59	40.31
5, 10 s	44.93	4.55	49.68	10, K s	60.58	2.4	37.99
5, J s	47.82	4.33	47.84	J, Q s	59.07	2.37	39.44
5, Q s	50.71	4.11	45.66	J, K s	61.47	2.18	37.76
5, K s	53.83	3.91	43.01	Q, K s	62.4	1.98	36.95
6, 7 s	42.82	5.08	49.57	A, A o	85.84	0.63	13.53
6, 8 s	43.81	4.84	48.98	2, 2 o	48.70	2.95	48.35
6, 9 s	45.15	4.55	48.49	3, 3 o	52.24	2.94	44.82
6, 10 s	46.8	4.28	48.13	4, 4 o	55.61	2.60	41.79
6, J s	48.57	4.06	47.44	5, 5 o	59.05	2.42	38.54
6, Q s	51.67	3.86	44.98	6, 6 o	62.02	2.18	35.80
6, K s	54.8	3.67	42.19	7, 7 o	65.07	1.99	32.94
7, 8 s	45.68	4.5	47.54	8, 8 o	68.26	2.06	29.67

7, 9 s	46.99	4.25	46.97	9, 9 o	71.46	2.01	26.53
7, 10 s	48.65	3.97	46.59	10, 10 o	74.36	1.69	23.94
7, J s	50.45	3.74	45.93	J, J o	77.06	1.32	21.63
7, Q s	52.52	3.55	44.62	Q, Q o	79.79	1.28	18.93
				K, K o	82.53	0.93	16.54

A few notes:

- Notice, using symmetry, we bring the number of states down to 169: since we are only analyzing the preflop odds, our hand combinations can be broken down into “suited” (2 hole cards of the same suit; we symbolize this by ‘s’), “off-suit”(2 hole cards of different suits; we symbolize this by ‘o’), and “pairs” (2 hole cards of the same value; obviously, they have to be of different suits).
- WIN%, TIE%, LOSS% each represent probability of winning, tying, or losing in a showdown respectively.

Optimal Policies and Risk Aversion

We have shown that given K_1 , K_2, the button position, and some belief of the structure of the transition model, we can calculate $U(fold)$ and $U(action)$. Calculation of $U(fold)$ is trivial. Below, we calculate $U(jam)$ for agents of varying risk aversion when K_1 , K_2 = *4000,* and the agent is posting the small blind for the round.

CARDS	U(0)	U(0.1)	U(0.25)	U(0.45)
A, 2 o	417.04	142.87	-268.38	-816.71
A, 3 o	446.69	173.01	-237.51	-784.87
A, 4 o	481.07	208.01	-201.58	-747.70
A, 5 o	518.55	246.20	-162.33	-707.04
A, 6 o	518.87	246.15	-162.94	-708.39
A, 7 o	560.66	288.56	-119.58	-663.78
A, 8 o	595.23	323.51	-84.06	-627.49
A, 9 o	623.27	351.75	-55.53	-598.57
A, 10 o	688.59	418.51	13.38	-526.79
A, J o	721.63	451.99	47.52	-491.76
A, Q o	749.82	480.15	75.63	-463.72
A, K o	792.38	523.24	119.52	-418.77
2, 3 o	-399.16	-663.52	-1,060.06	-1,588.79
2, 4 o	-371.96	-637.04	-1,034.65	-1,564.79
2, 5 o	-327.86	-593.44	-991.81	-1,522.98
2, 6 o	-340.04	-605.36	-1,003.35	-1,533.99
2, 7 o	-336.66	-602.15	-1,000.39	-1,531.38
2, 8 o	-273.35	-539.13	-937.79	-1,469.35
2, 9 o	-209.31	-475.82	-875.59	-1,408.61
2, 10 o	-147.33	-414.93	-816.33	-1,351.53
2, J o	-64.95	-334.42	-738.61	-1,277.53
2, Q o	37.52	-233.01	-638.79	-1,179.83
2, K o	184.52	-88.61	-498.30	-1,044.55
3, 4 o	-282.81	-548.73	-947.60	-1,479.44
3, 5 o	-247.45	-513.95	-913.70	-1,446.69
3, 6 o	-261.52	-527.60	-926.72	-1,458.88
3, 7 o	-255.15	-521.38	-920.72	-1,453.18
3, 8 o	-242.78	-508.89	-908.07	-1,440.31
3, 9 o	-169.86	-436.44	-836.31	-1,369.47
3, 10 o	-109.94	-377.71	-779.37	-1,314.91
3, J o	-21.03	-290.50	-694.71	-1,233.66
3, Q o	72.29	-198.05	-603.55	-1,144.23
3, K o	241.44	-31.69	-441.40	-987.67
4, 5 o	-177.77	-445.04	-845.94	-1,380.47
4, 6 o	-191.31	-458.47	-859.20	-1,393.51
4, 7 o	-186.59	-453.78	-854.57	-1,388.96
4, 8 o	-168.50	-435.15	-835.12	-1,368.41
4, 9 o	-146.94	-413.87	-814.27	-1,348.13
4, 10 o	-78.18	-346.02	-747.78	-1,283.45
4, J o	7.08	-262.25	-666.25	-1,204.92
4, Q o	106.94	-163.32	-568.71	-1,109.24
4, K o	265.01	-7.40	-416.01	-960.83
5, 6 o	-110.15	-377.80	-779.26	-1,314.55
5, 7 o	-110.13	-378.01	-779.85	-1,315.62
5, 8 o	-94.24	-361.64	-762.74	-1,297.54
5, 9 o	-66.98	-334.38	-735.47	-1,270.27
5, 10 o	-48.35	-316.43	-718.55	-1,254.71
5, J o	43.65	-225.70	-629.73	-1,168.43
5, Q o	145.77	-124.19	-529.14	-1,069.06
5, K o	299.33	27.29	-380.78	-924.87
6, 7 o	-40.09	-308.47	-711.05	-1,247.82
6, 8 o	-22.97	-290.87	-692.72	-1,228.51
6, 9 o	-1.57	-269.56	-671.55	-1,207.52
6, 10 o	19.68	-248.81	-651.55	-1,188.54
6, J o	65.38	-204.35	-608.95	-1,148.42
6, Q o	178.37	-91.77	-496.99	-1,037.28
6, K o	338.33	66.37	-341.58	-885.51
7, 8 o	44.08	-224.42	-627.17	-1,164.17
7, 9 o	67.38	-201.05	-603.71	-1,140.57
7, 10 o	89.45	-179.41	-582.69	-1,120.40
7, J o	131.51	-138.30	-543.01	-1,082.63
7, Q o	198.00	-72.36	-477.89	-1,018.61
7, K o	371.21	99.42	-308.26	-851.84
8, 9 o	138.92	-129.61	-532.40	-1,069.45
8, 10 o	156.47	-112.75	-516.57	-1,055.01
8, J o	203.54	-66.13	-470.65	-1,010.00
8, Q o	267.82	-2.35	-407.62	-947.96

8, K o	402.72	130.83	-277.02	-820.80
9, 10 o	228.85	-40.10	-443.51	-981.40
9, J o	272.47	3.06	-401.05	-939.87
9, Q o	348.58	78.34	-327.01	-867.48
9, K o	467.23	196.39	-209.87	-751.55
10, J o	339.04	69.59	-334.59	-873.50
10, Q o	396.16	126.41	-278.21	-817.71
10, K o	531.38	260.99	-144.58	-685.35
J, Q o	436.31	166.85	-237.33	-776.25
J, K o	542.53	272.23	-133.21	-673.80
Q, K o	580.93	311.03	-93.82	-633.61
A, 2 s	496.48	219.63	-195.66	-749.37
A, 3 s	530.88	254.72	-159.52	-711.83
A, 4 s	563.86	288.40	-124.80	-675.72
A, 5 s	600.09	325.29	-86.92	-636.52
A, 6 s	599.28	324.06	-88.76	-639.19
A, 7 s	643.24	368.64	-43.25	-592.45
A, 8 s	682.32	408.20	-2.99	-551.25
A, 9 s	716.32	442.56	31.94	-515.57
A, 10 s	790.41	518.23	109.97	-434.39
A, J s	822.77	551.21	143.86	-399.27
A, Q s	856.16	585.32	179.07	-362.60
A, K s	889.95	619.99	215.06	-324.85
2, 3 s	-374.31	-650.09	-1,063.77	-1,615.34
2, 4 s	-340.11	-616.29	-1,030.56	-1,582.93
2, 5 s	-298.79	-575.45	-990.43	-1,543.75
2, 6 s	-248.37	-521.59	-931.43	-1,477.88
2, 7 s	-235.29	-509.47	-920.74	-1,469.09
2, 8 s	-154.79	-430.34	-843.67	-1,394.77
2, 9 s	-76.71	-353.73	-769.26	-1,323.29
2, 10 s	2.27	-276.71	-695.16	-1,253.11
2, J s	87.44	-193.04	-613.77	-1,174.74
2, Q s	190.92	-90.08	-511.58	-1,073.58
2, K s	306.17	25.33	-395.92	-957.59

3, 4 s	-200.84	-473.64	-882.84	-1,428.45
3, 5 s	-163.56	-436.96	-847.07	-1,393.88
3, 6 s	-170.20	-443.84	-854.30	-1,401.58
3, 7 s	-154.94	-429.40	-841.10	-1,390.02
3, 8 s	-127.25	-402.75	-816.01	-1,367.01
3, 9 s	-38.55	-315.34	-730.51	-1,284.08
3, 10 s	38.00	-240.81	-659.02	-1,216.63
3, J s	126.93	-153.03	-572.96	-1,132.86
3, Q s	225.95	-54.70	-475.67	-1,036.97
3, K s	348.50	68.75	-350.87	-910.37
4, 5 s	-93.66	-367.59	-778.50	-1,326.37
4, 6 s	-101.36	-375.68	-787.17	-1,335.81
4, 7 s	-86.40	-361.53	-774.21	-1,324.46
4, 8 s	-52.51	-328.21	-741.77	-1,293.19
4, 9 s	-16.26	-293.22	-708.66	-1,262.58
4, 10 s	71.11	-207.67	-625.85	-1,183.43
4, J s	158.66	-121.24	-541.10	-1,100.91
4, Q s	259.85	-20.49	-441.00	-1,001.69
4, K s	380.17	100.74	-318.40	-877.24
5, 6 s	-22.93	-297.24	-708.71	-1,257.32
5, 7 s	-11.73	-287.10	-700.15	-1,250.89
5, 8 s	20.65	-255.37	-669.41	-1,221.46
5, 9 s	61.76	-215.11	-630.42	-1,184.17
5, 10 s	99.78	-179.00	-597.16	-1,154.70
5, J s	196.12	-83.66	-503.33	-1,062.88
5, Q s	299.33	19.45	-400.38	-960.14
5, K s	416.61	137.55	-281.03	-839.15
6, 7 s	59.01	-216.63	-630.08	-1,181.35
6, 8 s	91.13	-185.01	-599.22	-1,151.50
6, 9 s	128.56	-148.66	-564.49	-1,118.94
6, 10 s	169.33	-109.57	-527.92	-1,085.72
6, J s	219.37	-60.75	-480.94	-1,041.19
6, Q s	332.56	52.64	-367.24	-927.09
6, K s	453.18	174.49	-243.55	-800.93

7, 8 s	158.50	-117.96	-532.64	-1,085.56
7, 9 s	196.90	-80.38	-496.31	-1,050.88
7, 10 s	238.45	-40.38	-458.62	-1,016.28
7, J s	288.47	8.45	-411.58	-971.61
7, Q s	357.27	76.94	-343.55	-904.20
7, K s	488.95	210.40	-207.43	-764.54
8, 9 s	267.96	-9.05	-424.57	-978.58
8, 10 s	305.00	26.21	-391.99	-949.58
8, J s	360.06	80.59	-338.62	-897.57
8, Q s	427.61	147.81	-271.89	-831.49
8, K s	520.18	241.66	-176.12	-733.17
9, 10 s	377.39	99.23	-318.01	-874.33
9, J s	428.64	149.81	-268.43	-826.08
9, Q s	497.93	219.06	-199.24	-756.97
9, K s	589.74	312.22	-104.05	-659.07
10, J s	497.82	219.15	-198.85	-756.19
10, Q s	560.06	281.13	-137.28	-695.15
10, K s	656.34	379.40	-36.02	-589.92
J, Q s	596.06	317.83	-99.51	-655.97
J, K s	679.10	401.70	-14.39	-569.18
Q, K s	714.38	437.69	22.64	-530.75
A, A o	1,668.23	1,436.34	1,088.50	624.73
2, 2 o	203.47	-78.28	-500.89	-1,064.38
3, 3 o	347.56	67.22	-353.29	-913.96
4, 4 o	477.78	198.89	-219.44	-777.21
5, 5 o	613.87	337.42	-77.25	-630.15
6, 6 o	730.08	456.14	45.24	-502.63
7, 7 o	850.56	579.86	173.81	-367.58
8, 8 o	981.93	715.88	316.80	-215.31
9, 9 o	1,110.98	850.02	458.56	-63.37
10, 10 o	1,222.79	966.39	581.78	68.96
J, J o	1,324.64	1,072.71	694.81	190.94
Q, Q o	1,435.31	1,189.48	820.74	329.08
K, K o	1,539.65	1,299.59	939.50	459.39

A few notes:

- Notice, using symmetry, we bring the number of states down to 169: since we are only analyzing the preflop odds, our hand combinations can be broken down into "suited" (2 hole cards of the same suit; we symbolize this by 's'), "off-suit"(2 hole cards of different suits; we symbolize this by 'o'), and "pairs" (2 hole cards of the same value; obviously, they have to be of different suits).
- U(x) refers to the utility for jamming in this scenario when the agent in question has a risk aversion parameter of $\lambda^h = x$.
- Notice, the higher the risk aversion parameter, the lower the utility to jamming. This makes sense because jamming is a volatile action; therefore, the more we "dislike" volatility, the lower the utility.
- Agent chooses to jam if the aforementioned utility calculation is higher than -*SB* (or -*BB* as the case maybe).
- Green cells represent positive jam utility; Red cells represent negative jam utility.

RESULTS

The effect of risk aversion in Heads Up Texas Hold'em games can be understood as the relationship between "Exploiters" and "Explorers". Risk neutral players are exploiters; upon coming across a hand of positive expected value, these players exploit the opportunity at hand and jam. The don't care about the magnitude of the expected value -- just, that its positive! Risk averse players, on the other hand, are explorers: in order to jam, a risk averse players require that the expected value is positive and has high (risk discounted) magnitude. Failing either, the risk averse player will explore her options: she folds her hand in hope of a better hand in the future.

As a result, we expect that the optimal amount of risk aversion in low capital share scenarios is favorable; our agent doesn't have too many bets, so it better pick the right bet! On the flip side, in scenarios of high/equal capital share, high risk aversion may cause a player to wait too long for a great hand.

We were cheeky above in that we claimed that "optimal amount of risk aversion is favorable" -- obviously, the optimal amount of risk aversion is favorable! Our point is that a "risk aware" player (that is a player that is cognizant of the riskiness of her bets) is better off than a player that is "risk ignorant" (that is a player that doesn't account for risky bets). We propose possible methods of finding the "optimal amount" of such risk aversion in the "Next Steps" section.

	Risk Aversion = 0		**Risk Aversion = 0.40**	
Capital Structure	**Wins**	**Losses**	**Wins**	**Losses**
50%-50%	49.902	50.0980	50.0980	49.902
75%-25%	73.0840	26.9160	26.9160	73.0840
25%-75%	25.0703	74.9297	74.9297	25.0703

Table 1: Results for 5,00,000 tournament simulations between a risk neutral player (λ^h = 0) and a risk averse player (λ^h = 0.40).

	Risk Aversion = 0.10		**Risk Aversion = 0.25**	

Capital Structure	Wins	Losses	Wins	Losses
50%-50%	50.002	49.9980	49.9980	50.002
75%-25%	75.0570	24.9430	24.9430	75.0570
25%-75%	25.0050	74.9950	74.9950	25.0050

Table 2: Results for 5,00,000 tournament simulations between a risk sensitive player (λ^h = 0.10) and a moderately risk averse player (λ^h = 0.25).

	Risk Aversion = 0		Risk Aversion = 0.65	
Capital Structure	Wins	Losses	Wins	Losses
50%-50%	54.002	45.9980	45.9980	54.002
75%-25%	73.0510	26.9490	26.9490	73.0510
25%-75%	28.0323	71.9677	71.9677	28.0323

Table 3: Results for 5,00,000 tournament simulations between a risk neutral player (λ^h = 0) and a severely risk averse player (λ^h = 0.65).

NEXT STEPS

In this section we propose possible extensions to our work.

- The first obvious step of extension seems to be to extend our analysis beyond a Jam/Fold strategy. In choosing to either jam or fold, the agent loses on the opportunity to "check" (i.e., to be zero). Therefore, we miss out and fold on all hands when our opponent bets zero as well!
- We calculate *Pr(opponent folds)* as a minimax calculation assuming that the opponent is a risk neutral player. In reality, this might be a naive assumption: different players have different levels of risk aversion, and therefore we might be constantly over (or under) estimating *Pr(opponent folds)*. We believe that a better strategy would be use supervised learning techniques to predict *Pr(opponent folds)*. Furthermore, Heads Up tournaments often see players facing the same opponents repeatedly; we can then learn an opponent specific measure of *Pr(opponent folds)*.

- Miltersen[6] shows that optimal play in cash games may be different from tournament plays: in a cash game, a player may choose to leave the table. This is not an option in the tournament setting; the player's next hand may have a negative payoff which may influence her decision to jam/fold. Miltersen solution to this problem applies for us as well; instead of finding *U(action)*, we find *U*(U(action))* where *U** is a mapping to the tournament utility from the cash utility of a hand. This mapping can be learnt using standard reinforcement learning procedures.
- In a tournament situation, if the tournament doesn't end with the current hand, he is forced to play another hand. This other hand may have a negative expected payoff for him (for instance, he may have to post the big blind while the small blind has the advantage with the given parameters). If it does have a negative expected payoff, he is better off trying to avoid playing it and this will influence his optimal strategy in the current hand towards trying to end the tournament, typically by playing somewhat looser.

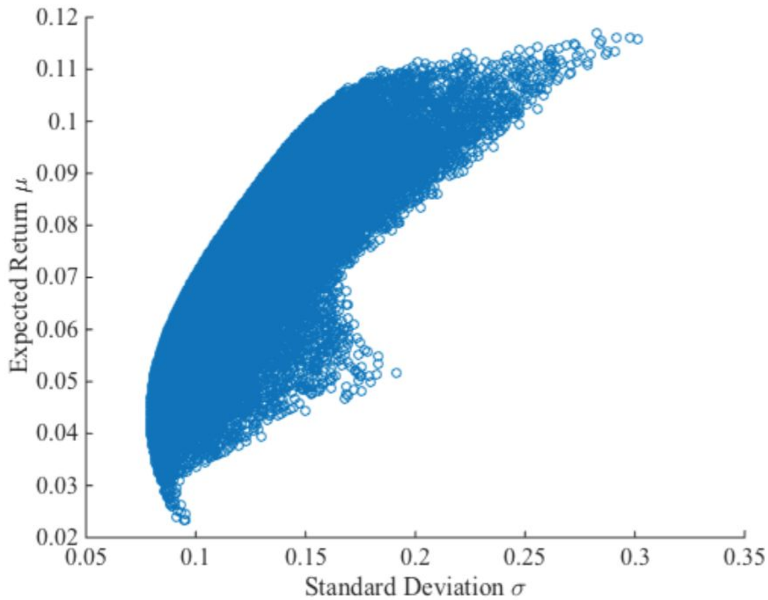
[6] Miltersen, Sorenson. "A Near Optimal Strategy for a Heads-Up Texas Hold'em Poker Tournament".

APPENDIX

1. RELATED WORKS

Mean Variance Preferences

Markowitz pioneered an approach to financial analysis that quantified risk in a portfolio and gave very concrete advice about which risky assets an investor should hold.



Markowitz started with the simple assumption that investors prefer higher returns and lower risk, as captured by the standard deviation or variance of a portfolio. Under this assumption, it makes sense to plot returns in "return risk" or "mean standard-deviation space" with expected returns on the y-axis and standard deviation on the x-axis. Since investors like higher returns and dislike risk (variance, or standard deviation), investors prefer portfolios that move in the northwest direction. The utility function for such investors is given as follows:

$$U(action) = E[action] - \lambda^h \sigma[action]$$

Where:

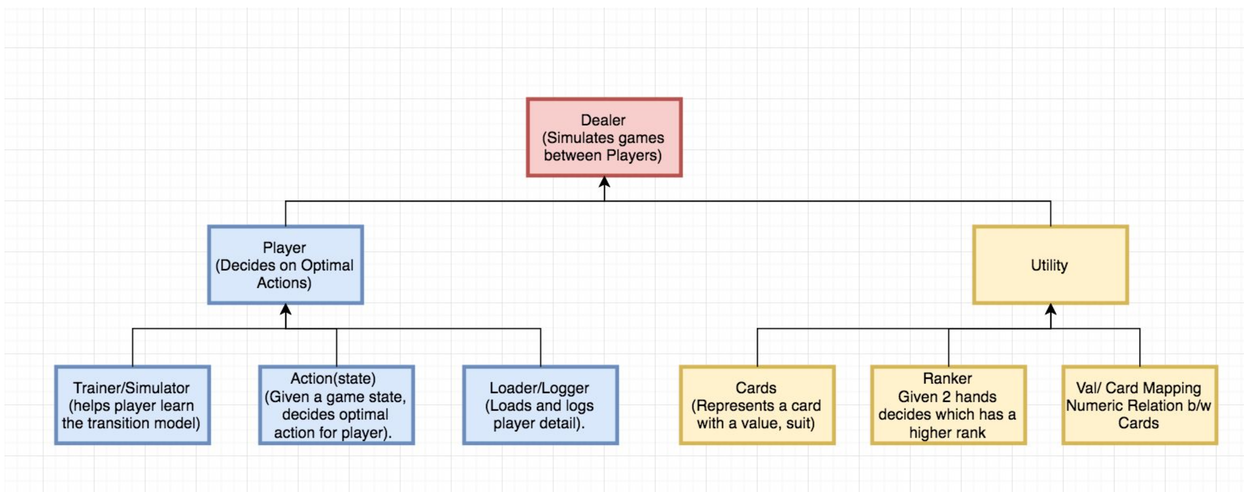
- $U(action)$ is the utility obtained from a given action.
- $E[action]$ is the expected payoff of the action.
- λ^h is the agent specific parameter which measures aversion to risk.
- $\sigma[action]$ is the standard deviation of the payoffs to the action. It measures the "riskiness" of the action.

Source: Phelan, Gregory. Asset Pricing, Williams College F'16

Poker Hand Rankings

Source: http://www.slickstermagazine.com/poker-hand-rankings/

2. IMPLEMENTATION & DOCUMENTATION



Information Flow Diagram for package

Dealer Class

The dealer class is responsible for simulating tournaments between players. Given two players, the dealer class can run a simulation of 5,000,000 tournament games between the two. The class then logs the relevant results. Relevant interface follows

```
### FUNCTION TO RUN SIMULATE 5,00,000 TOURNAMENTS
### @param riskAversion1 risk aversion of player
### @param riskAversion2 risk aversion of player
### @param K1 capital for player
### @param K2 capital for player
def Tournament(riskAversion1, riskAversion2, K1, K2):
```

Player Class:

The Player class is responsible for learning its transition model, saving/loading it from memory, mapping states to actions (i.e., its policy), and relevant data methods. Relevant interfaces follow:

```
### Class to represent one dictionary event
### That is the transition model for a given state
### @param win probability of winning
### @param loss probability of losing
### @param fold probability of folding
### @param tie probability of tieing
class Probility:


### Function to load transition model from memory
### @param filename file to load from
def loadTransitionModel(filename):

# FUNCTION TO MAP STATE TO ACTION
# STATE INFORMATION: @param K1, K2 capital structure
#                    @param A1, A2 antes
#                   @param card1, card2 player's cards
# @return 1 if JAM, -1 if FOLD
def action(self, K1, K2, A1, A2, card1, card2):
```

Utility Classes

The following section contains relevant interfaces for utility classes used by our agent. An interface for the following classes are available:

- Card

```
## Class for a Card Object
def Card:

       # CONSTRUCTOR
       # @param self pointer to self
       # @param suit suit of this card
       # @param val val of this card
       def __init__(self, suit, val):

       # @param other another Card object
```

```
    # @return True if and only if this.suit = other.suit &&
    #                                                     this.val = other.val
    def __cmp__(self, other):
```

- Deck

```
## Class for a Deck object
## Represents a deck of cards
def Deck:

    # CONSTRUCTOR
    # @return inits new deck of cards
    def __init__(self):


    # @return shuffles this deck of cards
    def shuffle(self):

    # @param card card to get from the deck
    # @return reference to card object if present in deck, None else
    def get(self, card):

    # @return first card of deck
    def getFirstCard(self):
```

- Suit

```
# Domain of suits
Suits = ["Spades", "Clubs", "Diamonds", "Hearts"]
```

- Values

```
# Domain of Card Values
Values = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
```

- Result (Enum)

```
class Result(Enum):
    win = 1
    tie = 0
    loss = -1
```

- Ranker

```
class Ranker:

    # @return true if and only if hand represents a royal flush
    def isRoyalFlush(hand):

    # @return true if and only if hand represents a straight flush
    def isStraightFlush(hand):

    # @return true if and only if hand represents a four of a kind
```

```python
def isFourOfAKind(hand):

# @return true if and only if hand represents a full house
def isFullHouse(hand):

# @return true if and only if hand represents a flush
def isFlush(hand):

# @return true if and only if hand represents a straight
def isStraight(hand):

# @return true if and only if hand represents a three of a kind
def isThreeOfAKind(hand):

# @return true if and only if hand represents a two pair
def isTwoPairs(hand):

# @return true if and only if hand represents a pair
def isPair(hand):

# @ return Result.win if handA > handB
#          Result.tie if handA = handB
#          Result.loss if handA < handB
def rank(handA, handB):
```