Game Theory: Texas hold'em Project

Instructed by *Tang Pingzhong*

Due on Jun 16, 2013

Wu yuexin, Kuang Zhonghong, Yang Sheng
Geek 10 2011012061/2011012357/2011012359

1 Participants

Team Name
Team Leader
None
Team Members
Wu Yuexin, Kuang Zhonghong, Yang Sheng
Contributions
Kuang Zhonghong: modeling and theoretical analysis.
Wu Yuexin: adapt the model into reality and find optimal parameters to maximaize profit. Coding.
Yang Sheng: design the overall architecture, implement the needed algorithms, and setup the origin bot on which optimization is done. Coding.

2 Our model : a incomplete information dynamic game

2.1 Dynamic game with one raise round

Two players are Alice and Bob, Alice go first and Bob go next.

Trivially, every player's type is uniformly distributed on $[0, 1]$.

Assumption 1: The tokens on the table is m currently. Each raise adds n.

Assumption 2: Alice's strategy is described as follow. If $a \in [0, A]$, check or fold; if $a \in [A, B]$, check or call; if $a \in [B, 1]$, raise and call.

Assumption 3: Bob strategy is descirbed as follow. If Alice check and $b \in [0, C]$, check; if Alice check and $b \in [C, 1]$, raise. If Alice raise and $b \in [0, D]$, fold; if Alice raise and $b \in [D, E]$, call; if Alice raise and $b \in [E, 1]$, raise.

Notice: Assumption 2 and assumption 3 are made because of **increasing difference property**.

Firstly, we need to compute C. At this threshold point, if Bob check, then his expected payoff is $-m + 2m\frac{C}{B}$; if Bob raise, this is a signal to Alice that $b \geq C$, then if Bob fold his payoff will be $-m$, if Bob call then his expected payoff is $-(m+n)+2(m+n)\frac{A-C}{1-C}$. Then we can use C to express A.

$$
\begin{aligned}
-m &= -(m+n)+2(m+n)\frac{A-C}{1-C} \\
n &= 2(m+n)\frac{A-C}{1-C} \\
A-C &= \frac{n(1-C)}{2(m+n)} \\
A &= C+\frac{n(1-C)}{2(m+n)} \\
A &= (1-\frac{n}{2(m+n)})C+\frac{n}{2(m+n)}
\end{aligned}
$$

Then Bob expected payoff is

$$
\begin{aligned}
&\Pr[\text{Alice fold}]m+\Pr[\text{Alice call}](-(m+n)+2(m+n)\Pr[\text{Bob's winning probability}|\text{Alice call}]) \\
=&\frac{A}{B}m+\frac{B-A}{B}(m+n)(-1+2\cdot 0) \\
=&\frac{1}{B}[Am+(A-B)(m+n)]
\end{aligned}
$$

Set

$$
-m+2m\frac{C}{B}=\frac{1}{B}[Am+(A-B)(m+n)]
$$

, we can use B to express C.

$$
\begin{aligned}
-m+2m\frac{C}{B} &= \frac{1}{B}[Am+(A-B)(m+n)] \\
-Bm+2mC &= (2m+n)A-Bn-Bm \\
2mC &= (2m+n-\frac{2mn+n^2}{2(m+n)})C+\frac{2mn+n^2-2Bmn-2Bn^2}{2(m+n)} \\
\frac{2Bmn+2Bn^2-2m^2-n^2}{2(m+n)} &= (n-\frac{2mn+n^2}{2(m+n)})C \\
2Bmn+2Bn^2-2m^2-n^2 &= (2mn+2n^2-2mn-n^2)C \\
2Bmn+2Bn^2-2m^2-n^2 &= n^2C \\
C &= \frac{2Bmn+2Bn^2-2m^2-n^2}{n^2}
\end{aligned}
$$

Obviously, at point D, it is indifferent to fold or call. Then we can use B to express D.

$$
\begin{aligned}
-m &= -m - n + 2(m+n)\frac{D-B}{1-B} \\
n &= 2(m+n)\frac{D-B}{1-B} \\
\frac{n(1-B)}{2(m+n)} &= D - B \\
D &= B + \frac{n(1-B)}{2(m+n)} \\
D &= (1 - \frac{n}{2(m+n)})B + \frac{n}{2(m+n)}
\end{aligned}
$$

And at point E, it is indifference to call or raise. If Bob call, then his expected payoff is $-(m+n)+2(m+n)\frac{E-B}{1-B}$. If Bob raise, then his expected payoff is $-(m+2n)+2(m+2n)\frac{E-B}{1-B}$. Then we can use B to express E, which is $E = \frac{1+B}{2}$.

For Alice, it is indifferent to check or raise at point B. If Alice check, then her expected payoff is

$$
\begin{aligned}
&\Pr[\text{Bob check}]\text{expected payoff}|\text{Bob check}] + \Pr[\text{Bob raise then Alice call}]\text{expected payoff} \mid \text{Bob raise then Alice call} \\
=&Cm + (1-C)(-(m+n) + 2(m+n)\frac{B-C}{1-C}) \\
=&Cm + (C-1)(m+n) + 2(m+n)(B-C) \\
=&-nC - (m+n) + 2(m+n)B
\end{aligned}
$$

If Alice raise, then her expected payoff is

$$
\begin{aligned}
&\Pr[\text{Bob fold}]m + \Pr[\text{Bob call}]\text{expected payoff} \mid \text{Bob call} + \Pr[\text{Bob raise}]\text{expected payoff} \mid \text{Bob raise} \\
=&Dm + (E-D)(m+n)(-1+2\cdot 0) + (1-E)(m+2n)(-1+2\cdot 0) \\
=&Dm + (D-E)(m+n) + (E-1)(m+2n) \\
=&(2m+n)D + nE - (m+2n)
\end{aligned}
$$

Then,

$$-nC - (m+n) + 2(m+n)B = (2m+n)D + nE - (m+2n)$$

Now we get five equations

$$A = (1 - \frac{n}{2(m+n)})C + \frac{n}{2(m+n)} \tag{1}$$

$$C = \frac{2Bmn + 2Bn^2 - 2m^2 - n^2}{n^2} \tag{2}$$

$$D = (1 - \frac{n}{2(m+n)})B + \frac{n}{2(m+n)} \tag{3}$$

$$E = \frac{1+B}{2} \tag{4}$$

$$-nC - (m+n) + 2(m+n)B = (2m+n)D + nE - (m+2n) \tag{5}$$

Rewrite this equations in matrix form

$$\begin{bmatrix} 1 & 0 & \frac{n}{2(m+n)}-1 & 0 & 0 \\ 0 & 2(m+n) & -n & -(2m+n) & n \\ 0 & \frac{2(m+n)}{n} & -1 & 0 & 0 \\ 0 & \frac{n}{2(m+n)}-1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} \begin{bmatrix} \frac{n}{2(m+n)} \\ -n \\ 1+\frac{2m^2}{n^2} \\ \frac{n}{2(m+n)} \\ -1 \end{bmatrix} \tag{6}$$

We need to solve A, B, C, D, E.
Plug the equations into **Mathematica**™, we get

$$\begin{aligned} A &= \frac{1}{2}(2+\frac{3m}{n}-\frac{4m}{m+n}-\frac{m(9m+2n)}{4m^2+5mn+2n^2}) \\ B &= 1+\frac{4m(m-n)(m+n)}{n(4m^2+5mn+2n^2)} \\ C &= \frac{6m^3+2m^2n+mn^2+2n^3}{4m^2n+5mn^2+2n^3} \\ D &= 1+\frac{m}{n}-\frac{m(7m+4n)}{4m^2+5mn+2n^2} \\ E &= \frac{2m^3+4m^2n+3mn^2+2n^3}{4m^2n+5mn^2+2n^3} \end{aligned}$$

If $m = 10, n = 20$, then $A = \frac{26}{33}, B = \frac{8}{11}, C = \frac{15}{22}, D = \frac{9}{11}, E = \frac{19}{22}$.
We find that if $m \geq n$, then $A, B, C, D, E \geq 1$.

2.2 Dynamic game with two raise round

Actually, it is too difficult for us to model the dynamic game with two round-raise. So we just use the model we get in **Dynamic game with one raise round** and add one new rule: if my position is 1 (that means I lose with probability 0), then I raise at the second raise-round, if there is one. This is the reference method we use in **flop**, **turn** and **river**.

2.3 Adapting the model

Since the model needs to much assuption, it cannot be put into actual BOT. So we need to do some modification. First, the conclusion give us some intuition of being "over rational" of nearly no check, and have a high probability of folding. Then we put these parameters into the bot, and run several times to see the outcome.

At first, we put much probability on "check", but this version is among the worst of all the agents we had implemented. Then we lower this probability step by step, and seems to get better result as the probability goes down. Suprisingly, the best parameter we found was closer to the theretical result then we had expected.

2.4 Deserted Work

The assignment didn't mentioned of the existance of evalHandTables, so we decided to write a function to evaluate which hand is better. Since this function may be called many times in the game, we tried to optimize this function. Instead of outputing which hand is better, we output some kind of rank(i.e. an array of 6 integers). We find the code highly efficient, even if called millons of time.

However, the existence of evalHandTables means all the work is deserted. The function provided in evalHandTables is much more efficnent then our function with the help of hard coded table. But writing this code helped us understanding Texas Poker, and of the project.

We also planned to use the potential model from some paper. But the algorithm is not explicitly explained and need so much detail to be implemented. Another problem is that the algorithms mentioned in the paper all need much more computational power then is available to us. Merely the preprocess takes more then several weeks of computation, which is far beyond our capability.

2.5 Winning probability

All the steps we calculate winning probability. Since calculating winning probability may need too much time, we find the table for preflop on the Internet. As for the floping stage, calculating also needs so much time. So we use **Monte Carlo method** to calculate this winning probability. To help debugging, we use 1000 rounds on our machine. In the real situation, we use at least 10000. We even use 100000 if time permits.

2.6 Overall Architecture

We use a switch in the main function to work each stage. And do not distinguish between first hand and second hand.

3 Reference

Pre-Flop Hole Card Winning Probabilities: `http://www.cs.indiana.edu/~kapadia/nofoldem/index.html`