Texas Hold 'em Strategy Design

Yung-Jen Cheng, Peiheng Hu, Sail Wu, Xide Xia

AM207 Advanced Scientific Computing: Stochastic Optimization Methods

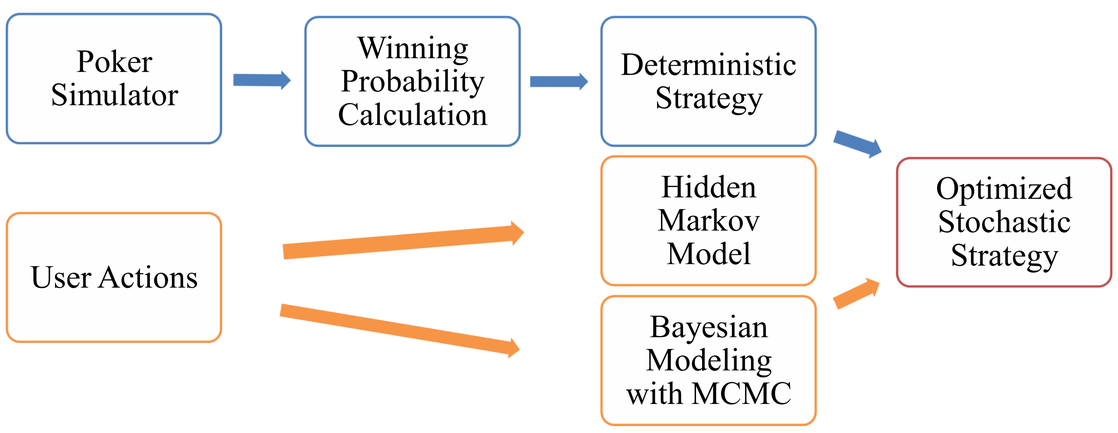
Spring 2015

Introduction

Poker, Texas hold 'em, is a game of imperfect information, making it impossible for anyone to conclude the final outcome of the hand. Admittedly, we can derive nearly deterministic probability of a given hand from millions of simulation results. However, the way how people react and the psychological components involved in betting make poker one of the most popular card games in the world. This project aims to implement a system which allows the designed computer program to model the style of the player, obtain observations of the style of that player and update the style of the player to play in the game of heads-up unlimited Texas hold 'em.

Approach

Deterministic and Stochastic Combination



Winning Probability

The winning probability is the probability to win this game, given your hole cards and the known board cards. In the river stage, all five bard cards are known so we compute this probability by permuting all possible opponent's hole cards and count winning times over total (45*44/2=990) combinations. In the turn stage, the method is similar but we need to permute one extra card since only four board cards are known. See Fig.1 for example.

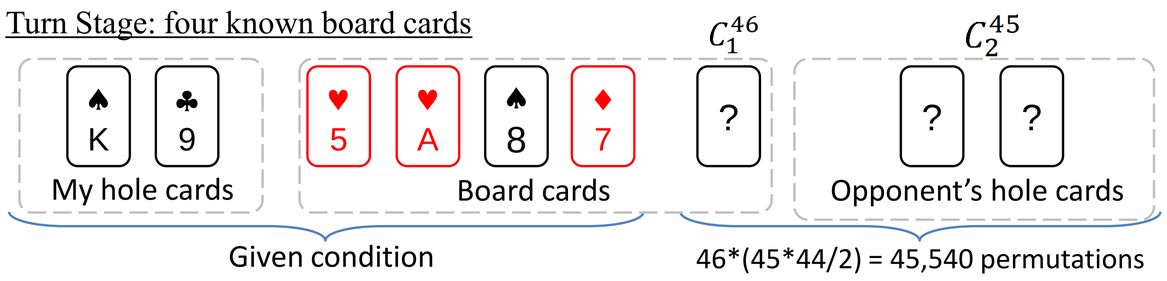


Figure 1. Compute winning probability in turn stage.

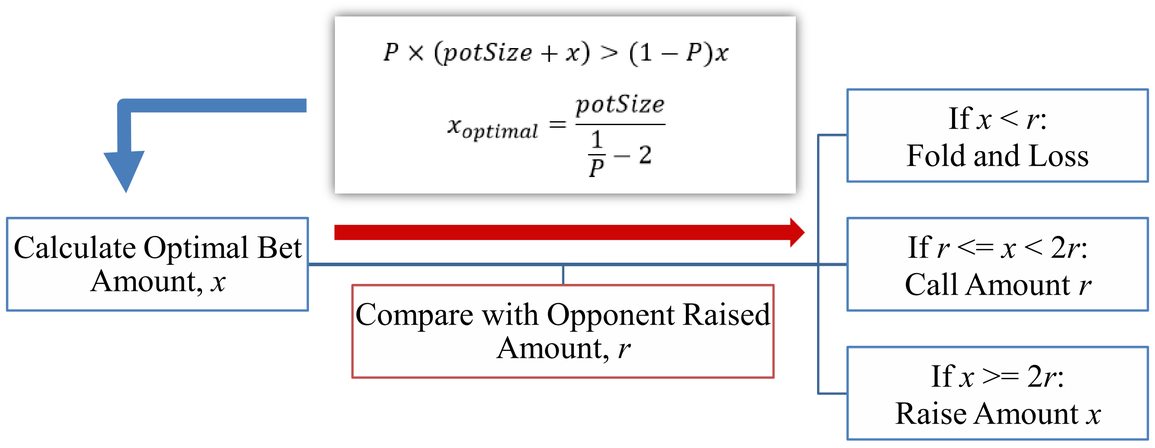
In the flop stage, the number of permutations grows to (47*46/2)*(45*44/2) = 1,070,190, which is too large to compute on the fly. Thus, we only sample one tenth of opponent's hole cards to approximate it.

Deterministic Model

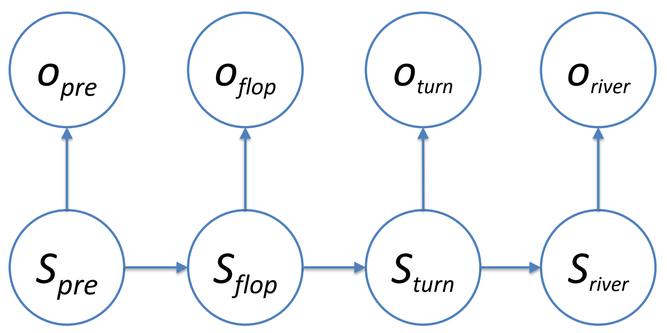
With the probability of winning computed from permuting all the combinations, a fixed/deterministic strategy can be deduced.

- Scenario 1 winning probability > 0.5: Better off to raise more money
- Scenario 2 winning probability < 0.5: Denote winning probability as P, current money in pot is potSize. We want to solve the optimal value to raise, which is x.

Action Decision Rule



HMM Hand Strength Estimation



Transition Probability

	Weak	Medium	Strong
Weak	0.50	0.33	0.16
Medium	0.33	0.33	0.33
Strong	0.16	0.33	0.50

Observation States Count for Emission Probability

	(Call, Low)	(Call, Med)	(Call, High)	(Raise, Low)	(Raise, Med)	(Raise, High)
Weak	50+	12+	8+	15+	10+	5+
Medium	15+	20+	15+	15+	20+	15+
Strong	5+	10+	15+	15+	30+	25+

- Hand strength state, S:{Weak, Medium, Strong} in each stage based on winning %
- Observed State, O:{(Call, L), (Call,M), (Call, H), (Raise,L), (Raise,M), (Raise,H)}
- L, M, and H indicates player's betting size, Low, Medium or High, relative to Pot size
- Emission Probability is calculated based on player's historical betting records

We used Viterbi Algorithm to estimate the player's most likely hand strength in the current stage. After estimating the player's hidden hand strength, we adjusted the winning probability accordingly.

Bluff Tendency Bayesian MCMC Estimation

We model a player's bluff status with probability $f(\theta_1 + \theta_2 t)$, conditioned on $\Theta = (\theta_1, \theta_2)$. We used $f(z) = 1/(1 + \exp(-z))$ to denote a player's likelihood to bluff in each action. t is the player's betting size relative to pot size. Due to limited prior information for each player, we chose a very flat prior: $\theta_1, \theta_2 \sim \mathcal{N}(0, 10^2)$. Thus, Prior can be expressed as:

$$P(\theta_1, \theta_2) = P(\theta_1) \times P(\theta_2) = \frac{1}{\sqrt{2\pi}\sigma_1\sqrt{2\pi}\sigma_2} \exp\left(-\frac{\theta_1^2}{2\sigma_1^2} - \frac{\theta_2^2}{2\sigma_2^2}\right) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{\theta_1^2}{2\sigma_1^2} - \frac{\theta_2^2}{2\sigma_2^2}\right)$$

Where $\sigma_1 = \sigma_2 = 10$

We expressed the likelihood for each action as:

$$Likelihood_i = \frac{1}{1+\exp(-\theta_1-\theta_2 t_i)}^{y_i} \times \frac{\exp(-\theta_1-\theta_2 t_i)}{1+\exp(-\theta_1-\theta_2 t_i)}^{1-y_i}$$

Where $y_i \in \{0, 1\}$ indicates if the player is bluffing at each action based on if his current winning probability greater *0.5*. The posterior likelihood can be expressed as:

$$P(\Theta|D) \propto P(D|\Theta) \times P(\Theta)$$

$$P(\Theta|D) \propto \prod_{i=1}^{N} \left(\frac{1}{1+\exp(-\theta_1-\theta_2 t_i)}^{y_i} \times \frac{\exp(-\theta_1-\theta_2 t_i)}{1+\exp(-\theta_1-\theta_2 t_i)}^{1-y_i}\right) \times \exp\left(-\frac{\theta_1^2}{2\sigma_1^2} - \frac{\theta_2^2}{2\sigma_2^2}\right)$$

We then used MCMC rejection sampling to estimate the player's bluff probability, and incorporated that to adjust our winning probability in the overall betting strategies.

Conclusions

- Our project successfully created an AI poker player that can determine the optimal betting strategies that based on both statistics and human behaviors to maximize its profits
- We obtained the statistically optimal betting decision based on calculating the winning percentage for each given hand after permutating all the possible scenarios of the opponent hands
- The human behaviors are captured by the historical records of the players and we used HMM to estimate player's hand strength and Bayesian MCMC to deduce the whether the player is bluffing at each action.
- The resulted code have conducted thousands of runs to be tested on both AI poker players for debugging and parameter tunings.

School of Engineering and Applied Sciences • Institute for Applied Computational Science