Building a Pineapple AI

Sen Lin, Vinson Luo, Glenn Yu

Introduction

Pineapple is a variant of Open-face Chinese Poker played between 2 players. The game is played over 5 rounds. Each player maintains 3 visible hands -- a Top Hand of 3 cards, a Mid Hand of 5 cards and a Bottom Hand of 5 cards -- and a discard pile which is faced down.

In the first round, each player draws 5 cards. Player 1 decides how to allocate the 5 cards between the 3 hands and puts them down on the table. Player 2 does the same.

For each round in the next 4 rounds, players draw 3 cards each. Player 1 then chooses 2 cards to place into his hands and discards the last card face-down. Player 2 follows.

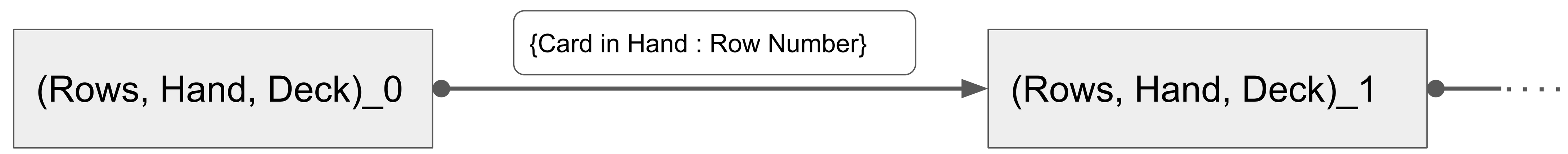
At the end of the rounds, Players count the total points gained by their play according to the scoring rules.

This is a zero-sum game between players over multiple rounds with large amount of adversarial decision-making under uncertainty. We aim to create a competitive AI player for this game.

Problem & Model

Baseline Models: We employed Random Policy and modifications with hard-coded heuristics as baseline models.

MDP w/ Q-learning and Feature Extraction (Solitare): This is represented by the following state-action setup. We preloaded the probability data and trained the weights with epsilon-greedy search and 1000 randomly simulated games.



$$\forall(s,a,r,s'):\ \hat{Q}_{\text{opt}}(s,a;w) \leftarrow (1-\eta)\hat{Q}_{\text{opt}}(s,a;w) + \eta(r+\gamma\hat{V}_{\text{opt}}(s'))$$

$$\text{where}\quad \hat{V}_{\text{opt}}(s') = \max_{a'\in A(s')} \hat{Q}_{\text{opt}}(s',a';w)$$

$$\hat{Q}_{\text{opt}}(s,a;w) = w\cdot\phi(s,a)\quad,\quad \phi_i(s,a)[\text{Hand}] = \mathbf{P}(\text{Row } i \text{ making Hand given (s,a)})$$

Oracle Evaluation Function (Solitare): Instead of using V_opt, we used an evaluation function equal to the highest score of M perfect play simulations of the remaining draws. This is implemented using a CSP solver with MCV-based variable ordering.

Modifying for Adversarial Game Performance: We employed a modified min-max framework. We simulate possible ending hands of the opponent and optimize against an adversarial "oracle play".

Evaluation and Findings

Solitare Models	Bust %	Royalties per Hand	FL %
Baseline	37.4%	0.40 ± 0.06	0.1%
Heuristic Never-Bust	0.3%	1.82 ± 0.11	1.2%
Pure Oracle Play	0%	25.2 ± 0.07	N/A
Q-learning w/ Features	21.1%	0.25 ± 0.04	0.1%
Oracle Eval. Function	15.2%	5.32 ± 0.19	14.7%

We simulated 1000 plays for each trained model and evaluated each model on the bust probability and the average royalties earned when the play did not bust.

Evaluation and Findings (Cont'd)

We modified the evaluation framework for adversarial play. Utilities measure actual game points earned for Policy 1 v. Policy 2.

Policy 1	Policy 2	Utility
Oracle Eval	Heuristic NB	2.31 ± 0.14
Heuristic NB	Baseline	3.78 ± 0.13
Oracle Eval	Adversarial Oracle Eval	0.21 ± 0.16

Both the Q-learning and the Oracle model quickly learned not to bust. However, the Q-learning model did not manage to learn the formation of complex hands and reported lower than expected average royalties. Ultimately, this underperformed.

The greatest challenge was exploring or generalizing across the enormous state space that is the game. We resorted to using "human heuristics" to design generalizing features, but even that was insufficient given the sparsity of "high royalty" hands.

Using a better feature model with a larger training set may be able to improve the results, but we were limited by resources available.

References

Kachushi, an artificial intelligence for the game of Open Face Chinese Poker, Retrieved from:https://tinyurl.com/ycetx35