

ZSY

Leon Lin (leonl)

争上游 (ZhengShangYou, or “Competition Upstream”) is a Chinese card game that is part strategy, part luck. Each player gets about 18 cards that they must get rid of to win, and they get rid of cards by matching patterns. Simplified game rules are on the next page. The game can be played with 2-4 players and one deck of cards, 4-7 players with two decks, but I’ll start out with just trying a two player version.

Like chess and go, there are patterns to be seen that, if accurately modeled, convey a large advantage in game play. The state space is enormous—possibly on the order of the factorial of the number of cards in play. As far as I could tell, this particular game has not been studied before for automation. And, unlike those games, the initial states are random and a very small portion of them can even be unwinnable.

A year and a quarter ago, I was in a 3-person CS 229 team that tackled ZSY. We used a TD-learning algorithm with hand-picked features that played many games against a random and a greedy agent we designed. While it did beat those purely manually designed agents, it could only beat humans about 30% of the time. There was no neural network involved and the hand-picked features didn’t work that well.

Our failure was two-fold. First, the features we chose could not possibly capture the game state. Second, the features were highly correlated (in fact, knowing several of them could sometimes entirely determine the others) which possibly was why the weights kept exploding—we couldn’t figure out a way to deal with the correlations at the time so we just implemented a massive weight decay that meant that little end-game information traveled back to the initial states.

At the most basic, a several layer neural network can be used on the features we extracted. However, there are better neural network architectures that might help with this project. Perhaps a CNN over the raw cards could pick out a pattern or an RNN could be useful for the sequential nature of the game. I am particularly interested in learning more about how Alpha-Go was designed.

In the old project, we built a simulator. It simplified the history of cards played to a mere pile of cards, but I could improve on this by taking the order the cards were played into account. Whichever methods I try, I will be training agents against each other with a simulator. The old project’s learning agent beat the best manually designed agent about 80% of the time. The manually designed agents never beat a human. At the very least, this project’s agent should beat the old one’s. I will run human trials with myself and several people experienced in the game with the ultimate goal of it winning more than 50% of the games. Human training data will be extremely sparse as I will be collecting all of it, so while it may be used for training it will primarily be used for testing. The training and testing distributions, then, will be fundamentally different.

Rules for simplified ZSY:

Two players are dealt 18 cards randomly from a deck of 54 cards (13 per number, 2 jokers). The goal is to get rid of all the cards in ones hand. A coin is flipped to determine who starts the first round.

That player that starts a round has these options to play:

- **Single:** 1 card
- **Double:** 2 cards of the same number
- **Triple:** 3 cards of the same number
- **Bomb:** 4 cards of the same number
- **Chain:** a series of consecutively-numbered cards, for which each 'link' has at least two of that number. For example, 33444, JJQQKK, are valid patterns. 5556677778899 is, but 44566 is not because there's only one five and 7799 is not because it's not consecutive.

The next player must play cards that match the pattern exactly, but are higher. For example, if 777 was played, the next player could follow with 888, 999, QQQ, or so on. If 55666 was played, he could follow with 77888 or JJQQQ. Alternatively, the player can play a "Bomb" over any pattern, and those can only be beaten by higher bombs. Or, the player could pass.

When every player has passed, the last player to play some cards wins the round, and gets to start the next round, setting the new pattern. As soon as a player runs out of cards, that player wins the game.