

Proposal: Learning of a Competitive Strategy in Scheisskopf

Nile Livingston
Rohan Paranjpe
November 19, 2013

1 INTRODUCTION

Scheisskopf is a card game intended for 2 or more players. Despite notable effects of chance in the configuration of the deck, nontrivial competitive strategies exist that require attention to nonlocal constraints. The game is further complicated by the fact that each player possesses imperfect information about the location of cards in the deck. Our goal is to develop an agent that can learn to play competitively against a human opponent in a two-player version of the game.

2 RULES OF SCHEISSKOPF

2.1 DEALING AND PRE-GAME

Cards are dealt from a shuffled standard deck with 4 suits and 13 ranks, a total of 52 cards. Each player is dealt three cards face down (*down cards*), three cards face up (*up cards*), and a three card *hand*. The remaining cards form the *deck*. In the *pre-game* round, the players are allowed to swap cards in their hand with their up cards. The first player to finish swapping may then play a card, upon which all players must cease swapping and normal turn-taking begins.

2.2 TAKING YOUR TURN

A normal turn is as follows: the card(s) most recently played form the top of the *pile*. Let the card on top of the pile be of rank X . For most X (4, 5, 6, 8, 9, J, Q, K, A), the player taking her turn must play at least one card from her hand of rank $Y \geq X$. The exception to this rule is $X = 7$, upon which the player must play card(s) of rank $Y \leq 7$. Wild cards may be played on cards of any rank X . At the end of a turn, the player must draw exactly as many cards from the deck as needed to have 3 in their hand (if the player already has 3 or more cards in her hand, she does not draw). If the player draws card(s) of rank equal to the rank she just played, she may quickly play these before the opponent plays.

If the player cannot play any cards, she must pick up the pile into her hand and her turn is over. On the other hand, if the player can play, she must; one may not elect to pick up the pile. Multiple up cards of the same rank may be played simultaneously. If, at any point, all four cards of a rank are stacked continuously in the pile, the pile is discarded and the player who completed the set is allowed to play again. A player's up cards are unplayable until both her hand and the deck are empty. Similarly, a player's bottom cards are unplayable until both her hand and up cards are exhausted. Down cards must be played one at a time and without looking at the card before playing it.

2.3 ENDGAME

The goal of each player is to be the first to play all of her cards. This means exhausting the deck, one's hand, one's up cards, and *successfully* playing all of one's down cards (i.e. if a down card is played that is not playable on the current up card, the down card was played *unsuccessfully* and the player must pick up the deck into her previously empty hand).

2.4 CARD ORDERING

The rank ordering of the non-wild cards is as follows: $4 < 5 < 6 < 7 < 8 < 9 < J < Q < K < A$.

2.5 WILDS

2's, 3's, and 10's are wild. 2's do nothing special, 3's force the opponent to pick up the pile into her hand (the 3 played is discarded rather than picked up by the opponent), and 10's discard the entire pile. Both 3's and 10's force the opponent to skip their turn.

3 CHALLENGES TO MODELING COMPETITIVE PLAY

3.1 IMPERFECT INFORMATION

At any given point in the game, there are factors that are unknown to each player: the contents and ordering of the deck, the unobserved contents of the opponents hand, and each player's down cards. For human players, this uncertainty often extends to the contents of the discard pile and possibly most of the contents of the opponent's hand due to the limitations of memory. A computer agent thus has the advantage of perfect memory of what it has seen; it may know

the contents of the discard pile, as well as perhaps some of the opponent's hand. Where the computer agent is less naturally talented is in assessing risk (e.g. choosing between playing a non-wild or a 10 on a reasonably large pile, unsure of whether or not the opponent has a 3). Making potentially risky choices in situations with imperfect information necessitates an inferential system, whether explicit (e.g. a probabilistic inference system) or implicit (e.g. policies acquired through Q-learning).

3.2 COMPLEXITY OF STATE SPACE

In most card games, the size of the state space (where each state is represented explicitly as the location of every card) is combinatorially massive (even just the permutations of possible decks is $52! \approx 8^{67}$). In order to pose a tractable Q-learning problem to our agent, we will have to devise an effective featurization procedure that compresses Scheisskopf game states into new, sparser representations that take into account only strategy-relevant features (e.g. number of cards in opponent's hand, number of wild cards in agent's hand, etc.).

4 IMPLEMENTATION

Our implementation of the game environment and our agents will be in Python. We intend to use the Tkinter package to handle all GUI components of the project necessary for human playability. We will maintain version control of our code and related files through a GitHub repository.

5 SCHEDULE

5.1 PHASE I (WEEK 1)

During this phase of the project, we intend to fully implement the game environment. This will involve constructing a simple, functional GUI for the human player as well as implementing a default agent that will select actions randomly. Completion of this phase of the project will allow a human player to play an entire game of realistic Scheisskopf against a randomly-acting opponent.

5.2 PHASE II (WEEK 2)

During this phase, we intend to implement and test both a Q-learner and a reflex agent. The reflex agent will rank actions based on a series of well-defined, reasonably simple heuristics that match our intuitions as amateur-expert players of Scheisskopf. The Q-learner will be trained against this reflex agent. Because of the massive state space problem discussed earlier, we will have to featurize our states in order to pose a tractable Q-learning problem.

5.3 PHASE III(WEEK 3)

During this phase, we hope to improve upon our agent, possibly by exploring alternative learning algorithms, fine tuning the featurization of states, or by incorporating certain heuristics into the learning agent's strategy. It is during this time that we hope to perform assessment of various strategies to show that, in fact, our finalized agent plays respectably.

6 ASSESSMENT OF AGENT PERFORMANCE

Each model will be assessed relative to the performance of a random agent. That is, amateur-expert human players will compete in some number of games against an agent, and the agent's average winning rate will be computed from these trials. It is our expectation that the performance of the random agent should be quite low (hopefully less than 0.5) when playing against a reasonably practiced human player. We hope to develop a system that performs significantly better than the random agent. Ideally, our agent would be able to win against a skilled human player well over half the time. It is important to keep in mind, however, that competitive strategies in *Schesskopf* are not fool-proof; chance deck/dealing configurations can result in games that are very imbalanced (e.g. one player draws significantly more wild/high cards than the other). Due to this notion of chance, we expect that the performance of our agent will have an upper bound; it might be impossible to win some games.