Performance Comparisons

Table 1: Performance comparisons among agents

Win rates(Row v.s Col)	Baseline	RANDOM	GREEDY	MINIMAX
Custom Player(rounds = 10)	70%	100%	100%	75%
Custom Player(rounds = 100)	56.5%	95%	100%	60%

The custom player is based on opening book and minimax. Basically it works as follows: 1) for game states with ply count less than 4, choose an action according to the opening book; 2) for game states with ply count equal or beyond 4, choose an action according to minimax algorithm. The baseline agent is the same as the custom player, except that for opening moves in 1) a random action is chosen instead of one from opening book. How to collect statistics for the opening book is explained in the following Q&A section. Data from 2 million simulations is stored in the opening book.

Experiments in which custom player plays against other agents(Baseline, RANDOM, GREEDY, MINIMAX) are conducted over different number of rounds(10, 100). Customer player shows a better result against Baseline, which is expected because Custom player is an improved version of the baseline where opening moves is selected based on statistics rather than randomness. Customer player shows a high winning rates against RANDOM and GREEDY as these two algorithms sacrifice optimality for computational efficiency. Custom player shows a winning rate slightly over 50% against MINIMAX, as statistics from 2 million simulations on opening moves gives Custom player a slight advantage over MINIMAX with a simple heuristic function.

What is surprising is that Baseline agent performs slightly better than MINIMAX, since Baseline agent's opening moves within 4 plies are based on random selection while the random opening moves of MINIMAX are chose within 2 plies. This suggests that for a complicated problem MINIMAX with simple heuristic functions does not necessarily perform better than pure random selection on opening moves.

Q&A

Q: Describe your process for collecting statistics to build your opening book. How did you choose states to sample? And how did you perform rollouts to determine a winner?

A: In depth level less than 4 pliers from the empty state, randomly choose an action from what are available in each state and move to the resulting state. Repeat the above steps recursively until reaching the maximal level 4 from the empty state.

Beyond this point, we will perform rollouts by randomly selecting a sequence of actions down the game tree until there is a winner in the game. Propagate the results backwards and update the dictionary mapping game state, which is within 4 plier away from the initial empty state, to a dictionary that maps action to its winning counts.

Repeat the process above for many times. In this case, the number of possible states within 4 pliers from empty state could be as many as 9x11x(8^3) = 50688. We want to run simulations to cover as many states as possible and visit each state as many times as possible. After this is done, return an opening book that maps game states to its best action. In experiment, a data set of 2 million simulation is compiled into “data.pickle”

Q: What opening moves does your book suggest are most effective on an empty board for player 1 and what is player 2's best reply?

A: According the opening book stored in “data.pickle”, the best move on an empty board for player 1 is [row(2), col(3)] and the best reply of player 2 is [row[5], col[9].

Note that both row and col index start at zero.