Applying Machine Learning to the Board Game Pylos
Stanislav Fort\textsuperscript{1}, Lucia Gan\textsuperscript{2}, Allen Zhao\textsuperscript{3}
CS 229, Autumn 2016, Stanford University

Pylos

Pylos is two-player zero-sum game where the player that places their ball on the top of the pyramid wins. We chose Pylos for this machine learning application because it is a deterministic and strategic game that has not been explored yet. Most of the difficulty in creating an effective Pylos player lies in its massive state space: several special configurations of balls allow the game to loop infinitely, allow players to take a ball or two from anywhere on the board, and perform other interesting moves. Thus, given the large space and no clear winning strategy, this is a perfect problem for machine learning.

Game representation

The game is represented by a decision tree that includes every possible move that can be made in a game. Since an AI cannot practically search the entire tree of possible moves during a game, we used the minimax algorithm with alpha-beta pruning and gradual deepening.

Support Vector Machine

After playing various agents against each other, we processed the data using an SVM. We culled 3,000 data points generated from 3,000,000 evaluated moves between two Ordered Pruned Minimax Agents, aggregating the agents’ scores and end results (win/draw/loss). This aggregate value was used to find an estimator for the node values of each unique board state encountered during the run. We use a Gaussian kernel to train an SVM regression with a validation set of 100 examples.

Player Simulations

We gathered game data (number of wins/draws/losses) by hardcoding and playing the following 4 agents against each other:

2. Maximizing Agent: chooses move maximizing numerical difference between player ball counts.

Results

We allowed each hardcoded agent to play every other agent (and itself) 100 times, generating the below data. This data shows that Agent 4 was the most effective, so we used it to generate the data for the SVM model.

Our resulting SVM model for static evaluation function was then played against Agent 4 - our best hardcoded agent - for 100 games with different time limits for moves (which affects value estimation). In every case, the trained model won more often than it lost. Thus, we consider our project to create an effective Pylos player (with machine learning) a success.

\textsuperscript{1} sfort1@stanford.edu, \textsuperscript{2} luciagan@stanford.edu, \textsuperscript{3} arzhao@stanford.edu