An Adaptive-Learning Analysis of the Dice Game Hog Rounds

Lucy Longo

August 11, 2011

Overview

- The rules of Hog Rounds
- Adaptive-learning
- Modeling and analyzing the game
- Results and strategy

The Rules

- Played for a fixed number of rounds
- Two or more players play simultaneously
- A die is rolled and its value is added to the round score
- After each die roll, players independently decide to either:
  - Leave and add the current round score to their total score or
  - Stay and hope to gain more points from additional die rolls
- When a 1 is rolled, all players still in the round get no points for that round
- The player with the highest score at the end of the game wins

Adaptive-Learning

- In game theory, a method by which a computer can learn the strategy for a game
- Must be able to distinguish game states
- Random moves/choices are made at first
- Some combination of reward and/or punishment is used when the computer wins or loses, respectively

Modeling the Game

- Used a Python script to simulate game play and implement adaptive-learning
- A 3 or more player game can be simplified to a 2 player game
- Implemented adaptive-learning for both players
  - A player decides whether to stay or leave based on the current round score and the difference between their score and their opponent's score
  - Two matrices per player represent what the player has learned
  - One matrix for when the player is ahead, one for when they are behind

Adaptive-Learning with a Non-Determinate Game

- Non-determinate due to randomness caused by dice
- No perfect winning strategy
- Inevitable noise
- Weight the results of earlier games lower

Areas of Investigation

- Is there a generalizable strategy?
- Similarities/differences between the strategies for games with a different number of rounds
  - 1 round, 2 rounds, 3 rounds, 5 rounds, 10 rounds
- Effects of different initial weights for staying or leaving
  - 1/1 vs 10/10 vs 50/50 vs 100/100

Procedure

- Limited the games to 2 players
- Ran 100 million games each time
- Played games with the various number of rounds for each of the different initial weights
- Selected a base case and ran the program with games of the same number of rounds and same initial weights 20 times

Explanation of Results Figures

- Two figures per player demonstrate what the player learned
  - One for when they are ahead, one for when they are behind
- Vertical axis: score difference
- Horizontal axis: round score
- Black areas indicate times when the player should stay in the round
- White areas indicate when the player should leave the round
- Grey areas indicate indecision

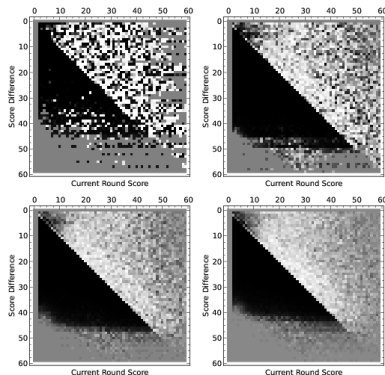# Effects of Different Initial Weights

3 rounds games:
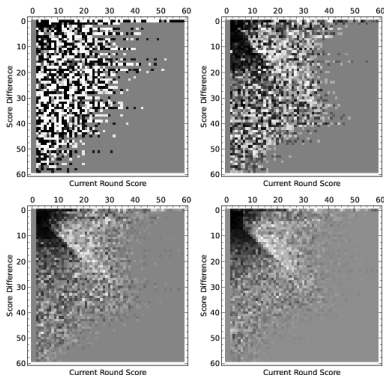
Figure: Ahead

Figure: Behind

Initial weight from top-left to bottom-right: 1/1, 10/10, 50/50, 100/100

Comparison of "Different" Players

- Players play simulatneously, so their strategies should be the same
- Three-round games with an initial weight of 50
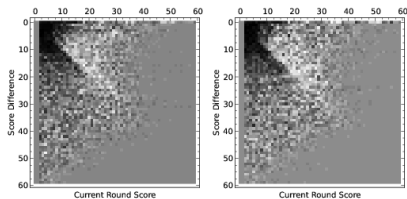- "Player 1" on the left, "player 2" on the right
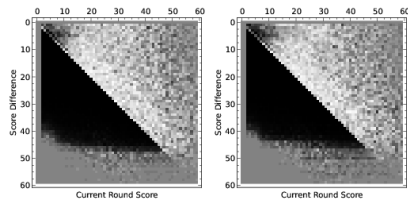

Figure: Ahead

Figure: Behind

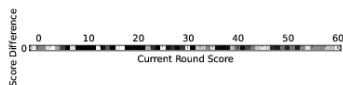# One-Round Games

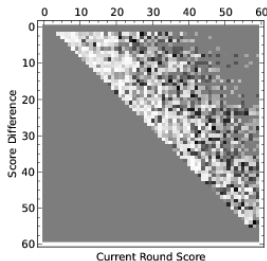Initial weights of 50/50

Figure: Ahead

Figure: Behind

- In a 1 round game, a player can only be ahead if they have already left the single round.
- A player can only be behind by a number of points greater than the current round score since the opponent must necessarily have already left the round

Multiple Round Games

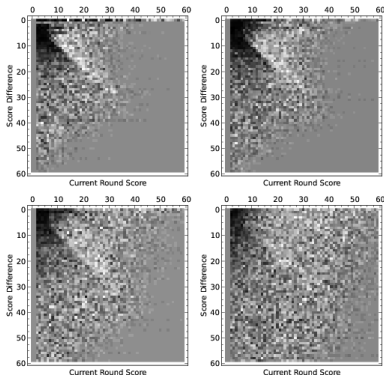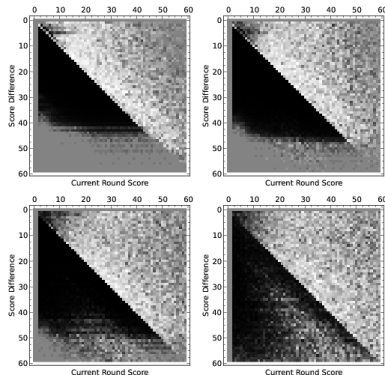Initial weights of 50/50

Figure: Ahead

Figure: Behind

Number of rounds from the top-left to the bottom-right: 2, 3, 5, 10

Base Case

- 3 round games with initial weights of 50/50
- 20 runs of 100 million games each
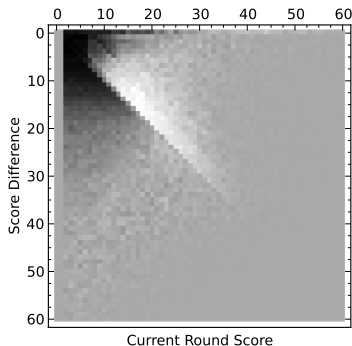- Averaged both players together (40 matrices per figure)
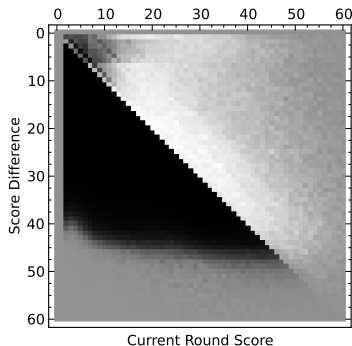

Figure: Ahead

Figure: Behind

Strategy Summary

- If you are winning (ahead):
  - Leave the round after the first roll
    unless you are only ahead by a small margin
- If you are losing (behind):
  - Stay in the round untill the round score equals or exceeds
    the difference between your score and your opponent's
  - If you are behind by only a small margin, you may want to
    stay in the round longer

Effectiveness of this Strategy

- Two players playing randomly against each other:
  - Each win approximately 43% of the time
  - Tie appoximately 13% of the time
- Using the strategy I found against a random player:
  - The non-random player wins approximately 70% of the time
  - Ties about 8% of the time