Effectiveness of Strategies to Play Battleship

Kuncheng Feng
Undergraduate at State University of New York at Oswego

Abstract

Battleship is commonly known as a 2 player guessing game, as at each term each player has to guess an location of enemy ships on an unrevealed board. However the game is not 100% based on guessing, as there are strategies that greatly improve one's chances of winning. In this research I developed symbolic AI to play the game heuristically with defined strategies, and recorded the effectiveness of those strategies. My results indicate that with some simple strategies one can easily dominate over a randomly playing player.

Introduction

The game of Battleship has been out for decades, and is mostly associated with children's board games. Classically the game is played by two players on opposing sides, each having a 10 by 10 grid board that is hidden to the other player, and each board has 5 ships of varying sizes placed on it. The goal of the game is to sink all the ships of the opposing player while keeping own's ships afloat. At each turn each player fires a salve at one of the cells, and the receiving end is supposed to announce whether or not it's a miss, hit, or hit and sink a ship.

At first glance the game is about luck, however after the reveal a hit much valuable information can be derived, as they can be used to infer enemy ship locations and greatly increase the next salvo's hit chance. Even if the salvo resulted in a miss, valuable information still can be derived, as they can be used to infer where the enemy ships are not located at. What is seemingly a guessing game now suddenly involves a lot of strategies, in fact there already exists much serious research that tackles this exact question. In this research I developed a few symbolic AIs that play with strategies, by greatly making use of that feedback information to play the game, as well as recording the effectiveness of those AIs, in particular how likely it is able to win against various opponents, and in addition with other game play statistics.

Background

In order to conduct this research, I turned to developing a program that mimicked two players playing games, with players mostly playing heuristically, that is playing by a set of rules, and have the program report back the results. The reason I used programs to play games instead of manually playing them is because computers are inherently much faster than humans, with proper setup a computer can play hundreds of games or even more. And it's hard to force humans to play a game consistently, and humans subconsciously have preferences over how to play the game. In fact a search behavior done by Hickey and Vegh, where they use financial incentives to encourage participants to quickly find a target cell, they found that humans do not search at purely random.

	Zone	*fe*	*fo*
Quadrants	Upper Left	25	27.1
	Upper Right	25	28.6
	Lower Left	25	23.3
	Lower Right	25	21.0

(Hickey)

The fe is the expected mean of each quadrant being explored, and fo is the actually observed mean of each quadrant being explored. Based on this they found that people are more likely to explore the upper right section of a board and are the least likely to explore the lower right section of a board. Because of this subconscious preference and possibly other more, humans are ruled out of my strategy statistic findings.

For the players in this program, I choose to develop heuristic ones due to the constraints that I face. It takes a lot to train an AI by reinforcement learning, on a quest for reinforcement learning Battleship AI, Tamburro "... train an agent to play battleships in 1,000,000 episodes on a 5x5 board and one cruiser ship" (Tamburro). And found that "The best model was saved at step 900,000. The policy needs about 7 moves on average to complete a game." (Tamburro).

Meanwhile the media popular deep learning AIs are too high of a learning curve to start this project, plus sometimes symbolic AI offers a simple and good enough approach. Although doing a different research, finding photosensitive epilepsy triggers in videos and gifs, Holly Emblem made a case that while 2D Convolutional Neural Network can be trained and detect the triggers to a fantastic degree, a simple heuristic approach can also achieve perfect scores, "with accuracy at 100%, recall at

100% and precision at 67%" (Emblem). The author also gave three advantages of heuristics approach: "Firstly, heuristics are inherently more explainable", "Secondly, heuristics can be implemented quickly", and borrowing words from a Google developer "Finally, as Zinkevich notes if you have very little, or no data, heuristics can be developed from previous experiences such as data from different subject areas, user research and even gut feel" (Emblem). The three reasons fit my situation well: an explainable pattern, quick implementation, no data, plus it is recommended by my professor. Thus I choose the heuristic approach.

Program Description

At the planning stage of my research, I thought of three sub strategies to develop for how to play battleship. First group focus on strategies of exploring an unrevealed board, second group focus on strategies of where to fire after achieving an hit, third group focus on strategic ship placements that allows longer survival. And then lastly I would combine the most effective of the three sub groups to make the ultimate strategy to play the game.

In the first stage, I wished to do exactly like the research done by Bennage, where he tested out the search patterns "to determine if the pattern used to pick points makes a significant difference on the outcome". (Bennage, n.d.). And then I would move on to a slightly more complicated search strategy that depends on the enemy ships that are still afloat. Something like the algorithm tested out by Schwartz where "The algorithm will need to use the pattern for the smallest boat still on the board. … However, when the Patrol Boat is eliminated, the shot pattern will switch to the pattern of the next-smallest ship." (Schwartz).

At the second stage, I wished to develop heuristics on where to fire next after achieving hits. Usually explore around the spot that is hit, but to a varying degree. On the third stage I would explore the effects of ship placement as a strategy. Lastly I would combine the best of the three areas to make an ultimate symbolic AI.

Unfortunately for me, the program takes longer than anticipated to set up, the infrastructure required to mimic a game being played required a bit of development. Most of the tested strategies revolve around what to do after achieving a hit, while other tactics have not been implemented, there is no ultimate strategy resulting in the pursuit of this research.

However I do like to share an "ultimate strategy" found by others. When exploring the unrevealed board, the best strategy is not following any set of predefined patterns, it is actually evaluating the current state of the board to find the cells with highest probability of containing a ship. In a research called "The Linear Theory of Battleship", Alemi stated that "on the whole there is greater probability to

get a hit near the center of the board than near the edges, an especially low probability of getting a hit in the corners.” simply because “there are a lot more ways to lay down a ship such that there is a hit in a center square than there are ways to lay a ship so that it gives a hit in a corner” (Alemi). In a research done by DataGenetics they put this strategy into millions of games, and found that “No game took longer than 73 moves to complete, and approximately one in every million games played with random boards was a perfect game (completed in 17 moves, each a hit and no misses).” and “The medium game length with this algorithm in 42 moves cf. 97 moves with a purely random shooting pattern, and 64 moves with a parity filtered hunt/target algorithm.” (DataGenetics). The probabilistic heuristic approach is certainly the most effective strategy, as on average it takes less than half the moves to finish a game than playing randomly, and still takes significantly less moves than a filtered target algorithm, which is the algorithm I used for my strategies.

Program Performance

The program I developed is written in Common Lisp and is interacted through Command Prompt or terminals. It includes a function that allows the human player to play against the heuristic AIs, this was intended to test out if the game is working properly.

Interactive demo

```
Your markers and your board:
     A   B   C   D   E   F   G   H   I   J          A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 0 |   |   |   |   |   |   |   |   |   |   |    0 | 5 | 5 | 5 | 5 | 5 |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 1 |   |   |   |   |   |   |   |   |   |   |    1 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 2 |   |   |   |   |   |   |   |   |   |   |    2 | 4 | 4 | 4 | 4 |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 3 |   |   |   | o |   |   |   |   |   |   |    3 |   |   | o |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 4 |   |   | o | x | o |   |   |   |   |   |    4 | 3 | 3 | 3 |   |   |   |   | o |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   |   |   |   |   |   |   |   |    5 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 6 |   |   |   |   |   |   |   |   |   |   |    6 | 2 | 2 | 2 |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 7 |   |   |   |   |   |   |   |   |   |   |    7 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 8 |   |   |   |   |   |   |   |   |   |   |    8 |   |   |   | o |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 9 |   |   |   |   |   |   |   |   |   |   |    9 | 1 | 1 |   |   |   |   |   |   |   | o |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
Enemy fired at H, 4
Enter target location: d 5
It's a hit!
```

```
Your markers and your board:
     A   B   C   D   E   F   G   H   I   J          A   B   C   D   E   F   G   H   I   J
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 0 |   |   |   |   |   |   |   |   |   |   |    0 | 5 | 5 | 5 | 5 | x |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 1 |   |   |   |   |   |   |   |   |   |   |    1 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 2 |   |   |   |   |   |   |   |   |   |   |    2 | 4 | 4 | 4 | 4 |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 3 |   |   |   | o |   |   |   |   |   |   |    3 |   |   | o |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 4 |   |   | o | x | o |   |   |   |   |   |    4 | 3 | 3 | 3 |   |   |   |   | o |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 5 |   |   |   | x |   |   |   |   |   |   |    5 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 6 |   |   |   |   |   |   |   |   |   |   |    6 | 2 | 2 | 2 |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 7 |   |   |   |   |   |   |   |   |   |   |    7 |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 8 |   |   |   |   |   |   |   |   |   |   |    8 |   |   |   | o |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
 9 |   |   |   |   |   |   |   |   |   |   |    9 | 1 | 1 |   |   |   |   |   |   |   | o |
   +---+---+---+---+---+---+---+---+---+---+      +---+---+---+---+---+---+---+---+---+---+
Enemy fired at E, 0
Enter target location:
```

However the main function involves little human input, as the goal is to have AIs play against each other and gather the statistics of each. Here is an example of that:

AI vs AI Demo

```
[2]> (getStatistics)
Available AIs:
1 - RANDOMPLAYER
2 - RANDOMPLAYERPLUS
3 - RANDOMPLAYERPLUSPLUS
4 - TIERLISTPLAYER
5 - TIERLISTPLAYERPLUS
6 - TIERLISTPLAYERPLUSPLUS
Enter a corresponding number to choose AI 1: 6
Enter a corresponding number to choose AI 2: 2
Enter the number of iterations: 100
Simulating games ...

50 games played...
100 games played...
Game statistics:
Number of games played: 100
Average number of turns: 54.16759
Player 1 victories: 83
Player 2 victories: 17
Draws: 0
Average hits achieved by player 1: 16.943237
Average hits achieved by player 2: 12.352563


NIL
[3]>
```

In this program, I have developed 6 heuristic AIs to place the game of battleship, each following a slightly different set of rules. The Random Player plays the game randomly and serves as a benchmark for further comparisons. The Random+ will prefer to explore cells next to the ones that are already hit, and the Random++ strive to follow consecutive hits until a miss. Then there are 3 tier list players that separate cells into different exploration priorities, they will avoid the cells that are next to misses, and prefer the cells that are next to hits. The only difference between them is that the Tier+ version will move on to a different set of locations after sinking a ship, and the Tier++ version follows the strategy as the Tier+ version except it will never place ships right next to each other.

I then put them against each other for 1,000 simulated each and gather the statistics. Although it is common to see in other researches that the algorithms are put through millions of games to gather data, and demonstrated visually by Bennage with graphs of distribution to show that “why it is necessary to run through very large numbers of games of Battleship for analysis” (Bennage, 2019). My program simply does not have the performance capabilities to simulate millions of games quickly, it is already taking up minutes to perform 1,000 simulated games. If this research is to be put into serious production, the code needs to be optimized and maybe migrate to a different language for more complex development. As of now, my gathered results are as follows:

Statistics of each player:

Player	Random	Random +	Random ++	Tier	Tier +	Tier ++
Random	R: 46.4 % R: 44.1 % D: 9.5 % Turns: 95.7 R Hits: 16.7 R Hits: 16.6	R: 1.1 % R+: 98.5 % D: 0.4 % Turns: 68.8 R Hits: 12.8 R+Hits: 17.0	R: 8.2 % R++: 89.9 % D: 1.9 % Turns: 68.9 R Hits: 13.0 R++Hits: 16.9	R: 1.7 % T: 97.8 % D: 0.5 % Turns: 71.8 R Hits: 8.99 T Hits: 17.0	R: 1.0 % T+: 98.7 % D: 0.3 % Turns: 60.6 R Hits: 10.7 T+ Hits: 16.9	R: 0.9 % T++: 99.0 % D: 0.1 % Turns: 42.8 R Hits: 9.12 T++ Hits: 17.0
Random +		R+: 47.3 % R+: 51.4 % D: 1.3 % Turns: 61.4 R+ Hits: 15.5 R+ Hits: 15.8	R+ : 65.2 % R++: 33.3 % D: 1.8 % Turns: 66.9 R+ Hits: 16.4 R++Hits: 12.9	R+: 41.2 % T: 56.3 % D: 2.5 % Turns: 59.4 R+ Hits: 14.2 T Hits: 16.2	R+: 31.7 % T+: 66.8 % D: 1.5 % Turns: 45.3 R+ Hits: 14.1 T+ Hits: 15.9	R+ : 15.5 % T++: 83.3 % D: 1.2 % Turns: 60.2 R+ Hits: 11.9 T++Hits: 16.4
Random ++			R++: 50.0 % R++: 48.9 % D: 1.1 % Turns: 56.0 R++Hits: 14.5 R++Hits: 15.9	R++: 26.9 % T : 71.5 % D : 1.6 % Turns: 56.8 R++ Hits: 16.3 T Hits: 15.7	R++: 20.6 % T+: 77.9 % D: 1.5 % Turns: 52.0 R++ Hits: 16.1 T+ Hits: 15.9	R++: 25.3 % T++: 72.2 % D: 2.5 % Turns: 61.8 R++ Hits: 15.0 T++ Hits: 16.5
Tier				T: 48.4 % T: 49.2 % D: 2.4 % Turns: 50.3 T Hits: 14.5 T Hits: 16.6	T: 36.5 % T+: 62.1 % D: 1.4 % Turns: 47.8 T Hits: 13.9 T+ Hits: 16.1	T: 29.4 % T++: 69.0 % D: 1.6 % Turns: 49.2 T Hits: 14.4 T++ Hits: 16.6
Tier +					T+: 49.9 % T+: 48.2 % D: 1.9 % Turns: 51.9 T+ Hits: 15.7 T+ Hits: 13.7	T+: 56.3 % T++: 41.2 % D: 2.5 % Turns: 46.6 T+ Hits: 14.1 T++ Hits: 16.9
Tier ++						T++: 48.7 % T++: 48.7 % D: 2.6 % Turns: 46.6 T++ Hits: 16.8 T++ Hits: 14.7

Conclusion

The conclusion is that the game of Battleship, despite being seen as a guessing game, involves a lot of strategies. As each new implementation of the heuristic AI, it acts more and more similar to how humans play the game. From randomly playing to knowing what to avoid, infer where the next ship locations are, and reflect on what to do when ships sink. And we can see from the statistics that as the machines use more human-like strategies, they achieve victories more often. However the last version that avoided placing ships next to each other, although seen as a human strategy, is actually a detriment to survival chances.

I think what my research measures is slightly different from the ones done by experts, as all of their algorithms are put to test against a blank player that does not make moves, while mine are put against different players and risks of losing. However I do think that a measurement against a non playing player does serve as a better baseline than a random player. Also I like to note that I would have put my AIs to play millions of rounds if it's feasible for my program. If the research were to continue I would migrate the code into a more modern one for further development, and optimize it with more efficient data structures.

Bibliography

Alemi. (n.d.). The Linear Theory of Battleship. The Virtuosi. Retrieved from http://thevirtuosi.blogspot.com/2011/10/linear-theory-of-battleship.html

Bennage, C. (n.d.). In A Game of Battleship, Strategy Influences the Outcome Given You Are Doing It Right. Retrieved from https://www.rpubs.com/cbennage/battleship_optimal_strategy_research

Bennage, C. (2019, October 31). Changes of Distribution with Different Number of Games Played. Retrieved from https://rpubs.com/cbennage/manygames

Emblem, H. (n.d.). A Case for Heuristics: Why Simple Solutions Often Win in Data Science. Towards Data Science. Retrieved from https://www.cs.oswego.edu/~blue/ai_articles/Case_for_Heuristics.pdf

Hickey, A. E. Jr., & Vegh, A. (1961). Battleship: a study of search behavior. Southern Universities Press. Retrieved from https://journals.sagepub.com/doi/pdf/10.2466/pms.1961.13.1.35

Schwartz, A. (2022, April 14). Coding an Intelligent Battleship Agent. Retrieved from https://towardsdatascience.com/coding-an-intelligent-battleship-agent-bf0064a4b319

Tamburro, A. (n.d.). An Artificial Intelligence Learns to Play Battleships. Towards Data Science. Retrieved from https://towardsdatascience.com/an-artificial-intelligence-learns-to-play-battleship-ebd2cf9adb01

Datagenetics. (2011, December 3). Battleship: Advanced Strategy. Retrieved from http://www.datagenetics.com/blog/december32011/