The Maths Behind Battleships

J fraser

December 11, 2014

1 Introduction

Battleships is a strategic two player game, played on a 10x10 grid, in which the player has to eliminate all of the opposition's ships before theirs are destroyed. Different length ships are randomly positioned on the board for both players, and each player takes it in turn to guess a value on the grid which they believe the opposition's ship is located. The base game of battleships can be run from "battleships.py". An example of a game board can be shown in figure 1.

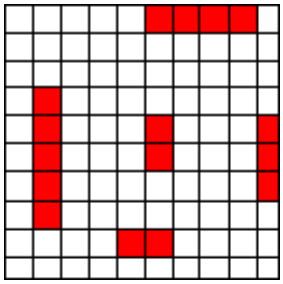

Figure 1: An example of a possible layout of a battleships game board.

2 Probability

The probability of achieving a perfect game in battleships, i.e hitting a ship with every move without missing once, is highly unlikely. In fact if the game has ships of lengths 5, 4, 3, 2 and 2, and the user picks each square completely at random, then the probability is given by:

$$\frac{k!(n-k)!}{n!} = \frac{16!(100-16)!}{100!} = \frac{16!84!}{100!} \approx 7.4302 * 10^{-19} \tag{1}$$

Where n is the number of elements in the sample space (on a 10x10 board there are 100 squares) and k is the number of correct squares (there are 16 squares on the board that contain a battleship).

2.1 Average attempts when random

In my python file labelled "NumberOfAttempts.py" the battleships game has been modified to allow the computer to chose a random coordinate on the grid

until the game has been won. It procedes to do this 100000 times. The number of attempts taken to complete each game are then placed into a list, and an average can be taken. This can be programmed in python and the graph plotted in sage.

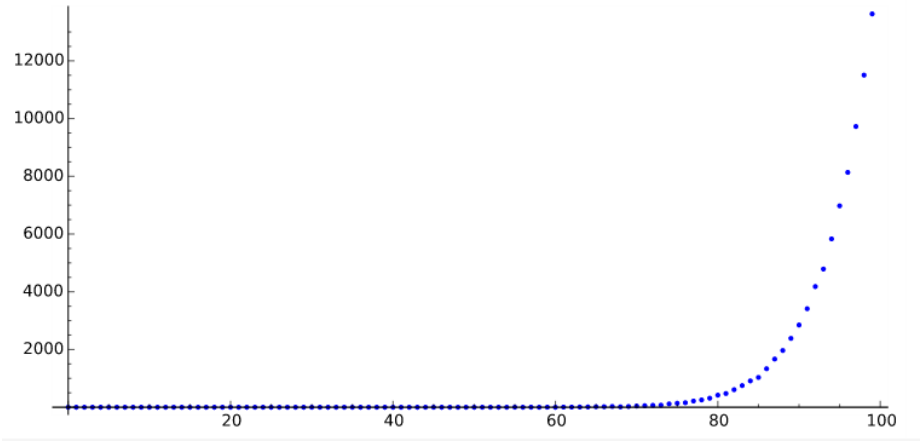

Figure 2: The graph to show the amount of times completed (Y axis) in x amount of attempts (X axis) for the computer player picking at random.

```
listOfAttempts = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 1, 0, 1, 0, 2, 0, 2, 0, 6, 5, 7,
                  10, 17, 18, 28, 13, 26, 42, 52, 65, 72, 116,
                  137, 157, 215, 253, 313, 419, 475, 608, 751,
                  912, 1033, 1334, 1669, 1967, 2386, 2849, 3413,
                  4181, 4787, 5832, 6977, 8136, 9726, 11506,
                  13626]

list_plot(listOfAttempts)
```

The listOfAttempts was obtained from running the code in NumberOfAttempts.py, and setting the number of games played to 100,000. The computer then playes the game 100,000 times, recording the number of attempts taken after each. For example, in this case, the computer completed the game in 100 attempts 13626 times, and in 95 attempts 5832 times.

2.2 A more "intelligent" simulation

To allow a better value for the average length of a battleships game, the computer player would have to act more like a human in certain situations. Such as, when there is a hit on a battleship, a person playing would check the tiles around it to try and take that battleship out of the game completely. This will mean that the computer will be able to complete the game in fewer turns than it would picking the tiles completely at random. This can be programmed in python and the graph plotted in sage.

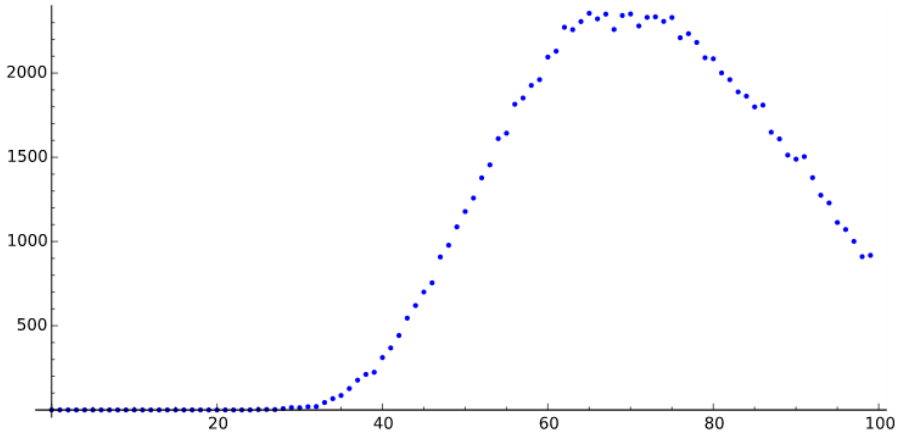

Figure 3: The graph to show the amount of times completed (Y axis) in x amount of attempts (X axis) for the more "intelligent" computer player.

```
listOfAttempts2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                   0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 2, 8, 14, 13,
                   19, 20, 44, 67, 86, 127, 177, 211, 224, 311,
                   368, 442, 545, 620, 700, 755, 908, 978, 1087,
                   1178, 1258, 1378, 1455, 1611, 1643, 1815, 1852,
                   1927, 1961, 2095, 2130, 2272, 2258, 2306, 2355,
                   2322, 2350, 2259, 2342, 2351, 2280, 2331, 2334,
                   2307, 2330, 2210, 2234, 2182, 2091, 2085, 2001,
                   1961, 1888, 1863, 1799, 1810, 1649, 1609, 1513,
                   1489, 1504, 1379, 1275, 1229, 1113, 1071, 1001,
                   910, 918]

list_plot(listOfAttempts2)
```

The listOfAttempts2 was obtained from running the code in "BattleshipsAverage.py", and setting the number of games played to 100,000. The computer then playes the game 100,000 times, recording the number of attempts taken after each. For example, in this case, the computer completed the game in 100 attempts 918 times, and in 95 attempts 1229 times.

3 Averages

3.1 E(x), mode and median

```
#The calculation of the mean using listOfAttempts (the method is
used for listOfAttempts2)
total = 0
counter = 0
for i in listOfAttempts:
    counter += 1
    total += i * counter
float(total/100000)
```

The expected value of x for the random model is 80.04 (2 d.p), the mode is 100 and the median is 95. For the more "intelligent" model , the mean is 71.26 (2 d.p), the mode is 66 and the median is 71

3.2 Skewness

As the mode <median <mean, we can see that the intelligent model has a positive skewness. Despite this, because of how close the mean, median and mode are, it could be modelled as a normal distribution, which could be used to find a good approximation of an event occurring. (1)

4 Conclusion

To conclude, the probability of winning the game of battleships at random is a lot lower than strategically choosing tiles depending on previous attempts. I have also found out that it can also be modelled as a normal distribution.

5 References

1. http://www.mathsisfun.com/data/skewness.html (Online, accessed 11th december 2014)