COSC343: Assignment 1 report

Guangjie Guo (3060110)
August 4, 2023

1 Introduction

In this assignment, what I need to do is to build an agent that shows intelligence to play the mastermind game. The task is about intelligently guessing a solution based on in place and in color scores of the prior guess, so that to get the target by as fewer steps as possible.

2 Main body

2.1 Algorithm

This part will explain my algorithm. Overall, at every step the agent will face a set of candidates that have the equal probability to be the target. For one candidate, we can calculate the possible patterns of in place and in color scores by comparing other candidates in the candidates set, and also get the number of cases in each pattern. If we choose this candidate as the next guess, the average size of the next candidates set will be the weighted average of the numbers of cases in all possible patterns. The weights are the probabilities of those pattern. The candidate with the minimum average score will be chosen as the next guess, which means the smallest candidates set in the next step in statistics. Of course, fewer candidates mean higher probability to guess the target and win the game earlier. I call this approach as miniaverage (It is similar with minimax, but I think my method is better than minimax).

2.2 Implementation

The following is how I implement it in detail.

1. First guess: At this step, all cases in the whole corpus could be the target. Therefore, for 6 colors and 5 code length, it is 7,776. Fully using the miniaverage strategy here will be too time-consuming. In addition, I notice that the optimal choice for the first guess is actually choosing an optimal code pattern. In this situation, only colors' number and the length for each color matter, and the positions and which colors chosen don't matter. For example, AABBC pattern has the same effect in statistics as BABAC or CCAFF has. Therefore, randomly choose dozens candidates, and compare them to the whole possible cases, then return the miniaverage one. In my experiment for 6 colors and 5 code length, these strategy can always return a guess in the optimal pattern at a relatively high speed, which is AABBC type. Furthermore, the same optimal first guess can be used in all the following games. Thus I store the self.firstguess when initiate the MastermindAgent class, and use it unchanged at every game of 100 games.

2. Next guess: First of all, according to the in place and in color scores produced

by comparing the previous guess and the target, find all the cases that can give the same in place and in color scores by comparing the previous guess within the previous candidates set (it will be the whole corpus for the first guess), and use these cases to compose of the current candidates set. Then utilize miniaverage strategy to generate the next guess. However, if the size of the current candidates set is too large, it is still a time-consuming task to fully implement the miniaverage strategy. As a compromise, a threshold will be set. If the size of the set is larger than the threshold, then just like the first guess, a manageable amount of candidates (100 candidates in my experiment) will be randomly chosen, and compare them within the current candidates set, then return the miniaverage one. If the size is smaller than the threshold, just do the miniaverage within the whole candidates set. For example, I set the size threshold as 1000 to satisfy the limitation for running time. Finally, assign the current candidates set to the previous candidates set. Iterate this process until guess the target.

3 Conclusion

In conclusion, using this algorithm, in my mac for 6 colors and 5 code length the average score is around 4.75, and the running time is around 5 mins. It is worth noting that there is a trade-off between the running time and the average score. There are two parameters that can be tuned, threshold and the number of candidates chosen randomly when meeting the threshold. In theory, increasing any of the two parameters can result a better score, but in the cost of longer running time. Therefore, it is an art about choosing the proper settings for obtaining a good score in a period of limited reasonable running time.