Artificial Intelligence 2023/2024

First assignment: Informed and adversarial search strategies

Submission: April 2, 2024

1 Introduction

The material related to this assignment can be found in the books suggested for the course and the slides. Besides, there is quite a lot of material available about adversarial games on the internet (beware to choose wisely).

This work is to be submitted via Moodle. It will be developed mainly during practical classes, but the students are expected to complement it with extra-class work.

Students must be organized in groups of 3.

In this assignment, you will learn how to design and implement a relatively simple program that is capable of playing connect-four with a human.

2 Connect Four: The Game

(**This assignment was originally proposed in the context of an AI course in Harvard. Unfortunately, the site** http://isites.harvard.edu/fs/docs/icb.topic623248.files/Asst3/asst3c.pdf is retired and not anymore available.)

Connect Four is a two-player strategy game similar to tic-tac-toe. It is played using 42 tokens (usually 21 red tokens for one player and 21 black tokens for the other player), and a vertical grid that is 7 columns wide. Each column can hold a maximum of 6 tokens. The two players take turns. A move consists of a player dropping one of his/her tokens into the column of his/her choice. When a token is dropped into a column, it falls until it hits the bottom or the top token in that column. A player wins by creating an arrangement in which at least four of his/her tokens are aligned in a row, column, or diagonal. For example, consider the state of the game shown in Figure 1.

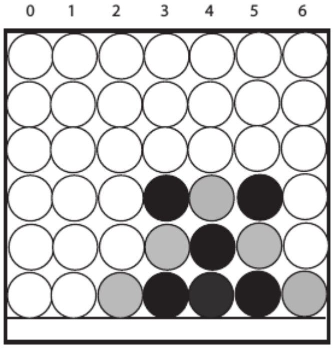

Figure 1: Sample Connect Four Board

The two kinds of tokens here are Black and Gray; it is Gray's turn to move. If Gray drops the token into column 3 or column 5, he/she wins. It is quite possible for a game of Connect Four to end in a draw, i.e., in a state where all 42 tokens have been used, the grid is full, but there are not four tokens of either colour aligned in any direction at any location.

3 Game Interface

The interface for the game can be something like the one shown in Figure 2.

```
-------
-------
-------
-------
X--O-XO
X-OXOXO

It is now X's turn.
Make a move by choosing your coordinates to play.
```

Figure 2: Example of interface for Connect Four

After X makes a move, the computer takes this new board with X move added and uses one of the two algorithms to be implemented. When the computer finishes choosing the best move among the possible ones, it will then exhibit a new board with the computer's move (in this case a new 'O' will show in the position chosen by your program), and wait for the human to play.

4 Work to develop

The goal of this assignment is to implement the A* and the Monte Carlo Tree Search (MCTS) algorithms. A* is an informed but non-adversarial strategy, therefore it does not take into account the fact that an adversary will change the state of the system in the next steps. On the other hand, MCTS does this.

For example, given the configuration of Figure 2, player X can drop in any of the 7 columns, and your program needs to decide which is best.

4.1 A* Algorithm

The speed with which your program plays will depend largely on how efficient the evaluator is. For some games, writing an evaluator is trivial; e.g., for tic-tac-toe, checking whether the middle square on the 3x3 board is occupied by X, O, or neither, and assigning the state a value of +0.5, -0.5, and 0, respectively, is perfectly sufficient. For other games such as chess or Go, however, good evaluators are few and far between. It turns out that the following simple evaluation function for Connect Four (based on R. L. Rivest, Game Tree Searching by Min/Max Approximation, AI 34 [1988], pp. 77-96) performs surprisingly well:

a win by X has a value of +512,

a win by O has a value of -512,

a draw has a value of 0,

otherwise, take all possible straight segments on the grid (defined as a set of four slots in a line horizontal, vertical, or diagonal), evaluate each of them according to the rules below, and return the sum of the values over all segments, plus a move bonus depending on whose turn is to play (+16 for X, -16 for O).

The rules for evaluating segments are as follows:

```
-50 for three Os, no Xs,
-10 for two Os, no Xs,
- 1 for one O, no Xs,
  0 for no tokens, or mixed Xs and Os,
  1 for one X, no Os,
 10 for two Xs, no Os,
 50 for three Xs, no Os.
```

While this evaluation function is guaranteed to work, it is not necessarily the best or the easiest to implement efficiently. Notably, it does not favour quicker wins. That is, if X can win in both 23 and 17 moves, there is no incentive for him to win faster.

4.2 MCTS Algorithm

For the implementation of the MCTS, vary the number of selected child nodes and use the Upper Confidence Bound for Trees (UCT) function to evaluate each branch.

4.3 Submission of the solution

The solution should be delivered in Moodle by April 2, 2024, at 23:59:59;

You should submit the following materials:

- Final code solution, as a notebook;
  - you should document your notebook, explaining your decisions and discussing the results obtained;
- Link for a video summary. This is a team video, but each member should participate in it. This is a very short and to-the-point video (maximum of 5 minutes), summarizing the following (you can use your notebook as background):
  - the problem;
  - your solution;
  - the results.
- Filled auto-evaluation file provided by Professors.

4.4 Presentation of the solution

Students must present their work during the practical class in the week that starts on the 4th of April, 2024 for practical classes on Thursday and 8th of April, 2024 on Monday, using the notebook (you do not need to use any additional documentation/slides).

4.5 Evaluation Criteria

Your work will be evaluated on the following criteria:

- 30% for the informed strategy implementation;
- 30% for the adversarial strategy implementation;
- 30% technical Skills: overall technical evaluation of the solution from a data science point-of-view;
- 10% soft-Skills: essentially - your communication skills;

4.6 Classification

This assignment represents 20% of the grade for the course (4 values). If you implement more than what is requested you can get additionally a maximum of 1 value, that can complement the grade obtained globally for the project's part.

4.7 Some Tips

Be creative in your solution! Think of how you can use certain approaches in an unusual way for example.

- Consider implementation constraints: understand the challenge well and identify any specific constraints regarding this challenge;
- Mention the constraints you are considering for the solution in the notebook;
- Work as a team: The time is very short, so we suggest that you distribute tasks well amongst the team;