Artificial Intelligence 2024/2025

Assignment: Adversarial search strategies and Decision Trees

Deadline Submission: May 18, 2025

1 Introduction

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

This work is to be submitted via Moodle. It will be developed partial 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 available anymore.)

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.

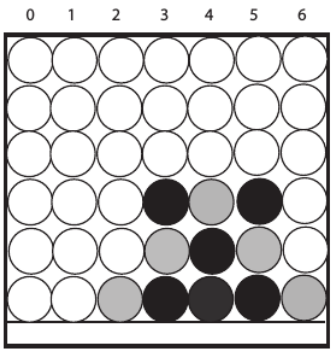

Figure 1: Sample Connect Four Board

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. A game of Connect Four can 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.

Your implementation will need to support the following three game scenarios:

1. human vs. human
2. human vs. computer
3. computer vs. computer (2 different algorithms)

4 Work to develop

The goal of this assignment is to implement the Monte Carlo Tree Search (MCTS) algorithms and a decision tree for each given dataset learned using the ID3 procedure.

4.1 Monte Carlo Tree Search

MCTS is an adversarial search algorithm; therefore, it does take into account the fact that an adversary will change the state of the system in the next steps.

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.

You should implement the MCTS algorithm using the Upper Confidence Bound for Trees (UCT) as the evaluation function for each branch. In addition, you should analyse/explore different numbers of selected children for each node.

4.2 Decision Trees

The goal of this part of the assignment is to write a program that learns a decision tree from a given training dataset using the ID3 procedure.

In addition to learning a decision tree, your program must also be prepared to accept test examples, i.e., after generating your tree, you must be able to apply your tree to new examples and be able to classify them appropriately.

Importat note: You are not be allowed to use `scikit-learn` or other libraries to automatically define and train the decision trees. External libraries will be allowed only to manage and pre-process input data.

4.2.1 Datasets

In this assignment, you will be required to consider 2 different datasets. The first dataset (available on the Moodle platform) can be considered a "warm-up" for testing your decision tree implementation. The first dataset is the **iris dataset**: it contains **numerical** information about plants of three classes: iris setosa, iris virginica and iris versicolor. The attributes are petal length and width and sepal length and width. The task is to learn a decision tree that can tell to which class a plant belongs to, given these attributes.

Important note: The dataset iris contains numerical values. You need to implement a way of discretizing these values in order to minimize the size of your decision tree.

The second dataset is directly related to the game of connect four and it is not available on Moodle, as you will need to generate it. In particular, using your previous implementation of the MCTS algoirthm, you will need to generate a dataset of pairs $(state_i, move_i)$ where $state_i$ refers to the current state of the game and $move_i$ is the corresponding next move suggested by the algorithm. Using this dataset, you need then to train a decision tree using the ID3 procedure that will be able to decide the next move given the current game state.

4.3 Submission of the solution

The solution should be delivered in Moodle by May 18, 2025, at 23:59:59, Lisbon time;

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 the Professors.

4.4 Presentation of the solution

Students must present their work during the practical class in the last 2 weeks as presented in the planning for the AI course. For the presentation, you should use 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 adversarial strategy implementation;
- 30% for the decision trees;
- 30% technical Skills: overall technical evaluation of the solution from a data science point-of-view, including rigor in the performance evaluation of the proposed solution;
- 10% soft-Skills: essentially - your communication skills;

4.6 Classification

This assignment represents 30% of the grade for the course (6 values).

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;