

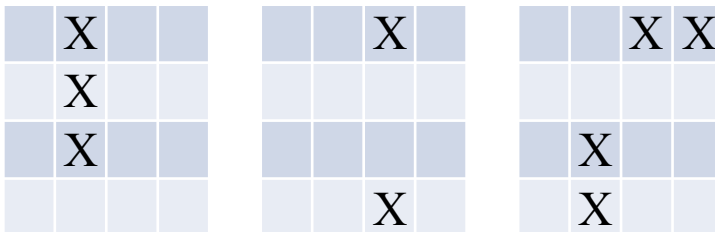
CS560 Fall 2019

Assignment 2: Reinforcement Learning

Due date: November 14, 11:59:59pm.

The goal for this assignment is for you to learn about Reinforcement Learning through the playing of the game of 3D Tic Tac Toe. The board that you will be playing on is 4 by 4. You will need to design a your program to learn how to play the game better by playing a sizable number of games against itself. Each game should be used to “improve” your programs play. Again you may work in groups up to three people.

In its simplest form your program should work to calculate a utility function for each square as it determines which squares are more valuable (more likely to lead to a win) than others. To play a really good game you will want to augment that computation with a something that will handle cases like 3 in a row, 2 in a row or potential double 3 in a row.



Each of these can lead to losses by O and may not be covered by the utility function associated with each square. Clearly you should first focus on the utility function for each square before trying to handle these cases, which can be solved either through learning or by doing a relatively small search.

Your program which should be able to be run directly from a shell, and should take 3 numbers as arguments that describe the number of trials that should be run to learn. The numbers should be increasing in value. After each number of trials (where each trial is the playing of a full game), you should print out the current utility values for each of the squares to 3 significant digits).

So for example:

```
> learn4x4x4 100 500 2000
```

will print out the utility values after 100 trials, after 500 trials and after 2000 trials.

If you are really happy with your program, I invite you to add an interface to your program that will allow it to play a game against an opponent. If there is more than one such program, we will hold a small tournament to see which team has built the best program.

As we all learned during the previous homework, come see me early on if you are having problems getting your program to work. I am happy to help keep you moving in the right direction but can't do that if you don't ask for help.

As we will be running into the end of the semester, I do not expect to extend the deadline for this assignment.

Submission Instructions

You should submit this homework by inviting me to your GitHub repository.

Your submission must include:

1. Your **source code** and a **README** file indicating how to compile and run your code (again the your program must be able to be run directly from the shell). Code should be well commented and it should be easy to see the correspondence between what's in the code and what's in the report.
2. A **report** in PDF format called HW2.pdf with the names of all group members indicated on top. The report should describe your implemented solutions in a way that it should be possible to understand your solution without having to look at your source code. In particular you should describe how you update the utility for each square as the result of each trial, how you have chosen which moves to make during the learning process and how (if at all) you have chosen to handle special cases like those I listed above.
3. All group reports need to include a brief statement of individual contribution, i.e., which group member(s) was/were responsible for which parts of the solution and submitted material.

Late policy: Assignments are due at 11:59pm on the assigned date (e.g. an assignment submitted at 12:01am after the deadline will count as 1 day late). Assignments will be accepted up to 1 week late at a penalty of 10% per day.

Academic integrity: All code and written responses for assignments should be original within your group. To protect the integrity of the course, we will be actively checking for code plagiarism (both from current classmates and the internet).