

Artificial and Computational Intelligence Assignment 2

Problem statement 1 : Gaming

Catch-Up with numbers

Catch-Up has five rules of play:

1. Two players take turns at choosing numbers from the set of natural numbers, $\{1, 2, 3, \dots, n\}$. The highest natural number n in this set is agreed on before the game. Once a number has been chosen, it is deleted from the set and cannot be chosen again.
2. Call the player to make the first move P1, and the second player P2. I will refer to them both as "it". At the outset, P1 chooses one of the n original numbers.
3. Thereafter, P1 and P2 successively choose one or more of the remaining numbers, but each must stop—and turn play over to the other player—when the sum of its choices up to that point equals or just exceeds its opponent's previous sum.
4. The goal of the players is to have a higher sum than an opponent at the end of play—and by as much as possible—or that failing, to have the same sum. If neither of these goals is achievable, a player prefers to lose by as small amount as possible.
5. The game ends when all numbers have been chosen. If one player has a higher sum than the other, that player wins. If not the game ends in a tie.

For example, assume $n=5$, so the numbers at the start are $\{1, 2, 3, 4, 5\}$. Sample choices of players are as below:

P1's first choice:

P1 chooses one of the 5 numbers at random. Each number has the same probability of being chosen.

P2's first choice:

For purposes of illustration, assume that P1 chooses $\{3\}$. Then there are eight possible subsets of the remaining numbers whose sum equals or exceeds 3:

$\{4\}, \{5\}, \{1, 2\}, \{1, 4\}, \{1, 5\}, \{2, 1\}, \{2, 4\}, \{2, 5\}$

P2 chooses one of these possibilities at random. Again, all possibilities have an equal probability of being picked.

In six of these cases, the subsets comprise two numbers, wherein the first number—either 1 or 2—is less than 3, so a second number (the second number in each subset) is needed to make the sum for P2 equal to or greater than 3. After P2 chooses one of the eight subsets at random, either two or three numbers remain.

P1's second choice:

P1 will choose again, and at random, a subset of the remaining numbers such that, when they are added to P1's present score of 3, equals or exceeds P2's score. Depending on the numbers that P1 chooses when it makes a second choice, P2 may or may not have a second choice that equals or exceeds P1's last total.

In summary, when a player randomises, it chooses with equal probability any of the subsets of available numbers that equal or exceed an opponent's last total.

Reference: <https://plus.maths.org/content/game-stymies-ai>

Problem statement 2: Logic

We have been gathering vital data from engineering students as part of a thorough investigation of the effect of 40-100 on people's success. We ask the following extremely objective questions in a totally voluntary survey that all students are obliged to complete:

- Do you attend parties? [party: yes/no]
- Are you smart? [smart: yes/no]
- Are you innovative? [innovative: yes/no]
- Will you be up-to-date to finish your assignments? [assignment: yes/no]
- Do you have a laptop? [laptop: yes/no]
- Does your project succeed between 40-100? [project: yes/no]
- Have you derived your project goal successfully? [goal: yes/no]
- Do you feel happy now? [happy: yes/no]

You can get the comma-separated survey results from following link.

https://drive.google.com/file/d/1qHqCFLfBt1EZ7BQO7h904WqHp3iCskkS/view?usp=s_haring

Evaluations will be based on the following.

1. Use Min-Max algorithm and implement the game in PYTHON (35% marks)
2. Derive the rules from the given decision tree and code as Prolog rules. (35% marks)
3. Interactive implementation. Dynamic inputs-based run of the game with step wise board display and error free game ending. (15% marks)
4. Interactive implementation. Dynamic inputs-based run of the logic expert system with step wise options display and error free recommendation & ending. (15% marks)

Important Note:

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word document for explaining the theory part. Do not include the theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.