SMU - Assignment 1

March 21, 2019

1 Task statement

You are going to learn the rules of a card game from observing plays. You know the following: The game is played with the deck of 8 cards. There are two players and both draw 3 cards. Which player won is determined only be the cards they drew. The rules may be unfairly biased in favor of one of the players. You repeatedly observe what cards are dealt and who won. From this, you are to learn to predict which player will win depending on which cards are dealt.

2 Implementation

Download the zip file from the turial page. You shoud put your implementation in the `agent.py` file, you are free to examine any other files provided, but you may not modify them. The file `assignment1.py` is an executable controller script. The file `agent.py` should contain a class called `Agent` implementing two methods: `receiveSample` and `receiveReward`. The method `receiveSample` receives an input `cards` which has the format: `[(`$card_1$`, `$player_1$`), (`$card_2$`, `$player_2$`), ..., (`$card_6$`, `$player_6$`)]` where $card_i \in \{0, ..., 7\}$ and $player_i \in \{0, 1\}$. The method should return the number of the player (0 or 1) which is predicted to win the game. The controller then calls the `receiveReward` method with the `reward` parameter value being either 0 or -1 indicating whether the answer from `receiveSample` was correct or not. The `receiveReward` should process this feedback and return a boolean value indicating whether another sample is requested or the learning is considered complete.

The learning goal is to be able to give a correct answer to at least 95% of the observed samples in at least 95% runs of the implementation.

3 Scoring

This assignment is worth 15 points maximum. A correct implementation of a learning algorithm is worth 6 points and 4 more points if it does not store observations (except the last observation) in memory during learning, i.e. if you implement the on-line version of the algorithm. The remaining 5 points are awarded if the number of requested samples is within the correct theorethical bounds for the PAC learning parameters above.

4 Technical notes

The implementation of the learning process should obviously use methods taught in this course.

The code will be tested on Python 3.7. You may use any modules from the python standard library. It's recommended to have a look at the `itertools` library. Optionaly, you may also use the `numpy` library achieve better speed. The implementation must run in a resonable time for the theachers to be able to evaluate it.

5 Changes

There were originially 32 cards in the deck and 9 cards for each player. These numbers have been reduced in order to make the running times faster.