COMP47490 Assignment 2

Deadline: Nov 29, 2022. Late submission penalties apply after this date as per details given in the first lecture.

Instructions

Submit your assignment as one Jupyter notebook file (not a DOC/DOCX/ODT/ZIP/PDF file) via the module Brightspace page.

Exam should be completed individually. ***Any evidence of plagiarism will be reported to the CS plagiarism committee and it can result in a Fail grade***.

Please keep the whole code in a single Jupyter notebook. In your notebook, please split the code and explanations into many little cells so it is easy to see and read the results of each step of your solution. Please remember to name your variables and methods with self-explanatory names. Please remember to write comments and where needed, justifications, for the decisions you make and code you write. Your code and analysis is like a story that awaits to be read, make it a nice story please. Aim to keep the notebook clear and concise, with the key code and discussion.

The objective of the first question is to use the ensemble learning functionality to identify the extent to which classification performance can be improved through the combination of multiple models. Experiments will be run on a dataset extracted from US Census data. The data contains 14 attributes including age, race, sex, marital status etc, and the goal is to predict whether the individual earns over $50k per year. For this question, download your dataset file from Brightspace (My Learning -> Assignment2 Datasets). So, if your student number is 12345678, then download census_12345678.csv. When downloading your dataset, please ensure that your student number is correct. **Submissions using an incorrect dataset will receive a 0 grade.**

The objective of the second question is to explore the usage of reinforcement learning to learn strategies for combinatorial tasks.

This is an open-ended assignment -- Feel free to take the exploration and the discussion deeper than what is asked to get bonus points.

Question 1

Using your dataset, perform the tasks below using Python and Scikit-learn.

(a) Clean and prepare the dataset for machine learning analysis. You can do basic feature engineering to make your techniques scalable, but there is no need to go overboard with the dataset cleaning. Carefully consider the evaluation measure(s) that you use for this exercise and justify why you selected the particular evaluation measure(s). [10 marks]

(b) Evaluate the performance of three basic classifiers on your dataset: a decision tree with depth at most 3, a neural network with at most 10 hidden nodes and 1-NN. You can do basic parameter tuning, but there is no need to go overboard. The goal in this step is simply to create better than random classifiers. [5 marks]

(c) Apply ensembles with *bagging* using the three classifiers from Task (b). Investigate the performance of each of these classifiers as the ensemble size increases (e.g., in steps of 2 from 2 to 20 members). Using the best performing ensemble size, investigate how changing the number of instances in the bootstrap samples affects classification performance. [10 marks]

(d) Apply ensembles with *random subspacing* using the three classifiers from Task (b). Investigate the performance of each of these classifiers as the ensemble size increases (e.g., in steps of 2 from 2 to 20 members). Using the best performing ensemble size, investigate how changing the number of features used when applying random subspacing affects classification performance. [10 marks]

(e) Based on the lectures, which set of classifiers is expected to benefit more from bagging techniques than random subspacing and which classifiers benefit more from random subspacing? For your dataset, determine the best ensemble strategy for each of these classifiers. Discuss if this is in line with what you expected. Discuss if there is enough diversity in your ensemble and what else could you have done to improve the performance of your ensemble. [15 marks]

Question 2

You are given a set of 10 balls and 10 bins (numbered 0 to 9). Initially, all the balls are in the bin number 5 and your goal is to ensure that the balls are well distributed across the bins eventually. Ideally, all balls should be in different bins. Train a reinforcement learning (RL) agent that should consider all balls ***in turn*** and for each ball b, decide whether to keep the ball in the current bin $\mathscr{B}_b$, move it to the bin $\mathscr{B}_b - 1$ or move it to the bin $\mathscr{B}_b + 1$. Note that the action to move to bin $\mathscr{B}_b - 1$ is only permissible if $\mathscr{B}_b - 1 \geq 0$ and the action to move it to the bin $\mathscr{B}_b + 1$ is only permissible if $\mathscr{B}_b + 1 \leq 9$.

(a) Carefully decide the features to use in the state representation of your RL agent. Also, carefully decide the reward function for your RL agent and the length of an episode. Note that you can also have contextual features in your state representation, i.e., features that are specific to the ball with the current turn. [10 marks]

(b) Implement how the agent interacts with the environment (i.e., the step function, restart function, init function etc.) [10 marks]

(c) Compare the RL strategies PPO, DQN and A2C for this problem. Are they able to learn a consistent winning policy after (i) 50,000 episodes, (ii) 100,000 episodes and (iii) 200,000

episodes? Which policy learns the strategy quickest? Why do you think this is the case? [10 marks]

(d) For the best RL strategy, start with a neural network architecture consisting of 3 hidden layers of 64 neurons each and gradually decrease the number of neurons and number of layers. What is the minimal architecture that still allows you to successfully learn a winning strategy? For example, can you learn a winning strategy with a single hidden layer of 2 neurons? Why do you think this is the case? [10 marks]

(e) Compare the effect of different state representations and different reward functions on the ability of the RL agent to learn a winning strategy quickly. [10 marks]