

Course project (2023 Spring)

Problem description:

1. Write a python code to create a weighted directed graph with arbitrary number of nodes and edges, for which the edges number can be modified by some parameters.
2. By using dynamic programming, solve the problem of finding a shortest path from any given starting node to any given target node in the weighted directed graph. In other words, you need to find out the optimal policy and optimal trajectory. The solution should still work without re-do dynamic programming when changing the starting node and target node.
3. Solve the above problem by using one of these reinforcement learning techniques, such as MC, SARSA(0), SARSA(λ), Q-learning.
4. Considering value function approximation, solve the above problem.
5. Train a model based on value function approximation such that the obtained solution can work even if you offer a different graph. In other words, the obtained solution can be applied to different weighted directed graphs without re-training. (This is a bonus problem. It is not necessary to complete Sub-problem 5 for your final project. But if you can provide a solution, you will receive bonus marks.)
6. Write a test code to illustrate your solutions obtained from above and make some analysis on the advantages and disadvantages of all these solutions by considering applicability to large-scale graphs, memory used by different solutions, execution time, training time, etc.

Regulation and rules:

1. The course project takes 40 marks for the course (40%).
2. The deadline to submit your project report and code packages is 23:59PM of June 15, 2023. It is a firm deadline (Late submission will receive 0 mark).
3. When completing your course project, you are required to write a project report together with the codes for the project, and hand in before the due time. Base on the project report and the code package, the project will be marked.
4. The project report should be written in English.
5. The project report should be presented in the IEEE conference paper style and suggest to use LaTeX if possible. Refer to the following link <https://www.ieee.org/conferences/publishing/templates.html> for the LaTeX Template, or you can work in Overleaf (an online LaTeX editor). The project report should contain the project title, authors, abstract, keywords, I. Introduction, II. Problem formulation, III. Main results (Algorithms), IV. Experiments (Simulations), V. Conclusion and future problems, and References.

6. Use Jupyter lab for codes and related output as well as comments (markdown units). Hand in a complete code package including the data set, the .ipynb file with detailed description of dependencies, etc., so that the code can be checked and run on another computer without any problem.
7. The project and the codes should not be copied from others. Once it is noticed that the hand-in is copied from others including your classmates or online available work, you will receive 0 mark.
8. Mark criteria
 - (1) Creativity (10 marks): You have to have your own idea to solve the problem studied in the project, especially on how to formulate the problem as an RL problem, and highlight it in your report. Please be aware that the only way TA can understand your new inputs is from the project report. So please make sure that you have provided sufficient evidence to show your novelty and creativity in the project report.
 - (2) Completeness (10 marks): The project should be a self-contained one (not a part of a large project). It should have a clear problem formulation, followed by a complete solution and algorithm, as well as simulation (experiments) results, and concluding discussion.
 - (3) Presentation of project report (10 marks): The report should be well organized and clearly written. The problem under consideration and the developed solution (algorithm) should be clearly described. The simulation (experiment) results should be fully discussed and analyzed. Valuable conclusion should be provided. Any unclear points will get some marks off.
 - (4) Presentation of the codes and codes comments (10 marks): The codes should meet a good Python coding style and easy for reading. Also, the codes should be accompanied with clear and detailed code comments. Any confusion in understanding the codes may lead to some marks off.