CIS667 Semester Project Report Rubric

Submission Instructions

Prepare a single file called report.pdf according to the following guidelines and upload that file to the blackboard website. Only one person per group needs to upload the file, but be sure to include the names and NetIDs of every student in the group. The deadline is 11:59pm on Friday, December 13, 2019, the last day of the semester.

Report Guidelines

Prepare a report describing your project with the following structure and content:

- Front matter: Include a title, names and NetIDs of all students in your group.
- Introduction: Briefly summarize (at most 250 words) the most important content of the report: which game you used, which tree search algorithm, which machine learning model, and the most significant experimental results. At the end, include a link to all of your code on a publicly available website, like GitHub or BitBucket. This code should be sufficient to reproduce all experimental results and figures in your paper. The repository should include a README file explaining your code's dependencies and how to run it. ***If you do not include a link to this repository with code and README, or if your code crashes or cannot be run without reasonable software dependencies and hardware, the rest of the rubric can be overruled and you may receive a zero on this project.***
- Game: Explain the rules of the game you chose in more detail. What is the state of the board or video game screen at each turn, and how did you represent it as a data structure? What are the valid actions at each step? When is the game over and how is the winner/score determined? Include an example with a picture and text description that illustrates a typical step during game-play. Also be sure to explain how you varied the size of the problem instance in your experiments. Did you try different board sizes? Different numbers of pieces or players? Different numbers of cards in a deck? Different numbers of actions available? Include citations for any existing code or libraries you used in implementing your game.
- Tree search: Explain the tree search you used in more detail. Include pseudocode for the tree search algorithm and citations for any sources you used to obtain/understand that pseudocode. What is the typical branching factor of the tree (number of children per node), for different problem instances sizes? What is the typical number of nodes at each depth in the tree? What methods did you use to constrain the size of the search tree? Did you use alpha-beta pruning? Random sub-sampling? Limited depth? If it was random sub-sampling, which sampling distribution did you use? If it was limited depth, how did you evaluate the utility of each leaf node in your tree, if it was not deep enough to be a game-over state?
- Machine learning model: Explain the machine learning model you used in more detail. Include any mathematical formulas that define the model, and citations for any sources you used to implement/understand those formulas. What are the trainable parameters of your model? What update rule did you use to train them? How did you generate training data? How did you convert the data structure for a game state into an array of numbers that the machine learning model could use as input? What are the different model configurations you used in your experiments (e.g., number of neurons in a neural network hidden layer, or number of layers, or learning rate)?

- Baseline opponent: To evaluate the performance of your AI, you need to let it play against a baseline opponent. For example, the baseline could be a simple opponent that chooses actions uniformly at random at each turn. Describe in more detail the baseline opponent that you implemented.
- Experimental results: In this section you will report the performance of your AI when playing against the baseline, both with and without the machine learning component, for a range of problem instance sizes. Each group member is responsible for running the experiments on a different configuration of the machine learning model. There should be one sub-section for each student in the group. For each of these sub-sections, the group member responsible should provide the following information:
    - Which group member is responsible for this sub-section?
    - What was the configuration of the machine learning model that this group member used?
    - You must evaluate your configuration using 5 different problem instance sizes. For each problem instance size, you must use at least 25 complete plays of the game for training at every parameter update, and you must use at least 100 updates during learning. You must plot the "learning curve" of training on each problem instance size, which shows performance on the y-axis and update iteration on the x-axis. So there will be 5 learning curves in your results.
    - After training is done, you must compare the performance of your AI in two modes: using just tree search (no machine learning), and using tree search with machine learning. To evaluate performance in each mode, you should measure both win rate and efficiency. Win rate should be how often your AI wins, when playing against the baseline opponent. Efficiency should be measured by the number of nodes checked during the tree search. You should use at least 100 games at each problem instance size to measure win rate and efficiency. These results should be shown with the appropriate plots (such as histograms).
- Conclusion: Summarize what you did. What were the most significant results? In which configuration and mode did your AI perform the best? What was the most challenging part of this project? If you were to continue working on it, what do you feel is the highest priority for future work, and why?
- Bibliography: Include references for every source you consulted during this project, including research articles, textbooks, websites, and software libraries. You can choose the bibliography format, but it must be consistent and a standard format such as APA, MLA, Chicago, etc. https://scholar.google.com makes it very easy to obtain a properly formatted reference for a research article or textbook in the format of your choice.

Grading Rubric

The report and associated implementation will be worth 90% of your project grade (the other 10% was the proposal). That 90% will be distributed as follows:

Style and formatting (5%): use complete sentences and paragraphs with proper punctuation rather than bullet-point lists, use a spell-checker and grammar-checker, use consistent formatting for your citations and bibliography. ***If you do not know how to format citations and bibliography in a report, ask the instructor or TAs in advance of the deadline.***

Technical content (85%): For each of the required sections, make sure you answer all questions and provide all information requested above. Each missing answer will result in a proportional point deduction. For the experimental results section, each team member will receive their own separate, individualized score. For the other sections, all team members will receive the same overall score.

- Front matter: 1%
- Introduction: 5%
- Game: 10%
- Tree search: 10%
- Machine learning model: 10%
- Baseline opponent: 5%
- Experimental results: 39% (each student gets a separate grade for their own sub-section)
- Conclusion: 5%

Academic integrity: ***If any violation of academic integrity is suspected on your project, it will be reported to the university, and may result in a zero grade on the project.*** It is your responsibility to review and understand the university's academic integrity policy. Violations include, but are not limited to:

- Copying text from the internet into your report without proper attribution, even if you make some modifications to it. Any copied text must be copied exactly verbatim (without modifications), and it must be surrounded by quotation marks, and it must be indented relative to the rest of the text, and immediately before or after the quote, you must state the original author's name and include a citation to the associated reference in your bibliography.
- Copying code from the internet without proper attribution. Any code that you did not author yourself must have its source acknowledged in your bibliography, your README file, and in a comment in the relevant code files themselves.
- Fabricating results in your report. All data-points that you include in any plot or describe in the text must have been generated from actually running your code.