CMPSCI 687 Final Project

Due Tuesday December 12, 2017, 11pm Eastern Time

Overview: For the course project you will conduct a small study that you are free to select. The study should take roughly the same amount of time and effort as a homework assignment, although you are free to do more. The goal of this project is to allow you to investigate something within the RL field that we did not cover in class. You should submit a .zip file containing a .pdf write-up along with your source code. The write-up should be roughly 3 pages and should describe what you did and what results you obtained. If you are unsure whether the project that you are thinking of is sufficient, you may ask during office hours or post on Piazza.

Examples: Below we list some examples of projects that you could do that are roughly the minimum amount of work required to get full credit for the project, and which can give you an idea about what we might expect.

1. Select a problem in your field that can be modeled as an MDP. Describe the problem, how it fits the MDP formulation, and apply any RL method to the problem. For this project the contribution is the new environment, and so you could use existing RL algorithm implementations (like OpenAI Gym). You could then report how well the RL algorithm worked (along with how hyperparameters were optimized), even if the RL methods did not perform well in the end.

2. Select an RL algorithm that was not covered in class and implement it on at least two MDPs. Since the point of this project is to implement the method, you should not use existing code for the RL algorithms (but could use existing code for the environments that you test). Examples of algorithms that we have not covered in class that could be a good fit are INAC from the paper *Model-Free reinforcement learning with continuous action in practice*, NAC-LSTD from the paper *Natural Actor-Critic*, Baird's residual gradient algorithms presented in his paper, *Residual Algorithms: Reinforcement Learning with Function Approximation*, Baird's advantage updating algorithm, presented in his technical report, *Advantage Updating*, PGPE from the paper *Parameter-exploring Policy Gradients*, PI^2-CMA-ES from the paper *Path Integral Policy Improvement with Covariance Matrix Adaptation*, or R-learning (the average-reward counterpart to Q-learning) described in the paper *Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results*. In the write-up for a project of this sort you should include a brief description of the method and what it does, pseudocode for the method, discussion of how you tuned its hyperparameters, and results on at least two MDPs.