CS 5313/7313: Advanced Artificial Intelligence

Solving Markov Decision Processes and
Multi-armed Bandit Problems

World model available

Implement value iteration and modified policy iteration (Chapter 17 of textbook) and evaluate their efficiencies and scale-up properties for (a) the navigation problem introduced in Section 17.1 and (b) a modified version of the Wumpus World (with different pits have different probabilities of killing the player, step cost is either c_1 or c_2, $c_1 > c_2$, depending on whether the player is stepping into a location next to a pit/wumpus or not respectively, reward for picking up gold is G, which you lose if you drop the gold, and is W for returning to starting location with gold, where $W \gg G$). Submit code (you can use any language of your choice) and a write-up in paper form discussing the results with analysis of the performance differences between the algorithms.

World model unknown

Implement Q-learning and SARSA (Chapter 22, page 802–803, of textbook) and evaluate efficiency and scale-up properties for the following navigation problems.

Implement an exploring reinforcement learning agent that uses direct utility estimation. Make two versions: one with a tabular representation and one using the function approximator in Equation 22.9 (page 804) (refer to slides 30-33 of UCBcs188Sp21-lec33.pptx included in the material for Week 7). Compare their performance in three environments:

1. The 4x3 world described in the Chapter 17.
2. A 10x10 world variant with no obstacles and a +1 reward at (10,10).
3. A 10x10 world variant with no obstacles and a +1 reward at (5,5).

Include color-coded policies in your report for the continuous space experiments.

Multi-armed Bandit problems

Implemnt the *UCB algorithm* (see slide 13–14 of the UWaterloo-cs486-lecture22.pdf included in the material for Week 7; one difference: all actions should be tried once before the Repeat loop) to learn to choose the option with the highest expected utility from the provided simulator. Evaluate the effect of decision accuracy on varying the number of actions, A, and the number of samples h, as a multiple, m, of the number of actions. Compare the performance of the UCB algorithm with the ϵ-greedy algorithm (vary ϵ over 0.1, 0.2, and 0.3) in terms of regret (query the simulator for the parameters of each arm and choose the one with the highest mean as the optimal action). Plot the estimated best action by the different algorithms after periods of m action choices.

Submit code and a write-up in paper form discussing the results with analysis of the performance differences between the algorithms.