Introduction to Robot Intelligence (CSCI-UA 480-072) Homework 6

Instructor: Lerrel Pinto

April 20, 2023

Submission Instructions

You must submit solutions to both the theory and coding portion of this homework to be eligible for full credit on this assignment.

Please navigate to the "Assignments" page of the course website (**linked here**) in order to download or copy the coding portion of the assignment.

You are strongly encouraged to typeset your answers to the theory questions below using LaTeX, via the course homework template (**linked here**). You must submit your answers to the coding problems by filling out the provided iPython notebook. We encourage you to use Google Colab to write and test your code.

This problem set is due on **May 5, 2023, 11:59 PM**. When you have completed both portions of the homework, submit them **on the course Gradescope as two separate files, with the coding portion in `.ipynb` format** by the due date. **No other forms of submissions will be accepted. Late submissions will also not be accepted.**

You may not discuss the questions in this problem set with other students.

Theory Questions

Question 1: A-Star Gone Wrong

Dijkstra's algorithm finds the minimal cost path between a source s and target vertex t in some weighted graph $G = (V, E),\ c : E \mapsto \mathbb{R}^+$ by performing a series of *relaxations*.

Each *relaxation* reflects the central "invariant hypothesis" of the algorithm.

This hypothesis can be written as follows: if, for each previously visited node w, $d(w)$ reflects the absolute distance from the source vertex s to w, and for each unvisited node y, $d(y)$ reflects the shortest distance from s to y while traveling **only over previously visited vertices**, then when newly visiting some vertex z such that $z = \arg\min_{v \in unvisited}\{d(v)\}$, we can maintain this property by setting $d(v) = \min\{d(v), d(z) + c(z, v)\}$ for all vertices $v \in V$ such that there is an edge in $(z, v) \in E$.

In the **A-Star** algorithm, we extend Dijkstra's algorithm by introducing a **heuristic function** $h : V \mapsto \mathbb{R}$ which estimates the cost of the cheapest path from each vertex to the target t. We then employ this heuristic function to improve the run-time of the algorithm by performing *relaxations* just as we do in Dijkstra while maintaining heuristic distances $d^*(v) = d(v) + h(v)$ for all vertices $v \in V$ as we perform search over the graph.

However, instead of selecting new vertices to visit during search via the criterion, $z = \arg\min_{v \in unvisited}\{d(v)\}$, we employ the heuristic estimate instead $z = \arg\min_{v \in unvisited}\{d^*(v)\}$.

With a "good" heuristic, A-Star can out-compete Dijkstra's algorithm, requiring no more than a single pass over all vertices in the graph to find the optimal shortest path.

1. The choice of a "bad" heuristic function can **compromise the run-time** of A-Star, requiring more than a single pass over vertices in the graph to produce the correct shortest path! Describe how this might happen. **What property must a heuristic function have in order to compromise the run-time of the algorithm?** You may construct a simple graph example to support your reasoning, if you wish.

2. The choice of a "bad" heuristic function can also **compromise the optimality** of A-Star, resulting in the algorithm returning *sub-optimal* (i.e. not truly) shortest paths. Describe how this might happen. **What property must a heuristic function have to break the optimality of the algorithm?** Again, you may construct a simple graph example to support your reasoning, if you wish.

3. Which of these properties, fast run-time or optimality, do you think is more important in the context of real-world robotics? Explain your answer.

Question 2: Life through Bayes-Colored Glasses

Use Bayes Theorem to compute answers to the following questions. Show your work.

1. It's April 2023 and the absolute probability of a New Yorker currently being infected with COVID-19 is 0.001. The probability of a New Yorker

being infected *given* that they've received an extra booster vaccination in the last 6 months is 0.001, while the probability of a New Yorker having received a recent booster *given* that they are currently sick with COVID-19 is 0.01. Compute the absolute probability that a New Yorker has recently received a booster.

2. A robotics lab is loading up a large number of cheap robot arms from a new vendor. Unfortunately, 10% of their experiments with the arms seem to be failing! A graduate student in the lab estimates that the probability of an experiment failing given that there is a defect in the arm is 50%, and that the probability of an arm having a defect given that the experiment run with it failed is 25%. Compute the absolute probability of a defect.