

Machine Learning Engineer Nanodegree

Capstone Proposal - Robot Motion Planning

Tien Pham

July 4, 2018

Plot and Navigate a Virtual Maze

Domain Background

This project was inspired from Micromouse competitions originated in the 1970s, wherein a robot mouse is given the task of running multiple times in a given maze and plotting the best path from a corner of the maze to its center. The robot mouse discovers the maze during the first run and using the planned best path it has previously learned to reach the center of the maze in subsequent runs.

In this project, a simplified model of the world is provided along with specifications for the maze and robot. The goal of this project is to obtain the fastest times possible in a series of test mazes by controlling the navigation of a virtual robot in a virtual maze.

Problem Statement

On any given maze, the robot must complete two runs. The starting point and the goal room are "known" to the mouse before it starts. The starting point is always in a left corner and is always surrounded by three walls. The goal room is four center blocks of the maze. In the first run, the robot is allowed to freely roam, explore, and analyze the maze to determine the best path plans to reach the center of the maze. To finish the first run, it must enter the goal room during its exploration. After entering the goal room, the robot is free to continue exploring the maze and may choose to end its exploration at any time. The robot is then moved back to the starting position and orientation for its second run. Its objective now is to go from the start position to the goal room in the fastest time possible. The robot's score for the maze is equal to the number of time steps required to execute the second run, plus one-thirtieth the number of time steps needed to execute the first run.

A maximum of one thousand time steps is allowed to complete both runs for a single maze. On each time step of the simulation, the robot may choose to rotate clockwise or counterclockwise ninety degrees, then move forwards or backwards a distance of up to three units.

Datasets and Inputs

The information of mazes is provided in a text file. On the first line of the text file is a number describing the number of squares on each dimension of the maze n , n is even and in a range $[12, 16]$. On the following n lines, there will be n comma-delimited numbers describing which edges of the square are open to movement. Each number represents a four-bit binary that corresponds four-side, in this order: up, right, down, left. A bit value of 1 if there is no wall, and a bit value of 0 if an edge is closed (walled).

The robot has three obstacle sensors mounted at the front, right and left which detect the number of open squares in the direction of the sensor. These sensors detect the number of open squares in the direction of the corresponding sensor, then store in the form of a list of three numbers.

A 2-digit pair will define the location of the robot mouse as the coordinate $[x, y]$. The robot decides next location with two values indicating the robot's rotation and movement on that timestep. Rotation is an integer taking one of three values: -90, 90, or 0, indicating a counterclockwise, clockwise, or no rotation. Movement is also an integer in the range $[-3, 3]$.

Solution Statement

Each trial run tries to solve a different problem. During the first trial, the main problem to solve is a detailed knowledge of the maze layout and possible paths reaching to the goal. The robot will explore the maze in the first run where it will learn and map the structure of the maze, and find all possible paths to the goal. The target solution in the second run would be reaching the goal following an optimal path in the fastest time or precisely in a minimum number of steps.

The number of total movements taken along with the whole time of the maze navigation is continuously assessed. After the exploration trial run, the knowledge of the number of steps needed to reach the goal for all possible paths should be stored, and the chosen route after optimization round should be the shortest path with the fewest steps. We can perform multiple runs to test if our robot has learned the optimal policy and the path for each run should remain the same if it is indeed the best path to the goal.

Benchmark Model

(approximately 1-2 paragraphs)

In this section, provide the details for a benchmark model or result that relates to the domain, problem statement, and intended solution. Ideally, the benchmark model or result contextualizes existing methods or known information in the domain and problem given, which could then be

objectively compared to the solution. Describe how the benchmark model or result is measurable (can be measured by some metric and clearly observed) with thorough detail.

Evaluation Metrics

(approx. 1-2 paragraphs)

In this section, propose at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metric(s) you propose should be appropriate given the context of the data, the problem statement, and the intended solution. Describe how the evaluation metric(s) are derived and provide an example of their mathematical representations (if applicable). Complex evaluation metrics should be clearly defined and quantifiable (can be expressed in mathematical or logical terms).

Project Design

(approx. 1 page)

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

Before submitting your proposal, ask yourself. . .

- Does the proposal you have written follow a well-organized structure similar to that of the project template?
- Is each section (particularly Solution Statement and Project Design) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your proposal?
- Have you properly proofread your proposal to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?

