Machine Learning Engineer Nanodegree

Capstone Proposal

Kenneth Mora
June 18th, 2017

Proposal

Domain Background

This project is inspirated in Micromouse competition that it is an event where a small robot mice solves a 16x16 maze.[1] In this competition the robot mouse must navigate inside a maze from a corner until the center. The first time, the idea is to explore the maze after to reach the goal and in the next try the mouse will follow the shortest path towards the goal using the learned previously.
This problem is interesting in the motion planning area because allow to explore how a robot autonomous can search and reach a goal in an optimal way. There are many paper related with this competition for example there is one explaining how to do a diagonal maze solver for micromouse [2]. This is relevant because commonly mouse moves following straight lines but it would be better moves in diagonal because the path would be shorter.

Problem Statement

This problem is about a robot mouse navigating a maze with the goal to reach the center following the best path. The first run the mouse explores the maze for a maximum of one thousand steps creating a map. Always the robot starts from the corner. In the next run it tries to find the shortest path for reach the goal in the center in the less number of steps using the learned in the first run.

Datasets and Inputs

The maze is drawn as a grid of squares with size NxN, N is even. Mazes are provided as text file, each line in the file represent a row of the maze containing a list of numbers. Each number depicts a square indicating how are its edges because it can be either closed (walled) or open (no wall) in any sides, for example the square with the number 10 has a wall on top and bottom side but is open in the left and right side [3]. There are 3 mazes with the following sizes 12x12, 14x14 and 16x16 and they are completely closed in its outer edge hence the mouse can't leave.
The virtual mouse starts the first run of each maze without knowing anything about the maze because the idea is it learns. In each step the virtual robot checks its three sensor (forward, left, right) to know if there is obstacle to advance and it can rotate left or right or continues in the same direction depending on its readings too.

Solution Statement

In the first run, the idea is to test differents algorithms to map the maze exploring with the virtual mouse. The algorithms would be: move randomly, move randomly with dead end path detection, counting areas visited and adding heuristic values measuring the distance to the goal. In the second run the idea is to use the A* algorithm to find the shortest path until the center using the mapped area of the first run.

Benchmark Model

The benchmark is on how to map the maze. For the first run is to take the mouse movement randomly for having a base metric then it must be compared with some improvements like detecting dead end path so the robot avoids to take a closed path, adding a counter of points visited to avoid repeat of the same path many times and adding heuristics values to get distances to all locations to the center (goal) in the maze to choose the best (shortest) path.

Evaluation Metrics

In the first run the algorithms would be compared by checking if it reached the goal in the minimum number of moves allowed, taking note of the numbers of movements necessary to reach the goal and the coverage. In the second run the A* search must be measured by the path length and number of moves. 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 required to execute the first run. A maximum of one thousand time steps are allotted to complete both runs for a single maze.

Project Design

The starter code has functions made to load maze from files, read its structure, visualize it and it has a sensor validation of the possible direction and rotation that the mouse could take. Therefore the first thing to do is create the basic random move for the virtual robot in the first run and a structure to store the path mapped on it. Here it's important to set the initial position for the robot and create a detector for the center so to know when it arrived to the goal. After that it is necessary to develop the A* algorithm to use the structure stored in the second run so it's possible to get the firsts complete scores of a solution.
Then the idea is to improve the mapping in the first run to be more efficient and smart the run, 3 improvements were proposed and it will be implemented in this order, first the end path detector, then the counter of area visited and finally the heuristic values. It's important to save the results to see how improved and then to compare between them. The last thing to do is create the report showing the results and summarizing the project with all work done.

References

[1] https://en.wikipedia.org/wiki/Micromouse
[2] http://ieeexplore.ieee.org/document/6852576/
[3] https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub