Udacity Machine Learning Nanodegree Capstone Project Proposal

Intelligent Mouse

Lei Mao

3/1/2017

Domain Background

Micromouse is a contest where small robot mice (micromouse) solve a maze (**Figure 1**) [1]. It is very popular in US, UK and Japan among the young juniors. In Micromouse contest, the players are going to test their micromouse to solve the maze. To win this contest, a good combination of artificial intelligence and robotics is required.

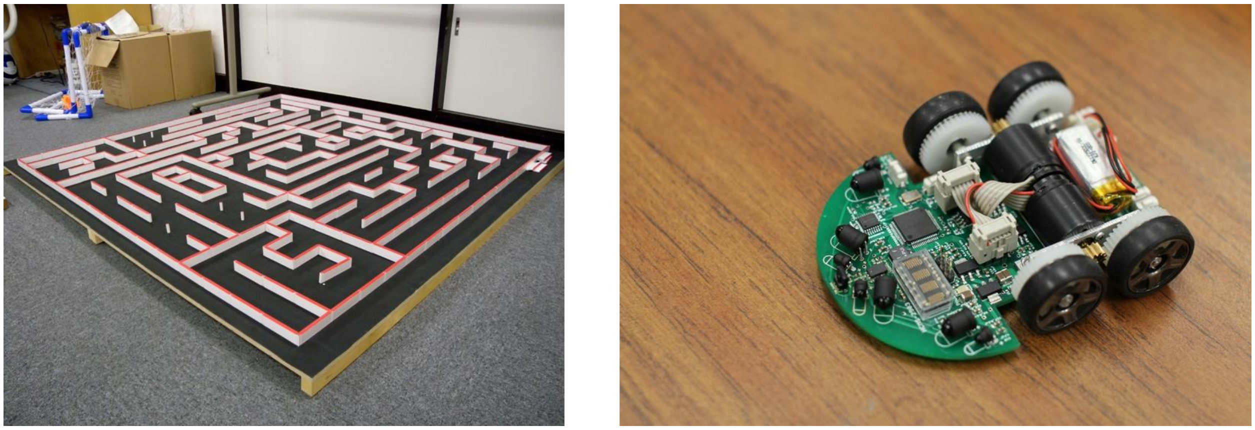

Figure 1. Example of maze (left) and robot mouse (right) in Micromouse contest.

Extended from Micromouse contest, the techniques used in micromouse could also be employed in real-world applications, such as map navigation and outerspace exploration. Designing such intelligent system that helps robots making optimal decisions based on the information collected from the environment would be an extremely fancy thing to do.

Problem Statement

Built on Micromouse contest, “Plot and Navigate a Virtual Maze” is one of the Machine Learning Nanodegree capstone project candidates proposed by Udacity [2]. However, I found that the original problem statement and assumptions oversimplified the contest, therefore the application of micromouse in real-world problems might become limited. I would like to generalize the problem a little bit so that the micromouse developed is more adaptable and functional.

The mission of this project is to design an intelligent micromouse that can explore maze of any kind of size and shape efficiently, and make optimal decisions going from any starting point to any destinations designated in the maze. The micromouse shall not know any information about the maze beforehand, such as the size and shape of the maze, and the location of its starting point and destination. It needs to collect information using its sensors to learn the maze throughout the game.

The virtual intelligent micrmouse will be tested in virtual mazes. It has four obstacle sensors and one destination sensor. The four obstacle sensors return the distances to the obstacles in the front, left side, right side and back, of the micromouse. The destination sensor informs the micromouse when it reaches the destination, which is analogous to mouse finding food in the maze. In each time step, the micromouse collects information from the sensors, chooses moving direction, and moves forward a distance from 0 to 3 units. If the micromouse hits the wall, it stays where it is. After each such movement, one time step has passed and the next time step begins.

The micromouse is allowed to have two runs in the maze. In the first run, starting from the location designated, the micromouse explore the maze trying to find the destination, and return to the starting point. In the second run, the micromouse tries to reach the destination as fast as possible. In both runs, the number of time steps is counted and will be used for score calculation.

Datasets and Inputs

The inputs for the micromouse are only limited to the information returned from the four obstacle sensors and the destination sensor. To simulate micromouse, virtual mazes have to be prepared. The inputs for the program, but not micromouse, is a file containing the wall information of the maze. The coordinates of starting point and destination point in the maze also have to be designated by people. It should be emphasized again that all these maze information are not available to the micromouse beforehand.

Solution Statement

There are two runs during the contest. In the first run, the micromouse needs to collect maze information as much as possible. In the second run, with the maze information learned, the micromouse calculates the optimal path to the destination it found it the first run and the corresponding action series that minimizes the number of time steps. So the micromouse requires an efficient exploration strategy in the first run, and a maze solver which gives optimal solutions in the second run.

In the first run, in order to maximize the exploration efficiency, the exploration strategy of the micromouse has to be rational and the randomness has to be minimized. A rational solution to the exploration problem is that the micrmouse always tries to visit the closest unvisited maze square costing minimum time steps. This rationale can be justified by Markov Decision Process and solved by Bellman Equation.

In the second run, given the maze information learned from the first run, the micromouse should be able to design a path and corresponding action series that cost minimum time steps. This is also a Markov Decision Process and can be solved by Bellman Equation.

Benchmark Model

I would consider the Markov Decision Process model introduced in the Reinforcement Learning courses of Udacity Machine Learning Nanodegree as the benchmark model for this project. In that Markov Decision Process example, people have to find the shortest path to the destinations on square grids, which is extremely analogous to our maze problem. The Markov Decision Process can be solved by value iteration for Bellman Equation.

Evaluation Metrics

To assess the efficiency of micromouse, I would like to use a scoring function that is slightly different to the one used in the original "Plot and Navigate a Virtual Maze" project proposed by Udacity. The final score 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 both runs. For examples, if it takes 100 time steps to explore the maze and return, and 20 time steps to reach the destination, the score is 20 + (100 + 20) / 30 = 24. In this way, the scoring function is closer to the one used in real Micromouse contest [3].

Project Design

In the first run, the micromouse collects information from sensors, and constructs or updates the maze map in its memory. With this maze map, the micromouse will select a closest unvisited square that costs minimum time steps to reach as the short-term destination during exploration. Once the short-term destination is determined, the micromouse will calculate the optimal action series to reach that destination. The selection of short-term destination and the calculation of action series are basically Markov Decision Process problems, and can be solved by Bellman Equation. This process is repeated until the micromouse finds the destination.

In the second run, with the sufficient maze information learned during the first run, the micromouse can easily figure out the optimal path that costs minimum time steps, and calculate the optimal action series to reach the destination using Bellman Equation.

Reference

[1] https://en.wikipedia.org/wiki/Micromouse

[2] https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub

[3] http://micromouseusa.com/wp-content/uploads/2013/10/APEC-Rules.pdf