Robot Motion Planning Capstone Project

Sen Lyu

January 14, 2016

Domain Background

This is a project takes inspiration from Micromouse competition, which is an event that a small robot mice solves a certain size maze automatedly. The competition starts in late 1970s and becomes more and more popular. I am doing this project to give an algorithm for the robot to quickly learn how to find the shortest way to find the solution and costs the least time.

For me, I like the Artificial intelligence. While learning in the machine learning class, I like to make the computer to learn some different stuffs, for example, normal data, images, natural languages, and now the map. The things get more and more abstract, and more and more fun. And this project could be my first try on Artificial intelligence area.

Problem Statement

The problem is that there will be a maze with a certain size of cells. Each cell may have walls in each side, the wall will block the mice to get through. Each cell could have at most 4 walls, or no walls, except for the outmost layer, which will have walls in the outside, so the mice won't get out.

The mice will start from the left bottom cell of the maze. The mice will have two chance to get to the goal room and come back to the start point. The goal room will be a 2*2 square in the center of the maze, just enter this area will be enough. The first time is for the mice to learn the map, and the mice must get into the goal room. And the second time is for the mice to get to the center as fast as possible. The total of one thirties of the time cost of the first chance and the time cost of the second time will be the total time cost of an algorithm. And this project is to find the time as lower as possible.

Datasets and Inputs

The inputs will be simple. The algorithm only needs the information of the maze. The maze will be input as a n*n square, each cell will have a number in it. The number will be in the set of [0,15], each number represents a four-bit number that have a bit value of 0 if this side have a wall, 1 if no wall, in the sequence of up, right, bottom, left.

Solution Statement

My solution to this problem has three steps. Step one, using random walking to find the way to the goal. Step two, check whether the shortest road we find now is the shortest one, if it is not, then keep random walking and check again until it finds the shortest way. Step three, go and back in the shortest way.

Benchmark Model

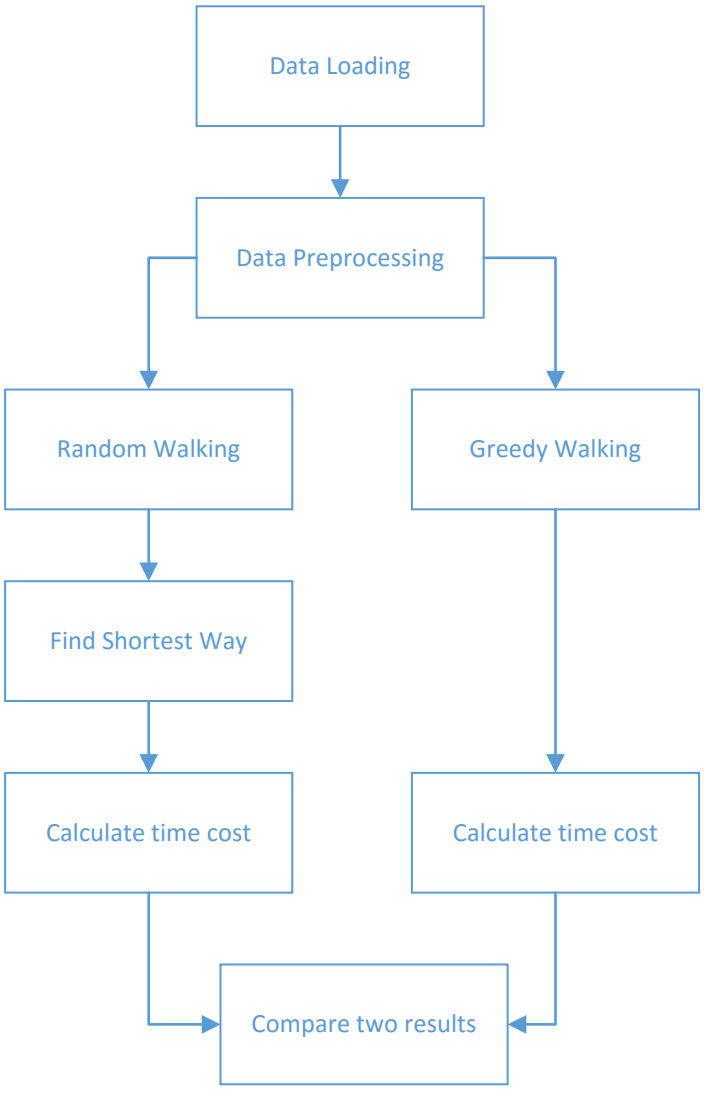
The Benchmark of the model is using greedy algorithm. The mice will choose to go to the direction according to the coordinate of the cell and the goal room, and the mice will try not to go back unless it has to. Once the mice get to the point it will come back right away and use this way as the shortest way.

Evaluation Metrics

The evaluation of the model is simple, we just focus on the time cost. If a model has a less time cost, it will be the better one. And if we assume the first step time cost to be C_1, the second step time cost to be C_2, the total time cost function will be:

$$\mathrm{TC} = \frac{1}{30} C_1 + C_2$$

Project Design



Data preprocessing

Need to change the number in each cell to the four possible actions, which are up, right, bottom, left.

Random walking

From the actions of the cell to get the action which the mice could take in this cell, and random choose one, then proceed.

Greedy walking

Assume the mice is in the cell (x,y), and in the next action it will be in the cell (x',y'). The center of the goal room is (6.5,6.5). From the action of the cell to get action which the mice could take in this cell, then find the action will minimize |x'-6.5|+|y'-6.5|.

Find the shortest way

When the mice get to the goal room, it will have a temporary shortest way from the start to the goal room. And all the cells in the map must be in one of this two situations, was passed by mice or was not passed by the mice. Then assume that all the cells which were not passed by mice to be wall-less, which means it can be reached by any direction from the other cells. Then find the shortest way again. If this time the shortest way was connected by the cells which passed by the mice, then this is the shortest way indeed, else the mice must keep walking and do it again when it find a new unpassed cell.