

CSE 512 – Winter 2018 – Homework Assignment 1

Instructor: Dr. Kerstin Voigt

Total number of points: 20.

In this assignment, you will complete a program that will **improve the dotbot** of our recent lectures and lab1. In particular, you will upgrade the dotbot program so that it will be able to extract knowledge from its specific “positive experience” that it has collected while learning to circle a given wall area.

Start with a copy of file `dotbot_learnHW1.py` and locate the section in the middle that asks for your completion. **Your code completions** are to process the information -- associations between “from”-coordinates with [“to”-coordinates, (...sensors s1 to s8 ...)] – and extract and compile from it a new data structure (“what_i_know”) which associates sensor readings with “good” next moves. Two listings last two pages at the end of this document are meant to give you some insight into this task.

When you have completed the **three member functions** correctly and load and run the program, you will set up a wall and place a robot for one long round of learning; after this round of learning, the user will be asked to four times in succession draw another wall, reset the robot, and restart the dotbot action. *Will the prior learning enable the dotbot to circle other walls as well? To what degree was it able to generalize from its specific positive experience?*

Experiment with four new walls in the following manner: (1) one wall with similar pattern, but of a larger size; (2) one wall with a pattern that is similar to the original, but somewhat different; (3) one very simple wall; (4) one wall that is more challenging than walls (1)-(3).

Repeat the experiment, but this time, let the initial “learning wall” be fairly complex (but no “tight spots”). Then test again with walls of types (1) to (4).

In order to receive credit for this assignment, submit: (1) a hardcopy of your `dotbot_learnHW1.py` file, (2) if possible, a video that captures the dotbot behavior in both experiments; if it is difficult to produce a video, be prepared to demo your program in the next lab session, (3) A written discussion (1/2 to 1 page) of the results/observations from your experiments; comment on both the upsides and downsides.

On the **due date, Jan 29, 2018**, submit your hardcopy with code and written discussion at the beginning of the lecture and make sure that your video is emailed by midnight of that day to both kvoigt@csusb.edu and wang303@coyote.csusb.edu. Otherwise, be ready to demo your program in the lab following the lecture. Submissions by teams of two students are permissible. Individual work is not discouraged 😊

Details for the dotbot learning program. To provide insight for HW1.

The data structure underlying this listing is already built and maintained by program Dotbot_learnHW1.py.

```
for xy in mybot.pos_exp_sensors.keys():  
    print "%s --> %s" % (xy, mybot.pos_exp_sensors[xy])
```

```
(220, 220) --> [(220, 200), (1, 0, 0, 0, 0, 0, 1, 1)]  
(140, 160) --> [(160, 160), (0, 1, 1, 0, 0, 0, 0, 0)]  
(180, 180) --> [(180, 200), (0, 0, 1, 1, 1, 0, 0, 0)]  
(220, 200) --> [(220, 180), (1, 0, 0, 0, 0, 0, 1, 1)]  
(120, 120) --> [(120, 140), (0, 0, 0, 0, 1, 0, 0, 0)]  
(200, 260) --> [(220, 260), (1, 1, 0, 0, 0, 0, 0, 0)]  
(120, 160) --> [(140, 160), (0, 0, 1, 0, 0, 0, 0, 0)]  
(160, 160) --> [(180, 160), (1, 1, 1, 0, 0, 0, 0, 0)]  
(180, 260) --> [(200, 260), (1, 1, 1, 0, 0, 0, 0, 0)]  
(180, 120) --> [(160, 120), (0, 0, 0, 0, 1, 1, 1, 0)]  
(160, 120) --> [(140, 120), (0, 0, 0, 0, 1, 1, 1, 0)]  
(180, 160) --> [(180, 180), (1, 1, 1, 1, 1, 0, 0, 0)]  
(140, 260) --> [(160, 260), (0, 0, 1, 0, 0, 0, 0, 0)]  
(160, 200) --> [(140, 200), (0, 0, 0, 0, 1, 1, 0, 0)]  
(220, 120) --> [(200, 120), (0, 0, 0, 0, 0, 1, 1, 0)]  
(220, 260) --> [(220, 240), (1, 0, 0, 0, 0, 0, 0, 0)]  
(200, 120) --> [(180, 120), (0, 0, 0, 0, 1, 1, 1, 0)]  
(160, 260) --> [(180, 260), (0, 1, 1, 0, 0, 0, 0, 0)]  
(240, 180) --> [(240, 160), (1, 0, 0, 0, 0, 0, 0, 0)]  
(240, 160) --> [(240, 140), (1, 0, 0, 0, 0, 0, 0, 1)]  
(120, 140) --> [(120, 160), (0, 0, 0, 1, 0, 0, 0, 0)]  
(140, 200) --> [(140, 220), (0, 0, 0, 0, 1, 0, 0, 0)]  
(240, 120) --> [(220, 120), (0, 0, 0, 0, 0, 0, 1, 0)]  
(140, 240) --> [(140, 260), (0, 0, 1, 1, 0, 0, 0, 0)]  
(140, 220) --> [(140, 240), (0, 0, 0, 1, 1, 0, 0, 0)]  
(220, 240) --> [(220, 220), (1, 0, 0, 0, 0, 0, 0, 1)]  
(220, 180) --> [(240, 180), (1, 1, 0, 0, 0, 0, 1, 1)]  
(180, 200) --> [(160, 200), (0, 0, 1, 1, 1, 1, 1, 0)]  
(240, 140) --> [(240, 120), (0, 0, 0, 0, 0, 0, 1, 1)]  
(140, 120) --> [(120, 120), (0, 0, 0, 0, 1, 1, 0, 0)]
```

```
>>>
>>>
```

The data structure underlying this second listing will be built up and maintained by some of your HW1 code ...

```
>>> for ss in mybot.what_i_learned.keys():
    print "%s --> %s" % (ss,mybot.what_i_learned[ss])
```

```
(0, 0, 0, 0, 1, 1, 1, 0) --> left
(0, 0, 0, 1, 1, 0, 0, 0) --> down
(0, 0, 0, 0, 0, 0, 1, 1) --> up
(0, 0, 0, 0, 1, 0, 0, 0) --> down
(0, 0, 0, 0, 1, 1, 0, 0) --> left
(0, 0, 1, 0, 0, 0, 0, 0) --> right
(0, 0, 1, 1, 0, 0, 0, 0) --> down
(1, 1, 0, 0, 0, 0, 0, 0) --> right
(0, 1, 1, 0, 0, 0, 0, 0) --> right
(1, 1, 1, 1, 1, 0, 0, 0) --> down
(1, 1, 0, 0, 0, 0, 1, 1) --> right
(1, 0, 0, 0, 0, 0, 0, 1) --> up
(0, 0, 0, 1, 0, 0, 0, 0) --> down
(1, 0, 0, 0, 0, 0, 0, 0) --> up
(0, 0, 0, 0, 0, 0, 1, 0) --> left
(1, 1, 1, 0, 0, 0, 0, 0) --> right
(0, 0, 0, 0, 0, 1, 1, 0) --> left
(0, 0, 1, 1, 1, 0, 0, 0) --> down
(0, 0, 1, 1, 1, 1, 1, 0) --> left
(1, 0, 0, 0, 0, 0, 1, 1) --> up
>>>
```