TRAIN A SMARTCAB HOW TO DRIVE

Omoju Miller
August 16, 2016

PROBLEM STATEMENT

In the not-so-distant future, taxicab companies across the United States no longer employ human drivers to operate their fleet of vehicles. Instead, the taxicabs are operated by self-driving agents—known as **smartcabs**—to transport people from one location to another within the cities those companies operate. In major metropolitan areas, such as Chicago, New York City, and San Francisco, an increasing number of people have come to rely on **smartcabs** to get to where they need to go as safely and efficiently as possible. Although **smartcabs** have become the transport of choice, concerns have arose that a self-driving agent might not be as safe or efficient as human drivers, particularly when considering city traffic lights and other vehicles. To alleviate these concerns, your task as an employee for a national taxicab company is to use reinforcement learning techniques to construct a demonstration of a smartcab operating in real-time to prove that both safety and efficiency can be achieved.

IMPLEMENT A BASIC DRIVING AGENT

QUESTION

- Observe what you see with the agent's behavior as it takes random actions. Does the **smartcab** eventually make it to the destination? Are there any other interesting observations to note?

At first, I set the action of the self-driving agent as forward so I could understand how the grid world worked.

```
# Select action according to your policy
action = random.choice(['forward'])
```

From this choice, it was evident the smartcab would never get to the destination, except in the case when we were lucky, and the goal just happened to be in an intersection ahead of the cab.

```
# Select action according to your policy
action = random.choice([None, 'forward', 'left', 'right'])
```

It was frustrating to watch. Most of the time, the agent never made it to the destination. There were a few times when it did; however, as expected, it *didn't learn* anything from making good decisions or bad decisions. There were no consequences for its behavior.

I decided to run a few trials of the simulation. For each trial, I captured whether it succeeded or failed by writing out the termination condition into a log file, one for success (smartCabTrialSuccess.txt) and another one for failure (smartCabTrialFailure.txt).

I ran a 100 simulation trials four times and got an average success rate of 19.3% and an average failure rate of 80.7%.

INFORM THE DRIVING AGENT

QUESTION

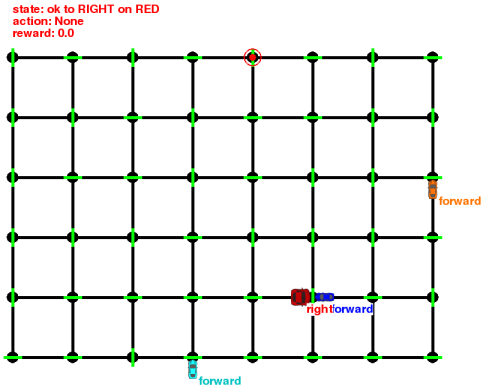

Figure 0.1: **Agent State: ok to RIGHT on RED**

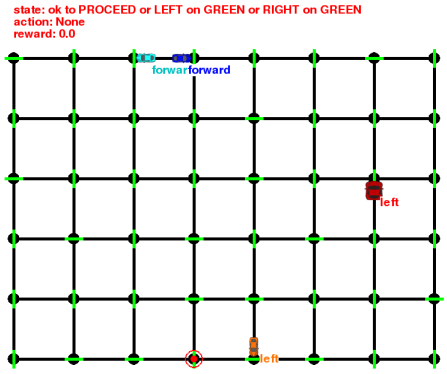

Figure 0.2: **Agent State: ok to PROCEED or LEFT on GREEN or RIGHT on GREEN**

- What states have you identified that are appropriate for modeling the smartcab and environment?
  States:
  - `ok to RIGHT on RED`
    At an intersection with a red light
  - `STOP RED light`
    At an intersection with a red light and there is traffic from the left
  - `ok to PROCEED or LEFT on GREEN or RIGHT on GREEN`
    At an intersection with a green light with no oncoming traffic and no traffic on the left

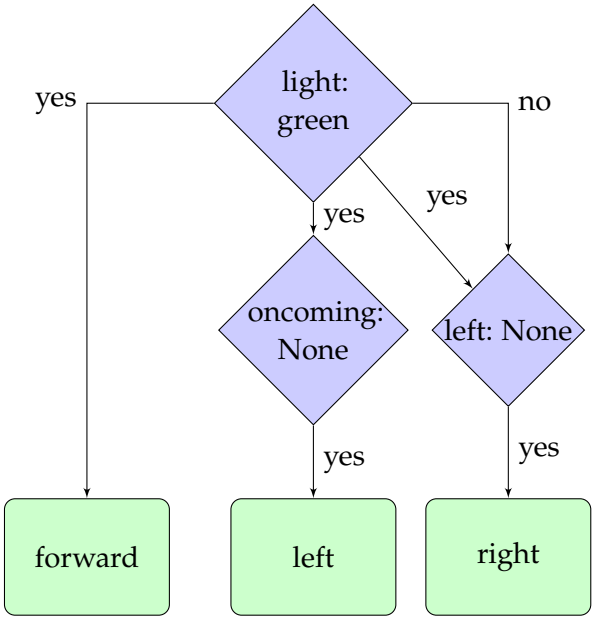

Figure 0.3: State transition diagram for smartcab at green light.

  - `ok to PROCEED or RIGHT on GREEN`
    At an intersection with a green light with no traffic on the left
  - `ok to PROCEED`
    At an intersection with a green light

- Why do you believe each of these states to be appropriate for this problem?

  There are basically two universal states that the smartcab can be in. Either it is in use, or it is not. In the case that it is not in use, it might be recharging, going through maintenance or just parked.

  When it is in service, there are a finite set of things that the smartcab can be doing. These states have to do with either movement or driving safety.

  I have idenitified the default state of the smartcab while its being use to be in the state "`ok to PROCEED`". 0.3

- Optional: How many states in total exist for the smartcab in this environment?

- Optional: Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?

IMPLEMENT A Q-LEARNING DRIVING AGENT

QUESTION

- What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?

IMPROVE THE Q-LEARNING DRIVING AGENT

QUESTION

- Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?
- Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?

CONCLUSION