Machine Learning Nano Degree

Project 4 – Train a SmartCab to drive

Daniel Fang 07/21/2016

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?*

When the agent only takes random actions, it seldom reached it final destination in the 100 trials. Besides, without taking rewards into consideration, it frequently break traffic rules. As shown in fig.1, the count of Succeed==True (which means the agent reached the destination in the trial) is about 10% of the total trials, and the median negative rewards is -11, which means the agent didn't obey the traffic rule at all.

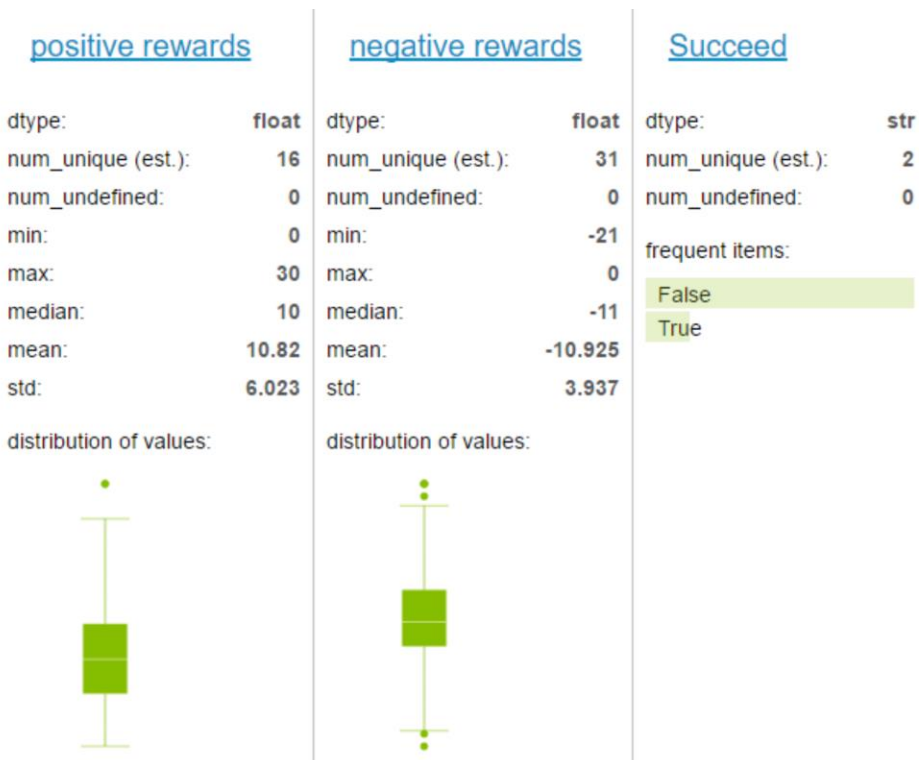

Fig.1 Results of random actions

Inform the Driving Agent

QUESTION: *What states have you identified that are appropriate for modeling the* ***smartcab*** *and environment? Why do you believe each of these states to be appropriate for this problem?*

The selected State model includes:

1) next_waypoint : it allows the agent to keep awareness of its position relative to its final destination, which is the most important information to finish the task as soon as possible.
2) light : The lights tell us the state of the traffic light which is important that we add to state to ensure that our agent obeys the rule of the traffic light.
3) incoming &left & right : The agent needs to know the state of vehicles coming in on the left and on the right to help decide when to give way to traffic.

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?*

Compared to random action agent, the Q-learning based agent is more intelligent in the way that its behavior became less erratic as the trials proceeded. Especially in the latter trials, the agent moved frequently towards its destination instead of moving the wrong directions. This is due to the Q-learning process, which 'teaches' the agent which action to take by selecting the highest rewards one.

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?*

The fine-tuning experiment is carried out on three variables: alpha, gamma and epsilon. The results are shown in the following table.

Experiment #	Alpha	Gamma	Epsilon	Success	Median_negative	Median_positive	Median_steps
1	0.9	0.9	0	55/100	-3.5	34	
2	0.9	0.4	0	100/100	0	22	12
3	0.9	0.2	0	99/100	0	24	13

4	0.4	0.9	0	65/100	-3	34	21
5	0.4	0.4	0	97/100	0	22	12
6	0.4	0.2	0	100/100	0	22	13
7	0.9	0.4	0.5	63/100	-4.5	22	21
8	0.9	0.4	0.1	93/100	-1	22	15

From the above results, the optimal parameter set is the experiment #2.

QUESTION: *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?*

As result of this process we determined as the Optimal Policy the one obtained by using a Discount Factor (gamma) of 0.2 and a Learning Rate (alpha) of 0.9. With this policy, our Driving Agent successfully arrived to its final destination within the given timeframe in 100 out of 100 opportunities. The median number of actions is 13. Its median negative rewards is 0, but it still have some minor negative rewards.