

CS 440 MP1 Part2

Jiaxin Lin jlin61

1. WorldWithoutThief

- a. When setting $\epsilon = 0$, the result of rewards is shown in Figure 1, the policy simulator result is in Figure 2. The result reward stays at 0, and the agent doesn't move when I reaches the top row of first column with no packages in hand.

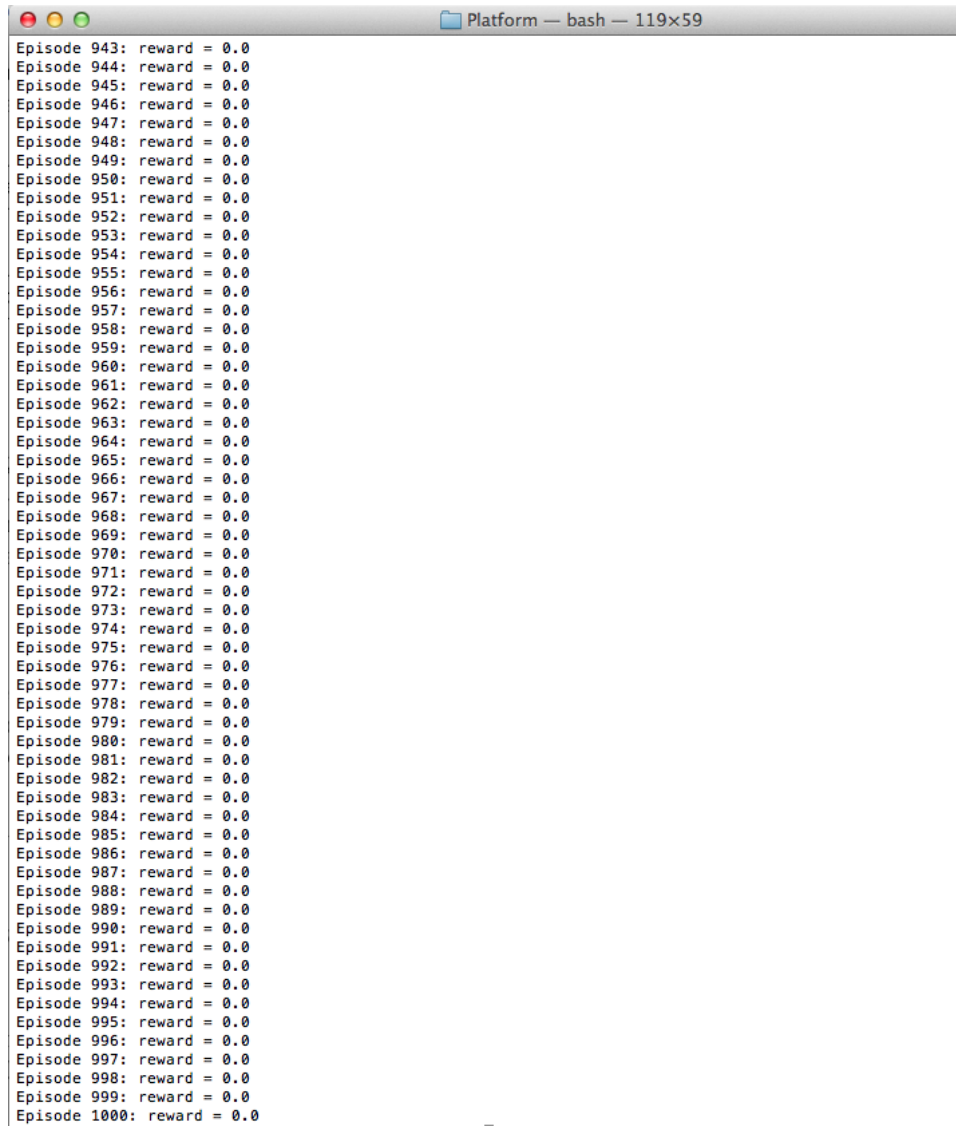
The primary reason for this is, after looking into policy.txt and WorldWithoutThief.java, the action for the robot at this state, which is $((0 \times 5 + 0) \times 2 + 1) \times 2 + 1 = 3$, has a policy of 0(North), hence it will stuck at the current state.

Also the several steps before this, which are state 23, 43, 63 and 83, all have policies of 0(North). This results in the agent's going north from the beginning and stuck at the top left corner.

The underlying reason for this is that since $\epsilon = 0$, there is no randomness for the agent when choosing the next step, instead it always choose the action with the max Q . Despite that it chooses randomly when there are multiple max Q , it reduces the chance for the agent to explore new actions. Since the agent receives penalty when dropping a package, going to the right part always have a negative Q , and since the agent is not "brave" enough (no randomly choosing step), it doesn't go to the right part, thus no positive rewards will be given.

This can also be easily shown by printing out the Q values after each episode (Figure 3 is an incomplete list of the Q values due to the size of pdf), where only 0 or negative values can be seen.

Figure 1

A terminal window titled "Platform — bash — 119x59" displays a list of episode rewards. The text is as follows:

```
Episode 943: reward = 0.0
Episode 944: reward = 0.0
Episode 945: reward = 0.0
Episode 946: reward = 0.0
Episode 947: reward = 0.0
Episode 948: reward = 0.0
Episode 949: reward = 0.0
Episode 950: reward = 0.0
Episode 951: reward = 0.0
Episode 952: reward = 0.0
Episode 953: reward = 0.0
Episode 954: reward = 0.0
Episode 955: reward = 0.0
Episode 956: reward = 0.0
Episode 957: reward = 0.0
Episode 958: reward = 0.0
Episode 959: reward = 0.0
Episode 960: reward = 0.0
Episode 961: reward = 0.0
Episode 962: reward = 0.0
Episode 963: reward = 0.0
Episode 964: reward = 0.0
Episode 965: reward = 0.0
Episode 966: reward = 0.0
Episode 967: reward = 0.0
Episode 968: reward = 0.0
Episode 969: reward = 0.0
Episode 970: reward = 0.0
Episode 971: reward = 0.0
Episode 972: reward = 0.0
Episode 973: reward = 0.0
Episode 974: reward = 0.0
Episode 975: reward = 0.0
Episode 976: reward = 0.0
Episode 977: reward = 0.0
Episode 978: reward = 0.0
Episode 979: reward = 0.0
Episode 980: reward = 0.0
Episode 981: reward = 0.0
Episode 982: reward = 0.0
Episode 983: reward = 0.0
Episode 984: reward = 0.0
Episode 985: reward = 0.0
Episode 986: reward = 0.0
Episode 987: reward = 0.0
Episode 988: reward = 0.0
Episode 989: reward = 0.0
Episode 990: reward = 0.0
Episode 991: reward = 0.0
Episode 992: reward = 0.0
Episode 993: reward = 0.0
Episode 994: reward = 0.0
Episode 995: reward = 0.0
Episode 996: reward = 0.0
Episode 997: reward = 0.0
Episode 998: reward = 0.0
Episode 999: reward = 0.0
Episode 1000: reward = 0.0
```

Figure 2

```
wirelessprvnat-172-17-239-216:Platform jiaxinlin$ java PolicySimulator WorldWithoutThief policy.txt y 10000
# 1
#
###
#
R # 2
Carrying both packages
Total reward is 0.0.

# 1
#
###
R #
C # 2
Carrying both packages
Total reward is 0.0.

# 1
#
R ###
#
C # 2
Carrying both packages
Total reward is 0.0.

# 1
R #
#
###
#
C # 2
Carrying both packages
Total reward is 0.0.

R # 1
#
#
###
#
C # 2
Carrying both packages
Total reward is 0.0.

R # 1
#
###
#
C # 2
Carrying both packages
Total reward is 0.0.

R # 1
#
#
###
#
C # 2
```

Figure 3

```
Episode 10: reward = 0.0
Q values are:
State: 0      0.0      0.0      0.0      0.0
State: 1      0.0      0.0      0.0      0.0
State: 2      0.0      0.0      0.0      0.0
State: 3      0.0      0.0      0.0      0.0
State: 4      0.0      0.0      0.0      0.0
State: 5      0.0      0.0      0.0      0.0
State: 6      0.0      0.0      0.0      0.0
State: 7      0.0     -0.05     0.0      0.0
State: 8      0.0      0.0      0.0      0.0
State: 9      0.0      0.0      0.0      0.0
State: 10     0.0      0.0      0.0      0.0
State: 11    -0.05     0.0      0.0      0.0
State: 12     0.0      0.0      0.0      0.0
State: 13     0.0      0.0      0.0      0.0
State: 14     0.0      0.0      0.0      0.0
State: 15     0.0      0.0      0.0      0.0
State: 16     0.0      0.0      0.0      0.0
State: 17     0.0      0.0      0.0      0.0
State: 18     0.0      0.0      0.0      0.0
State: 19     0.0      0.0      0.0      0.0
State: 20     0.0      0.0      0.0      0.0
State: 21     0.0      0.0      0.0      0.0
State: 22     0.0      0.0      0.0      0.0
State: 23     0.0      0.0      0.0      0.0
State: 24     0.0      0.0      0.0      0.0
State: 25     0.0      0.0      0.0      0.0
State: 26     0.0      0.0      0.0      0.0
State: 27     0.0     -0.05     0.0      0.0
State: 28     0.0      0.0      0.0      0.0
State: 29     0.0      0.0      0.0      0.0
State: 30     0.0      0.0      0.0      0.0
State: 31    -0.05     0.0      0.0      0.0
State: 32     0.0      0.0      0.0      0.0
State: 33     0.0      0.0     -0.05     0.0
State: 34     0.0      0.0     -0.05     0.0
State: 35     0.0      0.0     -0.05     0.0
State: 36     0.0      0.0      0.0      0.0
```

- b. When setting $\epsilon = 0.1$, the result of rewards is shown in Figure 4. An incomplete list of agent policies are shown in Figure 5. It is obvious that this time the agent actually delivers packages and making positive rewards.

The main reason is that the agent is allowed to have a level of randomness when choosing action, instead of only choosing those with maximum current Q . The drawback for choosing maximum Q is that the

Q values are current estimation, not the true Q . Agent need to have some level of randomness to derive the error of the estimated Q .

Figure 4

```
wirelessprvnt-172-17-239-216:Platform jiixinlin$ javac *.java
wirelessprvnt-172-17-239-216:Platform jiixinlin$ java Simulator WorldWithoutThief QLearningAgent y 10000 1000 policy.
txt episodes.txt
Episode 1: reward = -18.0
Episode 2: reward = -16.0
Episode 3: reward = -13.5
Episode 4: reward = -7.0
Episode 5: reward = -11.0
Episode 6: reward = -11.5
Episode 7: reward = -6.5
Episode 8: reward = -10.0
Episode 9: reward = -2.5
Episode 10: reward = -6.5
Episode 11: reward = -9.0
Episode 12: reward = -4.0
Episode 13: reward = -4.0
Episode 14: reward = -3.0
Episode 15: reward = 0.5
Episode 16: reward = 1.5
Episode 17: reward = -3.0
Episode 18: reward = 2.5
Episode 19: reward = 12.0
Episode 20: reward = 169.5
Episode 21: reward = 101.0
Episode 22: reward = 145.0
Episode 23: reward = 151.0
Episode 24: reward = 150.0
Episode 25: reward = 191.0
Episode 26: reward = 118.5
Episode 27: reward = 176.5
Episode 28: reward = 169.0
Episode 29: reward = 172.5
Episode 30: reward = 54.5
Episode 31: reward = 151.0
Episode 32: reward = 159.0
Episode 33: reward = 78.0
Episode 34: reward = 228.0
Episode 35: reward = 168.0
Episode 36: reward = 133.0
Episode 37: reward = 156.0
Episode 38: reward = 88.5
Episode 39: reward = 197.5
Episode 40: reward = 230.5
Episode 41: reward = 83.5
Episode 42: reward = 101.0
Episode 43: reward = 127.5
Episode 44: reward = 128.5
Episode 45: reward = 91.0
Episode 46: reward = 64.5
Episode 47: reward = 76.0
Episode 48: reward = 26.0
Episode 49: reward = 173.5
Episode 50: reward = 41.5
Episode 51: reward = 78.5
Episode 52: reward = 173.5
Episode 53: reward = 166.0
Episode 54: reward = 145.5
Episode 55: reward = 195.0
Episode 56: reward = 176.5
```

Figure 5

```

Total reward is 0.0.

# 1
#
###
#
C # R
Only carrying the package for customer 1
Total reward is 0.0.

# 1
#
###
#
C # R2
Only carrying the package for customer 1
Total reward is 0.0.

# 1
#
###
#R
C # 2
Only carrying the package for customer 1
Total reward is 0.0.

# 1
#
###
R
C # 2
Only carrying the package for customer 1
Total reward is 0.0.

# 1
#
###
R#
C # 2
Only carrying the package for customer 1
Total reward is 0.0.

# 1
#
R###
#
C # 2
Only carrying the package for customer 1
Total reward is 0.0.

# 1
R#
###
#
C # 2
Only carrying the package for customer 1
Total reward is 0.0.

```

- c. When $\epsilon = 0.5$, the rewards goes down below 0, ranges fro 0 to -50. Whereas when $\epsilon = 0.1$ rewards of over 100 are frequently sighted.

By trying settings of $\epsilon = 0.3$, and $\epsilon = 0.5$, we can conclude that a very high level of randomness (0.5 is considered too high) will disorder the agent's action too much and lower the rewards. Thus the "greedy" part and "random" part need to be balanced at a certain level. (Figure 6 shows the rewards at $\epsilon = 0.5$)

Figure 6

```

Episode 479: reward = -18.5
Episode 480: reward = -10.0
Episode 481: reward = -46.0
Episode 482: reward = -3.0
Episode 483: reward = -10.5
Episode 484: reward = -32.5
Episode 485: reward = -37.0
Episode 486: reward = 2.5
Episode 487: reward = -37.0
Episode 488: reward = -13.0
Episode 489: reward = -47.0
Episode 490: reward = -36.5
Episode 491: reward = -38.5
Episode 492: reward = -34.5
Episode 493: reward = -16.0
Episode 494: reward = -17.0
Episode 495: reward = -48.5
Episode 496: reward = -32.5
Episode 497: reward = -24.5
Episode 498: reward = -30.5

```

2. WorldWithThief

- a. In WorldWithThief, rewards are below 0, Figure 7 shows the rewards. Figure 8 shows the policy simulator.

In this case the agent is stuck at the top left corner. The policy simulator has a similar situation as WorldWithoutThief and a low ϵ . The possible underlying reason is that, with a relatively low ϵ , the agent chooses an action with highest ϵ almost every time, thus it rarely goes to the right because of the slipper area gives a negative reward, which then lower the Q values associates with it at early training stage. The other reason is that, at the top left corner, policy has the action of North (0), which is according to the max Q , thus causing the stuck.

The reason for low rewards during the episodes is possibly because of the unpredictable thief. Since the agent doesn't where is the thief, it ignores it (similar to WorldWithoutThief), thus hits on the thief frequently.

Figure 7


```
Episode 967: reward = -16.5
Episode 968: reward = -10.5
Episode 969: reward = -9.5
Episode 970: reward = -9.0
Episode 971: reward = -19.0
Episode 972: reward = -12.0
Episode 973: reward = -13.5
Episode 974: reward = -10.5
Episode 975: reward = -13.0
Episode 976: reward = -16.5
Episode 977: reward = -13.5
Episode 978: reward = -11.5
Episode 979: reward = -8.5
Episode 980: reward = -11.5
Episode 981: reward = -13.0
Episode 982: reward = -14.0
Episode 983: reward = -13.0
Episode 984: reward = -16.0
Episode 985: reward = -14.5
Episode 986: reward = -14.5
Episode 987: reward = -13.5
Episode 988: reward = -12.0
Episode 989: reward = -8.0
Episode 990: reward = -9.5
Episode 991: reward = -12.5
Episode 992: reward = -15.0
Episode 993: reward = -11.5
Episode 994: reward = -17.0
Episode 995: reward = -12.5
Episode 996: reward = -12.5
Episode 997: reward = -13.5
Episode 998: reward = -14.5
Episode 999: reward = -13.0
Episode 1000: reward = -11.5
wirelessprvnat-172-17-239-216:Platform jiaxinlin$ █
```

Figure 8

```
total reward is 0.0.

#T 1
# ##

R#
C# 2
Carrying both packages
Total reward is 0.0.

# 1
#T##
R
#
C# 2
Carrying both packages
Total reward is 0.0.

# 1
R# ##
T
#
C# 2
Carrying both packages
Total reward is 0.0.

R# 1
#T##

#
C# 2
Carrying both packages
Total reward is 0.0.

R# 1
# ##
T
#
C# 2
Carrying both packages
Total reward is 0.0.

R# 1
#T##

#
C# 2
Carrying both packages
Total reward is 0.0.
```

- b. When setting the know thief as y, the agent is more "clever", and the rewards are far better than not know where the thief is. Now the thief's position is a component of the agent's state. The agent's Q value is optimized for every thief position, and choose actions accordingly. Figure 9 is the rewards.

Figure 9

```

Episode 852: reward = 278.5
Episode 853: reward = 293.5
Episode 854: reward = 268.0
Episode 855: reward = 273.5
Episode 856: reward = 285.0
Episode 857: reward = 287.0
Episode 858: reward = 295.5
Episode 859: reward = 286.0
Episode 860: reward = 260.5
Episode 861: reward = 294.5
Episode 862: reward = 272.5
Episode 863: reward = 277.5
Episode 864: reward = 274.5
Episode 865: reward = 265.0
Episode 866: reward = 280.5
Episode 867: reward = 292.5
Episode 868: reward = 287.0
Episode 869: reward = 285.5
Episode 870: reward = 307.0
Episode 871: reward = 275.5
Episode 872: reward = 269.5
Episode 873: reward = 263.0
Episode 874: reward = 287.5
Episode 875: reward = 279.0
Episode 876: reward = 276.5
Episode 877: reward = 275.0
Episode 878: reward = 288.5
Episode 879: reward = 282.0
Episode 880: reward = 273.5
Episode 881: reward = 275.0
Episode 882: reward = 286.0
Episode 883: reward = 276.5
Episode 884: reward = 248.0
Episode 885: reward = 290.0
Episode 886: reward = 263.5
Episode 887: reward = 279.5
Episode 888: reward = 299.0
Episode 889: reward = 269.5
Episode 890: reward = 279.5
Episode 891: reward = 288.5
Episode 892: reward = 278.5
Episode 893: reward = 271.5
Episode 894: reward = 277.0
Episode 895: reward = 287.5
Episode 896: reward = 271.0
Episode 897: reward = 282.0
Episode 898: reward = 295.5
Episode 899: reward = 280.0
Episode 900: reward = 274.5
Episode 901: reward = 282.5
_ _ _ _ _

```

- c. For this part, to determine the best ϵ and learning rate, I test the model with various learning rate, ranges from 0 to 5 and a step of 0.02. Within

every step, various ϵ is tested, ranges from 0 to 0.5 and a step of 0.02. For each iteration, rewards are averaged out by number of episodes. This is done by modifying the agent file and simulator file, and adding a Tester class to run iterations.

The result for this experiment: the best rewards are derived when $\epsilon = 0.02$ and learning rate is 0.15. The reward (average) is 306.57.

3. Using the setup of 3c, $\epsilon = 0.02$, learning rate = 0.15 and discount = 0.9, 10000 steps and 1000 episodes, Figure 10 shows the rewards from the first iteration. The reward started from about -80, grows rapidly during episodes 0 to 80 and become steady after 200 episodes, and stays at around 320. Figure 11 shows the rewards after doing 10 iterations and taking the average of each episodes by all iterations. This is because, with the iterations of episodes, Q values for each state and each action converges more and becomes more "accurate", which then results in the growth of rewards.

Figure 10

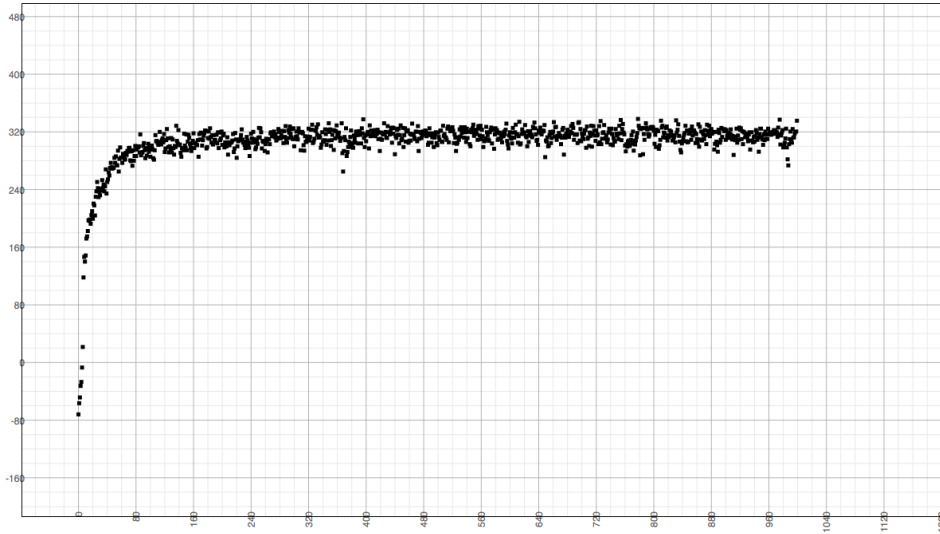


Figure 11

