Machine Learning in Robotics

Assignment 2

Surname: Li

First Name: Bowen

Matriculation Number :

03709969

Exercise 1

priors

pi

1x4 double

	1	2	3	4
1	0.2400	0.2972	0.2617	0.2011

means

mean

2x4 double

	1	2	3	4
1	-0.0432	-0.0194	0.0262	-0.0147
2	0.0446	-0.0166	0.0617	-0.0796

covariance matrix

covariance{1, 1}

	1	2
1	1.7479e-04	2.6154e-04
2	2.6154e-04	3.9754e-04

covariance{1, 2}

	1	2
1	7.4372e-04	-5.9168e-04
2	-5.9168e-04	6.1027e-04

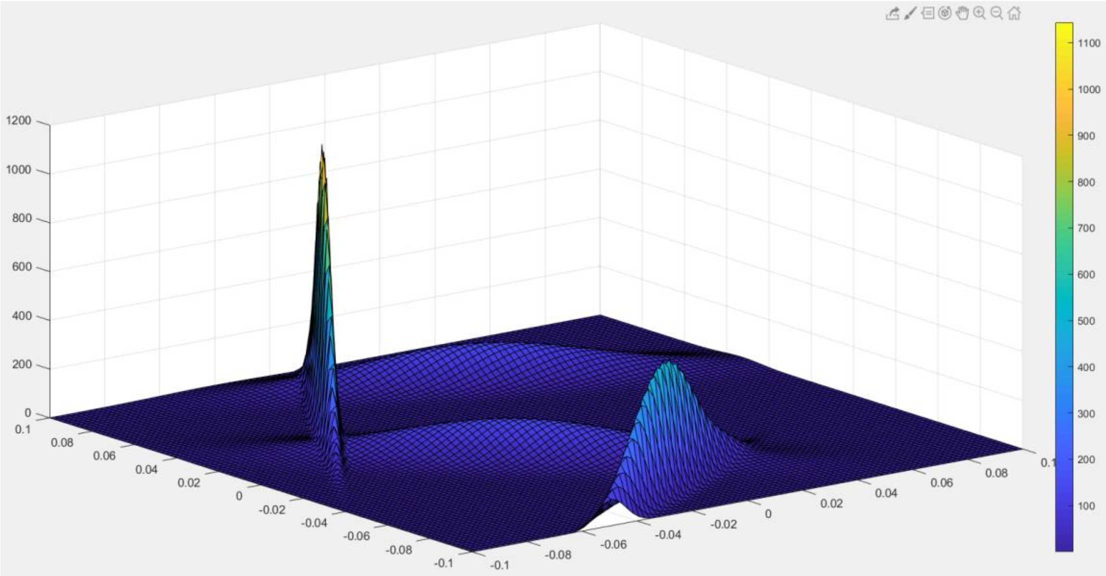
covariance{1, 3}

	1	2
1	0.0011	-4.2436e-04
2	-4.2436e-04	2.4312e-04

covariance{1, 4}

	1	2
1	3.9439e-04	2.1664e-04
2	2.1664e-04	1.2757e-04

density plot



Exercise 2

The log-likelihood value of 10 sequences:

log_likeli
10x1 double

	1
1	-511.4069
2	-570.6697
3	-387.9167
4	-427.3069
5	-437.5989
6	-426.1784
7	-473.3031
8	-400.2880
9	-377.1776
10	-401.0614

Since all values are smaller than -115, so all 10 sequences are classified as gesture 2.

The classification results:

gesture_label

10x1 double

	1
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2

Exercise 3

Task 2: Applying policy iteration

2.1 Report your reward matrix.

rew

16x4 double

	1	2	3	4
1	0	0	0	0
2	0	1	-1	-1
3	0	-1	-1	-1
4	0	0	0	0
5	-1	-1	0	1
6	0	0	0	0
7	0	0	0	0
8	-1	1	0	0
9	-1	-1	0	-1
10	0	0	0	0
11	0	0	0	0
12	-1	0	0	0
13	0	0	0	0
14	0	0	-1	1
15	0	-1	-1	1
16	0	0	0	0

2.2 What value of γ have you used and what is the result of increasing or decreasing γ.

In this task, γ=0.8. γ is a discount factor, it is used to balance immediate and future reward. Increasing γ indicates that more future steps are taken into account and vice versa.

2.3 Approximately how many iterations are required for the policy iteration to converge.

Approximately 30 iterations are needed to reach the convergence.

2.4 Attach the result of *WalkPolicyIteriation(s)* when starting from state 10 and 3.

Figure 1. starting from state 10

Figure 2. starting from state 3

Task 3: Applying Q-Learning

3.1 Report the values of ε and α.

ε = 0.2

α = 0.33

3.2 What happens if a pure greedy policy is used? Implement and compare with the ε-greedy policy. Does it matter what value of ε you use?

Pure greedy means ε = 0, and in this case, the algorithm will converge to local optimum. The agent only chooses the action with largest value. However, keeping a vaguely explorative / stochastic element in its policy (like a tiny amount of ε) allows it to get out of such states. With large ε, the agent tends to do the exploration action, it guarantees that the algorithm converges to global optimum but also needs more steps at the same time.

3.3 Approximately how many steps are necessary for the Q-Learning algorithm to find an optimal policy.

Approximately 100 steps are necessary. In the task T = 500.

3.4 Attach the result *WalkingQLearning(s)* when starting from states 5 and 12.

Figure 3. starting from state 5

Figure 4. starting from state 12