MACHINE LEARNING IN ROBOTICS
Assignment 2

Weiqi Luo 03697059

August 22, 2019

1 Exercise 1

1.1 Learned GMM Parameters

Component 1

$$\pi_1 = 0.2617 \quad \mu_1 = \begin{bmatrix} 0.0262 \\ 0.0617 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 0.0011 & -4.2436e-04 \\ -4.2436e-04 & 2.4312e-04 \end{bmatrix}$$

Component 2

$$\pi_2 = 0.2011 \quad \mu_2 = \begin{bmatrix} -0.0147 \\ -0.0796 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 3.9439e-04 & 2.1664e-04 \\ 2.1664e-04 & 1.2757e-04 \end{bmatrix}$$

Component 3

$$\pi_3 = 0.2972 \quad \mu_3 = \begin{bmatrix} -0.0194 \\ -0.0166 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 7.4372e-04 & -5.9168e-04 \\ -5.9168e-04 & 6.1027e-04 \end{bmatrix}$$

Component 4

$$\pi_4 = 0.2400 \quad \mu_4 = \begin{bmatrix} -0.0432 \\ 0.0446 \end{bmatrix} \quad \Sigma_4 = \begin{bmatrix} 1.7479e-04 & 2.6154e-04 \\ 2.6154e-04 & 3.9754e-04 \end{bmatrix}$$

1.2 Visualization

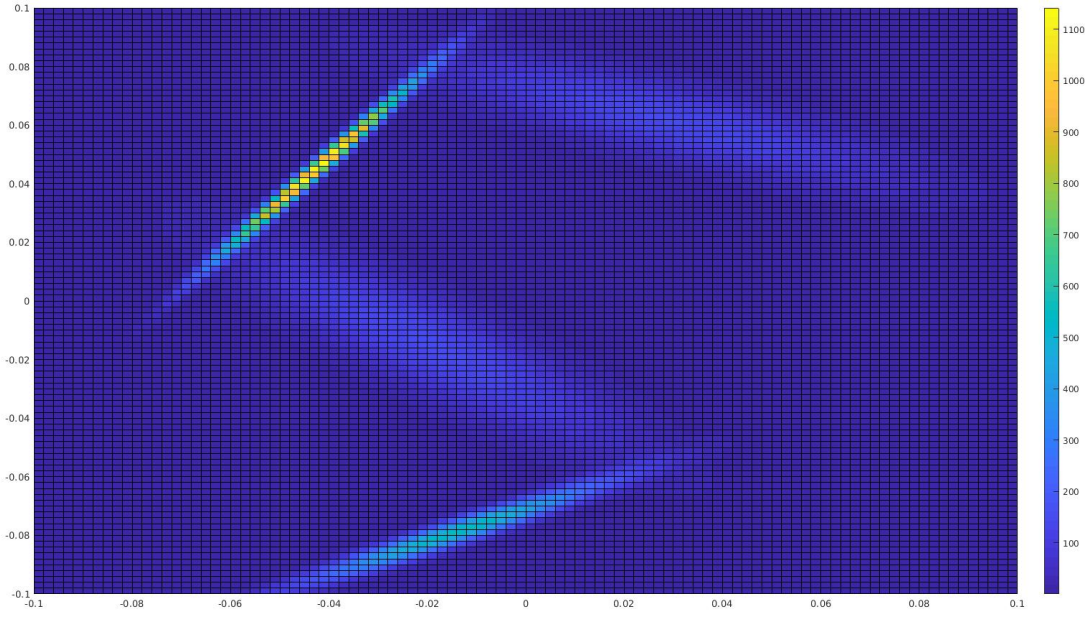

Figure 1: GMM Density Function Visualization

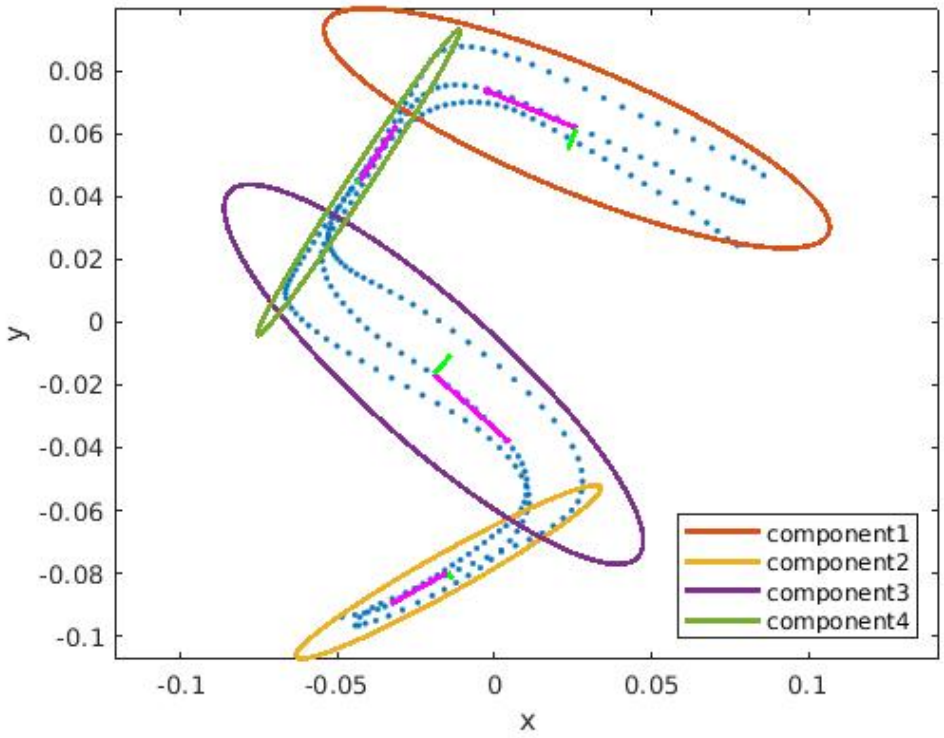

Figure 2: GMM Components Visualization

2 Exercise 2

2.1 Classification Result

All sequences in Test.txt belong to gesture2.

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

2.2 Visualization

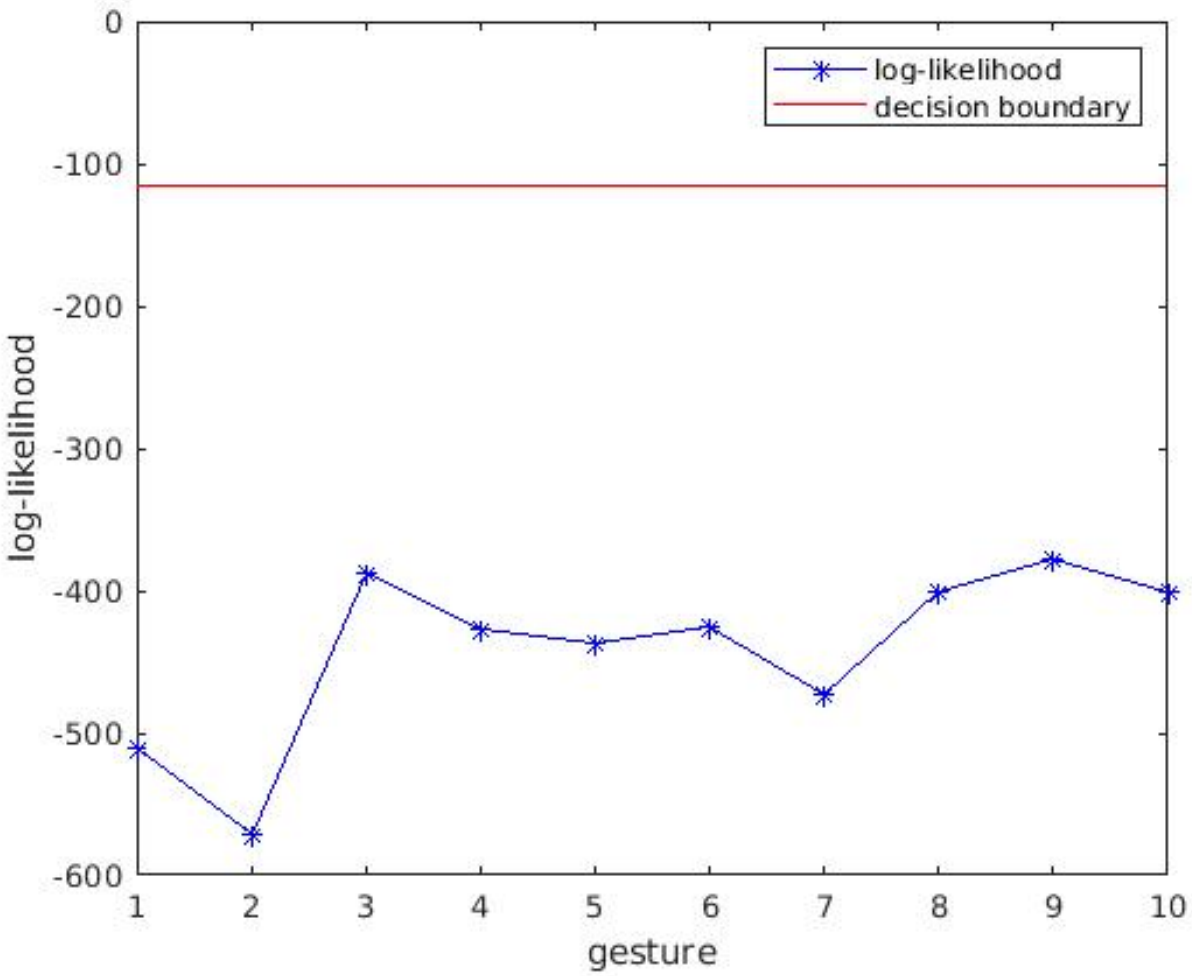

Figure 3: Log-likelihood visualization

3 Exercise 3

3.1 Applying Policy Iteration

3.1.1 Report reward matrix

Positive Rewards:
- Moving the Robot forward

Negative Rewards:
- Moving a leg forward or backward while it is still on the ground
- Raising one leg while the other is already in the air
- Moving the Robot backward

$$Reward\ Matrix\ = \begin{bmatrix} 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{bmatrix}$$

3.1.2 The choose of γ

The discount factor γ determines the importance of future rewards. A factor approaching 0 will make the agent short-sighted, while a factor of 0 will make it only considering current rewards, and the agent will fail to learn the optimal behavior. A factor approaching 1 will make it strive for a long-term high reward, however a large factor result in more iterations in oder to converge (see Fig.6). If the factor reaches 1 the resulting system of linear equations can not be solved because the matrix becomes singular.

I choose the value of $\gamma = 0.5$ according to Fig.6.

3.1.3 Iteration number

The iteration number required for the policy iteration to converge is approximately between 2 and 7 according to Fig.6.

3.1.4 Experiments Result

Initial State 10 State sequence: 10 , 14 , 2 , 3 , 4 , 8 , 5 , 9 , 13 , 14 , 2 , 3 , 4 , 8 , 5 , 9.

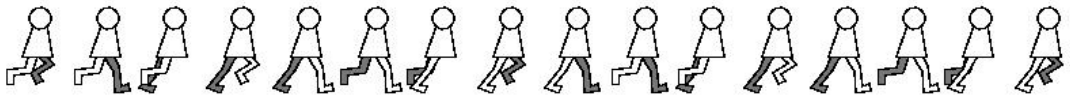

Figure 4: Policy iteration with initial state 10

Initial State 3 State sequence: 3 , 4 , 8 , 5 , 9 , 13 , 14 , 2 , 3 , 4 , 8 , 5 , 9 , 13 , 14 , 2.

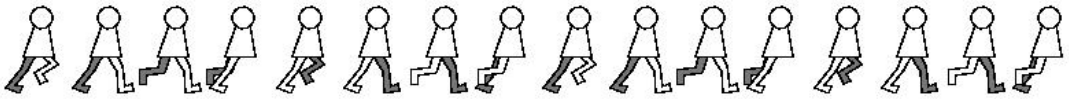

Figure 5: Policy iteration with initial state 3

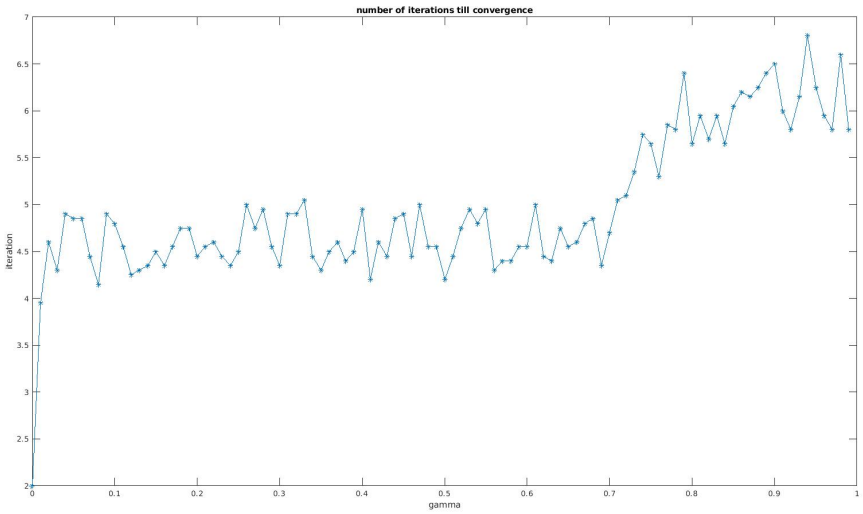

Figure 6: Average number of iterations for different discount factor γ

3.2 Applying Q-learning

3.2.1 The choose of ϵ and α

A learning rate α of 1 is selected. Since the reward and state transition matrix are deterministic, there is no need to consider several observations to incrementally approximate the average over all possible state.

A linearly decreasing ϵ is chosen to speed up the convergence. At beginning $\epsilon = 1$, which corresponds to a pure random policy, since the agent has no idea about the environment. As the agent collects more and more evidence, the policy shift towards a pure greedy policy.

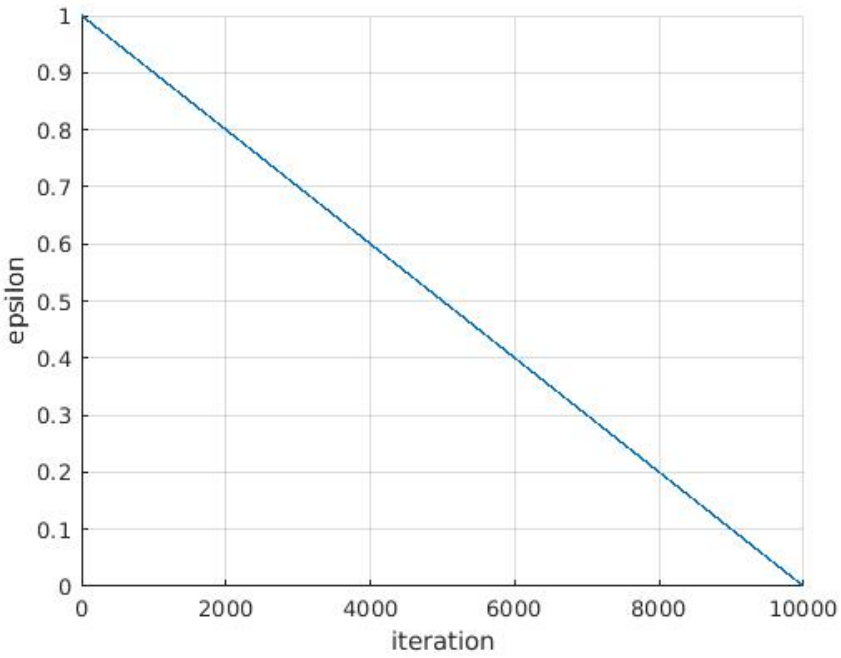

Figure 7: linearly decreasing ϵ with iteration

3.2.2 Compare of pure greedy policy and ϵ-greedy

In the case of pure greedy policy, the agent always performs the action witch correspond to the largest action-value function, no exploration is made by the algorithm. The learned policy will fail to converge at at the optimal policy.

If ϵ is too small, the system will fail to converge at at the optimal policy. If it is too large, it will take too many iterations to converge. A reasonable choise is to set ϵ decreasing with the number of iterations.

3.2.3 Iteration numbers

Approximately 1000

3.2.4 Experiments Result

Initial State 5 State sequence: 5 , 9 , 13 , 14 , 2 , 3 , 4 , 8 , 5 , 9 , 13 , 14 , 2 , 3 , 4 , 8.

Figure 8: Q learning with initial state 5

Initial State 12 State sequence: 12 , 9 , 13 , 14 , 2 , 3 , 4 , 8 , 5 , 9 , 13 , 14, 2 , 3 , 4 , 8.

Figure 9: Q learning with initial state 12