536

Xinyang Wang

189001002

1 HW4

1.1

We see that the class of each data point is determined entirely by the value of the X_k feature. so if we look at all the data points along the X_k axis, we will get the following picture.

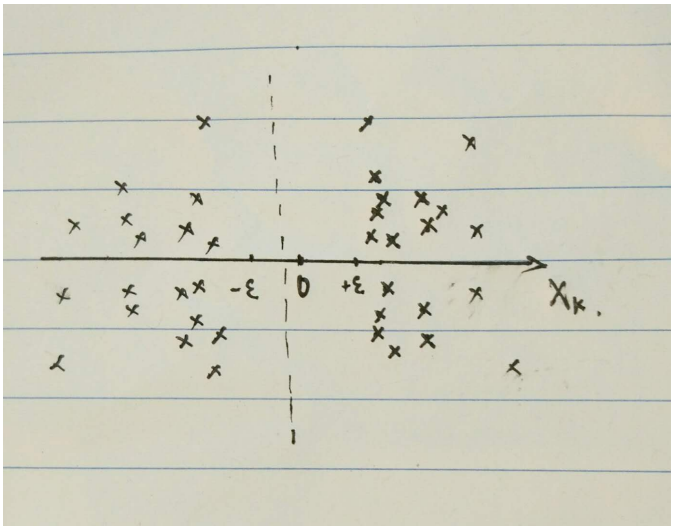

Figure 1: Q1

As we can see, Data points are dense when close to $\pm\epsilon$ and are sparse when $|X_k|$ getting bigger. That is because the distribution of X_k is EXP(1). And we can easily find that, any Hyperplane perpendicular to the X_k axis(like the dashed line in Figure 1) is a good perceptron, Which means that perceptrons are not unique in this case. And the best perceptron theoretically, is the Hyperplane that crosses 0 on the X_k axis, which guarantee the biggest margin between data points and hyperplane.

1.2

Generate a set of data of size $m = 100$ with $k = 20$, $\epsilon = 1$ and implement the perceptron algorithm on our data. Below is a result of one test, in which we draw the value of W_i. (w_0 corresponding to b, w_1-w_{20} coresopnding to x_1-x_{20})
As we can see, w_1 to w_{19} are relatively small, and w_{20} is much bigger, which means perceptron is almost perpendicular to x_{20}, but not as ideal as theoretically. And b is also not equal to 0. When m increases, w_1 to w_{19} will decrease gradually.

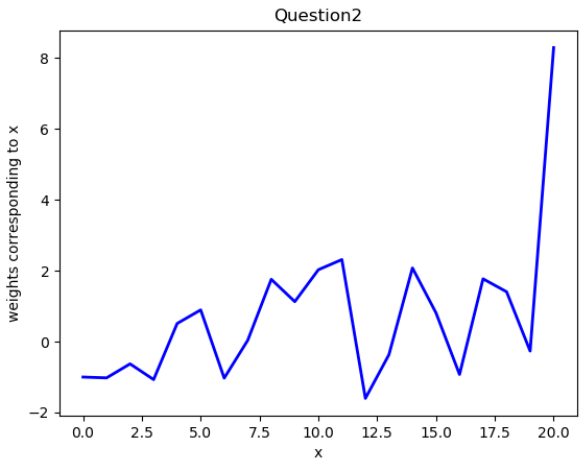

Figure 2: Q2

1.3

Fixing $m = 100$ and $k = 20$, we draw the changing steps as a function of ϵ in Figure 3. We can see that as the ϵ increases, the average step decreases. This is because ϵ represents the margin between separable data and the bigger the margin is, the more easily we get a perceptron.

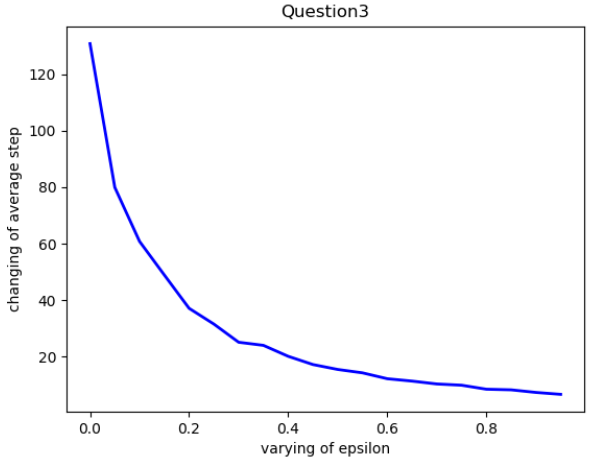

Figure 3: Q3

1.4

This time we fix $m = 100$ and $\epsilon = 1$, while changing the value of k. Below we draw the average step as a function of k in Figure 4. We can see that average steps increase in some degree, but not as much as k. I think this is reasonable: the bigger the k is, the higher dimension of data is, thus increasing the complexity of finding a perceptron.

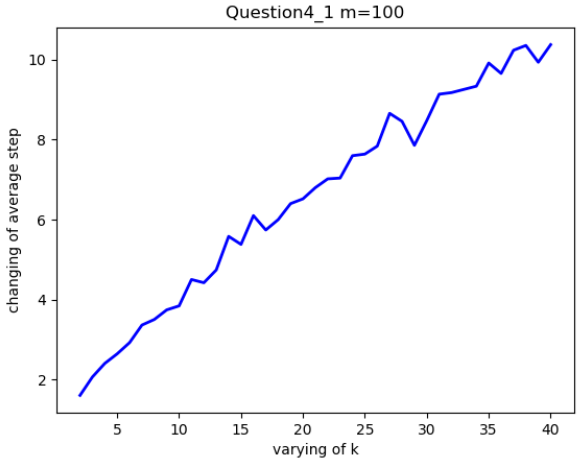

Figure 4: Q4-1

Then we fix m to equal 1000, and redo the experiments above. Below we draw the average steps as a function of k. This time we see that average steps increase a little. From my opinion, this is because we need to traverse more data and thus run into more mistakes (which means more steps).

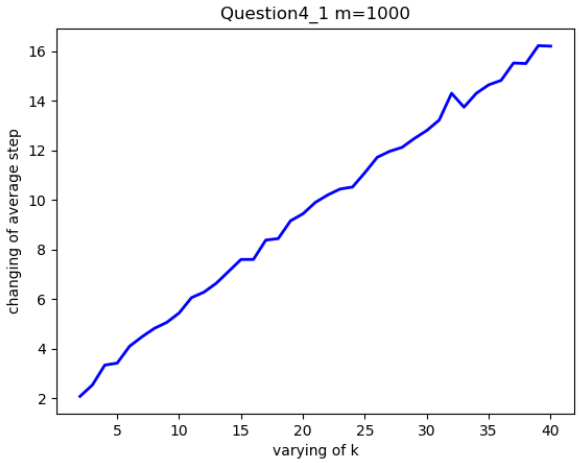

Figure 5: Q4-2

1.5

For $k = 2$, the definition of Y is a circle with center at (0,0). When data fall out of circle, Y equals +1 and when data fall into the circle, Y equals -1. Thus theoretically data can not be linearly separated.
When we run perceptron algorithm on these data, we record the value of b, w_1 and w_2 each time they are updated, and below we draw them. It is clear to see that they fluctuate all the time and never converge.
It is hard to say whether this algorithm will terminate(data are linearly separable in other words). In our algorithm, we use the loop number (How many times the code traverses all the data) and mistakes(if the number of mistakes exceeds m/20) as a sign to determine whether or not to terminate the algorithm. See more details in attached code.

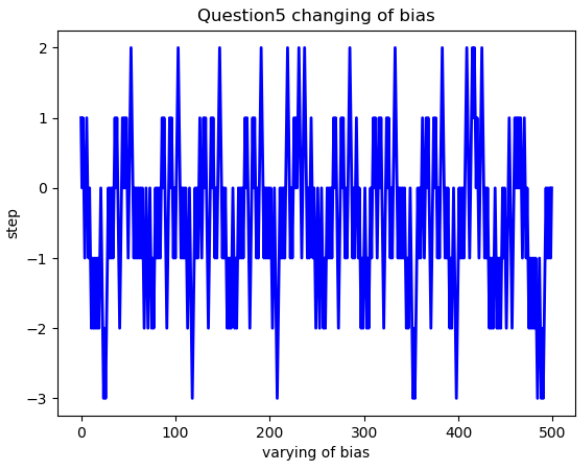

Figure 6: Q5-bias

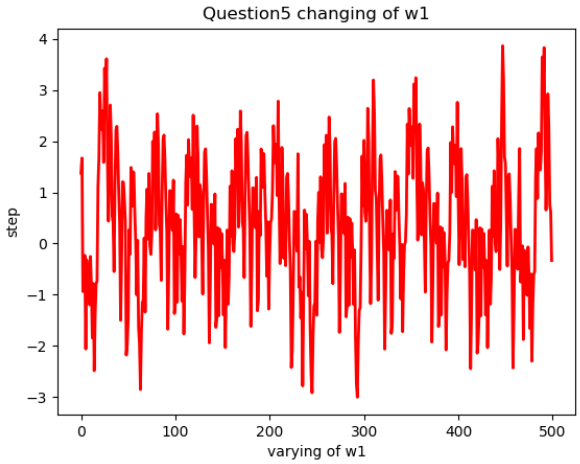

Figure 7: Q5-w1

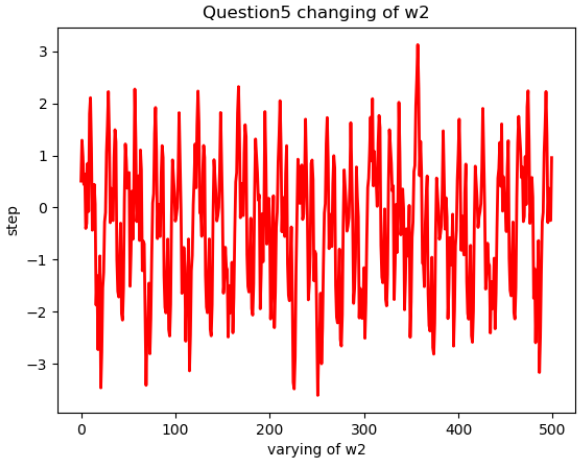

Figure 8: Q5-w2