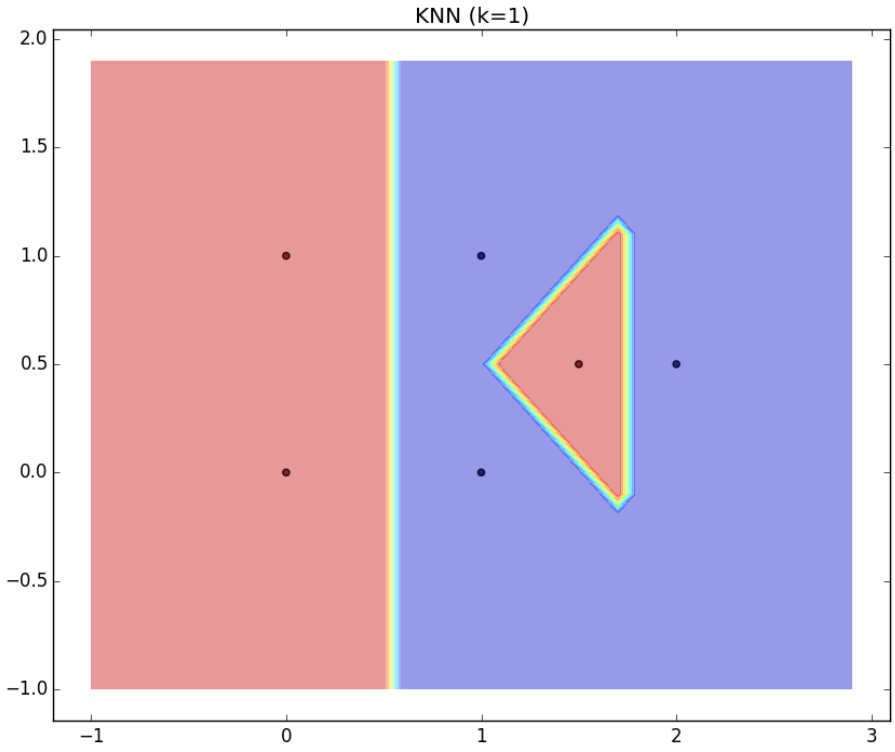
CS412 Assignment 5
Li Miao

Problem 1

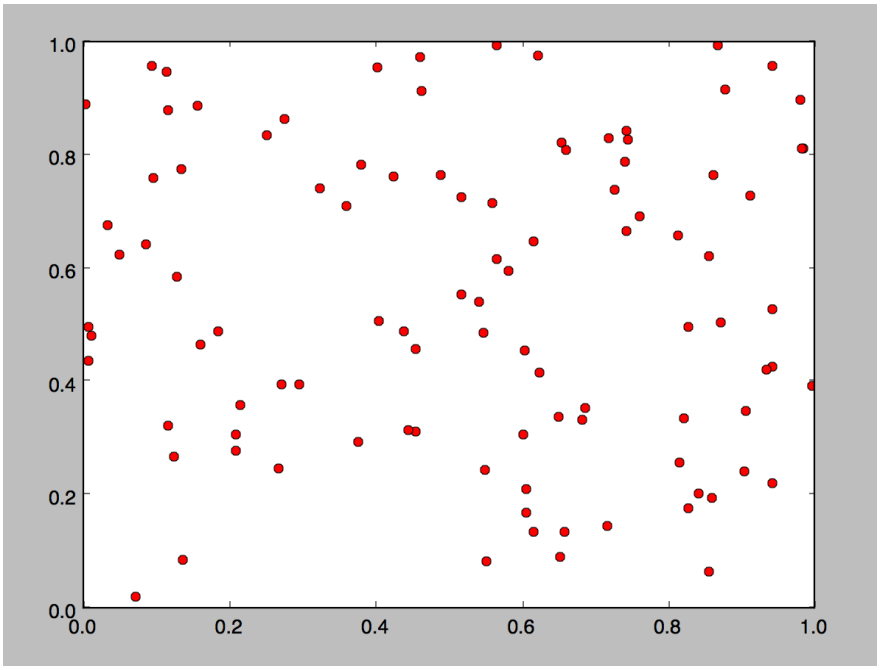
1. If we pick random unvisited point p, then the answer is no. Because we can not control which point we choose as next. But if we follow the order in the dataset, then the output would be the same.
2. We could find the k points which are most far apart and make them the initial centers. It would make the K-means algorithms more efficient. Because they are more representative than other points.
3. K-means is more efficient, K-medoids is much more expensive. Usually, PAM takes much longer to run than k-means. O(n^2*k*i) where k-means is O(n*k*i).
5. False. In high-dimensional space and when the data scale is really large, non-lazy learners are more efficient than lazy learners.

Problem 2

1.



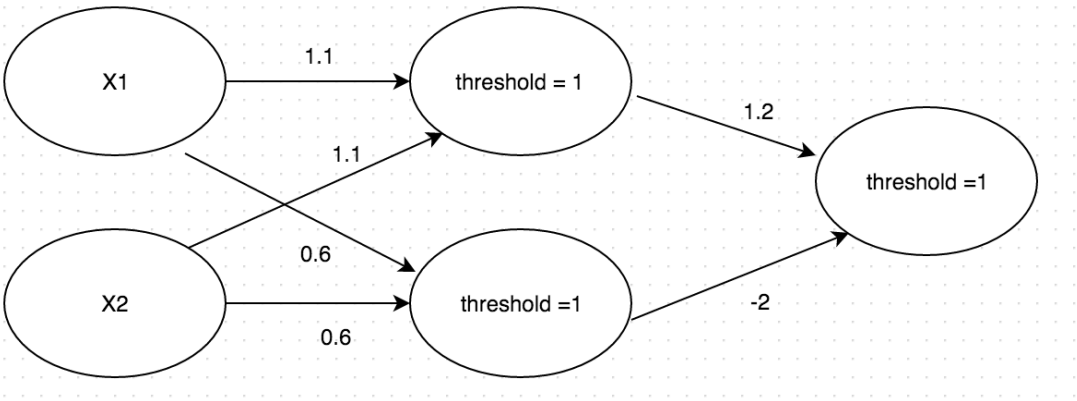
2. (x/1*100) >=3 ,where x is the volume of the cube centered at q. So x >= 0.03. Therefore the minimum side length is sqrt(0.03) = 0.1732051.



3. $\sqrt[100]{0.03} = 0.9655$
4. When applying kNN classifier to data in high-dimensional spaces, nearest neighbors becomes far away, and it becomes really hard classify new data points to the true classes. We can see that when d = 100, the minimum cube size is almost as the unit cube, almost cover the whole data space. And when d = 2, the minimum cube size is still acceptable.
5. Pros: Simple and powerful. Do not need to tune parameters for a model. And no training involved. New training examples can be added easily. When k is large, can do well in prediction.
   Cons: Expensive and slow. We must compute the distance to all training examples to determine the nearest k neighbors of a new point x.

Problem 3

1. Let W = (0.6,0.6), let threshold =1
   If $x_1 = 0$ & $x_2 = 0$ → 0*0.6 + 0*0.6 = 0 → less than 1, y = 0
   If $x_1 = 0$ & $x_2 = 1$ → 0*0.6 + 1*0.6 = 0.6 → less than 1, y = 0
   If $x_1 = 1$ & $x_2 = 0$ → 1*0.6 + 0*0.6 = 0.6 → less than 1, y = 0
   If $x_1 = 1$ & $x_2 = 1$ → 1*0.6 + 1*0.6 = 1.2 → larger than 1, y = 1
2. Let W = (1.1,1.1), let threshold =1
   If $x_1 = 0$ | $x_2 = 0$ → 0*1.1 + 0*1.1 = 0 → less than 1, y = 0
   If $x_1 = 0$ | $x_2 = 1$ → 0*1.1 + 1*1.1 = 1.1 → larger than 1, y = 1
   If $x_1 = 1$ | $x_2 = 0$ → 1*1.1 + 0*1.1 = 1.1 → larger than 1, y = 1
   If $x_1 = 1$ | $x_2 = 1$ → 1*1.1 + 1*1.1 = 2.2 → larger than 1, y = 1
3. Let w = -1, threshold = -0.5
   If x = 0 → 0 * -1 = 0 → larger than -0.5 → y = 1
   If x = 1 → 1 * -1 = -1 → less than -0.5 → y = 0
4. X1 xor X2 = (X1 or X2) and not (X1 and X2)

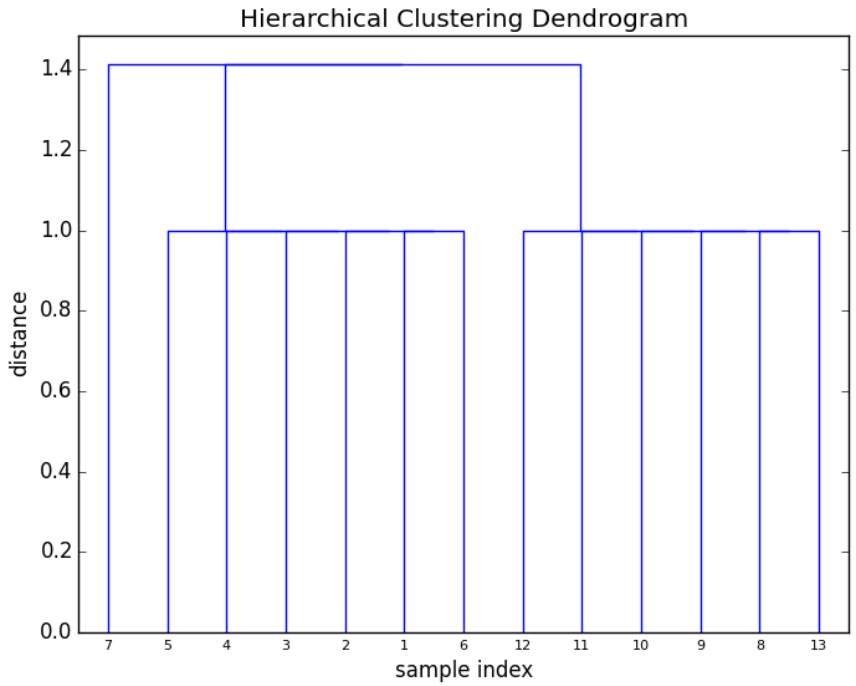


If $x_1 = 0, x_2 = 0 \rightarrow 0 < 1, y = 0$
If $x_1 = 0, x_2 = 1 \rightarrow 1.2 > 1, y = 1$
If $x_1 = 1, x_2 = 0 \rightarrow 1.2 > 1, y = 1$
If $x_1 = 1, x_2 = 1 \rightarrow -0.8 < 1 , y = 0$

5. No, it is not possible to implement XOR using a single layer neural network. As I mentioned, X1 xor X2 = (X1 or X2) and not (X1 and X2). So we need two layers.

Problem 4

1. 23

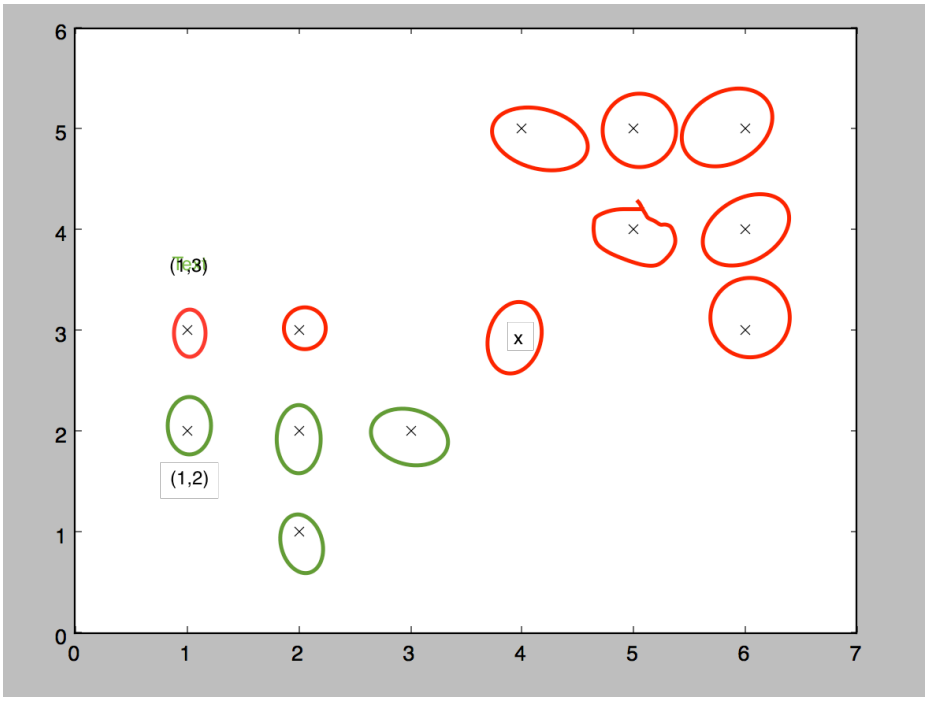


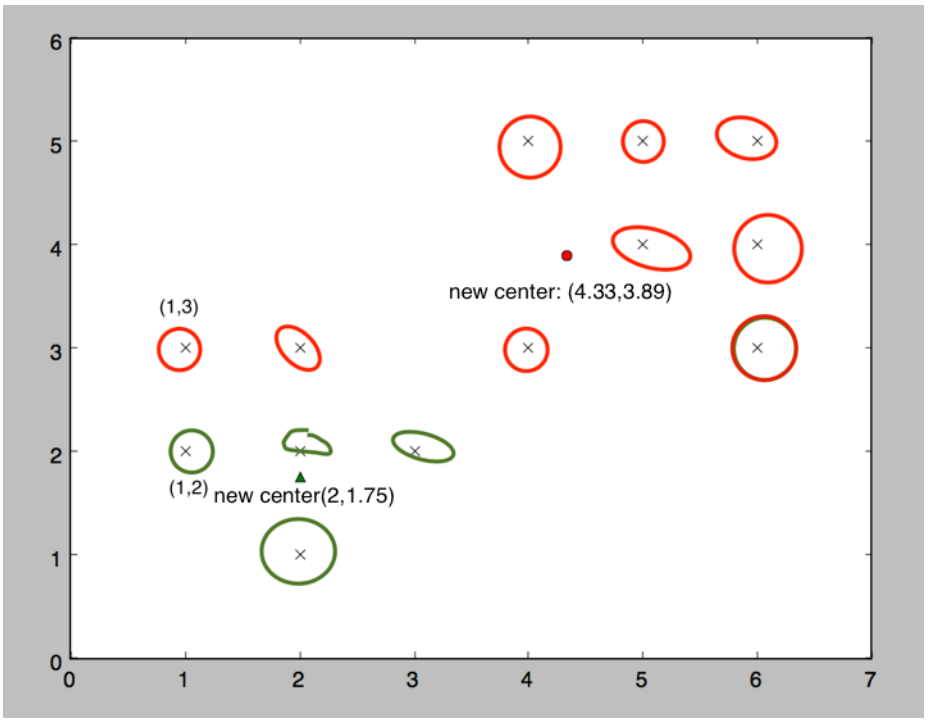
2. Numbers present the points
   Group 1: 7
   Group 2: 1,2,3,4,5,6
   Group 3: 8,9,10,11,12,13
3. Precision: 1/13 * (6 + 6 +1) = 1

Recall: 1/13 * (6 + 1/7 + 6*6/7) = 0.87

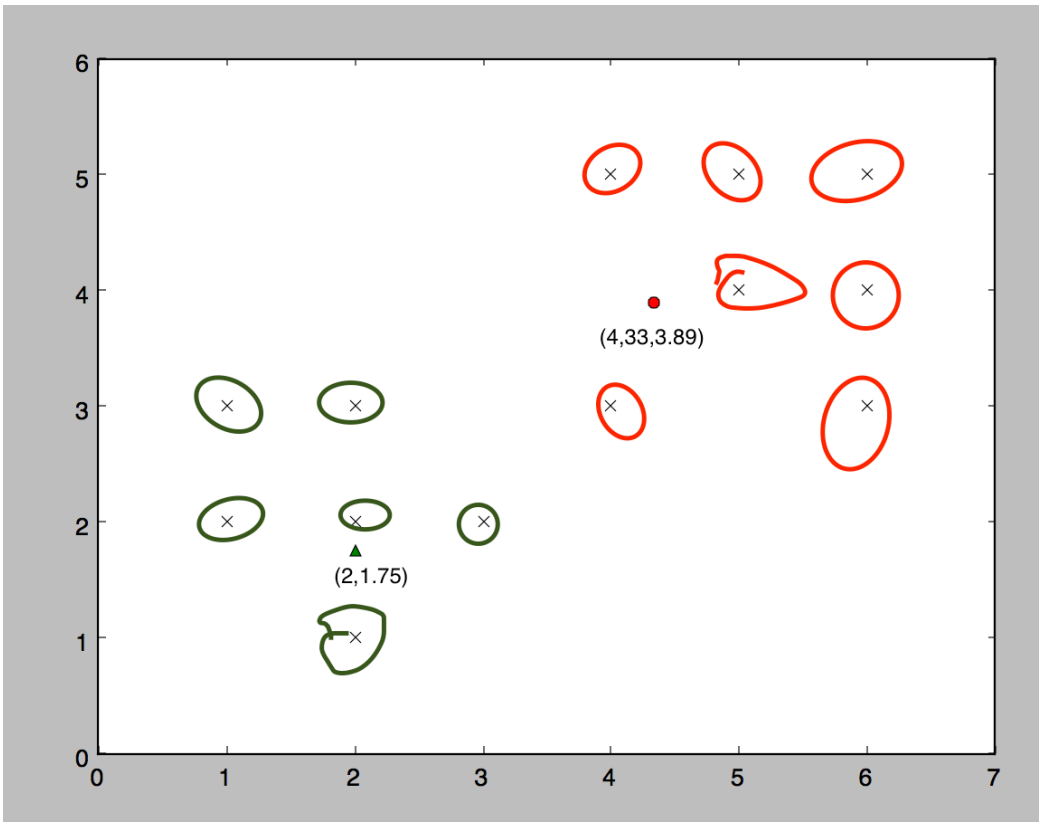
Problem 5

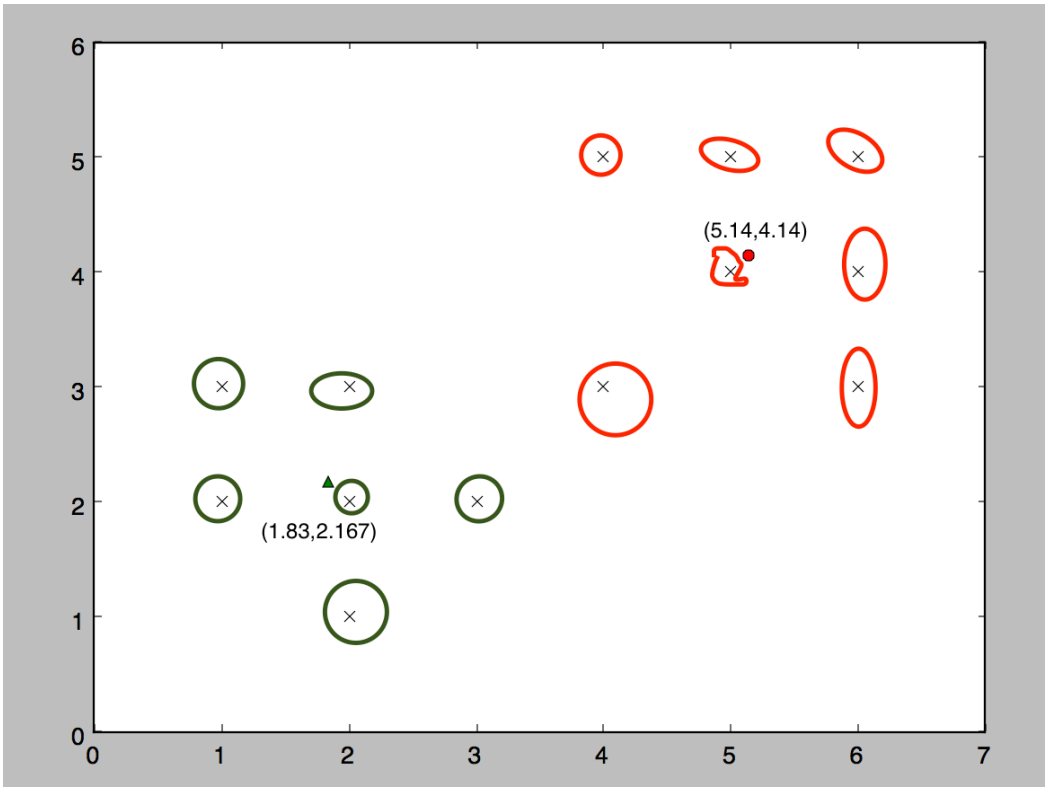
1. Itr = 1



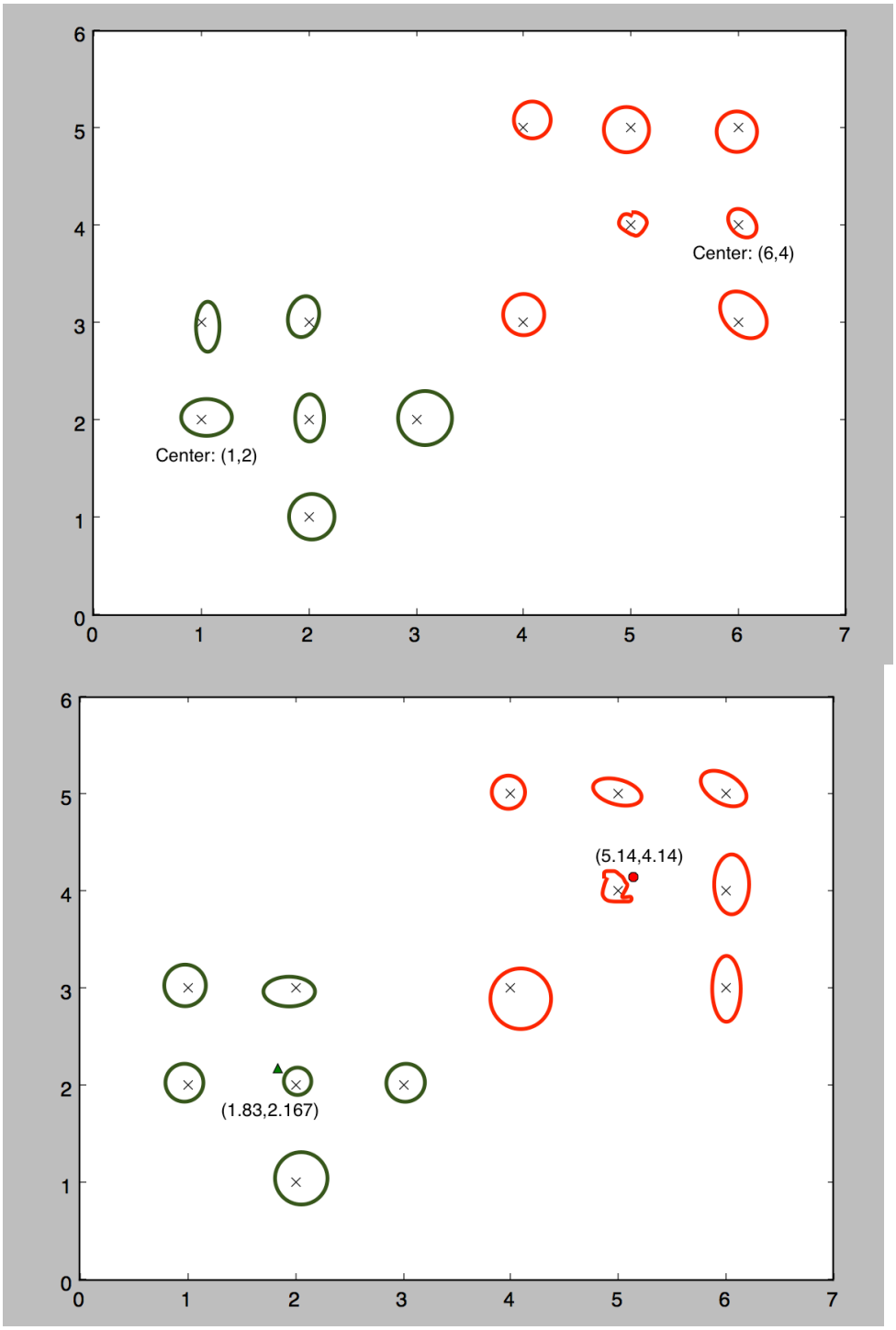


Itr = 2





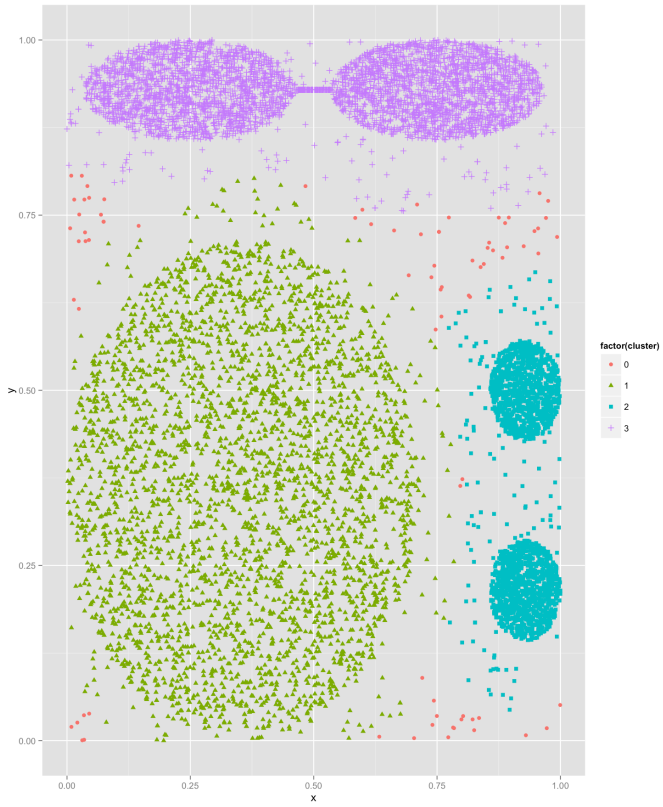
2. Itr 1



3. If k = 2, and all data are 2-dimentional, we should choose 2 points which are most far apart and make them the initial centers. Because if we choose to points which are close to each other, it makes not sense to cluster two groups centered by these two points. So when we choose the two farthest points, it could give us the highest possibility to find 2 dissimilar groups, and also become effective.

Problem 6

1. The scatterplot of my output



The scatterplot of ground truth

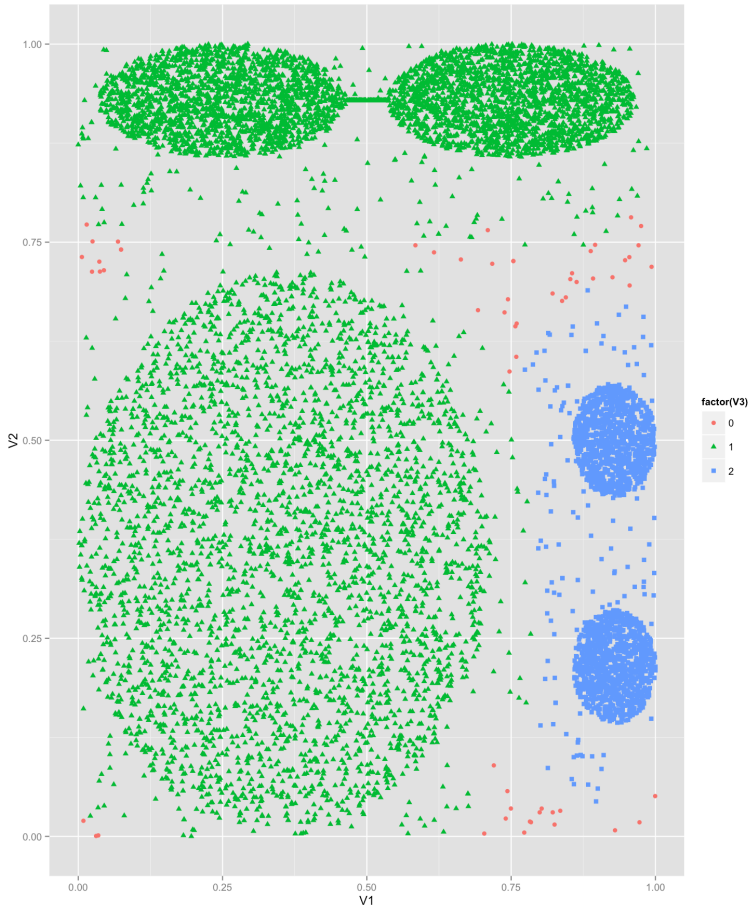
Question to ponder A: For my own python code, it takes less than 1 minute. In the worst case, the time complexity of DBSCAN is $O(n^2)$. In this case, all points within a distance larger than ε. So there are n clusters.
Question to ponder B: Compared with the ground truth, my result is not perfect in a way. Because I only get 3 clusters but the ground truth has 4 clusters. Therefore, some points are misclustered. Given the same MinPts, we should decrease ε to get more intuitive clustering results.

2. Table

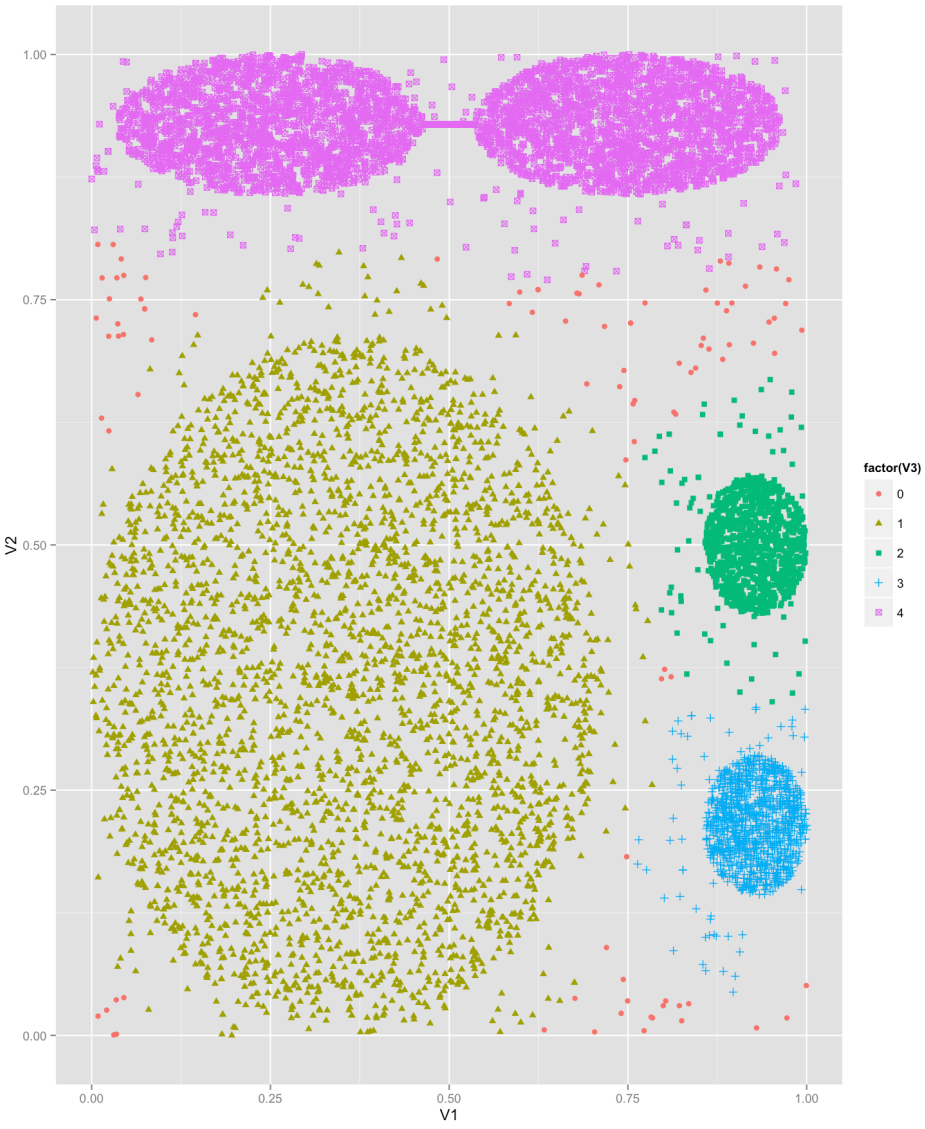
	# of clusters	# of outliers
ε = 0.065	3	82
ε = 0.07	2	60
ε = 0.06	4	97

Question to ponder C: I think ε = 0.065 is still the best. Because in all three experiments, when ε = 0.065, it is closer to the ground truth than others. To choose Minpts, we could choose Minpts >= D +1, where D is the dimensions of the data set. To choose ε, we should choose a small number. And we could plot the distance to the Minpts nearest neighbors for all points, then pick a reasonable small value.

ε = 0.07



ε = 0.06



3. Precision and recall

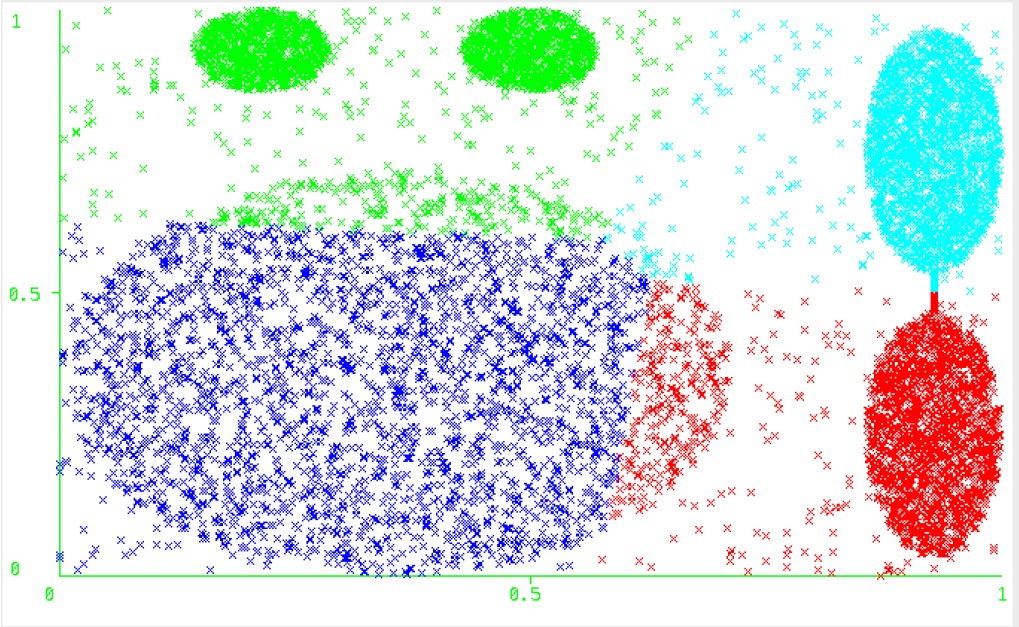
	precision	recall
Step1	0.85	1
Step2a	0.95	1
Step2b	0.97	1

Question to ponder D: We could use F-measure.

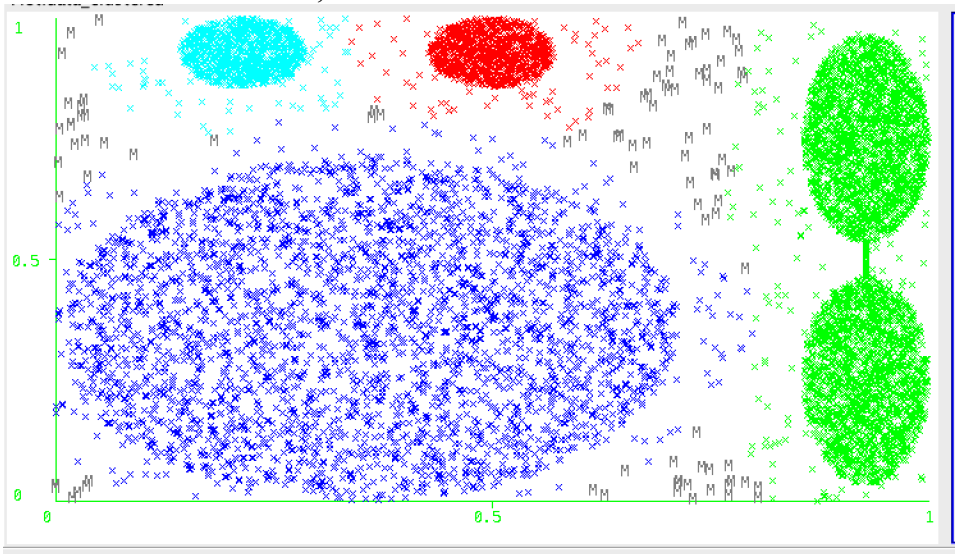
$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

P is precision and R is recall

4. SimpleKmeans: We choose to have 4 clusters



DBSCAN: MinPTs = 25, ε = 0.06



Question to ponder E: DBSCAN is more suitable than K-Means for the provided dataset. Because we can see in ground truth, there are noisy points, but K-means is not very good at detecting them. So DBSCAN is more suitable.