Homework 3

Xiao Wang

October 17, 2014

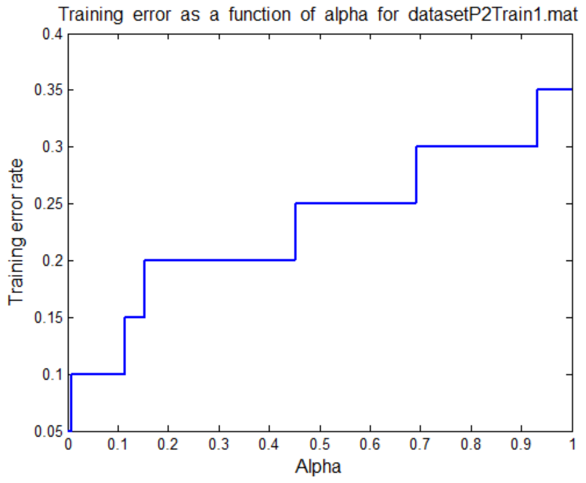
1

(a)

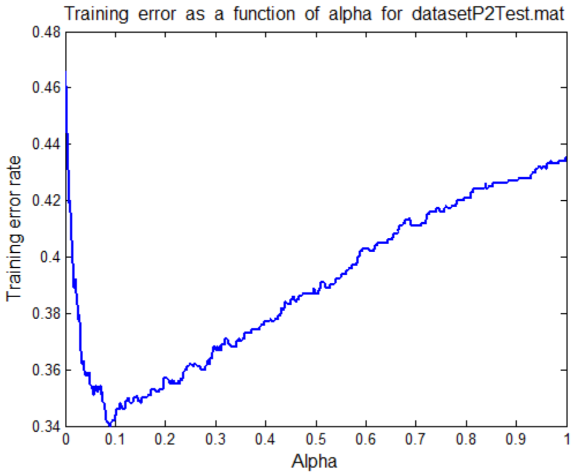
```
class1Mean =

    0.7969    0.4314   -1.4372


class2Mean =

    1.5947    4.1320   -3.6648


class3Mean =

    0.2426   -0.0509   -0.0850
```

```
class1Cov =

    2.5570   -0.5811    0.3888
   -0.5811    2.2111    2.1948
    0.3888    2.1948    2.9630


class2Cov =

    0.8019   -0.4215    0.1403
   -0.4215    2.3465   -0.3924
    0.1403   -0.3924    0.7130


class3Cov =

    8.4547   -4.7866    4.7287
   -4.7866    9.3591   -4.6097
    4.7287   -4.6097   11.0588
```

(b)



(c)



Answer:

The amounts of data in datasetP2Test.mat is much more than datasetP2Train1.mat. Therefore, plot (c) is much more smooth than plot (b). Plot (b) is a step function with its lowest point at the very beginning of α ($\alpha = 0$), and plot (c) is a curve with its lowest point at $\alpha = 0.089$.

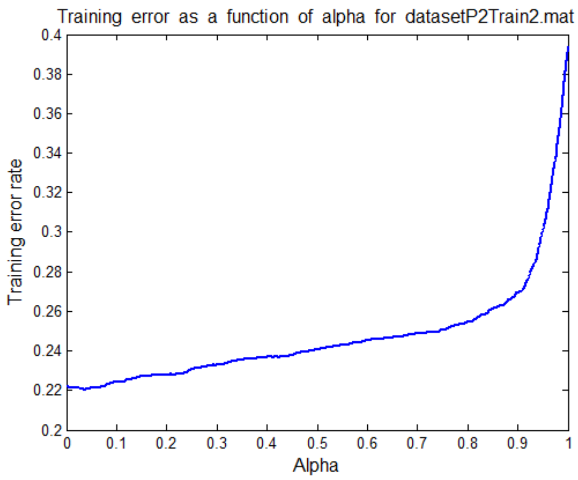
For plot (b), to yield best test performance, we should make α as small as possible (set $\alpha = 0$). For plot (c), to yield best test performance, we should set $\alpha = 0.089$.

(d)

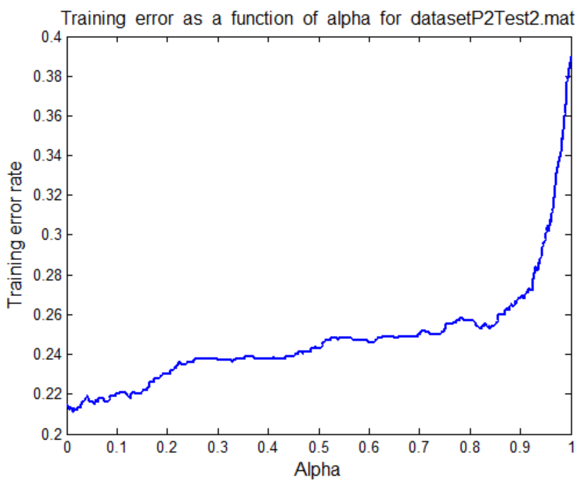
```
class1Mean =

   -0.0110    0.0054    0.0144

class2Mean =

    0.9931    4.9504   -2.9815

class3Mean =

    0.0254    0.0571   -0.0779
```

```
class1Cov =

    2.8264    0.0296   -0.0082
    0.0296    4.9327   -0.0831
   -0.0082   -0.0831    1.9919

class2Cov =

    1.0340    0.0419    0.0434
    0.0419    3.8829    1.0381
    0.0434    1.0381    5.9733

class3Cov =

    9.7938   -0.0261    0.1293
   -0.0261    9.1645    0.0610
    0.1293    0.0610   10.2140
```

(e)


Training error as a function of alpha for datasetP2Train2.mat
Training error rate
Alpha
0.4
0.38
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(f)


Training error as a function of alpha for datasetP2Test2.mat
Training error rate
Alpha
0.4
0.38
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Answer:

Speaking of overall shape, plot (e) and plot (f) is almost the same. However, plot (e) is more smooth than plot (f) because the amounts of data in datasetP2Train2.mat is much more than datasetP2Test.mat.

For plot (e), to yield best test performance, we should set $\alpha = 0.036$. For plot (f), to yield best test performance, we should set $\alpha = 0.013$.

For part (c), after shrinkage, the α which yields best test performance changes from 0 to 0.089, and the curve shape changes a lot. But for part (f), after shrinkage, the α which yields best test performance changes from 0.036 to 0.013, and the curve shape stays the same. Therefore, if the amounts of data in training group is less, the shrinkage would have a bigger influence on the result of test performance.

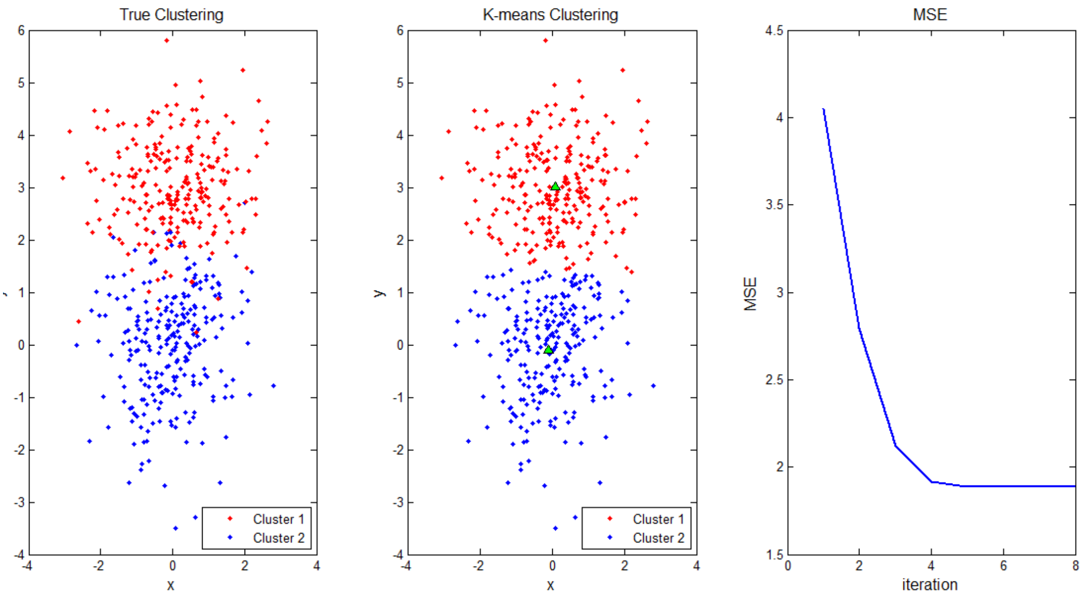
For small amounts of data, reasonable shrinkage will effectively help us to decrease the training error rate and yield best test performance. However, for big amounts of data, shrinkage will have little influence on the result.

2

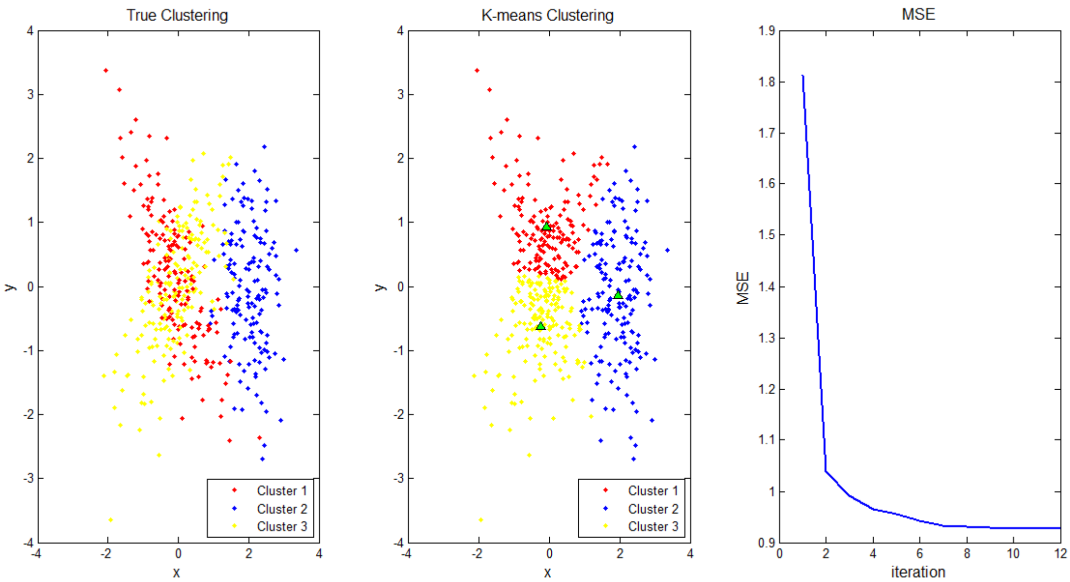
(a)

K-means function is in appendix.

(b)



(c)

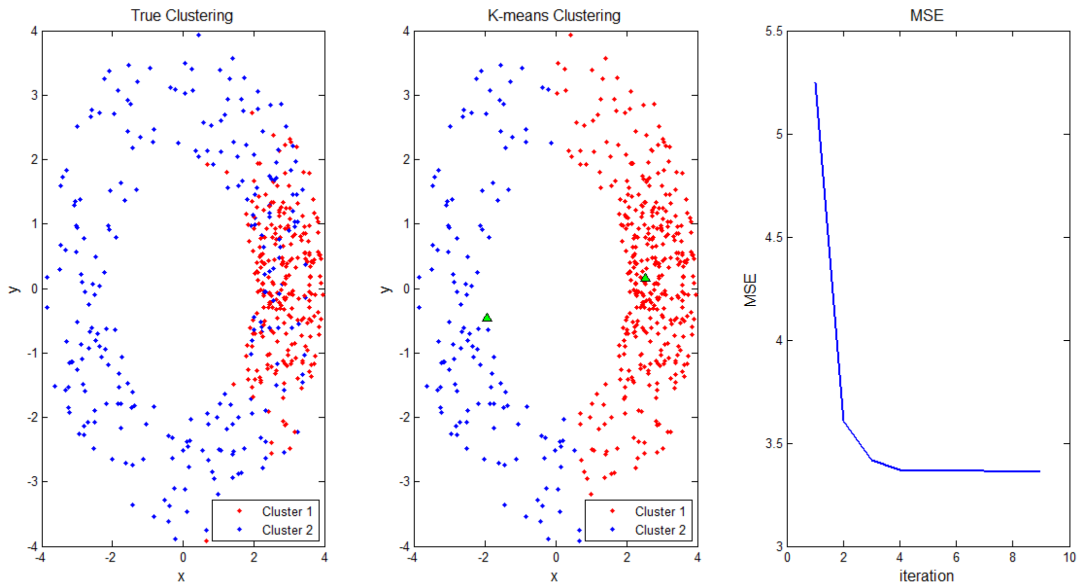


Answer:

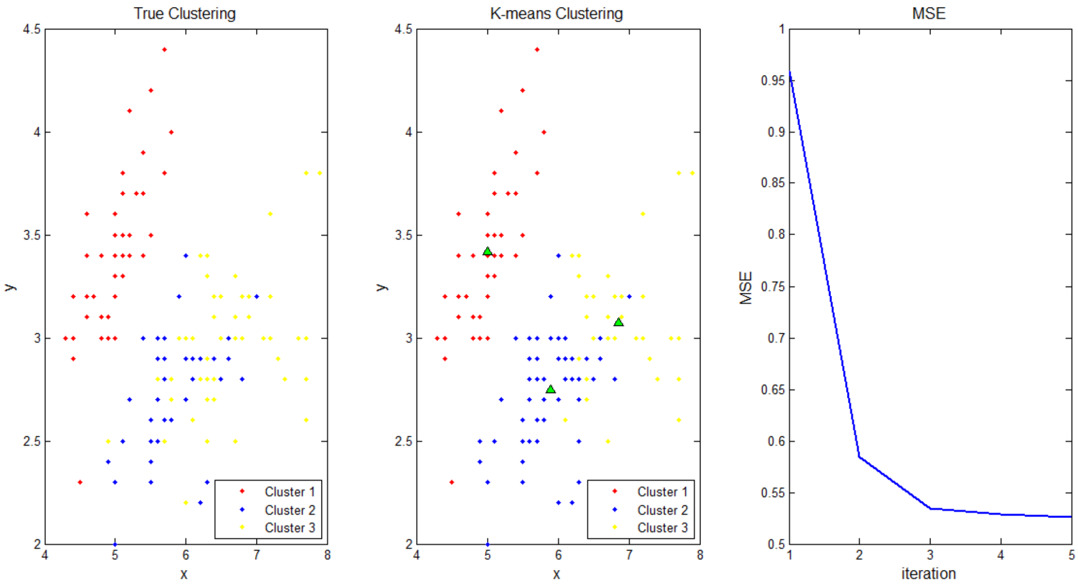
As we know, K-means is a clustering algorithm and clustering algorithms are unsupervised learning techniques for sub-dividing a larger dataset into smaller groups. The term "unsupervised" indicates that the data do not originate from clearly defined groups that can be used to label them a priori.

Therefore, K-means performs not very good under the situation that two or more groups have very similar means, especially when the data of a group are very decentralized from the mean, like in a long strip shape.

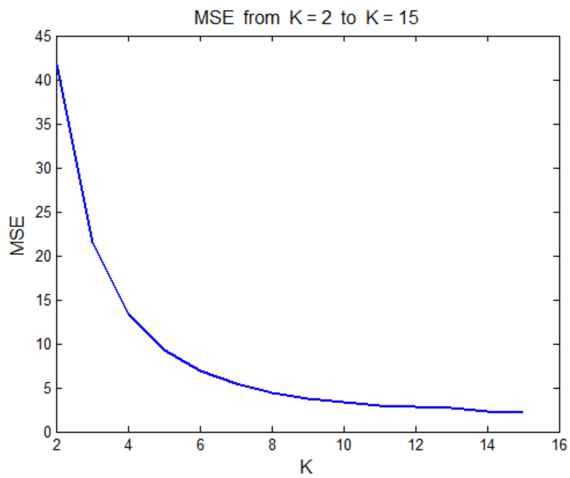
(d)


True Clustering
K-means Clustering
MSE
Cluster 1
Cluster 2
x
y
iteration

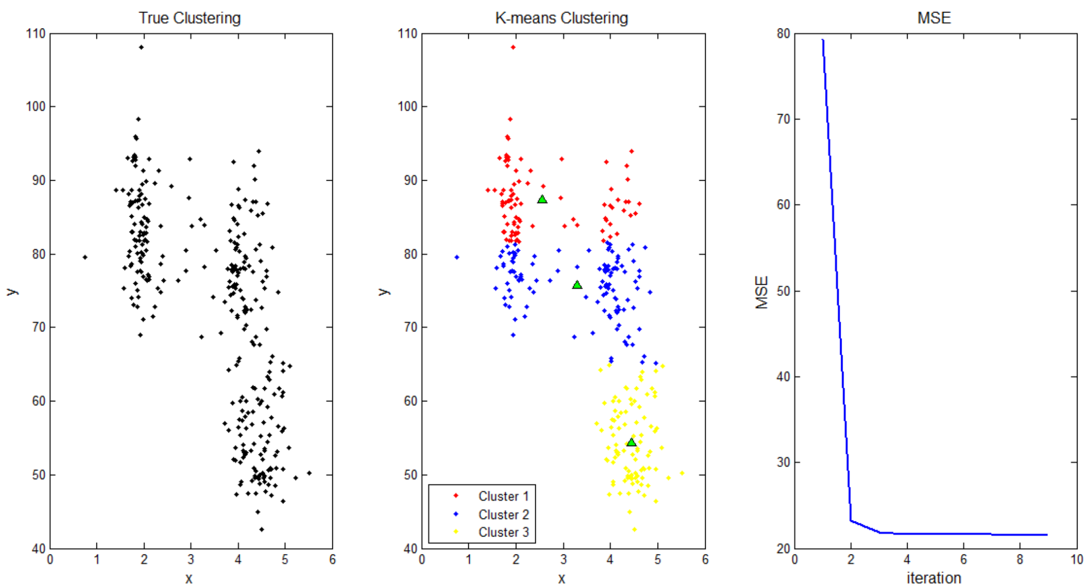
(e)


True Clustering
K-means Clustering
MSE
Cluster 1
Cluster 2
Cluster 3
x
y
iteration

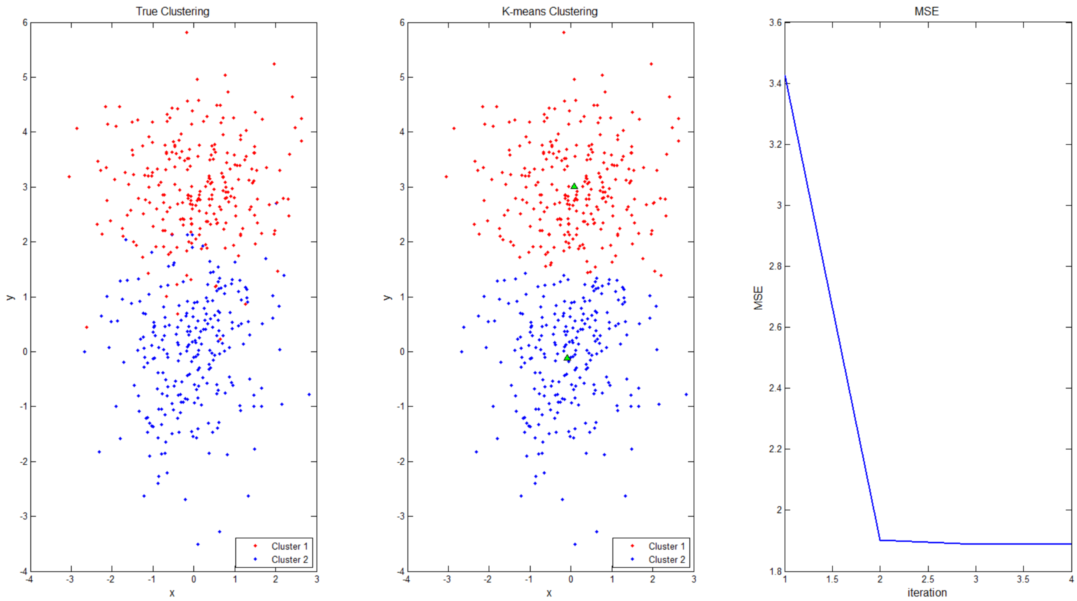
(f)

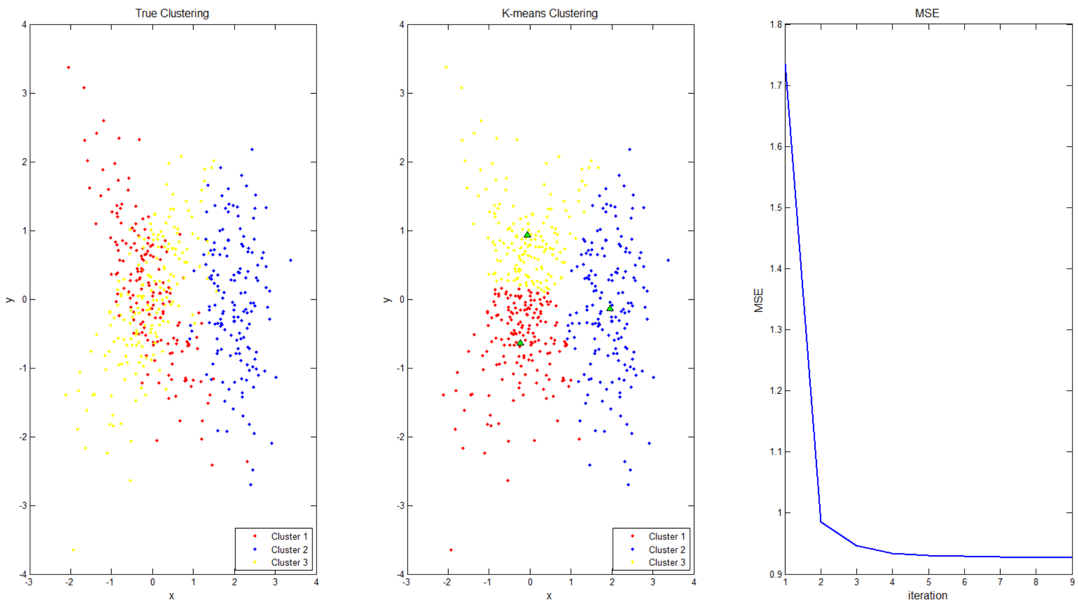


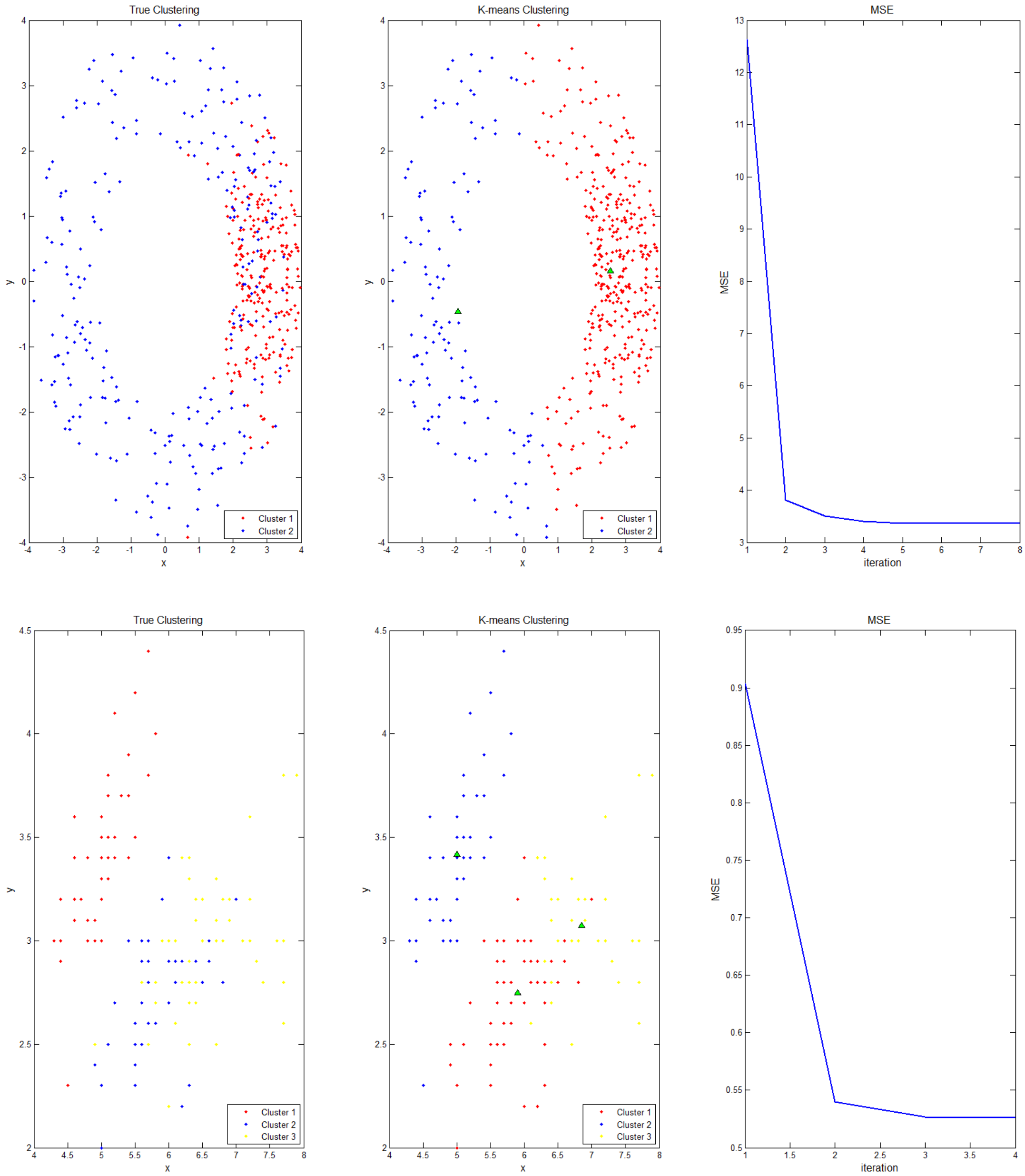
According to the figure above, we choose K = 3 as 'the knee', for which the rate at which the within-cluster MSE is decreasing sharply reduces.



(g) Bonus: implement the K-means++ algorithm to generate the plots from (b) - (e).







Answer:

The clustering performance stays the same when compared the K-means++ with the standard K-means (numberOfRuns = 10). However, the convergence speed of K-means++, generally speaking, is faster than K-means. In other words, K-means++ terminates faster while achieving a lower potential value. Because the K-means++ algorithm uses a smart and careful seeding technique. It makes the initial center points as far as possible from each other, which effectively avoids from merging clusters together.