111061702 常安彥

Pattern Recognition ex4

1. EM algorithm

   Sigma: [1 0.4; 0.4 1], [1 -0.6; -0.6 1], [1 0; 0 1]

   Initialization: mu = [2 2; 6 6; 12 2]

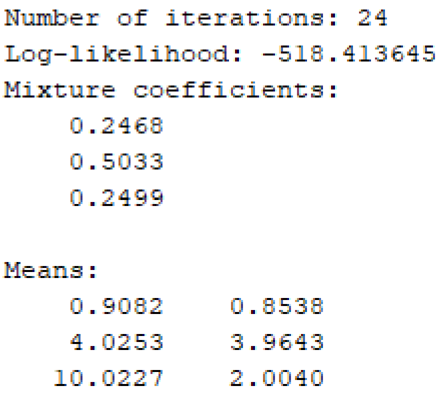
   sigma = [2 0; 0 2], [2 0; 0 2], [2 0; 0 2]

   Stopping threshold: convergence tolerant = 1e-6

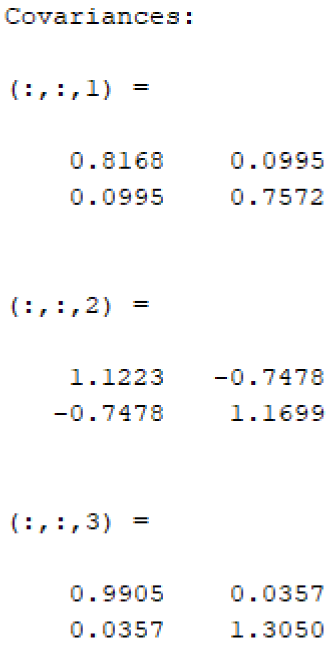
   if (log likelihood – last log likelihood > tol)

   or if iteration >= 100

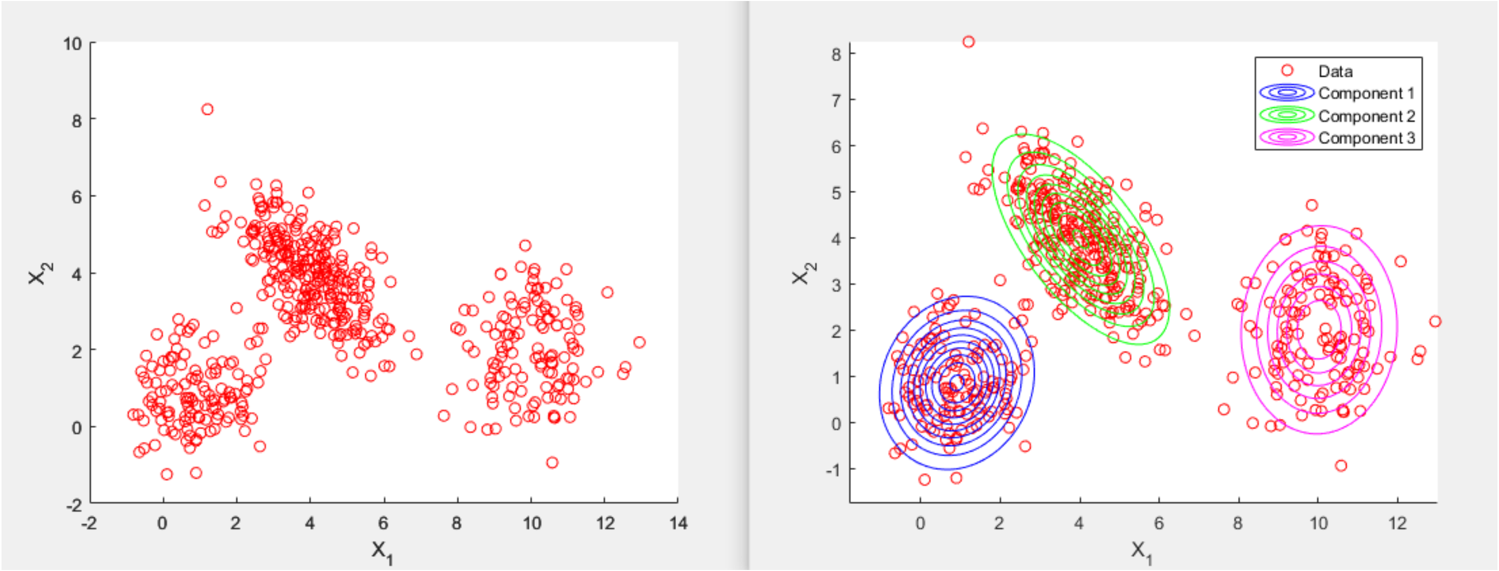
   (a) [1 1], [4 4], [10 2]



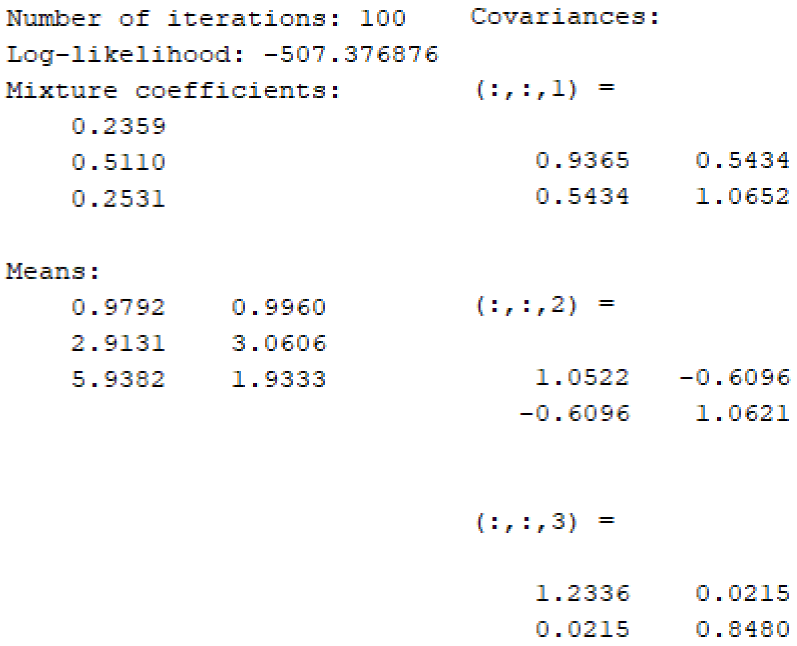
```
Number of iterations: 24
Log-likelihood: -518.413645
Mixture coefficients:
    0.2468
    0.5033
    0.2499

Means:
    0.9082    0.8538
    4.0253    3.9643
   10.0227    2.0040
```



```
Covariances:

(:,:,1) =

    0.8168    0.0995
    0.0995    0.7572


(:,:,2) =

    1.1223   -0.7478
   -0.7478    1.1699


(:,:,3) =

    0.9905    0.0357
    0.0357    1.3050
```

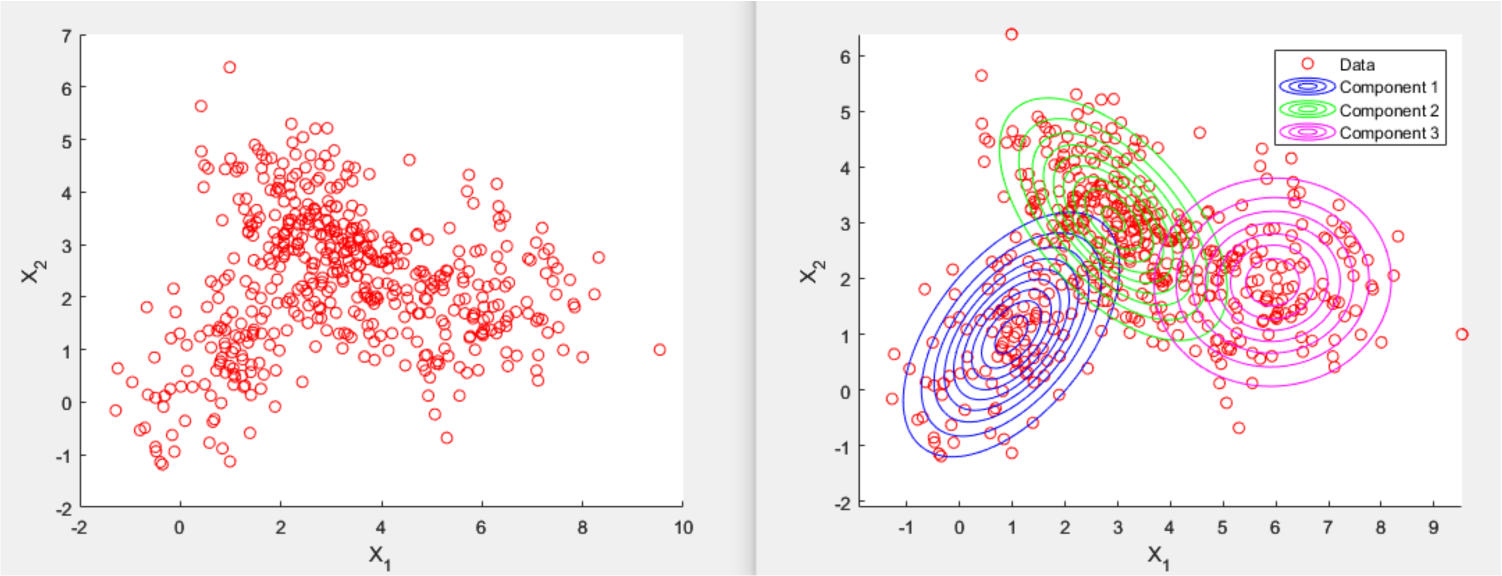


(b) [1 1], [3 3], [6 2]

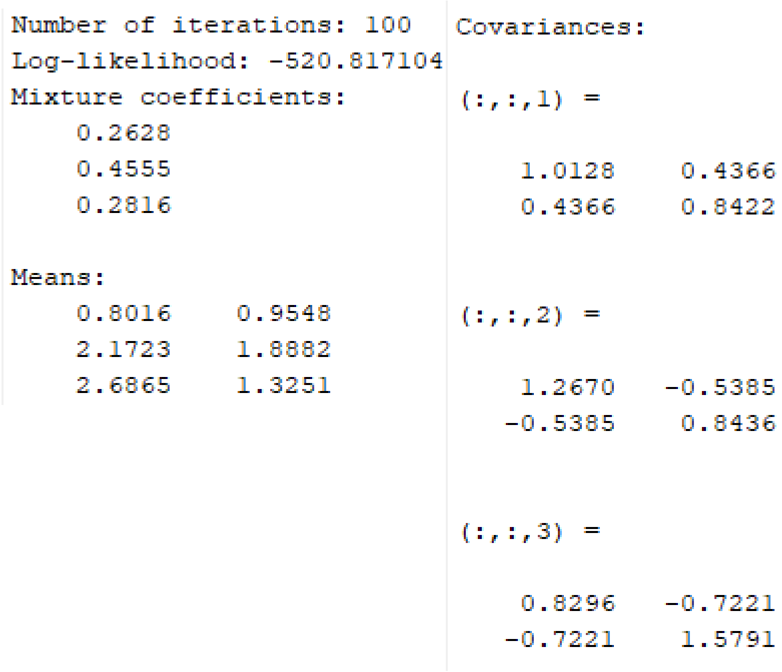


```
Number of iterations: 100
Log-likelihood: -507.376876
Mixture coefficients:
    0.2359
    0.5110
    0.2531

Means:
    0.9792    0.9960
    2.9131    3.0606
    5.9382    1.9333
```

```
Covariances:

(:,:,1) =

    0.9365    0.5434
    0.5434    1.0652


(:,:,2) =

    1.0522   -0.6096
   -0.6096    1.0621


(:,:,3) =

    1.2336    0.0215
    0.0215    0.8480
```

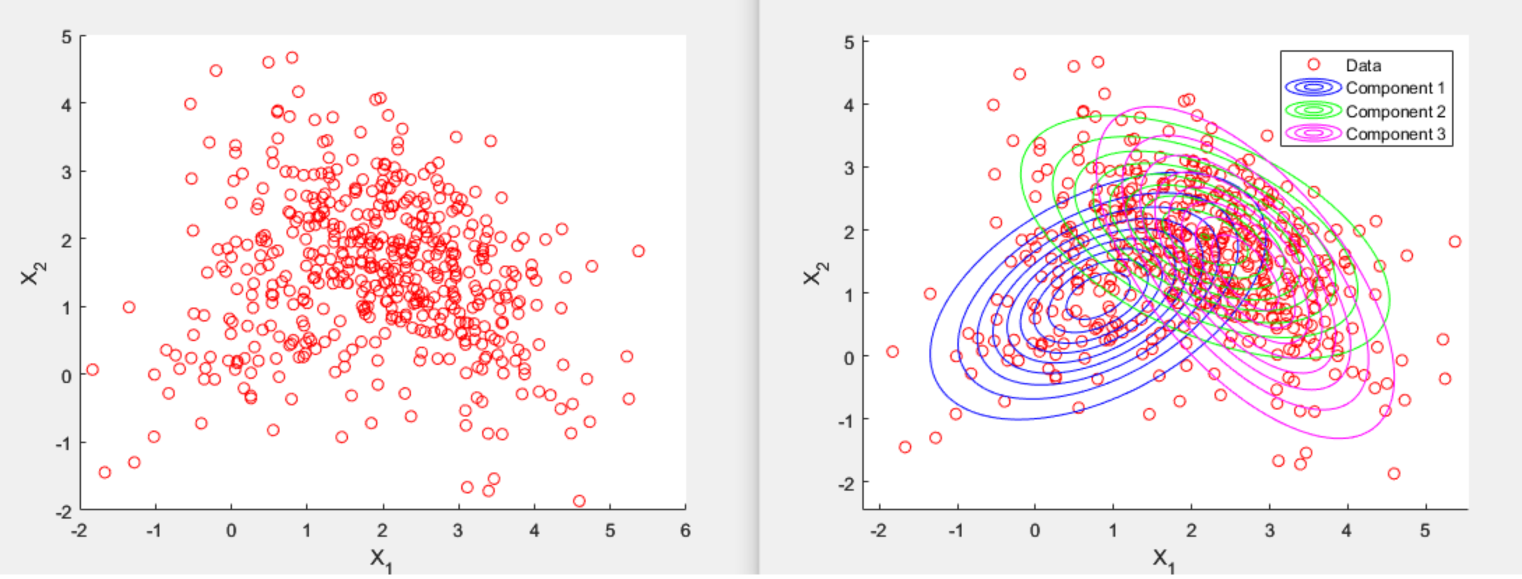


(c) [1 1], [2 2], [3 1]



```
Number of iterations: 100
Log-likelihood: -520.817104
Mixture coefficients:
    0.2628
    0.4555
    0.2816

Means:
    0.8016    0.9548
    2.1723    1.8882
    2.6865    1.3251
```

```
Covariances:

(:,:,1) =

    1.0128    0.4366
    0.4366    0.8422


(:,:,2) =

    1.2670   -0.5385
   -0.5385    0.8436


(:,:,3) =

    0.8296   -0.7221
   -0.7221    1.5791
```



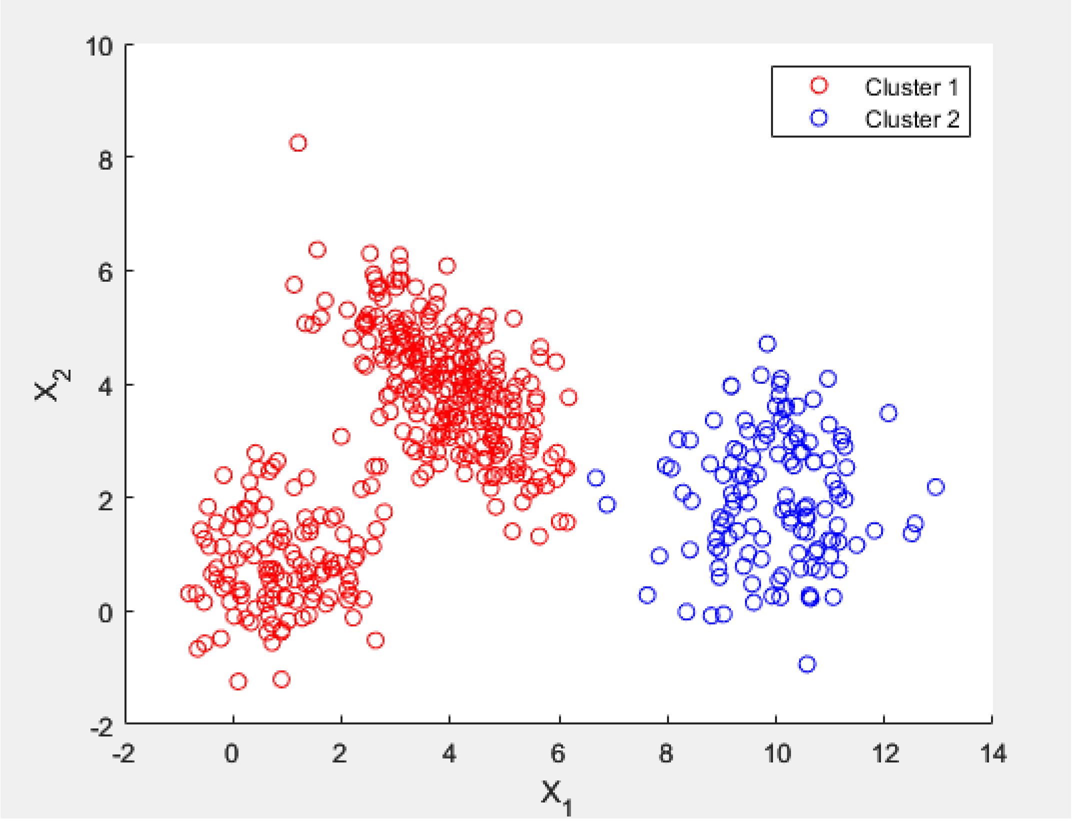
(d) Comparison

The difficulty is obviously increasing according to the scatter plots, which shows that the classes are significantly closer to each other.

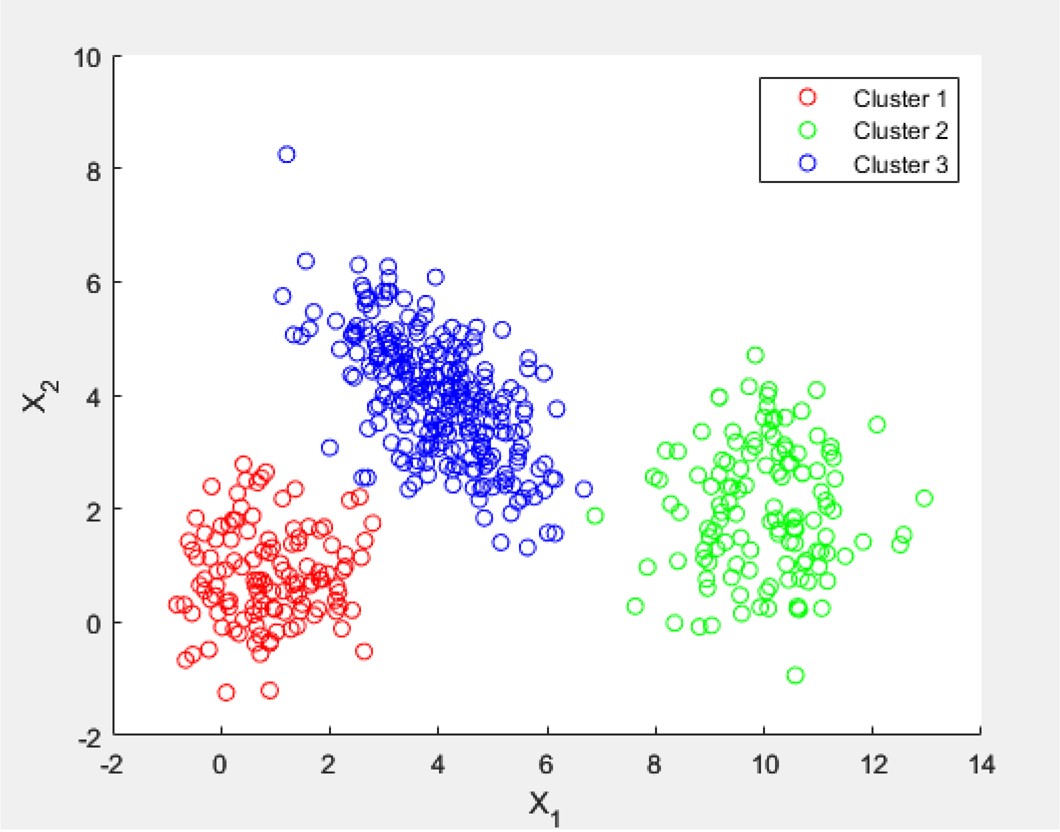
In all cases, the estimated means and covariances are all close to the true means and covariances, which indicate the good performance of EM algorithm. Nevertheless, the performance still became worse in case (c). Since the three classes stick together, the algorithm struggled clustering.

2. K-means

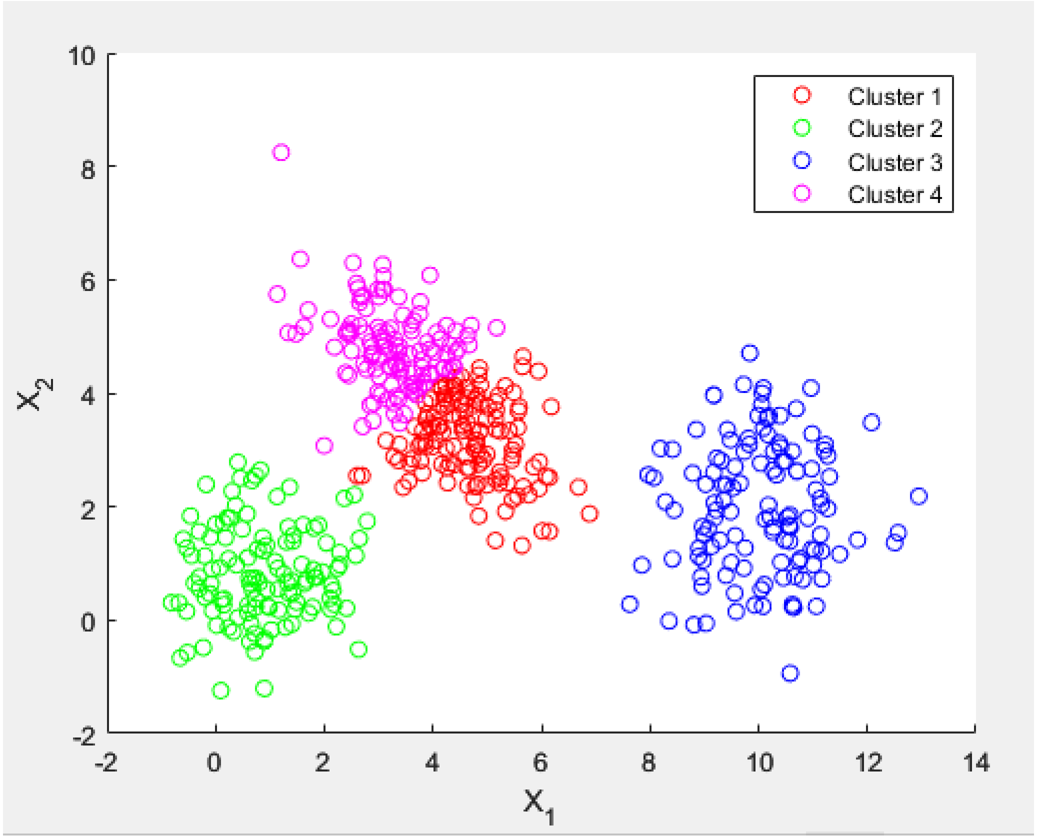
(a) K = 2



(b) K = 3



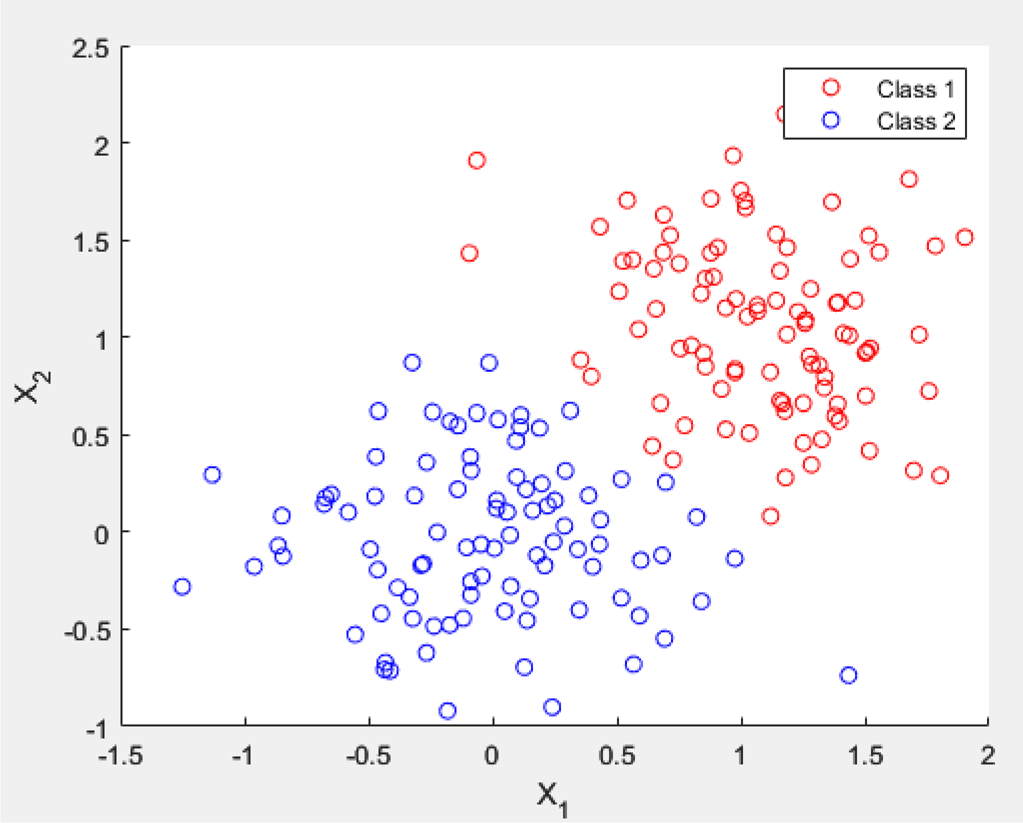
(c) K = 4



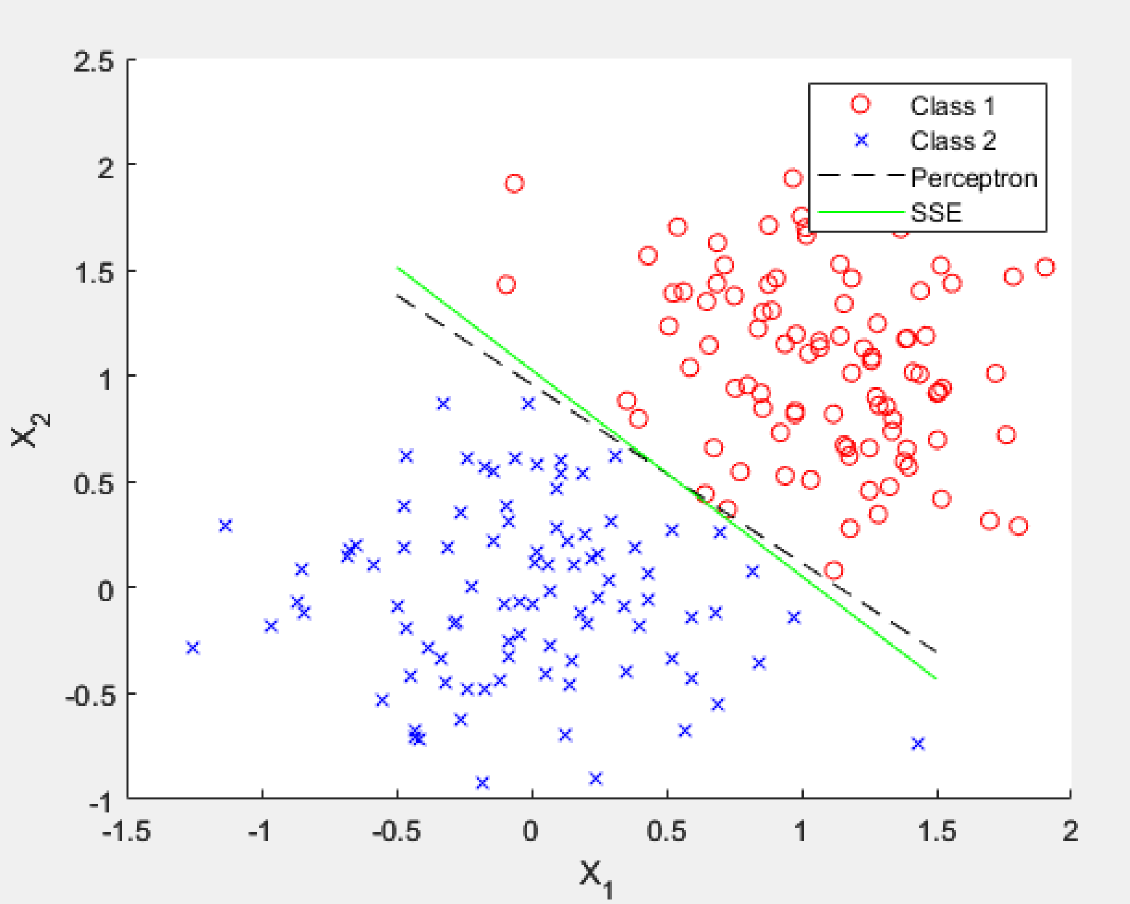
Comparison

K-means with k = 3 separated the clusters well, but with k =2 or 4, the results are less satisfactory. The prior one merged two clusters, and the posterior one split one into two.

3. Linear Discrimination



(a, b) Perceptron, Sum-of-Squared-Error Classifier

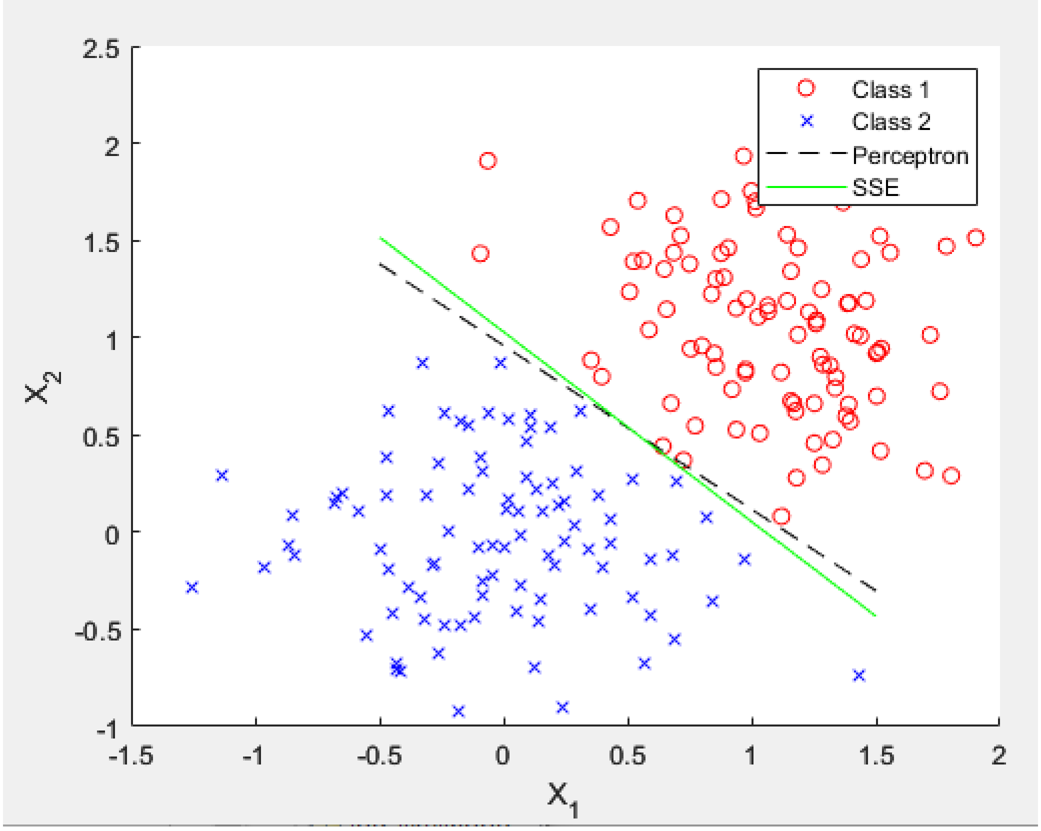


(d) Regularization: set a ‘Lambda’ for the loss calculation.

loss = ((1-ƛ)*sum(err.^2) + ƛ *sum(SSE_w.^2)) / n;

Early Stopping: set an ‘Epsilon’ for early stopping when loss is smaller than it.

Learning rate decay: set a ‘decay rate’ to decrease the learning rate over time for converging.



The result show tiny difference. Since the data is a simple case, and I iterated to 1000 iterations, the classifiers converged to similar results is reasonable.