Assignment 5

Computational Intelligence, SS2018

Team Members		
Last name	First name	Matriculation Number
Lee	Eunseo	11739623
Shadley	Alex	11739595
Lee	Dayeong	11739321

1 Classication/ Clustering

1.1 2 dimensional feature

1.1.1 Perform all of the above-mentioned tasks for the EM algorithm.

In the process of initializing the parameters, we used random function to select m points for calculating means. Therefore, the result is different for each process. We selected the best results from 10 trials.

- Compare the result with the labeled data set (i.e., consider labels as well). Make a scatter plot of the data and plot the Gaussian mixture model over this plot.

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2], dtype=int64)
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

Figure 1: label data

First array is EM algorithm classification. Second array is answer classification.
Three points are mis-classified.

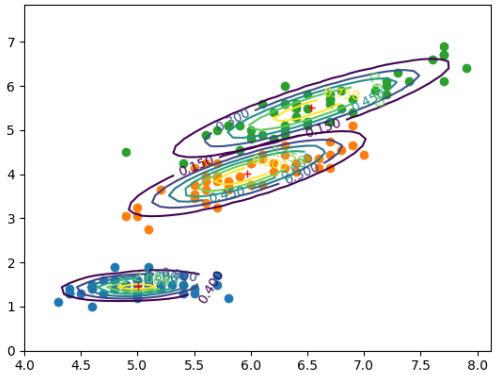


Figure 2: Three gaussian with scatter data points

EM algorithm suceeded to find three gaussian model and classified the data points well.

- For your tests, select the correct number of components (K = 3), but also check the result when you use more or less components. How do you choose your initialization $\theta 0$? Does this choice have an inuence on the result

  When the number of components is 2, the result is the following figure.

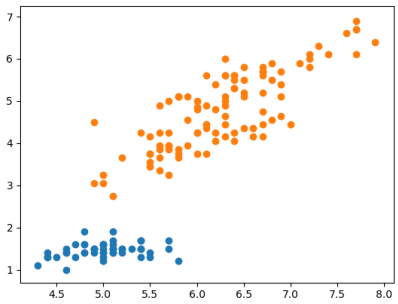


Figure 3: K = 2

When the number of components is 4, the result is the following figure.

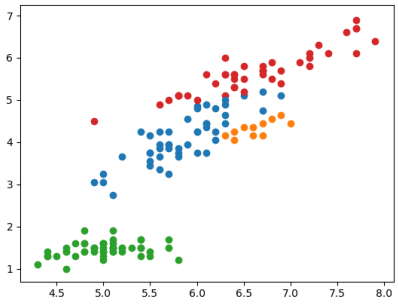


Figure 4: K = 4

For initialization $\theta 0$, we refered to the class pdf. The following figure is the reference class pdf.

4.2.1 Initialisierung

Eine Möglichkeit $\boldsymbol{\Theta}^0$ zu initialisieren ist:

1. α_m^0 auf uniforme Verteilungsfunktion $\alpha_m^0 = \frac{1}{M}$

2. $\boldsymbol{\Sigma}_m^0$ wird auf die Kovarianzmatrix $\boldsymbol{\Sigma}$ der Daten $\mathbf{X}$ gesetzt d.h. $\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$ wobei $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$.

3. Für $\boldsymbol{\mu}_m^0$ wählt man m Samples zufällig aus oder man verwendet den k-means Algorithmus.

Figure 5: The initialization process in the class pdf

The initialization process influences the EM algorithm result. When the randomly selected points, for calculating mean value, is well chosen, the result is accurate. That is, the result is accurate when the first random selected points are from first answer label group, the second random selected points are from second answer label group and the third random selected points are from third answer label group.

- plot the log-likelihood function over the iterations! What is the behavior of this function over the iterations?

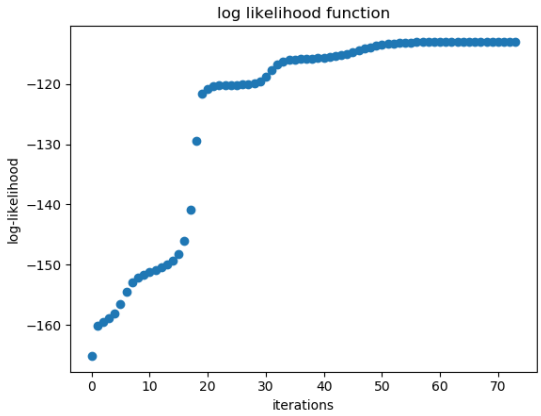


Figure 6: The log-likelihood function over iterations

As shown in Figure 4, the log-likelihood increases over iterations. That is, likelihood increased over iterations. And about 50th iteration, the function looks convering to the value, -112.96270776709655. Therfore, the process stops even though it didn't reach the max iteration number.

- Make a scatter plot of the data that shows the result of the soft-classication that is done in the E-step

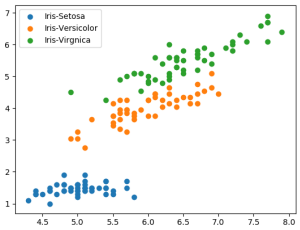


Figure 7: The EM algorithm soft-classification

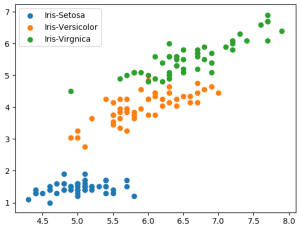


Figure 8: The answer classification

The EM algorithm classifies well the points when it is compared with the answer classification. EM algorithm fails to classify the points near the boundary of iris-Versicolor and iris-Virgnica.

1.1.2 Perform all of the above-mentioned tasks for the K-means algorithm

The algorithm for K-means initialization selects N different points at random from the dataset as the starting positions for N clusters. This ensures that all centers classify at least one point during the first iteration, producing better results.

- Compare the result with the labeled data set (i.e., consider labels as well). Make a scatter plot of the data and plot the Gaussian mixture model over this plot.

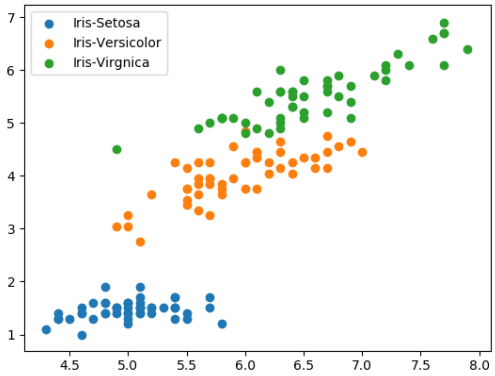


Figure 9: Original Data Labels

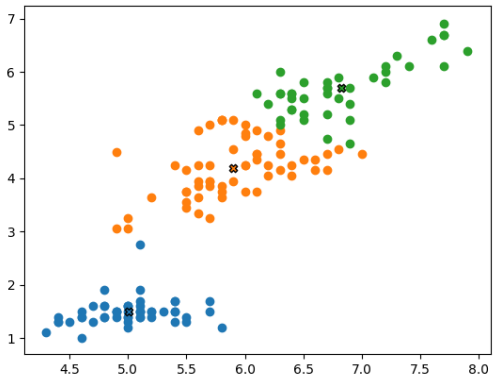


Figure 10: K-means clustering with 3 centers

Ultimately the K-means algorithm misclassified 13 samples, considerably worse than the EM algorithm, which we hypothesize is largely due to the limited explanatory power of the K-means approach. Which is to say, since K-means classifies solely on euclidean distance, it lacks the finesse of the EM algorithm, which can employ a covariance matrix to describe more complex structures.

- For your tests, select the correct number of components (K = 3), but also check the result when you use more or less components. How do you choose your initialization $\theta 0$? Does this choice have an inuence on the result

  The following image depicts a K-means clustering with 2 centers.

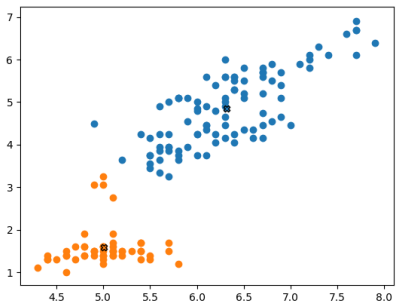


Figure 11: K = 2

With 4 centers, K-means produces the following.

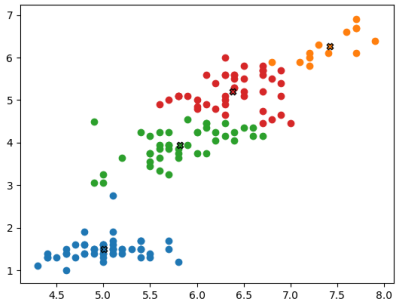


Figure 12: K = 4

The result of more or fewer clusters than 3, while clearly not ideal, is nonetheless fairly logical. One can see that, as more clusters are added, what was originally a single cluster begins to be broken up into smaller, similarly sized and shaped clusters. I also believe that K-means is a better algorithm for when distances between clusters are larger, since the algorithm seems to struggle especially with tightly packed clusters.

For initialization θ_0, we selected at random as many points from the dataset as we needed centers, and used those selected points as the starting centers. This has the beneficial for several reasons: first, it scales without modification for any number of centers less than the number of points, and second, it does not require the programmer to have any prior knowledge of the dataset they are analyzing. This approach also ensures that every cluster is responsible for at least one point during the first iteration.

- plot the cumulative distance over the iterations! What is the behavior of this function over the iterations?

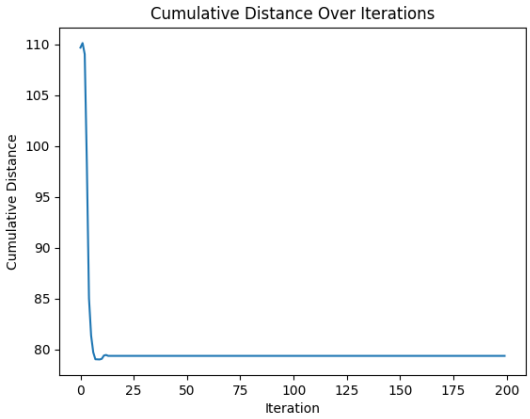
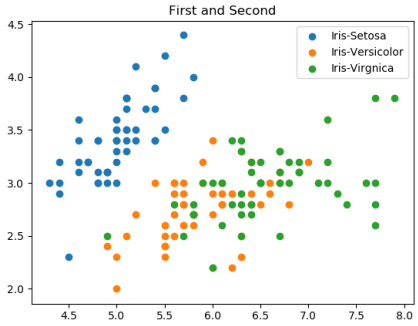


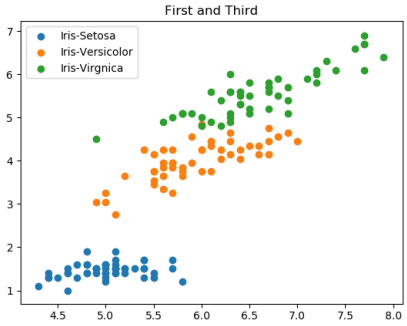
Figure 13: The cumulative distance over iterations

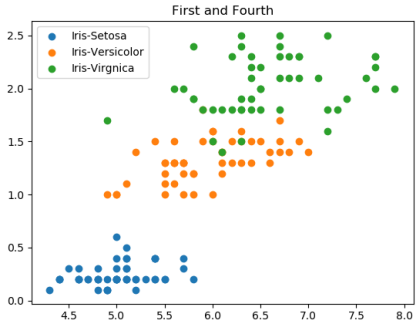
The cumulative distance sharply decreases as the algorithm starts, indicating a quickly improving result, and shortly converges onto a solution. Curiously, the cumulative distance does occasionally increase, indicating that across some iterations the result actually becomes slightly worse.

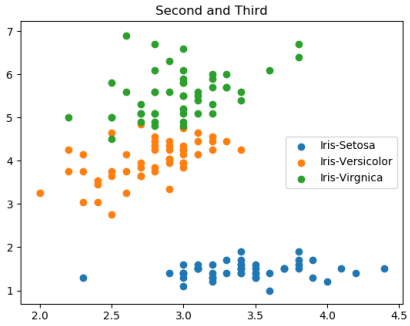
1.1.3 You may additionally choose any other pair of features; how would this change the classication accuracy

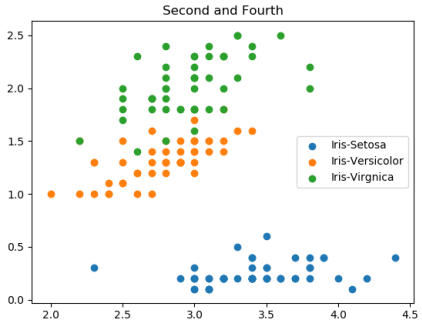
The following figures are the scatter plots when two of four data columns are selected.

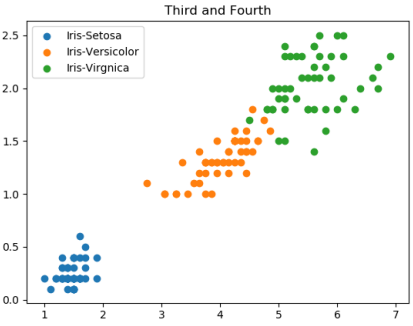












As shown in above figures, when first and third column are selected, we can draw a line to seperate the Iris-Virgnica and the Iris-Versicolor. That is, it is easy to differentiate these two groups. When other columns are selected, these two groups are mixed together. Therfore, EM algorithm also can not classify these two groups as different groups. For example, the worst one is when first and second columns are selected. When first and second columns are selected, the result is the following graph.

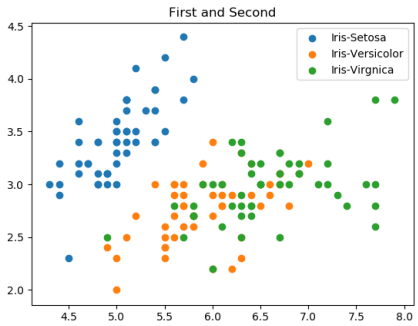


Figure 14: The answer classfication

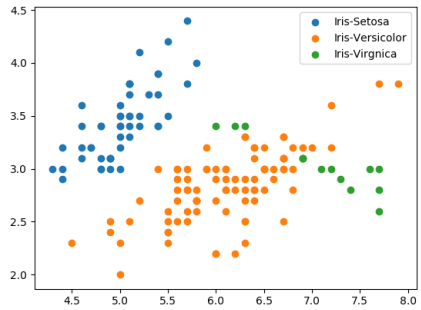


Figure 15: EM algorithm classification

As expected, the EM alogirthm result is not classified well when first and second columns are selected. That is, the EM algorithm accuracy decreased.
Therfore, the column selection influences the EM algorithm accuracy.

1.2 4 dimensional feature

1.2.1 EM algorithm tasks

- Compare the result with the labeled data set (i.e., consider labels as well). Make a scatter plot of the data and plot the Gaussian mixture model over this plot.

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int64)
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

Figure 16: label data

First array is EM algorithm classification. Second array is answer classification.
Four points are mis-classified.

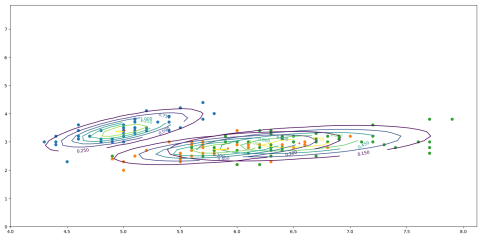


Figure 17: Three gaussian with scatter data points

EM algorithm suceeded to find three gaussian model and classified the data points well.

- For your tests, select the correct number of components (K = 3), but also check the result when you use more or less components. How do you choose your initialization $\theta 0$? Does this choice have an inuence on the result When the number of components is 2, the result is the following figure.

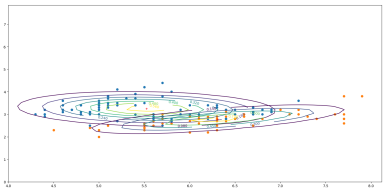


Figure 18: K = 2

When the number of components is 4, the result is the following figure.

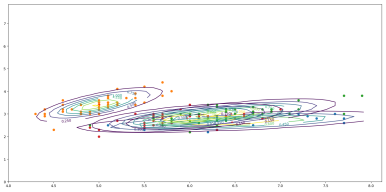


Figure 19: K = 4

Initalizaiton $\theta 0$ is the same with the dimension 2.
As written in section 1.1.1, the initialization process affects the result.

- plot the log-likelihood function over the iterations! What is the behavior of this function over the iterations?

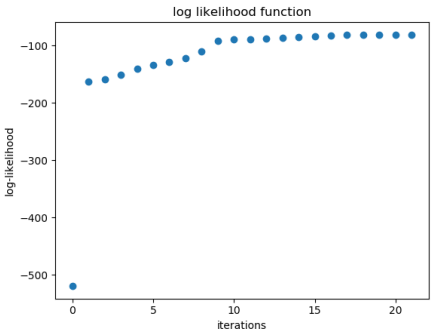


Figure 20: The log-likelihood function over iterations

As shown in Figure 4, the log-likelihood increases over iterations. That is, likelihood increased over iterations. And about 25th iteration, the function looks convering to the value, -87.41158191236445.. Therfore, the process stops even though it didn't reach the max iteration number.

- Make a scatter plot of the data that shows the result of the soft-classication that is done in the E-step

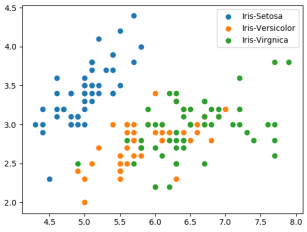


Figure 21: The EM algorithm soft-classification

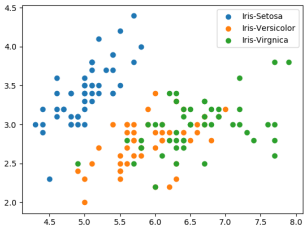


Figure 22: The answer classification

The EM algorithm classifies well the points when it is compared with the answer classification. EM algorithm fails to classify the points near the boundary of iris-Versicolor and iris-Virgnica.

1.2.2 How do the convergence properties and the accuracy of you classication change in comparison to scenario 2.1?

- EM-algorithm
  The convergence value of log likelihood function increased. In the scenario 2.1, the value was -112.96. However, in the scenario 2.2, the value increased to -87.41. Also, the total iteration decreased in the scenario 2.2. The value decreased from 75 to 26.
  Also, we compared the scenario 2.1 and 2.2 with each 10 trials. Overall, the EM classifies the data points better in 2.2 scenario. The accuarcy of 2.2 scenario is higher than that of 2.1 scenario. Because the EM algorithm in 2.2 scenario has more additional information about the dataset.

1.2.3 Within your EM-function confine the structure of the covariance matrices to diagonal matrices! What is the inuence on the result.

we confined the structure of the covariance matrices to diagonal matrices by setting non-diagonal matrices to 0. These following figures are the EM results with diagonal covariance matrices.

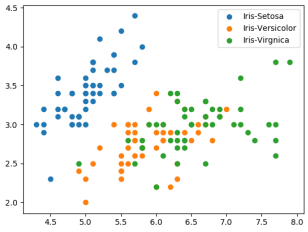


Figure 23: The answer classification

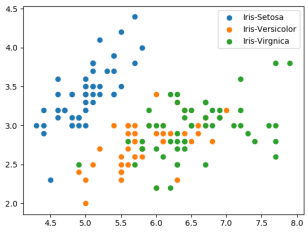


Figure 24: The EM algorithm soft-classification

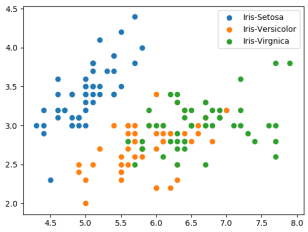


Figure 25: The EM algorithm soft-classification with diagonal covariance

As shown in the figure 23,24,25, the result with diagonal covariacne, looks similar to th others. That is, the diagonal convariance doesn't influence on the result.

1.3 Processing the data with PCA

1.3.1 How much of the variance in the data is explained this way?

- original variance(sum of eigenvalues) - 4.499157046979866
- associated eigenvalues - 4.15886089, 0.23573307
- the amount of explained variance - 0.9767594057574307

1.3.2 How does the performance of your algorithms compare to scenario 2.1 and scenario 2.2?

Scenario 2.1 has no difference at performance for both algorithms. Actually it takes more time to do PCA. The amount of explained variance is 1. The data from PCA is rotated original data that eigenvectors are the axes. The number of misclassified points was 2 with EM and 13 with K-means. It showed same behavior with scenario 1.

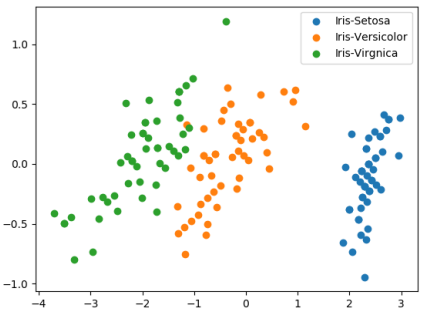


Figure 26: answer for pca scenario 1

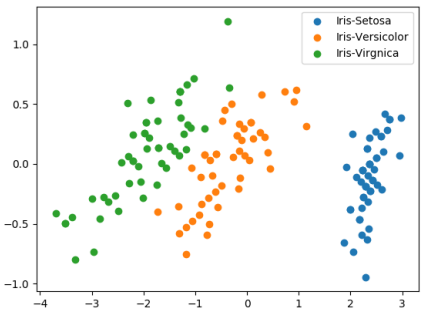


Figure 27: scatter plot for scenario 1. EM

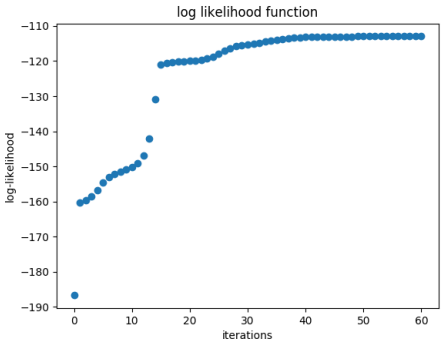


Figure 28: log-likelihood plot for scenario 1. EM

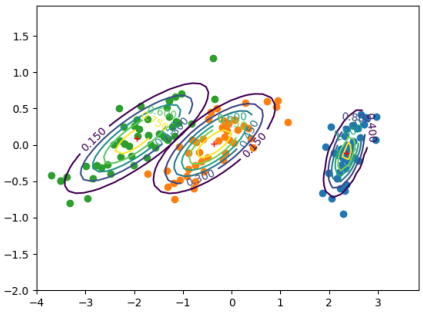


Figure 29: scatter plot with Gaussian for scenario 1. EM

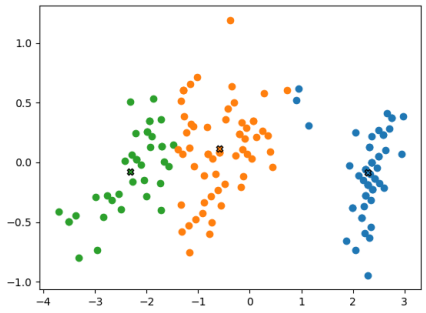


Figure 30: scatter plot for scenario 1. K-means

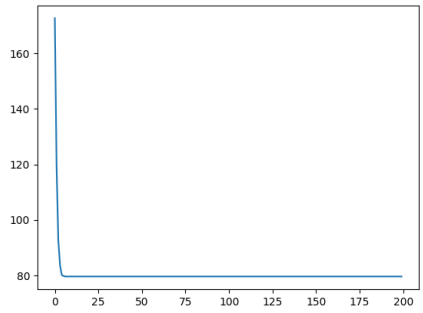


Figure 31: cumulative distance for k-means plot for scenario 1. K-means

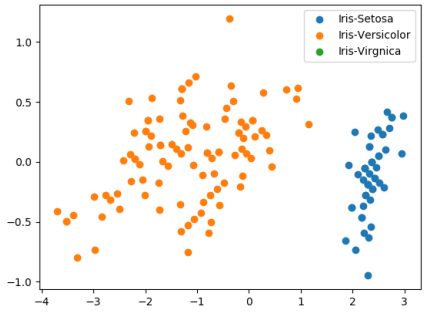


Figure 32: scatter plot for scenario 1. EM. K = 2

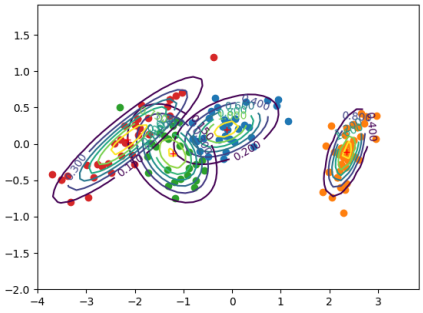


Figure 33: scatter plot for scenario 1. EM. K = 4

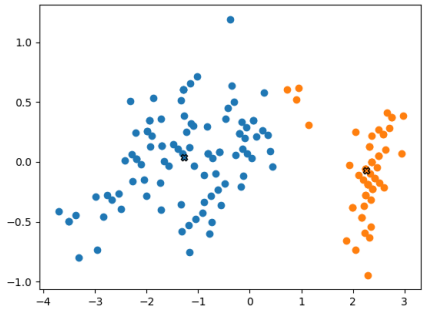


Figure 34: scatter plot for scenario 1. K-means. K = 2

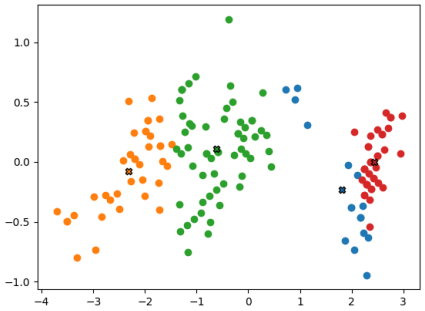


Figure 35: scatter plot for scenario 1. K-means. K = 4

EM showed better performance in scenario 2.2. PCA reduced the dimension of data to 2 so it took less time and showed more accuracy. The number of misclassified point was 3 in average. It is slightly less value compared to 4 in scenario 2. K-means also had shorter running time. The number of misclassified point was 9 in average.

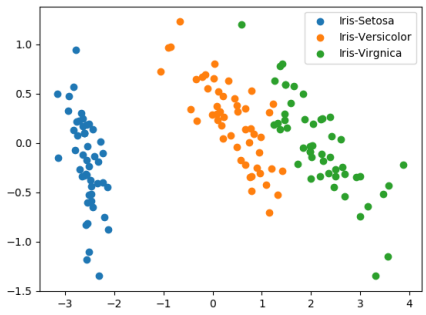


Figure 36: answer for pca scenario 2

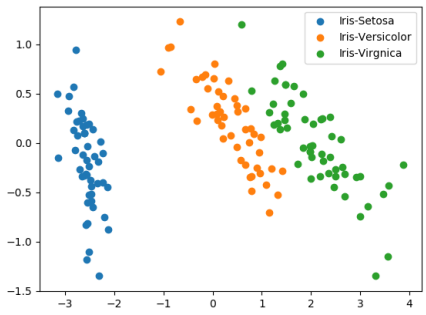


Figure 37: scatter plot for scenario 2. EM

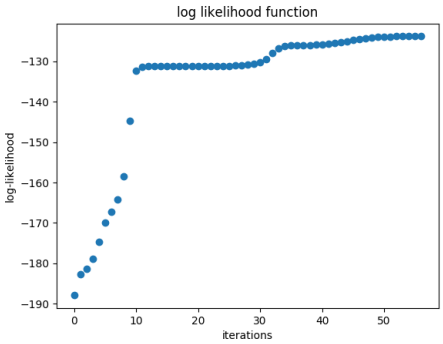


Figure 38: log-likelihood plot for scenario 2. EM

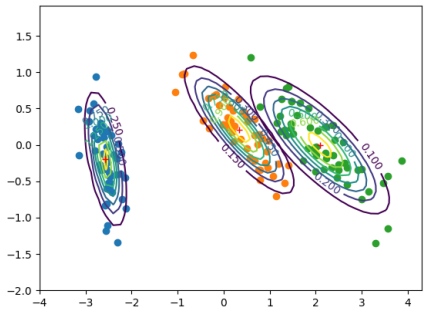


Figure 39: scatter plot with Gaussian for scenario 2. EM

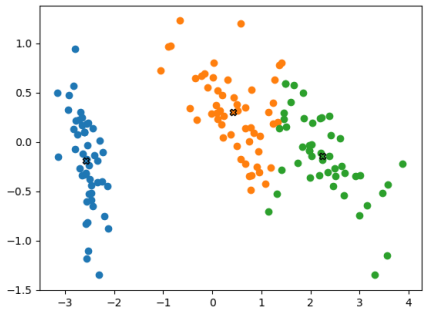


Figure 40: scatter plot for scenario 2. K-means

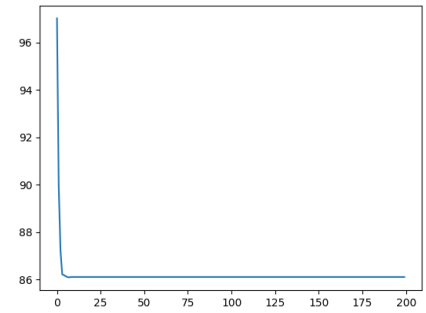


Figure 41: cumulative distance for k-means plot for scenario 2. K-means

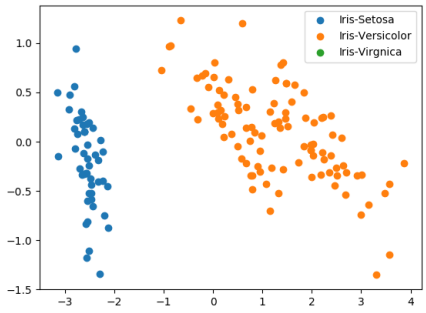


Figure 42: scatter plot for scenario 2. EM. K = 2

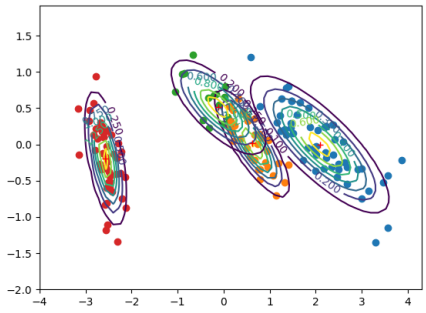


Figure 43: scatter plot for scenario 2. EM. K = 4

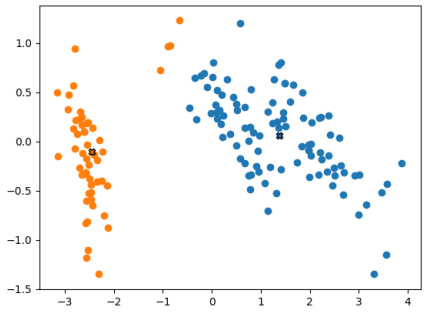


Figure 44: scatter plot for scenario 2. K-means. K = 2

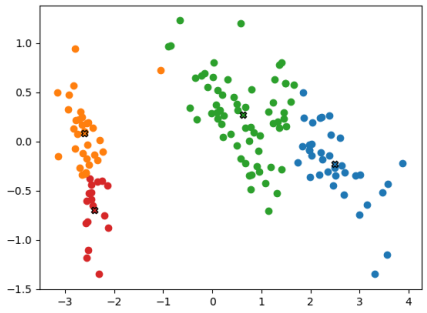


Figure 45: scatter plot for scenario 2. K-means. K = 4

1.3.3 Apply PCA with whitening, so that the transformed data has zero mean and a unit covariance matrix. How does this influence the choice of your initialization?

Compared PCA without whitening and PCA with whitening on scenario 2(4 dimension → 2 dimension), initialization result is below.

- EM
  - without whitening
    - α
      $\begin{bmatrix}0.33333333 & 0.33333333 & 0.33333333\end{bmatrix}$
    - mean
      $\begin{bmatrix}1.13021491 & -0.01001823 & 1.13587265 \\ 0.23478179 & 0.11992288 & -0.42093766\end{bmatrix}$
    - covariance
      $\begin{bmatrix} 4.15886089e+00 & 4.15886089e+00 & 4.15886089e+00 \\ -8.58001217e-16 & -8.58001217e-16 & -8.58001217e-16 \\ -8.58001217e-16 & -8.58001217e-16 & -8.58001217e-16 \\ 2.35733074e-01 & 2.35733074e-01 & 2.35733074e-01 \end{bmatrix}$
  - with whitening
    - α
      $\begin{bmatrix}0.33333333 & 0.33333333 & 0.33333333\end{bmatrix}$
    - mean
      $\begin{bmatrix}0.65215026 & -2.61884422 & -2.74090941 \\ -0.1535649 & -0.0246702 & -0.2064007\end{bmatrix}$
    - covariance
      $\begin{bmatrix} 1.72961239e+01 & 1.72961239e+01 & 1.72961239e+01 \\ -1.05433932e-15 & -1.05433932e-15 & -1.05433932e-15 \\ -1.05433932e-15 & -1.05433932e-15 & -1.05433932e-15 \\ 5.55700821e-02 & 5.55700821e-02 & 5.55700821e-02 \end{bmatrix}$

    The mean of sample means became close to 0 after whitening, assumed that the number of choosen samples in each class is same. The covariances also changed. Theoritically, the covariance matrix with 4 dimension should be identity matrix. However, our covariances is not covariance matrix with 4 data dimension. Whitening influenced somehow to initialize EM.

- K-means
  - without whitening - initial values
    $\begin{bmatrix}0.44172808 & 1.97553771 & -2.56665681 \\ 0.44933724 & -0.0330603 & -0.31820248\end{bmatrix}$
  - with whitening - initial values
    $\begin{bmatrix}-5.54268692 & 1.16714556 & -4.69918262 \\ 0.108882 & -0.08323771 & -0.65221037\end{bmatrix}$

Calcutated mean value of initial values in each dimension became close to 0 after whitening.

2 Samples from a Gaussian Mixture Model

2.1 Write a function Y = sample-GMM(alpha, mu, cov, N)

Our algorithm works first by constructing a discrete distribution from the alpha values of each Gaussian Distribution, then using this discrete distribution to assign each sample to a Gaussian Distribution, and finally placing the sample according to its assigned Gaussian Distribution. The probability of each Gaussian Distribution being selected for a new sample is equal to its corresponding alpha value.

2.2 Using a GMM of your choice($K > 3$), demonstrate the correctness of your function

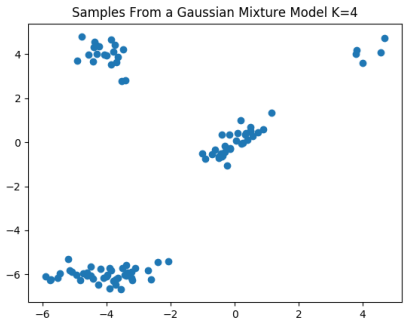


Figure 46: 100 Samples drawn from a Gaussian Mixture Model

This result demonstrates the correctness of our algorithm. For parameters we used the following alpha values: [0.05, 0.2, 0.3, 0.45] and following mean values: [(4, 4), (-4, 4), (0, 0), (-4, -6)]. You can see that the Gaussian Distribution centered at (4, 4) has a very low alpha (0.05), since it has so few points compared to the others. The covariance matrices are as follows.

$$\Sigma_1 = \begin{bmatrix} .2 & 0 \\ 0 & .2 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} .3 & 0 \\ 0 & .3 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} .3 & .25 \\ .25 & .3 \end{bmatrix}$$

$$\Sigma_4 = \begin{bmatrix} 1 & 0 \\ 0 & .1 \end{bmatrix}$$