University of Tehran

School of Electrical and Computer Engineering

Machine Learning

Homework 5: Dimension Reduction and Expectation Maximization

Author:
Alireza Javid

Student Number:
810198375

Contents

1 Problem 1: Basics of PCA

(a) In Figure 1 we can see the first and second principal components of PCA.

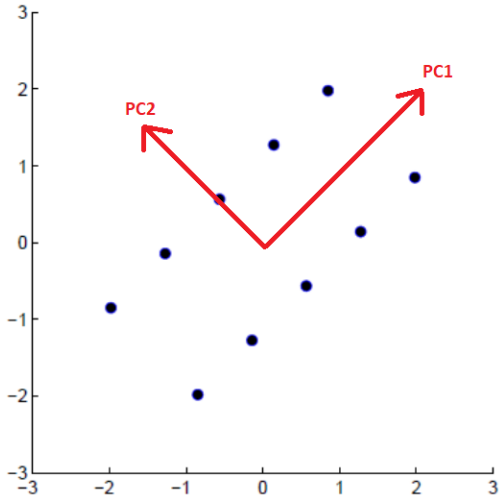


Figure 1: The first and second principal components of PCA

PCA is an unsupervised method used for dimensionality reduction, and it does not consider the labels or class information of the dataset when determining the principal components. The principal components are solely based on the variance and covariance structure of the input features. In PCA, the labels do not directly influence the computation of the principal components.

(b)
- Consider the following labeling.

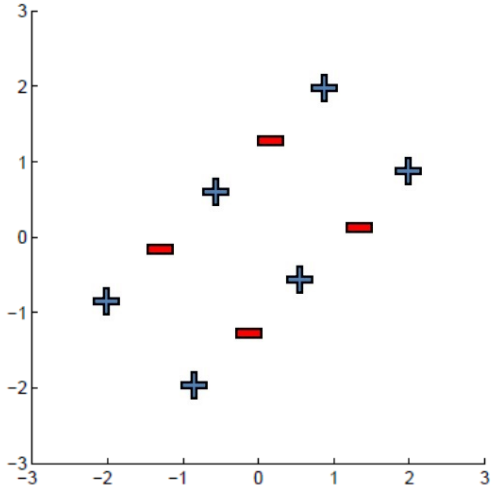


Figure 2: The first labeling scheme

In this scenario, the closest point to each red point in the 2D plane is a blue point, and vice versa, resulting in an accuracy of 0 when using the nearest neighbor (NN) classification. However, if we project each point onto the PC1 axis, the nearest point to each red point is another red point, and the

nearest point to each blue point is another blue point and the accuracy of NN is 1.0.

- Now consider this labeling.

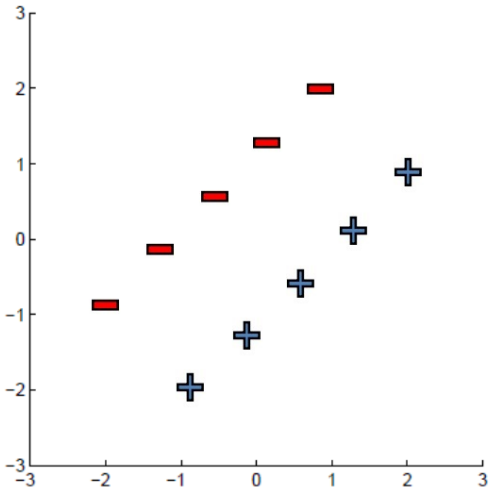


Figure 3: The second labeling scheme

In this case, the closest point to each red point in the 2D plane is another red point, and vice versa, resulting in an accuracy of 1.0 when using the nearest neighbor (NN) classification. However, if we project each point onto the PC1 axis, the nearest point to each red point is a blue point, and vice versa, and the accuracy of NN is 0.

2 Problem 2: Scattering Metric

$$J = \frac{1}{n_1 n_2} \sum_{y_i \in Y_1} \sum_{y_j \in Y_2} \big(y_i - y_j\big)^2 = \frac{1}{n_1 n_2} \sum_{y_i \in Y_1} \sum_{y_j \in Y_2} y_i^2 + y_j^2 - 2 y_i y_j$$

As we know based of $E(Z^2) + E(Z)^2 = Var(Z)$ we have $\sum_{y_i \in Y_l} y_i^2 = s_l^2 + n_l \times m_l$ where $l = 1, 2$.

So we can write:

$$J = \frac{1}{n_1} \sum_{y_i \in Y_1} y_i^2 + \frac{1}{n_2} \sum_{y_j \in Y_2} y_j^2 - 2 \frac{1}{n_1 n_2} \sum_{y_i \in Y_1} \sum_{y_j \in Y_2} y_i y_j$$

$$J = \frac{1}{n_1} \sum_{y_i \in Y_1} y_i^2 + \frac{1}{n_2} \sum_{y_j \in Y_2} y_j^2 - 2 \{\frac{1}{n_1} \sum_{y_i \in Y_1} y_i\} \{\frac{1}{n_2} \sum_{y_j \in Y_2} y_j\}$$

$$J = \frac{1}{n_1} \{s_1^2 + n_1 \times m_1\} + \frac{1}{n_2} \{s_2^2 + n_2 \times m_2\} - 2 \times m_1 m_2$$

$$J = (m_1 - m_2)^2 + \frac{1}{n_1} s_1 + \frac{1}{n_2} s_2$$

3 Problem 3: EM for Poisson Distribution and PCA First Principal Component

(a) Assume we have given observation $D = \{x_1, x_2, ..., x_N\}$. The hidden variable we wish to find is $\theta = \{p_1, ..., p_K, \lambda_1, ..., \lambda_2\}$ where $\sum_{i=1}^{K} p_i = 1$ and $0 \leq p_i \leq 1$ and Θ is a family of mixture models. The likelihood function is defined as

$$L(D;\theta) = \sum_{i=1}^{K} p_i f(D;\theta) \quad f(x;\lambda_1) = \frac{\lambda^x}{x!} e^{-\lambda}$$

If we presume x_i's are i.i.d random variables we have:

$$L(D;\theta) = \prod_{l=1}^{N} \sum_{i=1}^{K} p_i f(D_l;\theta)$$

We can logarithm from both sides and define the log-likelihood function as below.

$$\ell(D;\theta) = \log \prod_{l=1}^{N} \sum_{i=1}^{K} p_i f(D_l;\theta) = \sum_{l=1}^{N} \log \sum_{i=1}^{K} p_i f(D_l;\theta)$$

and we want to find θ^* as below.

$$\theta^* = \arg\max_{\theta \in \Theta} \ell(D;\theta)$$

We use the following notion for simplicity.

$$q(i,l) = p_i f(D_l;\lambda_i)$$

And using the conditional probability we have:

$$p(i \mid l) = \frac{q(i,l)}{\sum_{m=1}^{K} q(m,l)}$$

We call these the membership probabilities because they give the probability that D_l is a member of component i.

After this long formulation, we start to use the EM algorithm.

The EM algorithm begins with a set of estimates for p_is and λ_i and in each iteration makes better estimations.

In the E-step, we compute the expectation that each observation came from each component and we use the existing estimates to calculate new estimations.

$$\ell(D;\theta) = \sum_{l=1}^{N} \log \sum_{i=1}^{K} \frac{p(i \mid l)}{p(i \mid l)} q(i,l) \geq \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log \frac{q(i,l)}{p(i \mid l)} = b(\theta)$$

We've used Jensen's inequality in the above formula.

$$b(\theta) = \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log q(i,l) - \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log p(i \mid l)$$

The $p(i \mid l)$ is fixed so we define the Q function as follows:

$$Q(\theta) = \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log q(i,l)$$

We calculate the new parameters by differentiating Q with respect to each parameter and setting the derivative to 0.

- To update the Poisson parameters we have:

$$\frac{\partial Q}{\partial \lambda_i} = \frac{\partial}{\partial \lambda_i} \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log q(i,l) = \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \frac{\partial \log q(i,l)}{\partial \lambda_i}$$

  Recall that

$$\frac{\partial \log q(i,l)}{\partial \lambda_i} = \frac{\partial \log p_i f(D_l; \lambda_i)}{\partial \lambda_i} = \frac{\partial \log p_i \frac{\lambda_i^{D_l}}{D_l!} e^{-\lambda_i}}{\partial \lambda_i}$$
$$= \frac{\partial \log p_i + D_l \log \lambda_i - \log D_l! - \lambda_i}{\partial \lambda_i} = \frac{D_l}{\lambda_i} - 1$$

  So we have:

$$\frac{\partial Q}{\partial \lambda_i} = \sum_{l=1}^{N} p(i \mid l)(\frac{D_l}{\lambda_i} - 1)$$

  If we set the derivative to 0 we can evaluate the λ_i as below:

$$\sum_{l=1}^{N} p(i \mid l)(\frac{D_l}{\lambda_i} - 1) = 0 \rightarrow \lambda_i = \frac{\sum_{l=1}^{N} p(i \mid l) D_l}{\sum_{l=1}^{N} p(i \mid l)}$$

  Thus, we have our new estimate of λ_i

- To update the mixing probabilities we follow the same procedure:

$$\frac{\partial \log q(i,l)}{\partial p_i} = \frac{\partial \log p_i f(D_l; \lambda_i)}{\partial p_i} = \frac{\partial \log p_i \frac{\lambda_i^{D_l}}{D_l!} e^{-\lambda_i}}{\partial p_i}$$
$$= \frac{\partial \log p_i + D_l \log \lambda_i - \log D_l! - \lambda_i}{\partial p_i} = \frac{1}{p_i}$$

  Because p_is are the probability distributions we use Lagrange multiplayer β to constrain the p_i values.

$$\frac{\partial Q}{\partial p_i} = \frac{\partial}{\partial p_i} \sum_{l=1}^{N} \sum_{i=1}^{K} p(i \mid l) \log q(i,l) + \beta (\sum_{i=1}^{K} p_i - 1)$$

$$\frac{\partial Q}{\partial p_i} = \sum_{l=1}^{N} \frac{p(i \mid l)}{p_i} + \beta = 0$$

  By summing over k, we can evaluate β :

$$\sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \sum_{l=1}^{N} \frac{p(i \mid l)}{\beta} = \sum_{l=1}^{N} \frac{1}{\beta} = \frac{N}{\beta} = 1 \ \rightarrow \ \beta = N$$

$$p_i = \frac{\sum_{l=1}^{N} p(i \mid l)}{N}$$

  Thus, we have our new estimate of p_i

(b)

$$C = \sum_i \lambda_i p_i p_i^T$$

As we know

$$p_i^T p_j = \begin{cases} I & i = j \\ 0 & i \neq j \end{cases}$$

Thus we can write:

$$Cp_1 = \sum_i \lambda_i p_i p_i^T p_1 = \lambda_1 p_1 p_1^T p_1$$

We can multiply both sides by p_1^T and know $C = X^T X$

$$p_1^T X^T X p_1 = \lambda_1 p_1^T p_1 = \lambda_1$$

Recall that, the variance of projection of the data to the first component (Xp_1) is

$$Var(Xp_1) = (Xp_1)^T Xp_1 = p_1^T X^T X p_1$$

And finally, we have:

$$Var(Xp_1) = \lambda_1$$

4 Problem 4: EM Algorithm

(a) The first picture is better.

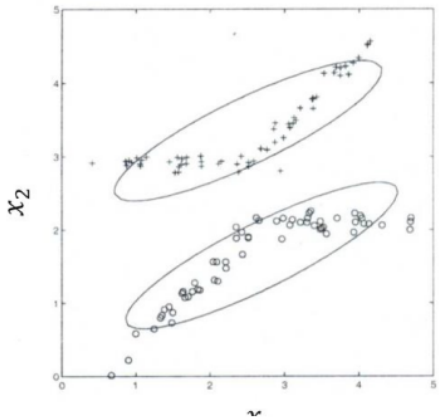


Figure 4: The Result of EM on Gaussian Mixture Model

Figure 4 demonstrates the improved discrimination of this model. The Gaussian models are clearly distinct, with no intersection, effectively separating both clusters.

(b) Look at the figure below.

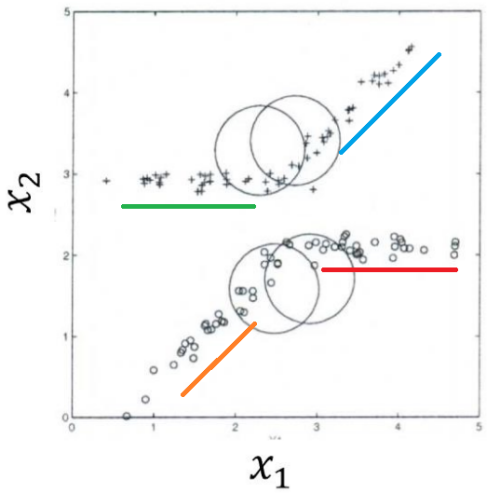


Figure 5: The Initial State of Gaussian Mixture Model

We have four distinct Gaussian models in our analysis. In each iteration of the EM algorithm, we update the parameters of these models based on the most probable points assigned to each Gaussian. In Figure 5, the colored regions represent the most probable points for each Gaussian model. This selection of points indicates where we expect the mean values to be in the next iteration.

The first pictures accurately illustrate this process.

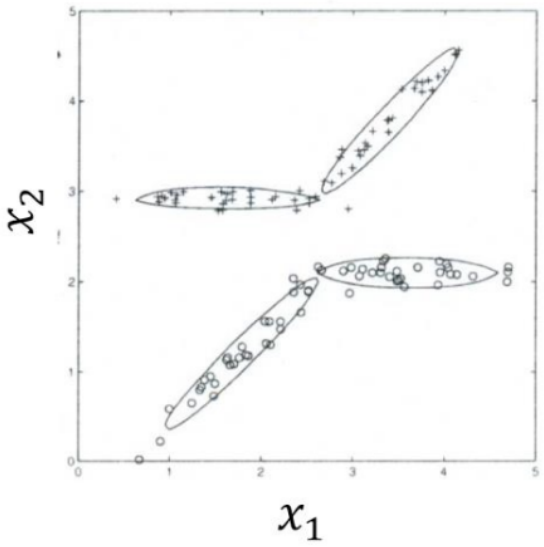


Figure 6: The Initial State of Gaussian Mixture Model

5 Problem 5: Exploring Data with PCA and LDA

5.1 Part 1: PCA for Emotion Detection

In this part, we use Principal Component Analysis (PCA) to investigate the emotion detection dataset.

(a) In this analysis, we plot the eigenvalues in descending order. It can be observed that after considering the first 40 eigenvalues, their values become close to zero.

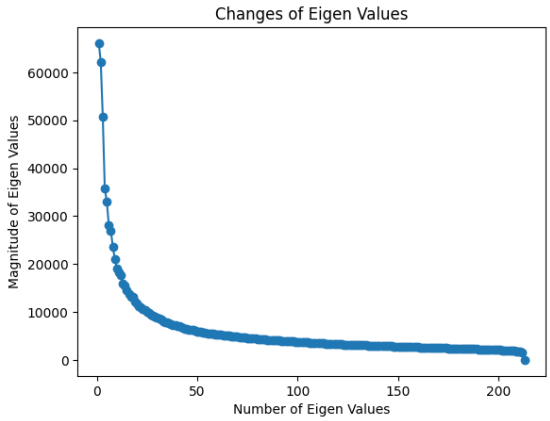


Figure 7: Changes of Eigen Values in PCA

We have utilized the "*explained_variance_ratio_*" attribute to determine the optimal number of principal components for compression. Our goal was to select the smallest number of components that can capture a cumulative explained variance of at least 90

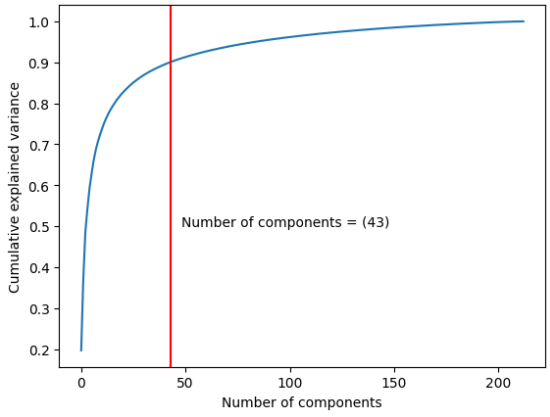


Figure 8: Optimal number of components to reach 90% cumulative explained variance

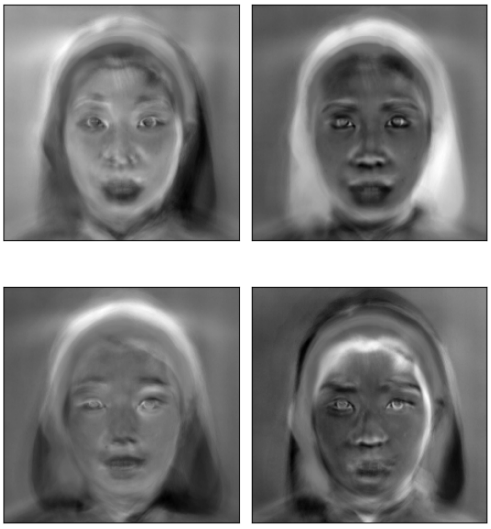
By examining the cumulative sum of the explained variance ratios, we observed that after including 43 principal components, the cumulative explained variance reached or exceeded the desired threshold of 90%. This indicates that these components contain the majority of the information present in the original dataset

and we effectively compress the data while retaining a significant portion of the dataset's variability.

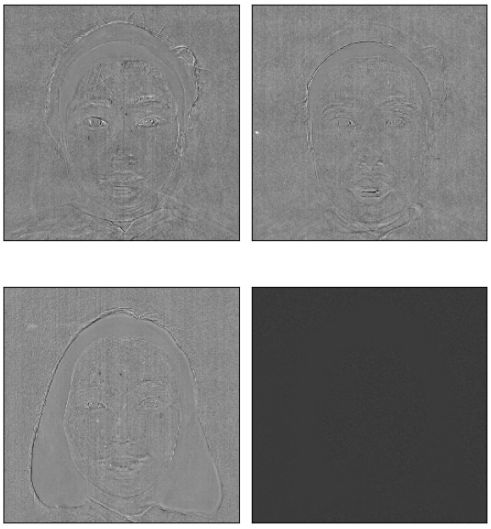
(b) In this part, we have visualized the eigenfaces corresponding to the four most significant principal components and the four least significant ones. The eigenfaces represent the patterns that contribute the most and the least to the overall variability in the dataset.

Observing the first four eigenfaces, we can discern meaningful facial features and patterns. These eigenfaces capture the fundamental characteristics that distinguish different faces in the dataset. However, as we move towards the least significant components, we notice that the corresponding eigenfaces exhibit more noise and do not provide meaningful representations of facial features. These eigenfaces may contain variations that are specific to individual images or noise present in the dataset.

This observation highlights the importance of selecting an appropriate number of principal components for dimensionality reduction. Choosing too many components may lead to the inclusion of noise or redundant information, while selecting too few components may result in a loss of significant patterns and details.



(a) Eigenfaces corresponding to the four most significant principal components



(b) Eigenfaces corresponding to the four least significant principal components

Figure 9: Eigenfaces of PCA

5.2 Part 2: LDA for Fashion-MNIST

In this part, we use Linear Discriminant Analysis (LDA) to investigate the Fashion-MNIST dataset. Generally, the LDA scatter matrix for each class is:

$$S_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

where $\mathbf{m}_i$ is mean value for class i. We define the within-class scatter matrix for LDA as

$$S_W = \sum_{i=1}^{c} S_i$$

and between-class scatter matrix as

$$S_B = \sum_{i=1}^{c} N_i (\mathbf{m} - \mathbf{m}_i)(\mathbf{m} - \mathbf{m}_i)^T$$

where $\mathbf{m}$ is the mean value of all the data and N_i is number of samples in class i.

(a) In LDA (Linear Discriminant Analysis), the eigenvalues of the separability matrix ($S_W^{-1} S_B$) represent the importance or discriminative power of the corresponding eigenvectors. When the eigenvalues are sorted in descending order, the first eigenvalue corresponds to the most discriminative component, the second eigenvalue corresponds to the second most discriminative component, and so on. The eigenvalues represent the amount of class-specific information captured by each eigenvector.

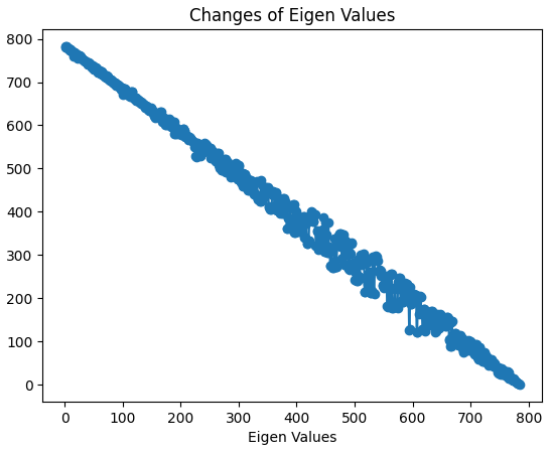


Figure 10: Changes of Eigen Values in LDA

By examining the changes in eigenvalues in descending order, we can gain insights into how much information is captured by each subsequent component. Generally, we observe that the initial eigenvalues have larger magnitudes, indicating they capture more discriminative information. As we move towards lower eigenvalues, the amount of class-specific information captured by each component decreases.

(b) In LDA, the separability measure is calculated as $trace(S_W^{-1} S_B)$. This measure quantifies the separability or discriminative power of the selected features in LDA.

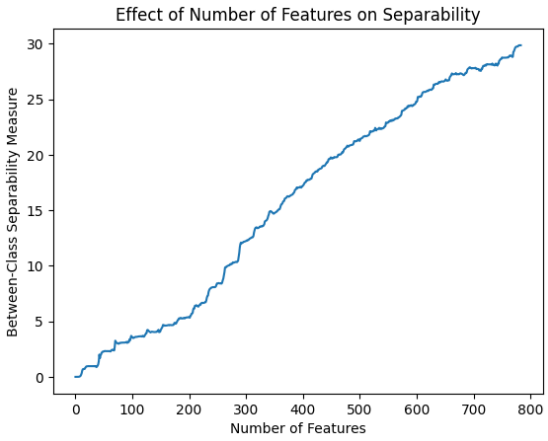


Figure 11: Effect of Number of Features on Separability in LDA

As the number of features increases, the separability measure tends to increase initially. This is because a larger number of features can potentially capture more discriminative information about the classes. However, there is a point beyond which adding more features may not necessarily improve the separability measure. In fact, including irrelevant or redundant features can introduce noise and decrease the separability. This is known as the curse of dimensionality. In this problem, the size of the dataset, is large enough that we do not encounter this issue.

You can see the code details of both parts in the attached file.

6 Problem 6: Gaussian Mixture Model

(a) In this part, we utilize the Gaussian Mixture Model (GMM) algorithm with K=2 to fit separate models for each class based on the RGB features. We then plot the models along with their corresponding contours. It is evident from the plot that the mean of one Gaussian distribution in the first model is very close to the mean of the other Gaussian distribution in the second model for a different class. This proximity of means suggests that there may be some overlap or similarity between the two classes in terms of their RGB feature distribution.

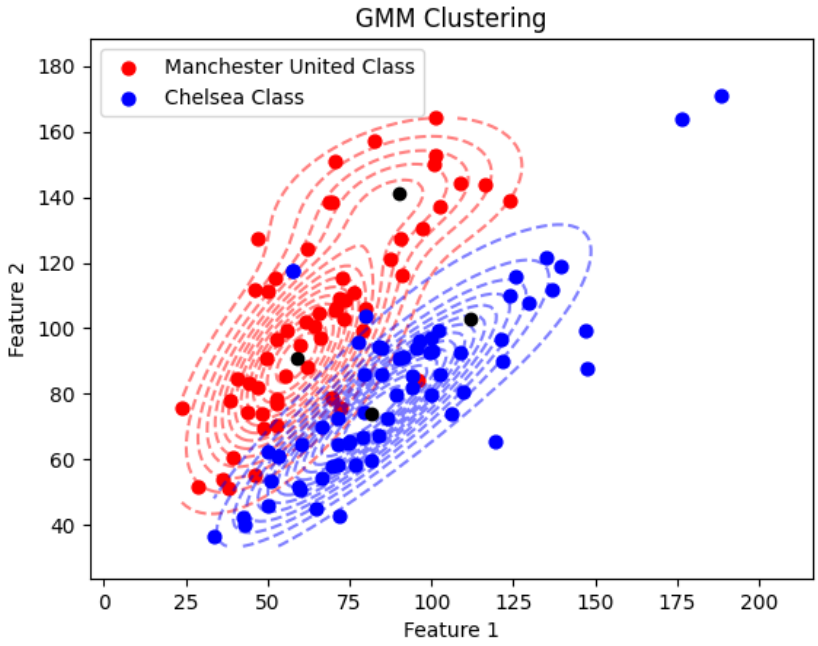


Figure 12: Gaussian Mixture Models

And we have GMM parameters in Figure 13.

```
Component 1 parameters:
Weight: 0.7554506728040803
Mean: [58.80821217 90.98118952]
Covariance matrix:
[[280.6650719  226.98783562]
 [226.98783562 416.69323113]]

Component 2 parameters:
Weight: 0.2445493271959197
Mean: [ 89.94134538 140.91862308]
Covariance matrix:
[[449.11249752 102.39718695]
 [102.39718695 169.8513045 ]]
```

(a) GMM Model for Manchester Class

```
Component 1 parameters:
Weight: 0.2703984412175727
Mean: [111.79949486 102.87932864]
Covariance matrix:
[[1520.58716804  667.66808135]
 [ 667.66808135  895.60626832]]

Component 2 parameters:
Weight: 0.7296015587824272
Mean: [81.61750163 74.00116811]
Covariance matrix:
[[555.7617593  426.9072995 ]
 [426.9072995  413.48403773]]
```

(b) GMM Model for Chelsea Class

Figure 13: Parameters of GMM Model

(b) To choose the best value of k for each model, we employ three different algorithms: AIC, BIC, and K-fold cross-validation.

(i) AIC (Akaike Information Criterion): AIC is a statistical measure that balances the goodness of fit and the complexity of the model. It quantifies the trade-off between these factors and aims to find a model that minimizes information loss while considering the number of parameters. In the case of GMM, we calculate the AIC value for different values of k and choose the value of k that corresponds to the lowest AIC value. Lower AIC values indicate a better fit and a more parsimonious model.

(ii) BIC (Bayesian Information Criterion): BIC is similar to AIC but incorporates a stronger penalty for model complexity. It takes into account the likelihood of the data and the number of parameters in the model. BIC tends to favor simpler models and can be particularly useful when dealing with smaller sample sizes. Similar to AIC, we calculate the BIC value for different values of k and select the value of k that corresponds to the lowest BIC value. Lower BIC values indicate a better fit and a more parsimonious model.

(iii) K-fold Cross-Validation: Cross-validation is a resampling technique used to assess the performance of a model. In the case of GMM, we can use K-fold cross-validation to evaluate the model's performance for different values of k. The dataset is divided into K subsets, and the model is trained and evaluated K times, each time using a different subset as the validation set. We calculate the average performance metric (e.g., accuracy, log-likelihood) for each value of k and choose the value that yields the best performance.

And here is the results:

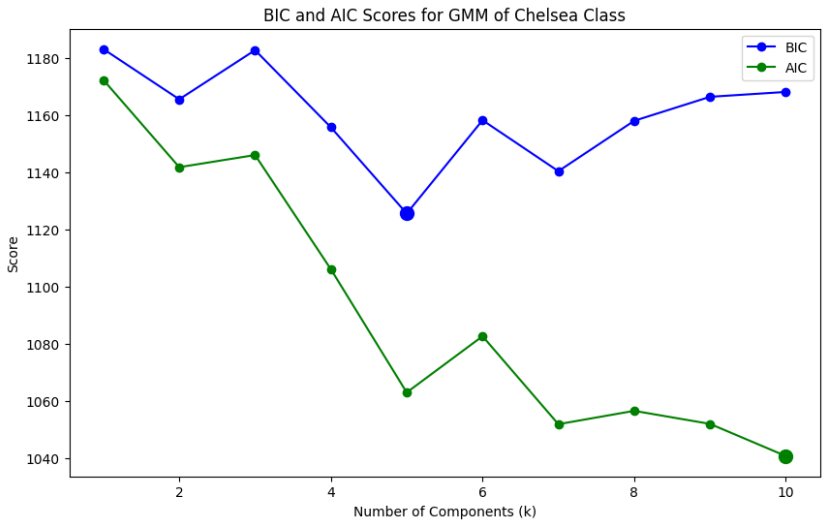


Figure 14: AIC and BIC Scores for Chelsea Class

So the Best value of k based on BIC is 5 and based on AIC is 10.

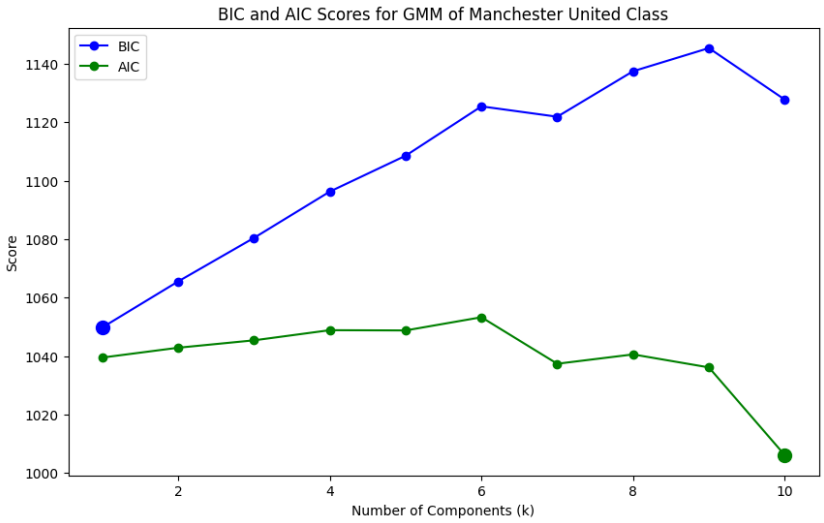
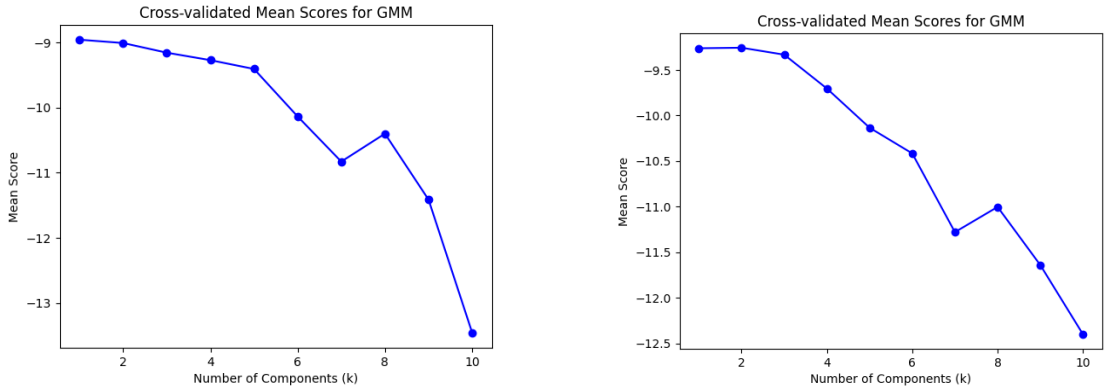


Figure 15: AIC and BIC Scores for Manchester Class

So the Best value of k based on BIC is 1 and based on AIC is 10.



(a) CV Scores for Manchester Class (b) CV Scores for Chelsea Class

Figure 16: Cross-validated Mean Scores for GMM Models

So the Best value of k based on the K-fold for the Manchester Class is 1 and for Chelsea Class is 2.

You can see the code details in the attached file.