ISYE 6740 HW 3

Mohammed Khalid Siddiqui

September 29, 2024

1 Conceptual Questions

1. Both KDE and Histograms are methods to help visualize and estimate the density of underlying data. One advantage of histograms is that they are pretty easy to implement and understand. They are also not computationally expensive compared to other methods of estimation. One of the downsides of using histograms is the importance of binning. Selecting different values for bin size as well as start and end units can result in drastically different bins and therefore estimates. In addition to this sensitivity, the block-y representation of the data can give a false sense of where some data points lie in relation to others. KDE provides a smoother representation of data and is less sensitive to bin-width choices than histogram. However, the choice of bandwidth can influence the level of smoothness or jaggedness. KDE is also more computationally expensive than histograms.

2. Gaussian Mixture Models consist of multiple Gaussian models which represent clusters or components of data. Using MLE to estimate GMM can be difficult because of the presence of latent variables which represent the component membership, the fact that the likelihood function is typically non-convex, meaning there are multiple local optima, as well as the fact there are weights of different Gaussian components and the mixing components (which sum up to 1) add a constraint to the MLE process which doesn't easily handle constraints like these directly.

   Instead, you can use the EM process which contains both an Expectation step and a Maximization step. During the E-step, you compute the posterior probabilities that each data point belong to each Gaussian component given current state of parameters. During the M-step, the algorithm updates parameters by maximizing the log-likelihood. These are repeated until convergence.

3. In the EM algorithm, the posterior probability that a given data point x belong to the k-th Gaussian component is known as the responsibility of the k-th Gaussian for the ith x or tau. You can first write out Bayes rule expressing the posterior probability as the prior times the likelihood. You then plug in the Gaussian probability density function and the mixing component to arrive at the closed-form solution below:

$$P(Z = k \mid x_i) = \frac{P(x_i \mid Z = k)P(Z = k)}{P(x_i)} \tag{1}$$

$$\tau_{ik} = \frac{\pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \cdot \mathcal{N}(x_i \mid \mu_j, \Sigma_j)} \tag{2}$$

$$\mathcal{N}(x_i \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\right) \tag{3}$$

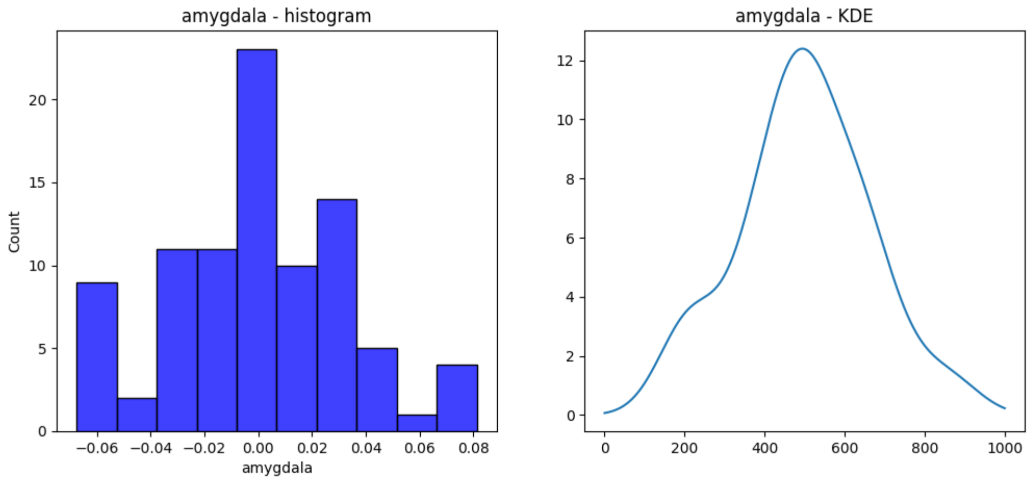
In the above equation, N is the probability density function of the k-th Gaussian component. Mu is the mean vector of the k-th Gaussian. Sigma is the covariance matrix of the kth Gaussian. d represents the dimensionality of the data. Pi is the mixing coefficient of the kth component. The sum of Pi for all k is equal to 1.

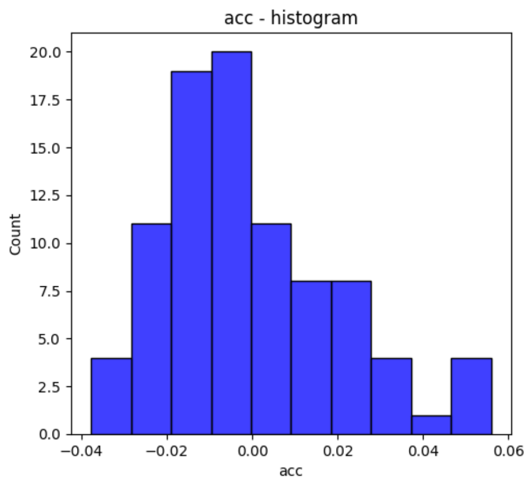
4. Choosing an appropriate kernel bandwidth when implementing KDE is important. A bandwidth that is too wide can result in an estimation which is too smooth, thereby not capturing behavior appropriately. Conversely, having too small of a bandwidth can result in over-fitting and extremely jagged estimation. There are several methods for choosing a bandwidth. One is the Silvernman's rule of thumb which provides a simple formula to use if implementing a Gaussian estimation. A more robust and generaliseable approach is to use cross validation. By training and testing multiple iterations across various segments of data, you can identify a more robust result regarding bin width. The drawback is the added computational effort required to do so. In some 2-dimensional and 3-dimensional instances, it is even possible to apply a visual inspection to determine via sanity check if the bandwidth makes sense.

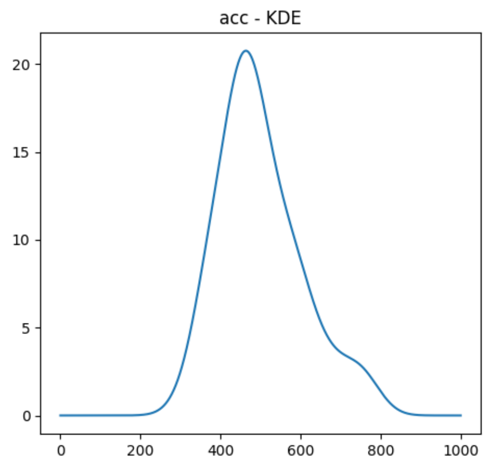
2 Density estimation: Psychological experiments

1. For this section, 10 was chosen as an appropriate number of bins. With 90 data points, a uniform distribution would have 9 points per bin. From the visual below, it appears that the amygdala and acc exhibit some form of Gaussian or skewed-normal distribution. For the KDE plot, the value of h (kernel bandwidth) was set using Silverman's rule below as specified in the lectures.

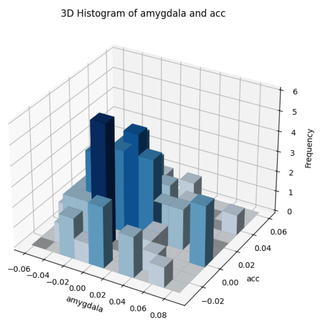
$$h \approx 1.06 \cdot R\sigma \cdot m^{-\frac{1}{5}} \tag{4}$$

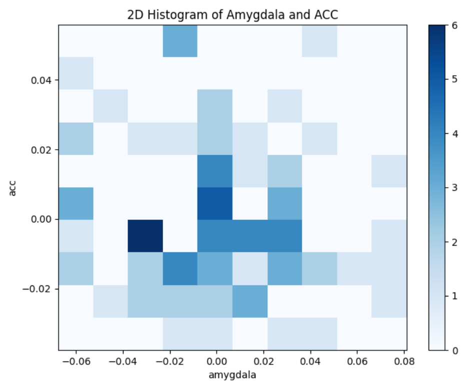




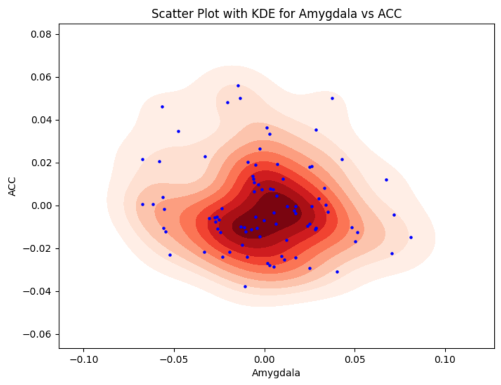


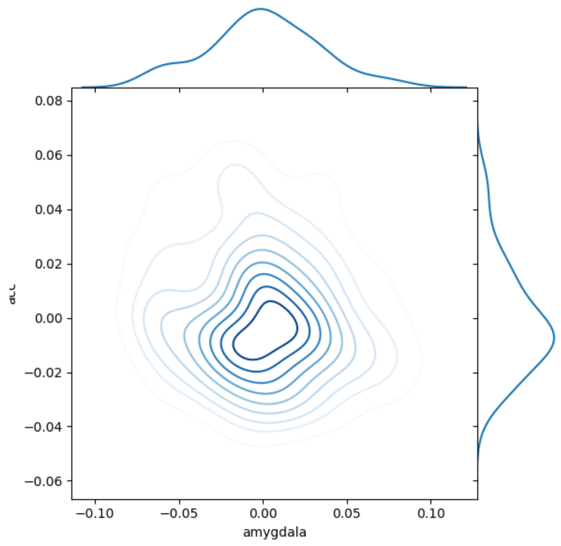
2. 2-D and 3-D rendering of the variable distributions as shown via histograms are provided below. Using 10 bins per dimension once again, it appears there is one major peak and smaller local optima.





3. Using the seaborn library, we were able to plot a kernel density estimation for the two dimensional denstiy function of amygdala and acc. Instead of directly setting a kernel bandwidth, the parameter bw_method was set to ’silverman’ meaning that the silverman method would be used to find the appropriate bandwidth for the kernel density estimation. The distribution appears to be unimodal based on the KDE and joint plots below. There seems to be one main peak.





In addition to the visual representation, to test for independence or even correlation between acc and amygdala, we ran 4 different tests. The first two are correlation tests. Pearson's correlation finds the degree to which there's a linear relationship between samples. Spearmans correlation looks at rank and finds the strength and statistical significant of monotonic relationship. Mutual Information score shows how much one feature can provide information about another. Lastly, the Jensen-Shannon divergence test measures the divergence between probability distributions of two variables.

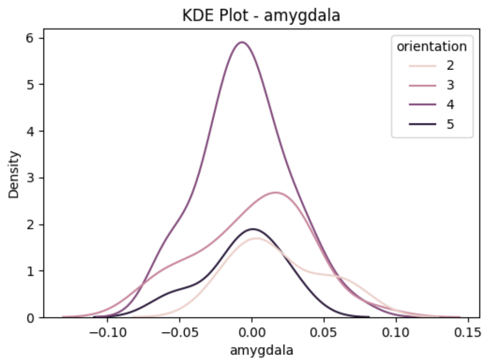
Statistic Test	**Value**	**P-value**	**Explanation**
Pearson Correlation	-0.128	0.227	Measures the linear correlation between amygdala and ACC. The small negative value indicates weak negative linear correlation, and the high p-value suggests the result is not statistically significant.
Spearman Correlation	-0.101	0.345	Measures the rank-based correlation between amygdala and ACC. Again, the small negative value indicates weak correlation, with a high p-value confirming non-significance.
Mutual Information Score	0.200	N/A	Measures the amount of information shared between amygdala and ACC. The low value suggests limited shared information between the variables, indicating potential independence.
Jensen-Shannon Divergence	0.142	N/A	Measures the similarity between the distributions of amygdala and ACC. The low value implies the distributions are not strongly diverging from each other.

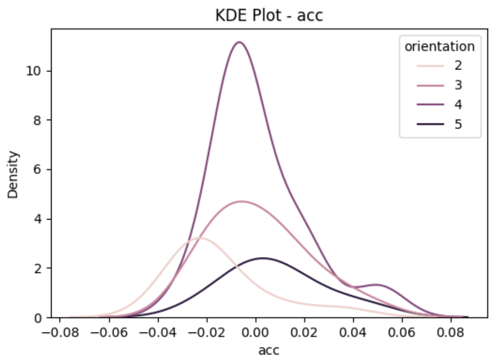
Table 1: Summary of Statistical Tests between Amygdala and ACC

It seems based on the statistical tests and visuals above, acc and amydgala appear to have either a very weak dependence or are likely independent. There doesn't seem to be a strong correlation or mutual information between them.

4. We're now interested in the conditional distribution of amygdala and acc individually based on political orientation. Again, we used Silverman's method for our kde plot and seaborn to search and find the optimal h. The density plots below indicate that orientation does seem to impact the distribution of

the other two variables.



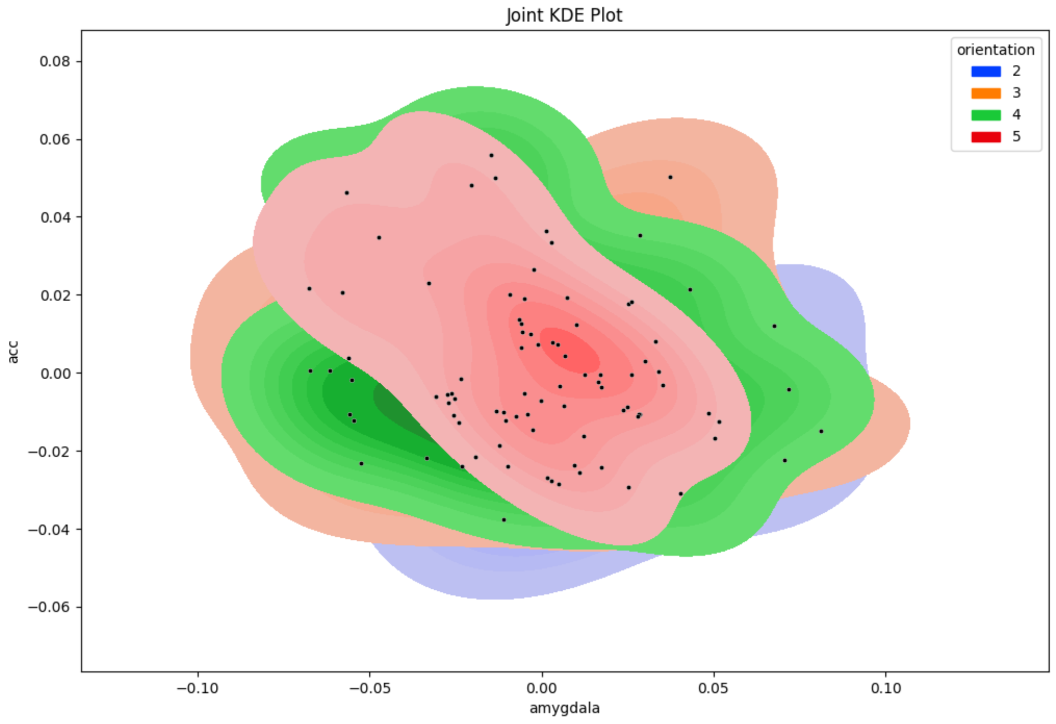


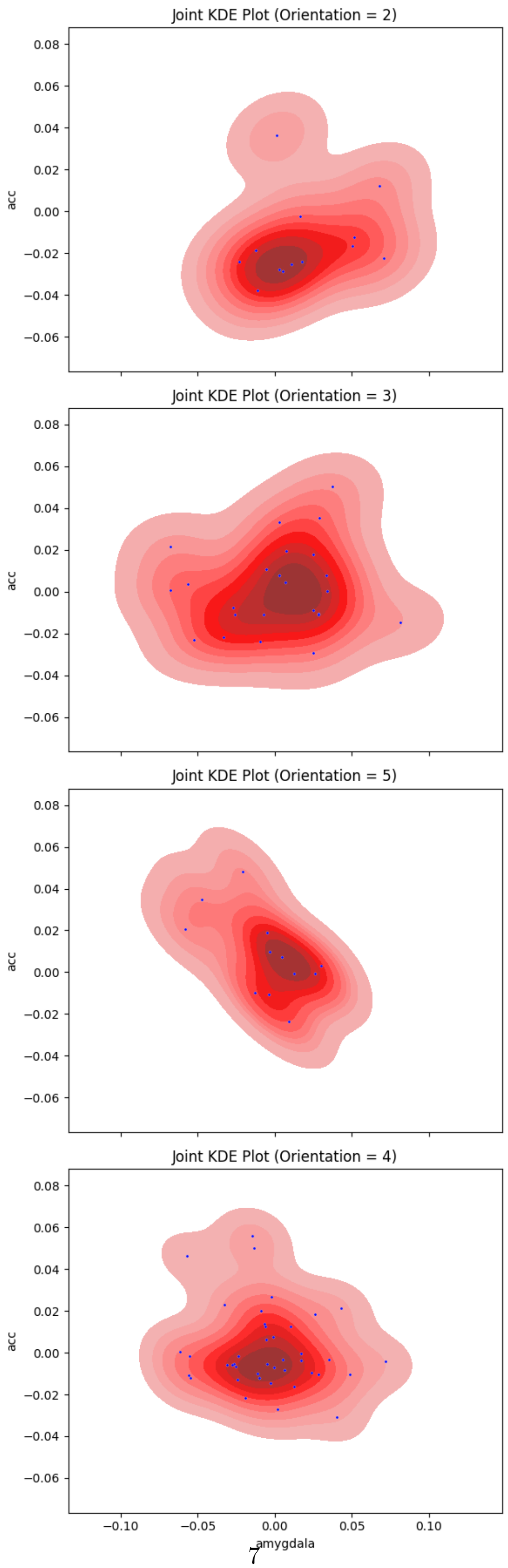
Based on the visuals above, it seems that acc for those with a political orientation of 2 if on average lower than for higher orientations. The conditional means shown below also seem to show that there is a major gap in acc between those who have a political orientation of 1 and those with 3 or 4. However, having the density estimations allows us to see the distribution more clearly and infer more complex relationships than simply observing the sample mean. for example, the width of the distributions and concentration of acc or amygdala around the mean for various orientation levels.

Orientation	**Amygdala**	**ACC**
2	0.019062	-0.014769
3	0.000588	0.001671
4	-0.004720	0.001310
5	-0.005692	0.008142

Table 2: Orientation Data with Corresponding Amygdala and ACC Values

5. Using a two dimensional KDE, we can see the sheer difference in variability among the different levels of political orientation for amygdala and acc. In addition to the visuals, we performed the Kruskal-Wallis test to test whether at least one of the orientation groups has a different distribution than the others.


Joint KDE Plot
orientation
2
3
4
5
acc
amygdala
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.10
−0.05
0.00
0.05
0.10



Joint KDE Plot (Orientation = 2)
Joint KDE Plot (Orientation = 3)
Joint KDE Plot (Orientation = 5)
Joint KDE Plot (Orientation = 4)
acc
amygdala
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.10
−0.05
0.00
0.05
0.10

Test	**H-statistic**	**p-value**
Kruskal-Wallis Test for Amygdala	4.8986	0.1794
Kruskal-Wallis Test for ACC	12.2702	0.0065

Table 3: Kruskal-Wallis Test Results

Based on the above values, it seems there is no significant test between the different amygdala groups. However based on the H-statistic and p-value, there does seem to be a statistically significant difference between the different orientation groups for acc.

3 Implementing EM for MNIST dataset

1. To implement our EM algorithm for the MNIST dataset classifying whether items were either a 2 or a 6, we first reduced the dimensions to the top 4 principle components. Then following the step by step instruction in the homework assignment and leveraging the provided demo-code, we implemented the EM model and ran it on the transformed image dataset to watch it converge into the appropriate clusters. You can see the visual representation of the clusters below:

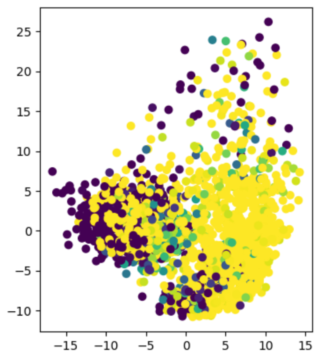


Figure 1: Iteration 0

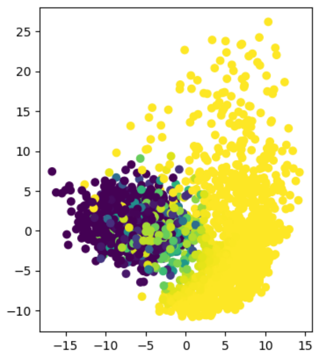


Figure 2: Iteration 7

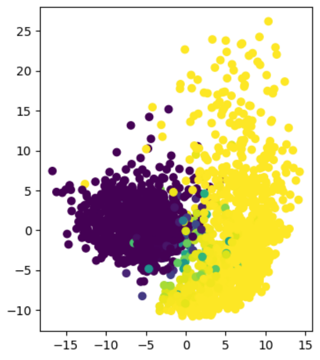


Figure 3: Iteration 14

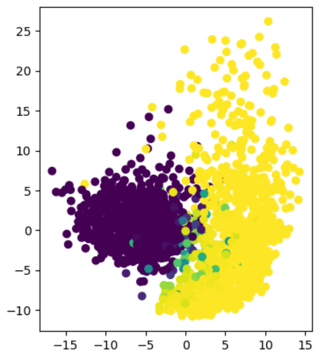
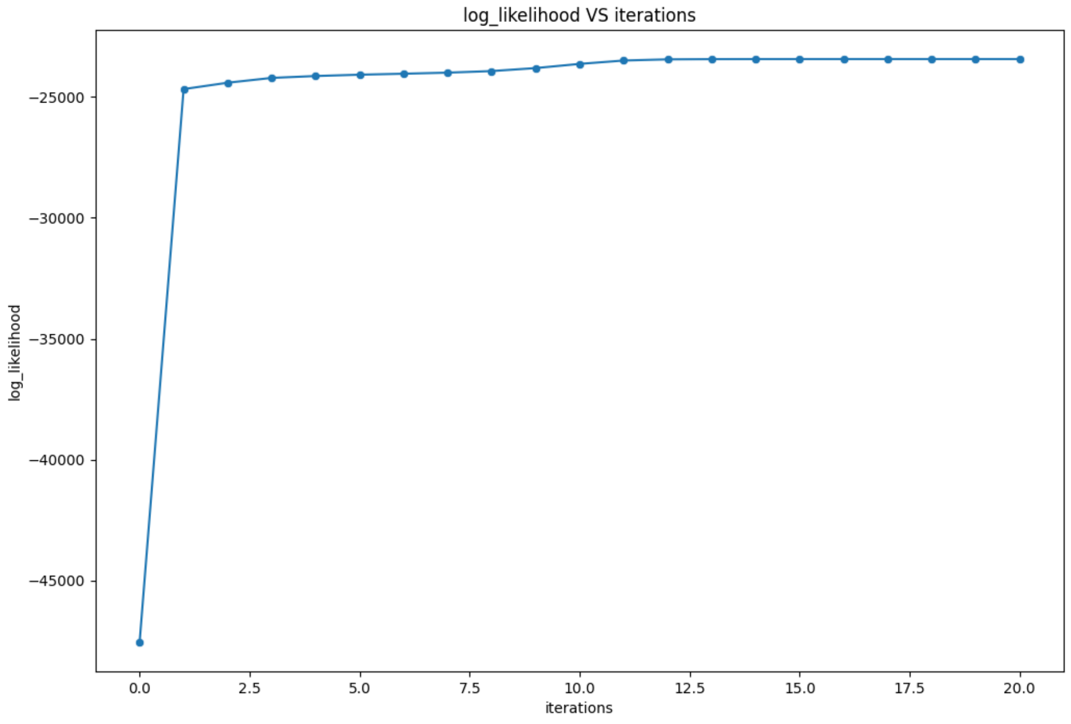


Figure 4: Iteration 20



You can also see the convergence through the log-likelihood function plotted below.

2. The mean images and numerical weight for each component are provided below. As you can see, given the slight differences in the input images and the reduced components via pca, the 'mean' image resembles a blurry version of the target labels for both Gaussian component 1 mapping to 6 and Gaussian component 2 mapping to the number 2.

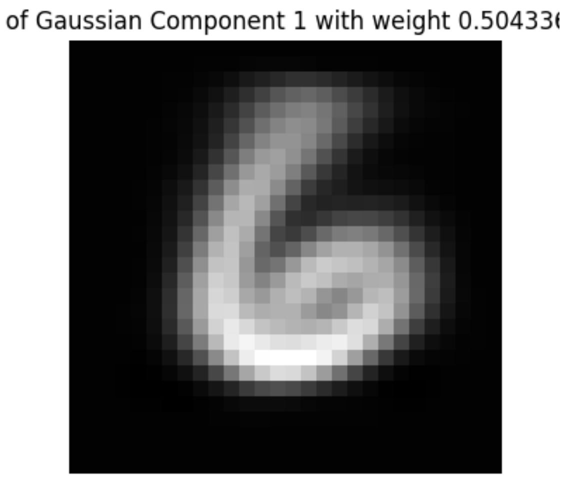


Figure 5: 'Gaussian Component 1 Weight: 0.5043364819082853

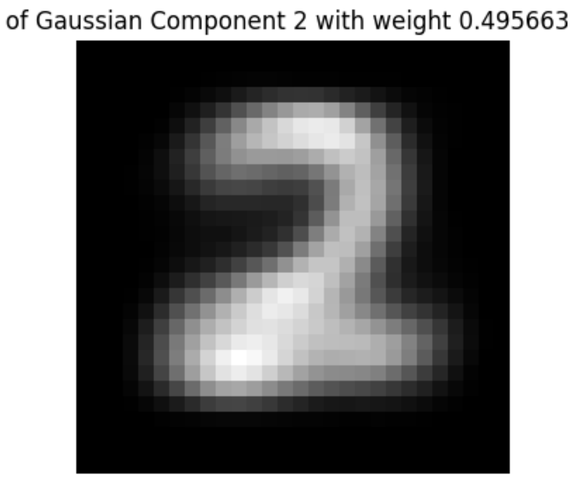
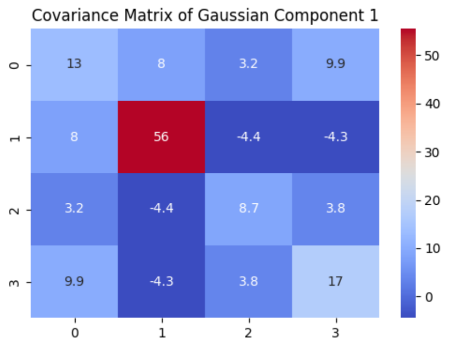
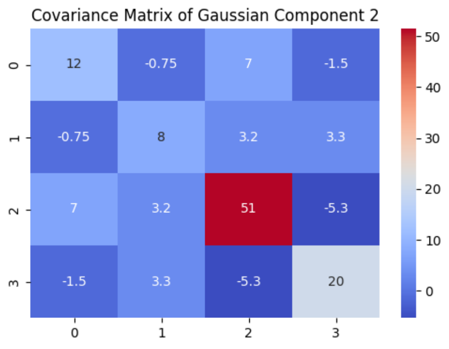


Figure 6: 'Gaussian Component 2 Weight: 0.49566351809171477

The heatmaps below show the respective covariance matrices for Gaussian components 1 and 2 corresponding to the numbers 6 and 2 respectively. As you can see the 2nd principal component has the highest variance for Gaussian component 1 and the 3rd principal component has the highest variability for Gaussian component 2. This means that whatever 'feature' is captured by the 2nd principal component for Gaussian component 1, there seem to be large differences. The same can be said for the 3rd principal component for Gaussian component 2. It does seem both Gaussian components 1 and 2 have a decent bit of variability among principal component 3 as well. For Gaussian component 1 it seems that principal component 1 has large variability as well and that principal components 1 and 3 have a pretty high covariance.For Gaussian component 2, it seems principal components 1 and 3 have moderately high covariance.





3. Inferring the labels for the images using tau and then comparing them with the ground truth labels, we were able to determine a misclassification rate for each label as well as the total. It seems GMM has a much lower misclassification rate overall as well as for both labels individually. It also seems that both kmeans and GMM misclassified 2's at over twice the rate as they misclassified 6.

Table 4: Misclassification Rates for GMM and K-means

Method	Total Misclassification Rate	Misclassification Rate for '2'	Misclassification Rate for '6'
GMM	0.0372	0.0601	0.0125
K-means	0.0543	0.0746	0.0324

Sources

1. https://stackoverflow.com/questions/64161106/how-to-plot-a-histogram-to-get-counts-for-all-unique-values
2. https://stackoverflow.com/questions/6986986/bin-size-in-matplotlib-histogram
3. https://stackoverflow.com/questions/11950375/apply-color-map-to-mpl-toolkits-mplot3d-axes3d-bar3d
4. https://stackoverflow.com/questions/30145957/plotting-2d-kernel-density-estimation-with-python
5. https://seaborn.pydata.org/generated/seaborn.kdeplot.html
6. https://www.geeksforgeeks.org/how-to-perform-a-kruskal-wallis-test-in-python/
7. https://stackoverflow.com/questions/46257627/scikit-learn-preprocessing-scale-vs-preprocessing-standardscalar
8. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
9. Demo Code Lecture 7