1. Class-Conditional Gaussians

1.1 Use Bayes' rule to derive an expression for $p(y = k/ \text{x}; \mu; \sigma)$

$$\mathrm{p(y = k|x, u, \sigma)} = \frac{p(x|y = k, u, \sigma) * p(y = k|u, \sigma)}{p(x|u, \sigma)}$$

$$= \frac{(\prod_{i=1}^{D} 2\pi{\sigma_i}^2)^{-0.5}\exp\{-\sum_{i=1}^{D}\frac{1}{2{\sigma_i}^2}(x_i - u_{ki})^2\}\pi_k}{p(x|u,\sigma)}$$

$$\mathrm{p(x|u, \sigma)} = \sum_{j=1}^{k} p(x|y = j, u, \sigma) * p(y = j|u, \sigma)$$

$$= \sum_{j=1}^{k}(\prod_{i=1}^{D} 2\pi{\sigma_{ji}}^2)^{-0.5}\exp\{-\sum_{i=1}^{D}\frac{1}{2{\sigma_i}^2}(x_i - u_{ji})^2\}\partial j$$

Thus: $$\mathrm{p(y = k|x, u, \sigma)} = \frac{\prod_{i=1}^{D}(2\pi{\sigma_i}^2)^{-0.5}\exp\{-\sum_{i=1}^{D}\frac{1}{2{\sigma_i}^2}(x_i - u_{ki})^2\}\pi_k}{\sum_{j=1}^{k}\prod_{i=1}^{D}(2\pi{\sigma_{ji}}^2)^{-0.5}\exp\{-\frac{1}{2{\sigma_i}^2}(x_i - u_{ji})^2\}}$$

1.2 $$\mathrm{p(x, y|\theta)} = p(x|y, \theta)p(y|\theta)$$

$$\mathrm{l(\theta; D)} = -\log p(x^1, y^1, x^2, y^2 \ldots . x^N, y^N|\theta)$$

$$\mathrm{l(\theta; D)} = -\sum_{n=1}^{N} \log p(x^{(n)}, y^{(n)}|\theta)$$

$$= -\sum_{n=1}^{N} \log[p(x^{(n)}|y^{(n)}, \theta)p(y^{(n)}|\theta)]$$

$$= -\sum_{n=1}^{N} \log\left[p\left(x^{(n)}\middle|y^{(n)}, \theta\right)\right] - \sum_{i=1}^{N} p(y^{(n)}|\theta)]$$

$$p\left(x^{(n)}\middle|y^{(n)}, \theta\right) = \prod_{i=1}^{D}(2\pi{\sigma_i}^2)^{-0.5}\exp\{-\frac{1}{2{\sigma_i}^2}(x_i^{(n)} - u_{ki})^2\}$$

Thus: $$\mathrm{l(\theta; D)} = -\sum_{n=1}^{N}\log\alpha_k^i - \sum_{i=1}^{D}(\log\frac{1}{\sqrt{2\pi\sigma_i^2}} + \frac{(x_i^{(n)} - u_{ki})^2}{2\sigma_i^2}]$$

1.3

$$\frac{\partial l(\theta; D)}{\partial\sigma_i^2} = \sum_{n=1}^{N}[\frac{1}{\sigma^2} - \frac{1}{\sigma^3}({x_i}^{(n)} - u_{ik})^2]$$

$$\frac{\partial l(\theta; D)}{\partial u_{ik}} = -\sum_{n=1}^{N}\frac{(x_i^{(n)} - u_{ki})}{\sigma_i^2}$$

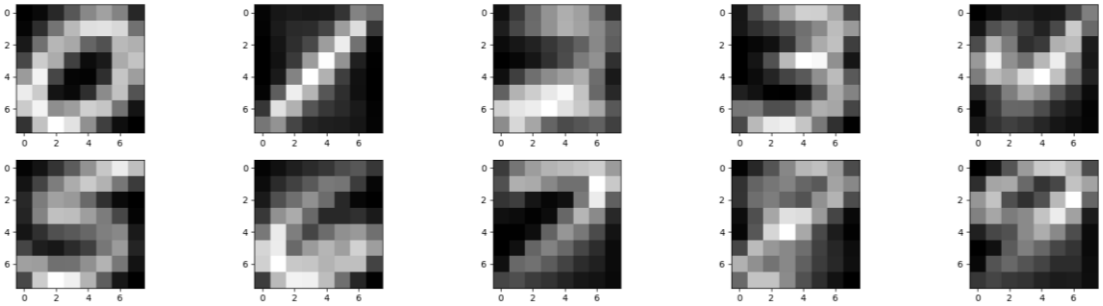
1.4

$$\mu_{ik} = \frac{\sum_{n=1}^{N} 1[y^{(m)} = k]x_i^{(n)}}{\sum_{n=1}^{N} 1[y^{(n)} = k]}$$

$$\sigma_{ik}^2 = \frac{\sum_{n=1}^{N} 1[y^{(m)} = k](x_i^{(n)} - u_{ik})^2}{\sum_{n=1}^{N} 1[y^{(n)} = k]}$$

2. Handwritten Digit Classification

2.0 Load the data and plot the means for each of the digit classes in the training data (include these in your report). Given that each image is a vector of size 64, the mean will be a vector of size 64 which needs to be reshaped as an 8 × 8 2D array to be rendered as an image. Plot all 10 means side by side using the same scale.



2.1 K-NN Classifier

1. Build a simple K nearest neighbor classifier using Euclidean distance on the raw pixel data.

(a) For K = 1 report the train and test classification accuracy.
For K=1, the train classification accuracy is: 1.0
For K=1, the test classification accuracy is: 0.96875
(b) For K = 15 report the train and test classification accuracy.
For K=15, the test classification accuracy is: 0.96075
For K=15, the train classification accuracy is: 0.963714285714

2. For K > 1 K-NN might encounter ties that need to be broken in order to make a decision. Choose any (reasonable) method you prefer and explain it briefly in your report.

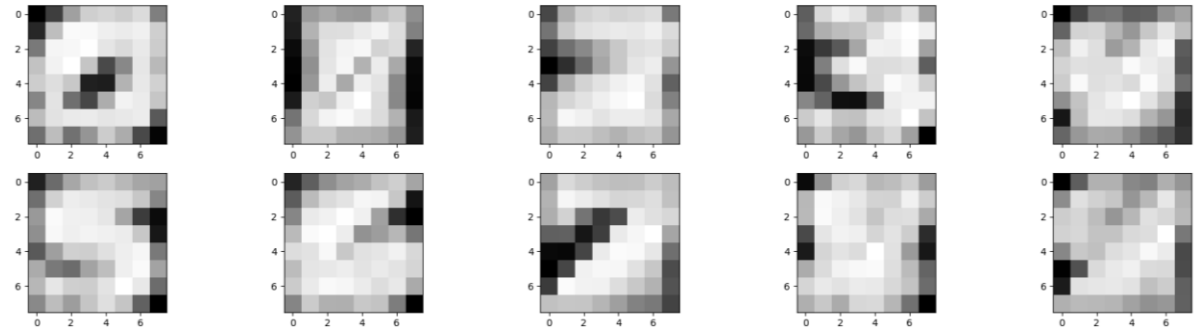
After selecting top K minimum distances of the data, let's assume K-NN meets ties between, for instance, target A and target B. We add up the distances of target A and distances of target B respectively, and choose the one with smaller sum.

3. Use 10 fold cross validation to find the optimal K in the 1-15 range. You may use the K-Fold implementation in sklearn or your existing code from Assignment 1. Report this value of K along with the train classification accuracy, the average accuracy across folds and the test accuracy.

The optimal K I found is 4 with train classification accuracy 0.986428571429, average accuracy cross folds 0.968142857143 and test accuracy 0.972.

2.2 Conditional Gaussian Classifier Training

1. Plot an 8 by 8 image of the log of the diagonal elements of each covariance matrix Σ_k. Plot all ten classes side by side using the same grayscale.



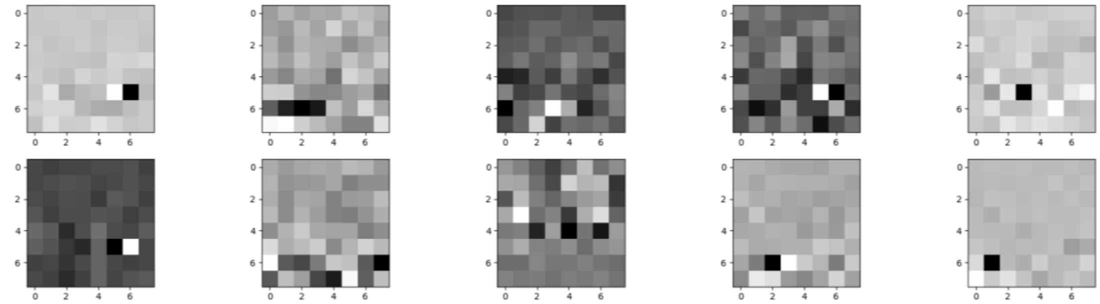
2. Using the parameters you fit on the training set and Bayes rule, compute the average conditional log-likelihood, i.e. $\frac{1}{N}\sum_{i-1}^{N}\log(p(y^i|x^i,\theta))$ on both the train and test set and report it.

The average conditional likelihood over true class labels of training data is: -0.124624436669
The average conditional likelihood over true class labels of testing data is: -0.196673203255

3. Select the most likely posterior class for each training and test data point as your prediction, and report your accuracy on the train and test set.

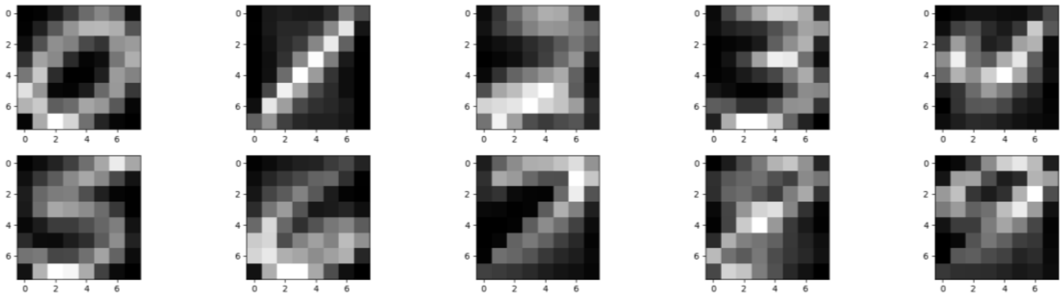
The accuracy for train data is: 0.9814285714290
The accuracy for test data is: 0.97275

4. Extra work for interested students: Compute the leading eigenvectors (largest eigenvalue) for each class covariance matrix (can use np.linalg.eig) and plot them side by side as 8 by 8 images.

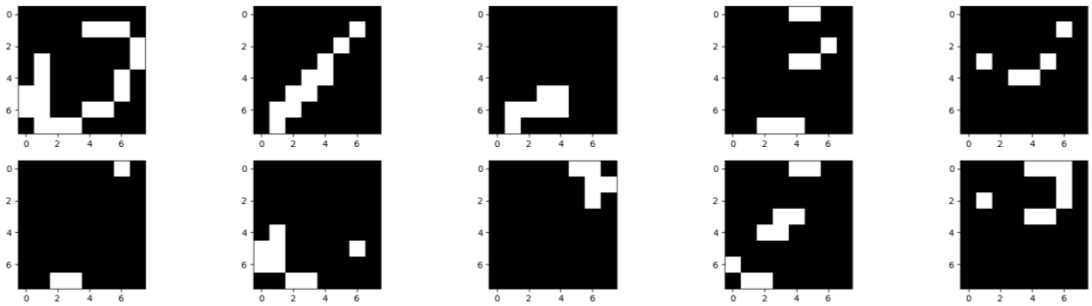


2.3 Naive Bayes Classifier Training

3. Plot each of your η(k) vectors as an 8 by 8 grayscale image. These should be presented side by side and with the same scale.



4. Given your parameters, sample one new data point for each of the 10 digit classes. Plot these new data points as 8 by 8 grayscale images side by side.



5. Using the parameters you fit on the training set and Bayes rule, compute the average conditional log-likelihood.
   The average conditional likelihood over true class labels of all training data is: -30.7935910549
   The average conditional likelihood over true class labels of all testing data is: -30.740525686

6. Select the most likely posterior class for each training and test data point, and report your accuracy on the train and test set.

The accuracy for train data is: 0.773285714286
The accuracy for test data is: 0.76325

2.4 Briefly (in a few sentences) summarize the performance of each model. Which performed best? Which performed worst? Did this match your expectations?

In this experiment, for KNN, the optimal k we choose through across validation is 4 with the test accuracy 0.972. The Conditional Gaussian Classifier performed the best with the test accuracy up to 0.97275 and Naïve Bayes Classifier performed the worst with test accuracy 0.76325.

This result matches my expectations because Naïve Bayes has the assumption that all the features are independent, which is not the case in real life.