CSC 411
Machine Learning & Data Mining
ASSIGNMENT # 2

Manasa Bharadwaj
Student**:** 1003892889
bhaman@cs.toronto.edu

Q1

$$p(y = k) = \alpha_k \tag{1}$$

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \left(\prod_{i=1}^{D} 2\pi\sigma_i^2 \right)^{-1/2} \exp\left\{ -\sum_{i=1}^{D} \frac{1}{2\sigma_i^2} (x_i - \mu_{ki})^2 \right\} \tag{2}$$

1. Use Bayes' rule to derive an expression for $p(y = k|x, \boldsymbol{\mu}, \boldsymbol{\sigma})$. [**Hint**: Use the law of total probability to derive an expression for $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})$.]

2. Write down an expression for the negative likelihood function (NLL)

$$\ell(\boldsymbol{\theta}; D) = -\log p(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, \cdots, y^{(N)}, \mathbf{x}^{(N)}|\boldsymbol{\theta}) \tag{3}$$

of a particular dataset $D = \{(y^{(1)}, \mathbf{x}^{(1)}), (y^{(2)}, \mathbf{x}^{(2)}), \cdots, (y^{(N)}, \mathbf{x}^{(N)})\}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$. (Assume that the data are iid.)

The answers for both are in next page.

Q1.1. $P(y=k|x,\mu,\sigma) = \frac{P(x|y=k,\mu_k,\sigma_k)\,P(y=k)}{P(x|\mu,\sigma)}$ (Bayes Rule)

Using Total Law of probability

$$P(x|\mu,\sigma) = \sum_y P(x|y=k,\mu_k,\sigma_k)P(y=k)$$

$$P(y=k|x,\mu,\sigma) = \frac{P(x|y=k,\mu_k,\sigma_k)\,P(y=k)}{\sum_y P(x|y=k,\mu_k,\sigma_k)\,P(y=k)} \quad -①$$

given $P(x|y=k,\mu_k,\sigma_k) = \left(\prod_{i=1}^{D}(2\pi\sigma_i^2)^{-1/2}\exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\right\}\right)$

Plugging in $\frac{①}{}$ $P(y=k)=\alpha_k$

$$P(y=k|x,\mu,\sigma) = \frac{\left(\prod_{i=1}^{D}2\pi\sigma_i^2\right)^{-1/2}\exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\right\}*\alpha_k}{\sum_{k=1}^{K}\left(\prod_{i=1}^{D}2\pi\sigma_i^2\right)^{-1/2}\exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\right\}*\alpha_k}$$

$y \rightarrow$ takes K values for K classes (assume $k=1 \rightarrow K$)

Q1.2. NLL $\Rightarrow$ $\ell(\theta;D) = -\log P(y^{(1)},x^{(1)},y^{(2)},x^{(2)},\ldots y^{(N)},x^{(N)}|\theta)$

since data is iid

$$= -\log\left(\prod_{i=1}^{N}P(x^{(i)},y^{(i)}|\theta)\right)$$

$$= -\sum_{i=1}^{N}\log(P(x^{(i)},y^{(i)}|\theta)) = -\sum_{i=1}^{N}\log\left(P(x^{(i)}|y^{(i)},\theta)\,P(y^{(i)}|\theta)\right)$$

$$= -\sum_{i=1}^{N}\log P(x^{(i)}|y^{(i)},\theta) + \log P(y^{(i)}|\theta)) \quad -①$$

given $P(x^{(i)}|y^{(i)},\theta) = \left(\prod_{j=1}^{D}(2\pi\sigma_j^2)^{-1/2}\exp\left\{-\sum_{j=1}^{D}\frac{1}{2\sigma_j^2}(x_j^{(i)}-\mu_{y^{(i)}j})^2\right\}\right)$

$P(y=k,\theta)=\alpha_k$

plugging in values in ① we get

$$NLL = -\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{D}-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma_j^2 - \sum_{j=1}^{D}\frac{1}{2\sigma_j^2}(x_j^{(i)}-\mu_{y^{(i)}j})^2\right) + \log(\alpha_{y^{(i)}})\right]$$

3. Take partial derivatives of the likelihood with respect to each of the parameters μ_{ki} and with respect to the shared variances σ_i^2.

Q1.3; 1.4 Partial Derivations Calculation & Values for μ_{kj} σ_j^2.

$$NLL = -\sum_{i=1}^{N}\left[-\frac{1}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{D}\left(\log \sigma_j^2 + \frac{1}{\sigma_j^2}(x_j^{(i)} - \mu_{y^{(i)}j})^2\right) + \log(\alpha_{y^{(i)}})\right]$$

$$\frac{\partial NLL}{\partial \mu_{kj}} = -\sum_{i=1}^{N} \mathbb{1}(y^{(i)}=k)\,(x_j^{(i)} - \mu_{kj}) = 0$$

$$\sum_{i=1}^{N} \mathbb{1}(y^{(i)}=k)\,\mu_{kj} = \sum_{i=1}^{N} \mathbb{1}(y^{(i)}=k)\,x_j^{(i)}$$

$$\mu_{kj} = \frac{\sum_{i=1}^{N} \mathbb{1}(y^{(i)}=k)\,x_j^{(i)}}{\sum_{i=1}^{N} \mathbb{1}(y^{(i)}=k)} \longrightarrow \mu_{kj}$$

MLE estimate for means

$$\frac{\partial NLL}{\partial \sigma_j^2} = -\sum_{i=1}^{N} -\frac{1}{2}\left(\frac{1}{\sigma_j^2} + \frac{(x_j^{(i)} - \mu_{y^{(i)}j})^2}{(\sigma_j^2)^2}\right) = 0$$

$$\sum_{i=1}^{N} \sigma_j^2 = \sum_{i=1}^{N} (x_j^{(i)} - \mu_{y^{(i)}j})^2$$

$$N\sigma_j^2 = \sum_{i=1}^{N} (x_j^{(i)} - \mu_{y^{(i)}j})^2$$

$$\sigma_j^2 = \frac{\sum_{i=1}^{N} (x_j^{(i)} - \mu_{y^{(i)}j})^2}{N}$$

MLE estimate for variance

Q1_4. Find the maximum likelihood estimates for μ and σ

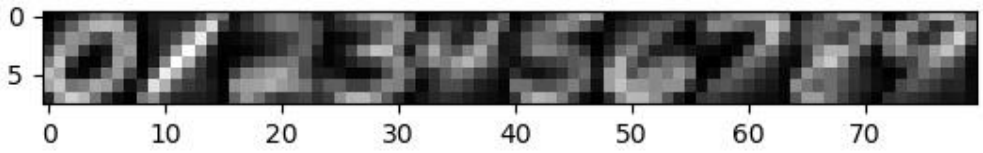
$\boldsymbol{\mu}$ is a K*D matrix, where each element is μ_{kj} as computed above.

$\boldsymbol{\sigma}$ is a 1*D vector where each element is σ_j^2 as computed above.

Since variance are shared among all classes, this is just a vector.

Q2_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.



Plot for means of pixel values

Q2_1. k-NN Classifier

1. (a) For K = 1 report the train and test classification accuracy.

   For k 1 Train error is 100.0% and Test Error is 96.875%

   (b) For K = 15 report the train and test classification accuracy.

   For k 15 Train error is 96.1% and Test Error is 95.925%

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.

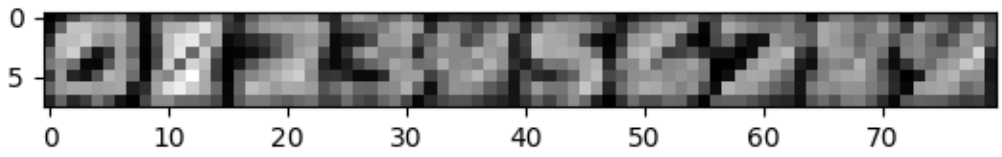
   I have reduced K till ties are broken. In worst case we reduce K till 1, and there are no ties for K=1.

3. Use 10-fold cross validation to find the optimal K in the 1-15 range. Report this value of K along with the train classification accuracy, the average accuracy across folds and the test accuracy.

   Best k 3 train error 98.5984126984127% cv error 96.51428571428572% test error 96.975%

Q2_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.



Plot of covariances

2. Using the parameters, you fit on the training set and Bayes rule, compute the average conditional log-likelihood, on both the train and test set and report it.

   Average conditional for training data -0.124624436669
   Average conditional for test data -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.

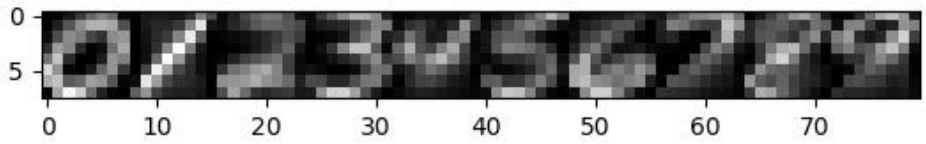
   Train accuracy 98.14285714285714%
   Test accuracy 97.275%

Q2_3. Naïve Bayes Classifier Training

1. Convert the real-valued features x into binary features b using 0:5 as a threshold: **Already done in starter code**

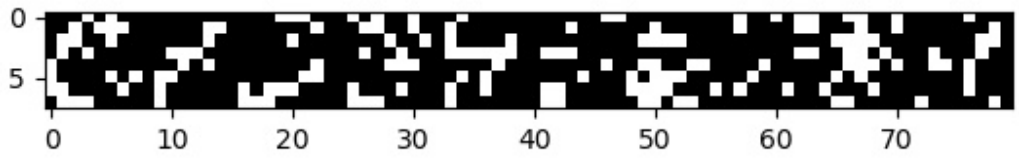
2. You should compute parameters eta 10*64 - **It is computed in code.**

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

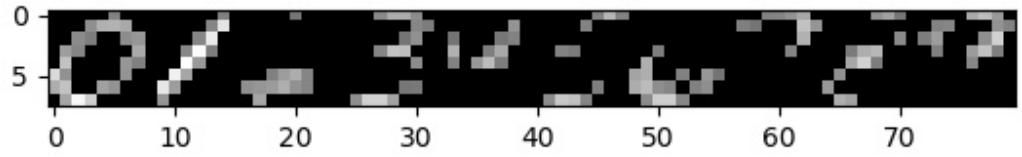


ETA 10*64 reshaped to (8,8)

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.

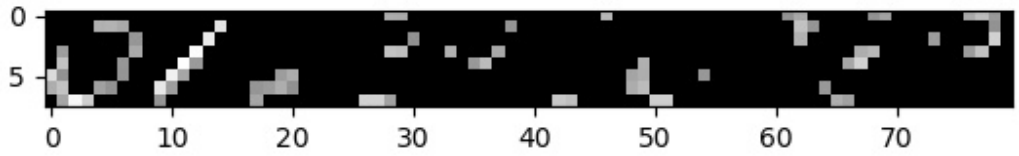


Plot of newly generated data using binomial distribution



Plot of newly generated data using threshold of 0.4

Plot of



newly generated data using threshold of 0.5

5. Using the parameters, you fit on the training set and Bayes rule, compute the average conditional log-likelihood, on both the train and test set and report it.

   Average conditional for training data -0.9437538618
   Average conditional for test data -0.987270433725

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

   train error 77.41428571428571
   test error 76.425

2.4 Model Comparison

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

Best: Conditional Gaussian Classifier
Worst: Naïve Bayes

k-NN performed very well, and close to Gaussian. But takes lot more time than the other two. This delay is as expected, since for each point we calculate seven thousand distances. This accuracy is better than what I expected. It is possible that given data doesn't have many outliers, causing k-NN to produce bad labels. Data is highly structured.

Gaussian performed the best. I think this is because the parameters mean and covariance model the feature values for a given class very well. This matched my expectations.

Naïve Bayes performed the worst. This could be due to converting features to binary values. This is a lossy conversion. This will take away the advantage which Gaussian classifier had. This is worse than what I expected. I expected this to be inferior to gaussian, but not by accuracy difference of close to 20% percentage.