

CSC2515 Assignment 2

Tianxiang Chen 999473181

I. QUESTION 1

1. The derivation of $p(y = k|x, \boldsymbol{\mu}, \boldsymbol{\sigma})$ is:

$$\begin{aligned}
 p(y = k|x, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \frac{p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) * p(y = k)}{p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})} \\
 &= \frac{p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) * p(y = k)}{\sum_{i=1}^k p(\mathbf{x}|y = i, \boldsymbol{\mu}, \boldsymbol{\sigma}) * p(y = i)} \\
 &= \frac{\left(\prod_{j=1}^D 2\pi\sigma_j^2 \right)^{-1/2} \exp \left\{ -\sum_{j=1}^D \frac{1}{2\sigma_j^2} (x_j^{(i)} - \mu_{kj})^2 \right\} * \alpha_k}{\sum_{i=1}^k \left(\left(\prod_{j=1}^D 2\pi\sigma_j^2 \right)^{-1/2} \exp \left\{ -\sum_{j=1}^D \frac{1}{2\sigma_j^2} (x_j^{(i)} - \mu_{ij})^2 \right\} * \alpha_i \right)}
 \end{aligned} \tag{1}$$

2. Given assumption of Naive Bayes and iid. of the data, the follows can be derived:

$$\begin{aligned}
 \ell(\boldsymbol{\theta}; D) &= -\log p(y^{(1)}, x^{(1)}, y^{(2)}, x^{(2)}, \dots, y^{(N)}, x^{(N)}|\boldsymbol{\theta}) \\
 &= -\log(p(y^{(1)}, x^{(1)}|\boldsymbol{\theta}) * (p(y^{(2)}, x^{(2)}|\boldsymbol{\theta})) * \dots * p(y^{(N)}, x^{(N)}|\boldsymbol{\theta})) \\
 &= -\log \left(\prod_{i=1}^n p(y^{(i)}, x^{(i)}|\boldsymbol{\theta}) \right) \\
 &= -\log \left(\prod_{i=1}^n (p(x^{(i)}|y^{(i)}, \boldsymbol{\theta}) * p(y^{(i)})) \right) \\
 &= -\log \left(\prod_{i=1}^n (p(x^{(i)}|y^{(i)}, \boldsymbol{\theta}) * p(y^{(i)})) \right) \\
 &= -\log \left(\prod_{i=1}^n \left(\left(\prod_{j=1}^D 2\pi\sigma_j^2 \right)^{-1/2} \exp \left\{ -\sum_{j=1}^D \frac{1}{2\sigma_j^2} (x_j^{(i)} - \mu_{y^{(i)}j})^2 \right\} * \alpha_{y^{(i)}} \right) \right) \\
 &= -\log \left(\prod_{i=1}^n \alpha_{y^{(i)}} \right) + \frac{n}{2} \sum_{j=1}^D \log(2\pi\sigma_j^2) + \sum_{i=1}^n \sum_{j=1}^D \frac{(x_j^{(i)} - \mu_{y^{(i)}j})^2}{2\sigma_j^2}
 \end{aligned} \tag{2}$$

3. Taking partial derivatives of the likelihood with respect to each of the parameters μ_{ki} :

$$-\frac{\partial \ell(\boldsymbol{\theta}; D)}{\partial \mu_{ki}} = -\mathbb{1}[y^{(i)} = k] \sum_{i=1}^n \sum_{j=1}^D \frac{(x_j^{(i)} - \mu_{y^{(i)}j})}{\sigma_j^2} \tag{3}$$

Taking partial derivatives of the likelihood with respect to the shared variances σ_i^2 :

$$-\frac{\partial \ell(\boldsymbol{\theta}; D)}{\partial \sigma_i^2} = \mathbb{1}[y^{(j)} = k] \left[\left(\frac{n}{2} \sum_{i=1}^D \frac{1}{\sigma_i^2} - \sum_{j=1}^n \sum_{i=1}^D \frac{(x_i^{(j)} - \mu_{y^{(j)}i})^2}{2\sigma_i^4} \right) \right] \tag{4}$$

4. Setting Eq.(3) to zero, the maximum likelihood estimates for μ is:

$$\begin{aligned} \mathbb{1}[y^{(i)} = k] \sum_{i=1}^n \sum_{j=1}^D \mu_{kj} &= \mathbb{1}[y^{(i)} = k] \sum_{i=1}^n \sum_{j=1}^D x_j^{(i)} \\ &= \sum_{i=1}^n \mathbb{1}[y^{(i)} = k] x^{(i)} \end{aligned} \quad (5)$$

Rearrange the equation, we get:

$$\mu_k = \frac{\sum_{i=1}^n \mathbb{1}[y^{(i)} = k] x^{(i)}}{\sum_{i=1}^n \mathbb{1}[y^{(i)} = k]} \quad (6)$$

Setting Eq.(4) to zero, the maximum likelihood estimates for σ is:

$$\mathbb{1}[y^{(j)} = k] \frac{n}{2} \sum_{i=1}^D \frac{1}{\sigma_i^2} = \mathbb{1}[y^{(j)} = k] \sum_{j=1}^n \sum_{i=1}^D \frac{(x_i^{(j)} - \mu_{y^{(j)}i})^2}{2\sigma_i^4} \quad (7)$$

Rearrange the equation, for each i in D , we get:

$$\sigma_i^2 = \frac{\sum_{j=1}^n \mathbb{1}[y^{(j)} = k] (x_i^{(j)} - \mu_{y^{(j)}i})^2}{n} \quad (8)$$

II. QUESTION 2

In this part, only the questions asked in the handout are answered. For details about the code implemented, please checked the *.py files uploaded.

0. The plot of the means for each digit class in the training set is shown below:

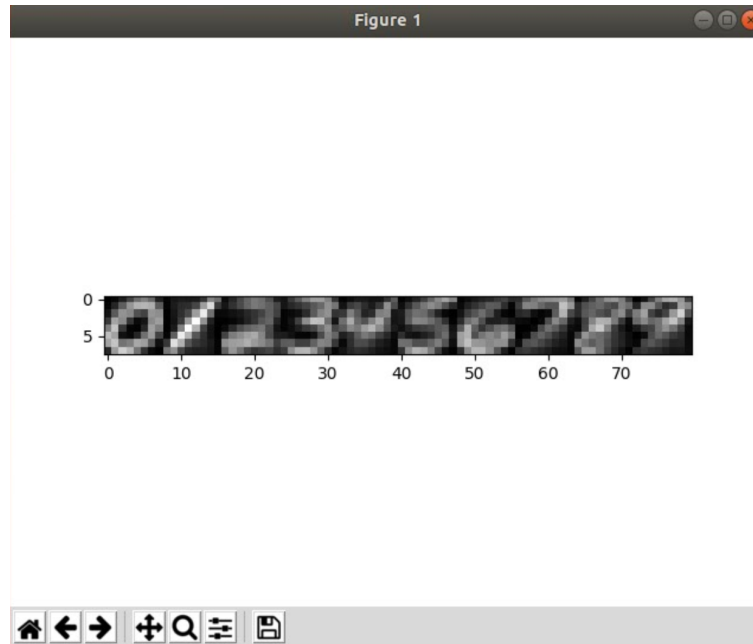


Fig. 1: Means for Each Digit Class in the Training Set

1. K-NN Classifier

For $K=1$ and $K=15$, the train and test classification accuracy are:

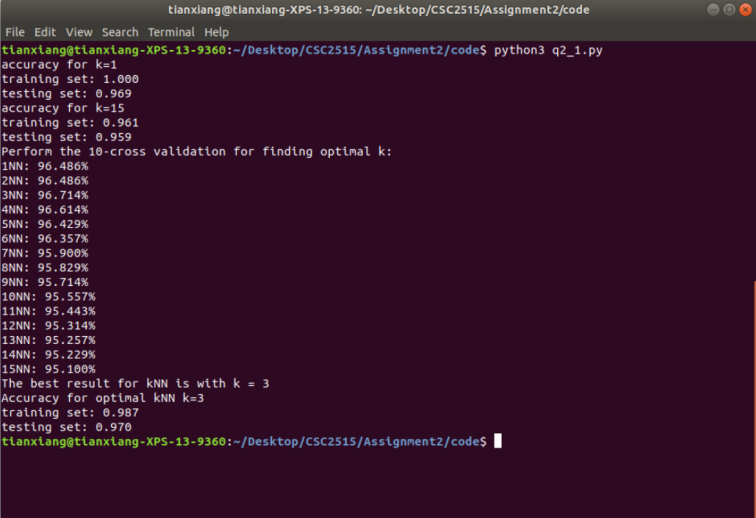
	Train Set Accuracy	Test Set Accuracy
$K=1$	1.000	0.969
$K=15$	0.961	0.959

For $K > 1$ K-NN might encounter ties that need to be broken in order to make a decision. The strategy I used is to decrease the K value by 1 and redo the " $K-1$ " nearest neighbour. This process is repeated i times until for a value " $K-i$ " there is no tiers.

Using 10 fold cross validation for finding optimal K in the 1-15 range:

	Result
Optimal K	3
Average accuracy across folds for $K=3$	0.967
Train Set Accuracy for $K=3$	0.987
Test Set Accuracy for $K=3$	0.970

Figure below shows the result from running the code.



```
tianxiang@tianxiang-XPS-13-9360: ~/Desktop/CSC2515/Assignment2/code
File Edit View Search Terminal Help
tianxiang@tianxiang-XPS-13-9360:~/Desktop/CSC2515/Assignment2/code$ python3 q2_1.py
accuracy for k=1
training set: 1.000
testing set: 0.969
accuracy for k=15
training set: 0.961
testing set: 0.959
Perform the 10-cross validation for finding optimal k:
1NN: 96.486%
2NN: 96.486%
3NN: 96.714%
4NN: 96.614%
5NN: 96.429%
6NN: 96.357%
7NN: 95.908%
8NN: 95.829%
9NN: 95.714%
10NN: 95.557%
11NN: 95.443%
12NN: 95.314%
13NN: 95.257%
14NN: 95.229%
15NN: 95.100%
The best result for kNN is with k = 3
Accuracy for optimal kNN k=3
training set: 0.987
testing set: 0.970
tianxiang@tianxiang-XPS-13-9360:~/Desktop/CSC2515/Assignment2/code$
```

Fig. 2: Q2.1 Result

2. Conditional Gaussian Classifier Training

Images of the log of the diagonal elements of each covariance matrix Σ_k is shown in Fig. 3.

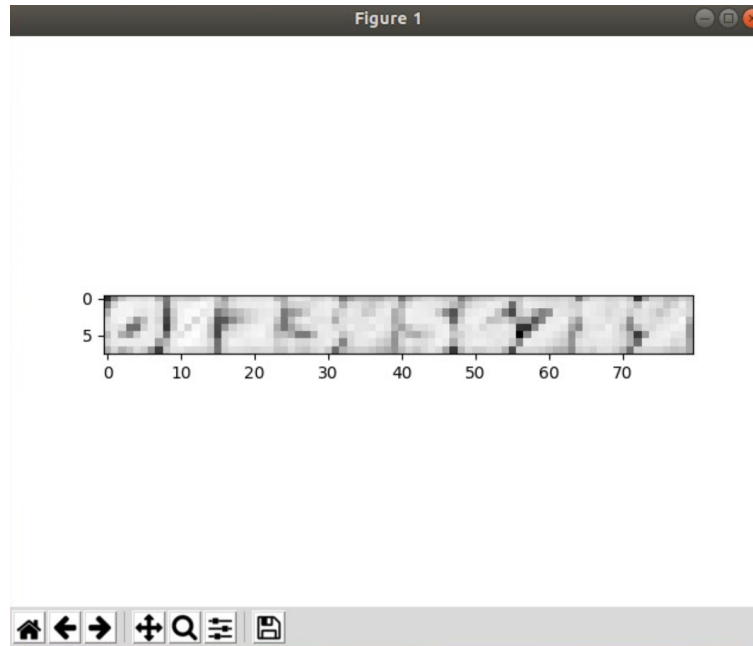


Fig. 3: Log of the diagonal elements of each covariance matrix Σ_k

Using the parameters fit on the training set and Bayes rule, the average conditional log-likelihood is:

training set: -0.1246

testing set: -0.1967

Select the most likely posterior class for each training and test data point as prediction, the accuracy on the train and test set is:

training set: 0.981

testing set: 0.973

3. Naive Bayes Classifier Training

η_k vectors as an 8 by 8 grayscale image is plotted in Fig. 4.

A new data sample for each class using the parameter from the generative model before is plotted in Fig. 5.

Using the parameters fit on the training set and Bayes rule, the average conditional log-likelihood is:

training set: -0.9438

testing set: -0.9873

Select the most likely posterior class for each training and test data point as prediction, the accuracy on the train and test set is:

training set: 0.774

testing set: 0.764

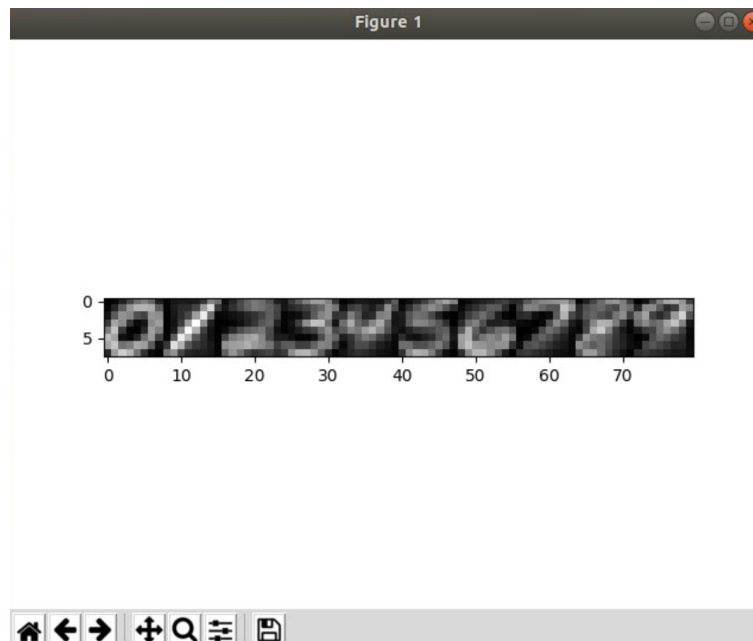


Fig. 4: η_k vectors

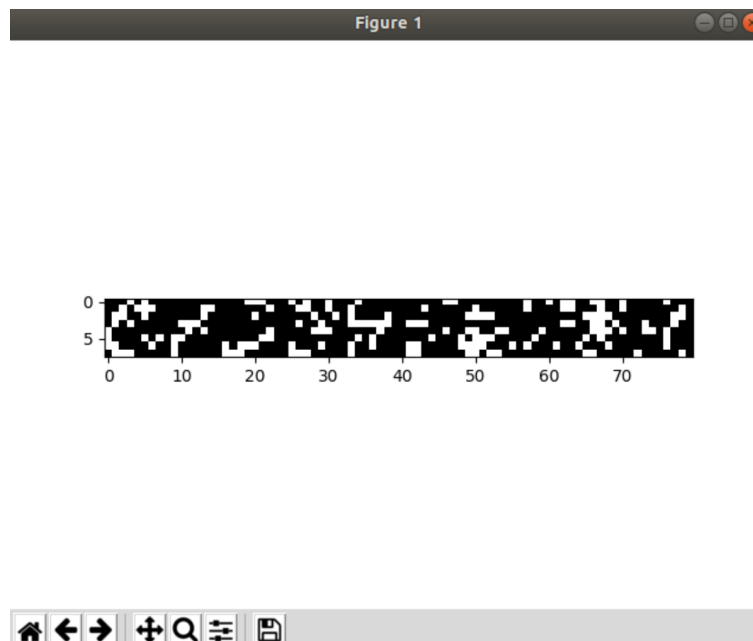


Fig. 5: A New Data Sample for Each Class

4. Model Comparison

For K-NN, we got train accuracy = 1 when $K=1$. This matches our expectation since it just tries to find the closest distance towards one point from the training set, which is the point itself. But for $K=1$, due to over-fitting, the testing and folded accuracy are not as good as training's. From the 10 fold cross validation, the optimal K is found as 3 in range $K=1-15$. One problem of K-NN is expensive at runtime. As indicated from lecture slides, the runtime is $O(kdN)$ for K-NN. For this assignment, since the input dimension is not that big, we could still get a reasonable runtime.

For Conditional Gaussian Classifier, it gives good accuracy. The model take the correlation of different feature into consideration. It has covariance matrixs as well as the mean vectors. These factors guaranteed that accuracy of this model.

For Naive Bayes Classifier, it makes a strong assumption that each feature is independent. Due to this factor, the accuracy

is not as good as the previous two model. But at the same time, it also relax the calculation and leads to a small runtime.

Overall, K-NN and the Conditional Gaussian Classifier have a good performance while the Naive Bayes Classifier has the worst performance but runs fastest among the 3 models.