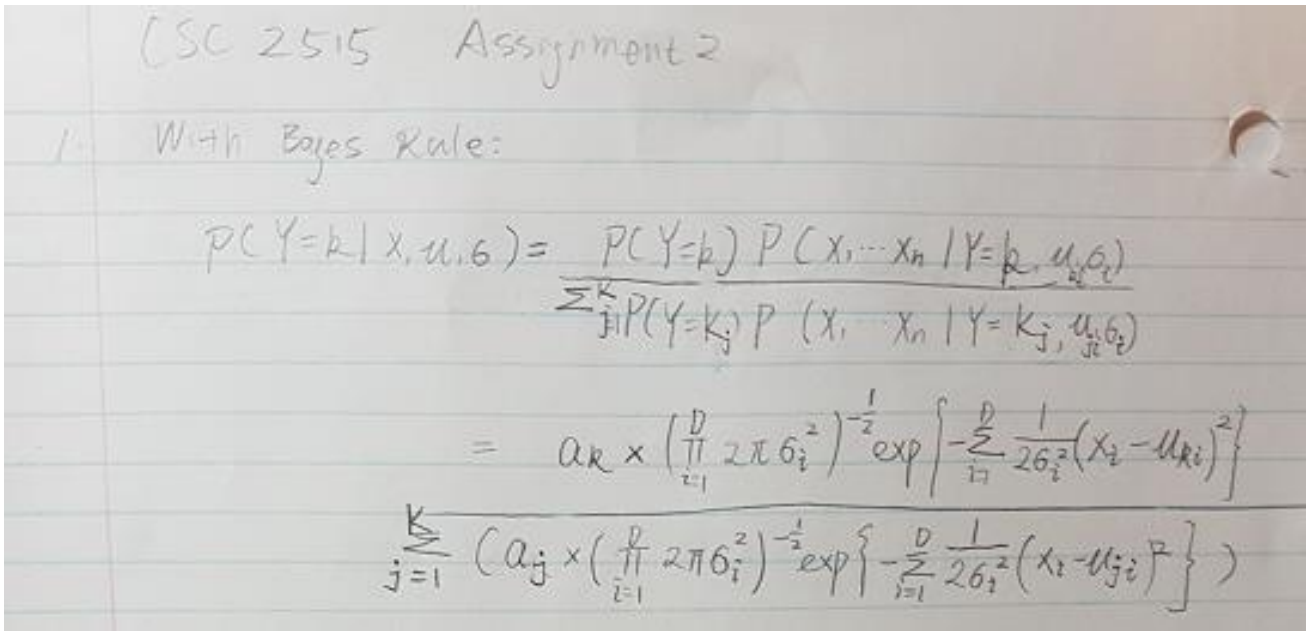
Q1.1:



CSC 2515 Assignment 2

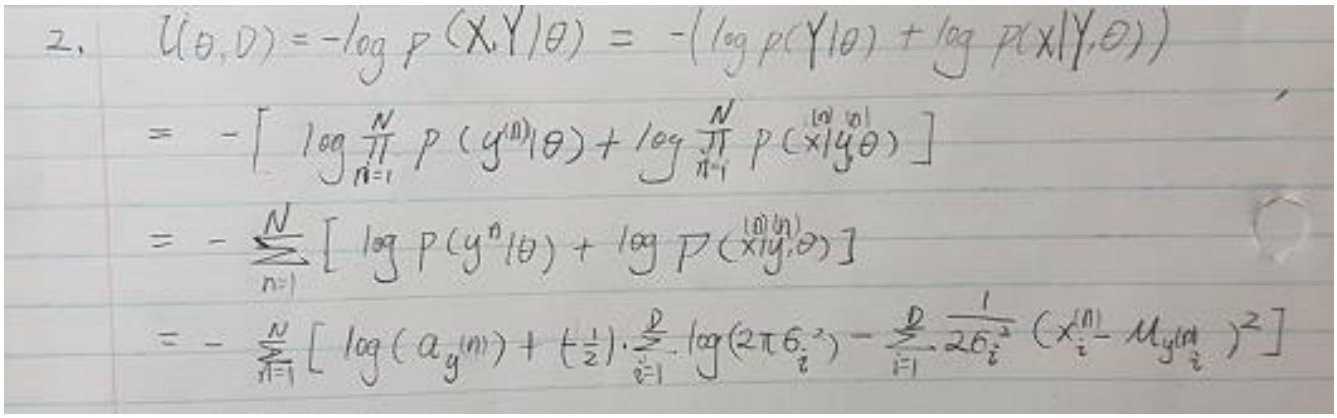
1. With Bayes Rule:

$$p(Y=k \mid x, u, \sigma) = \frac{P(Y=k)\, P(x_1 \cdots x_n \mid Y=k, u_k, \sigma_i)}{\sum_{j=1}^{K} P(Y=k_j)\, P(x_1 \cdots x_n \mid Y=k_j, u_{ji}, \sigma_i)}$$

$$= a_k \times \left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} \exp\left\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2\right\}$$

$$\sum_{j=1}^{K} \left(a_j \times \left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} \exp\left\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2\right\}\right)$$

Q1.2



2.

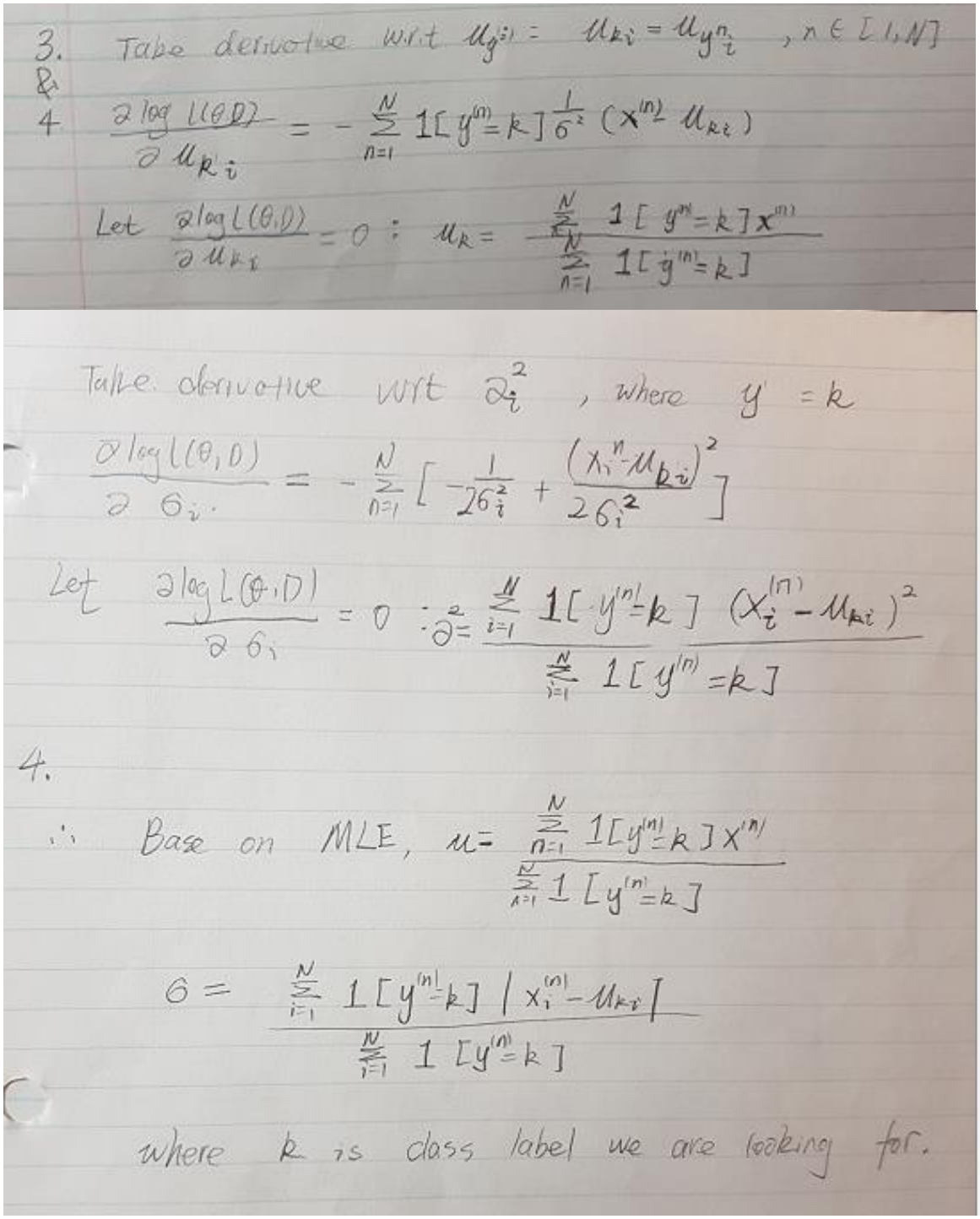
$$U(\theta, D) = -\log p(X, Y \mid \theta) = -(\log p(Y \mid \theta) + \log p(X \mid Y, \theta))$$

$$= -\left[\log \prod_{n=1}^{N} p(y^{(n)} \mid \theta) + \log \prod_{n=1}^{N} p(x^{(n)} \mid y^{(n)}, \theta)\right]$$

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

$$= -\sum_{n=1}^{N}\left[\log(a_{y^{(n)}}) + (-\tfrac{1}{2}) \cdot \sum_{i=1}^{D} \log(2\pi\sigma_i^2) - \sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i^{(n)} - u_{y^{(n)}i})^2\right]$$

Q1.3 & Q1.4:



3. Take derivative w.r.t $u_{ji} = u_{ki} = u_{y_i^n}$, $n \in [1, N]$

&

4 $$\frac{\partial \log L(\theta|D)}{\partial u_{ki}} = -\sum_{n=1}^{N} 1[y^{(n)}=k] \frac{1}{\sigma_i^2}(x^{(n)} - u_{ki})$$

Let $\frac{\partial \log L(\theta|D)}{\partial u_{ki}} = 0$: $u_k = \frac{\sum_{n=1}^{N} 1[y^{(n)}=k] x^{(n)}}{\sum_{n=1}^{N} 1[y^{(n)}=k]}$

Take derivative wrt σ_i^2, where $y = k$

$$\frac{\partial \log L(\theta|D)}{\partial \sigma_i} = -\sum_{n=1}^{N} \left[-\frac{1}{2\sigma_i^2} + \frac{(x_i^n - u_{ki})^2}{2\sigma_i^4} \right]$$

Let $\frac{\partial \log L(\theta|D)}{\partial \sigma_i} = 0$: $\sigma^2 = \frac{\sum_{i=1}^{N} 1[y^{(n)}=k] (x_i^{(n)} - u_{ki})^2}{\sum_{i=1}^{N} 1[y^{(n)}=k]}$

4.

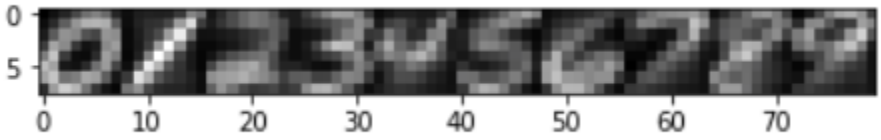
∴ Base on MLE, $u = \frac{\sum_{n=1}^{N} 1[y^{(n)}=k] x^{(n)}}{\sum_{n=1}^{N} 1[y^{(n)}=k]}$

$$\sigma = \frac{\sum_{i=1}^{N} 1[y^{(n)}=k] \, | x_i^{(n)} - u_{ki} |}{\sum_{i=1}^{N} 1[y^{(n)}=k]}$$

where k is class label we are looking for.

Q2.0

Plot all 10 means side by side using the same scale:



Q2.1

1.

```
Train accuracy for K=1: 1.0
Test accuracy for K=1: 0.96875
Train accuracy for K=15: 0.9637142857142857
Test accuracy for K=15: 0.96075
```

(a) According to the code, when K=1 train accuracy = 100%, test accuracy = 96.875%

(b) According to the code, when K=15 train accuracy = 96.37%, test accuracy = 96.075%

2.

Tie Breaking:

For the topk nearest points, record the count of each class, the sum of distance of each account to the test datum. Choose the class with the most vote. If multiple classes have the same votes, choose the one with smallest summation of distance to the test datum. If both the votes and distance summation are same, sort class labels in alphabetical order, choose the class label with the lowest alphabetical order.

3. Report optimal K along with the train, test and average accuracy.

In the code, the average accuracy across folds are recorded for all K values in 1-15 and the K with the highest score is chosen. It turns out K=4 is the optimal solution.

Below are the average validation scores on each fold:

Val_scores on each fold :
[0.96357142857142841, 0.96357142857142841,0.96542857142857152,
0.96628571428571441, 0.96357142857142863,
0.96471428571428564, 0.95942857142857163, 0.96014285714285708,
0.95700000000000007, 0.95685714285714274, 0.95685714285714274,
0.95514285714285718, 0.95414285714285718, 0.95214285714285718]

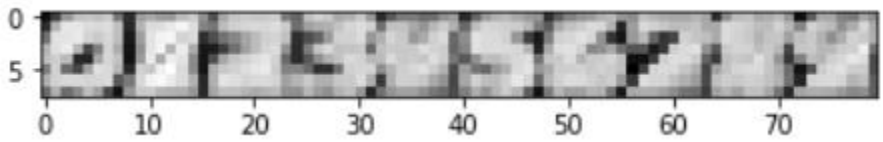
Optimal hyperparameter K is : 4

With the optimal K value, the train, test accuracy are shown below:

Train classification accuracy: 0.9864285714285714
Average Accuracy across folds: 0.96628571428571441
Test Accuracy: 0.972

Q2.2

1. Plot all ten classes side by side using the same grayscale:



2. Compute the average conditional log-likelihood:

   Below are the results from the code output:

   Average conditional log-likelihood on train set for each true label is: -0.12462443666863021
   Average conditional log-likelihood on test set for each true label is: -0.19667320325525584

3. Predict the data and report the accuracy on the train and test set:

   Below are the results from the code output:

   Accuracy on train set is : 0.9814285714285714
   Accuracy on test set is : 0.97275

Q2.3

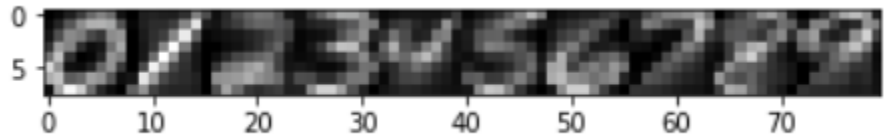
1. Convert the features into binary features:

   As in the code

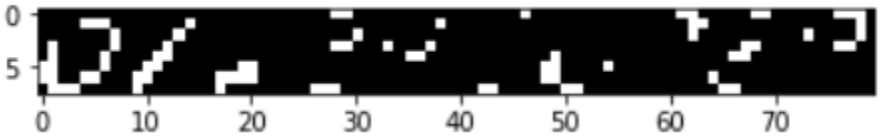
2. Train a Bernoulli Naïve Bayes classifier using MAP estimation.

   As in the code

3. Plot each of vectors:



4. Given the parameters, sample one new data point for each of the 10 digit classes. Plot theses new data points:



5. Compute the average conditional log-likelihood on both train and test set:

Below are the results from the code output:

```
Average conditional log-likelihood on train set is:
-0.9437538618002537
Average conditional log-likelihood on test set is:
-0.9872704337253584
```

6. Predict for each training and test data point, and report the accuracy:

   Below are the results from the code output:

```
Accuracy on train set is : 0.7741428571428571
Accuracy on test set is : 0.76425
```

2.4 Model Comparision

According to the accuracy score on both training and test data set obtained by the 3 methods above, it's clear that both *K-NN classifier* and *Conditional Gaussian Classifier* have relative high accuracy score. K-NN classifier and conditional Gaussian Classifier have test scores of 97.2% and 97.275% respectively. Meanwhile, the Naïve Bayes classifier has the lowest score of 76.4%. The performances match my expectation.

K-NN is a discriminative model, we make prediction based on the similarity between a test datum and training data, and select the most likely class. By tuning the hyperparameter K, we can get reasonable accuracy.

Conditional Gaussian Classifier is a generative model, it models the generative and conditional likelihood based on the train data. The implicit assumption of using a conditional Gaussian classifier is that there might be correlation between different classes, which is represented by the covariance matrix. We train the model to find the covariance matrix, and make prediction based on the MAP.

The Naïve Bayes classifier has the worst accuracy. This is because it's assumption doesn't hold for this example. Naïve Bayes model assumes the features in certain class are independent, however, the pixels (features) in certain class are highly correlated with each other. For example, in label '7', if knowing the pixel in the left top corner is on, it's highly likely the pixels will be on in other part of '7'. Thus, the accuracy score for Naïve Bayes classifier is the lowest in this example.