$\boxed{1}$ **1.** $$p(y=k \mid x,\mu,\sigma) = \frac{p(x \mid y=k,\mu,\sigma)\, p(y=k)}{p(x \mid y=k,\mu,\sigma)}$$

where $y \in \{1,2,\dots K\}$

$$= \frac{p(x \mid y=k,\mu,\sigma)\, p(y=k)}{\sum_{i=1}^{K} p(x \mid y=i,\mu,\sigma)\, p(y=i)}$$

2. $$\ell(\theta;D) = -\log(p(D \mid a,\mu,\sigma))$$

$$= -\log\left(\prod_{n=1}^{N} p(x^{(n)},y^{(n)})\right) = -\sum_{n=1}^{N} \log(p(x^{(n)},y^{(n)}))$$

where $$p(x^{(n)},y^{(n)}) = p(x^{(n)} \mid y^{(n)}=k)\, p(y^{(n)})$$

$$= \left(\prod_{i=1}^{d} 2\pi\sigma_i^2\right)^{-1/2} \exp\left\{-\sum_{i=1}^{d} \frac{1}{\sigma_i^2}(x_i^{(n)} - \mu_{ki})^2\right\} a_k$$

and $D = \{(y^{(1)}, x^{(1)}), (y^{(2)}, x^{(2)}), \dots (y^{(N)}, x^{(N)})\}$

3. $$\frac{\partial}{\partial \mu_{ki}} -\sum_{n=1}^{N} \log(p(x^{(n)},y^{(n)})) = -\sum_{n=1}^{N} \frac{\partial}{\partial \mu_{ki}} \log(p(x^{(n)},y^{(n)}))$$

and similarly for $\frac{\partial}{\partial \sigma_i^2} -\sum_{n=1}^{N} \log(p(x^{(n)},y^{(n)}))$, so we find the partial derivatives for one element $\log(p(x^{(n)},y^{(n)}))$.

$$\log(p(x^{(n)}, y^{(n)})) = \log\left[\left(\prod_{i=1}^{d} 2\pi\sigma_i^2\right)^{-1/2} \exp\left\{-\sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(x_i^{(n)} - \mu_{Ki})^2\right\} a_K\right]$$

$$= -\frac{1}{2}\log\left(\prod_{i=1}^{d} 2\pi\sigma_i^2\right) - \sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(x_i^{(n)} - \mu_{Ki})^2 + \log(a_K)$$

$$= -\frac{1}{2}\left(d\log(2\pi) + \sum_{i=1}^{d} \log(\sigma_i^2)\right) - \sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(x_i^{(n)} - \mu_{Ki})^2 + \log(a_K)$$

$$= f(\sigma_i^2, \mu_{Ki})$$

$$\frac{\partial}{\partial \mu_{Ki}} f(\sigma_i^2, \mu_{Ki}) = -\frac{\partial}{\partial \mu_{Ki}} \sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(x_i^{(n)} - \mu_{Ki})^2$$

$$= -\sum_{i=1}^{d} \frac{1}{2\sigma_i^2} \frac{\partial}{\partial \mu_{Ki}} (x_i^{(n)} - \mu_{Ki})^2$$

$$= -\sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(2\mu_{Ki} - 2x_i^{(n)}) = -\sum_{i=1}^{d} \frac{1}{\sigma_i^2}(\mu_{Ki} - x_i^{(n)})$$

$$\frac{\partial}{\partial \sigma_i^2} f(\sigma_i^2, \mu_{Ki}) = -\frac{1}{2} \frac{\partial}{\partial \sigma_i^2} \sum_{i=1}^{d} \log(\sigma_i^2) - \frac{\partial}{\partial \sigma_i^2} \sum_{i=1}^{d} \frac{1}{2\sigma_i^2}(x_i^{(n)} - \mu_{Ki})^2$$

$$= -\frac{1}{2}\sum_{i=1}^{d} \frac{1}{\sigma_i^2} + \sum_{i=1}^{d} \frac{1}{2\sigma_i^4}(x_i^{(n)} - \mu_{Ki})^2$$

Note: The summation $\sum_{i=1}^{d}(\ldots)$ is kept for clarity, but for a given i, μ_{Ki}, σ_i^2 only the corresponding element of the summation is non-zero.

4. To find MLE of μ and σ, set partial derivatives to zero.

$$\frac{\partial}{\partial \mu_{ki}} - \sum_{n=1}^{N} \log\left(p(x^{(n)}, y^{(n)})\right)$$

$$= \sum_{n=1}^{N} \sum_{i=1}^{d} \frac{1}{\sigma_i^2} \left(\mu_{ki} - x_i^{(n)}\right) = \sum_{i=1}^{d} \frac{1}{\sigma_i^2} \left[N\mu_{ki} - \sum_{n=1}^{N} x_i^{(n)} \right]$$

set $= 0$

$$N\mu_{ki} = \sum_{n=1}^{N} x_i^{(n)} \Rightarrow \mu_{ki} = \frac{1}{N} \sum_{n=1}^{N} x_i^{(n)} \quad \text{for } i \in \{1, 2, \dots, d\}$$

$$\frac{\partial}{\partial \sigma_i^2} - \sum_{n=1}^{N} \log\left(p(x^{(n)}, y^{(n)})\right)$$

$$= \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{d} \frac{1}{\sigma_i^2} - \sum_{n=1}^{N} \sum_{i=1}^{d} \frac{1}{2\sigma_i^4} \left(x_i^{(n)} - \mu_{ki}\right)^2$$

$$= \sum_{i=1}^{d} \left[\frac{N}{2\sigma_i^2} - \frac{1}{2\sigma_i^4} \sum_{n=1}^{N} \left(x_i^{(n)} - \mu_{ki}\right)^2 \right.$$

set $= 0$

$$N\sigma_i^2 = \sum_{n=1}^{N} \left(x_i^{(n)} - \mu_{ki}\right)^2 \Rightarrow \sigma_i^2 = \frac{1}{N} \sum_{n=1}^{N} \left(x_i^{(n)} - \mu_{ki}\right)^2$$

$$= \mathrm{Var}(X_i) \quad \text{for } i \in \{1, 2, \dots, d\}$$

Note: Again, the summation $\sum_{i=1}^{d}(\dots)$ is kept for clarity

Part 2.0

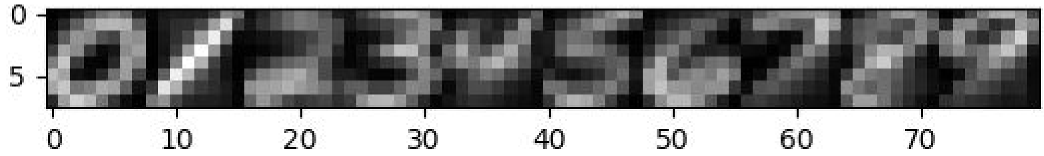

Figure 1: Means of each digit class in [0,9]

Part 2.1

1.

K	Train Classification Accuracy	Test Classification Accuracy
1	1.0	0.96875
15	0.9637142857142857	0.961

2. When there is a tie between two or more classes, the class (from the ones in conflict) that has the closest point is chosen.

3.

K	Cross Validation Accuracy
1	0.9644285714285715
2	0.9644285714285715
3	0.9651428571428571
4	0.9655714285714284
5	0.9634285714285715
6	0.9645714285714286
7	0.9607142857142856
8	0.9615714285714286
9	0.9579999999999999
10	0.9568571428571427
11	0.9555714285714286
12	0.9549999999999998
13	0.9531428571428571
14	0.9542857142857141
15	0.9497142857142858

The best (highest cross validation accuracy) value for K is 4, with accuracies:

Average Cross Validation Accuracy	0.9655714285714284
Train Set Classification Accuracy	0.9864285714285714
Test Set Classification Accuracy	0.97275

Part 2.2

1.

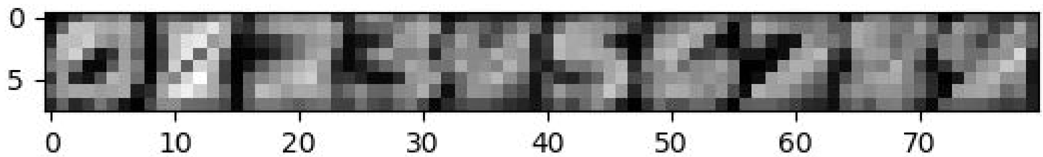

Figure 2: Diagonal elements of each covariance matrix for classes in [0,9]

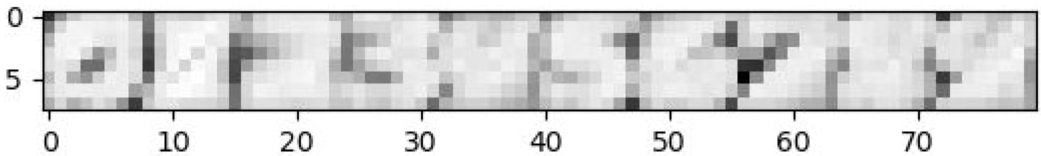

Figure 3: Log of diagonal elements of each covariance matrix for classes in [0,9]

2. Average conditional log-likelihood for Train set: -0.124624436669
   Average conditional log-likelihood for Test set: -0.196673203255

3. Classification accuracy on Train set: 0.9814285714285714
   Classification accuracy on Test set: 0.97275

Part 2.3

3.

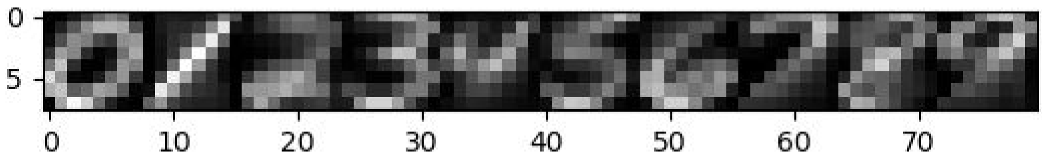

Figure 4: Eta matrix for each class in [0,9]

4.

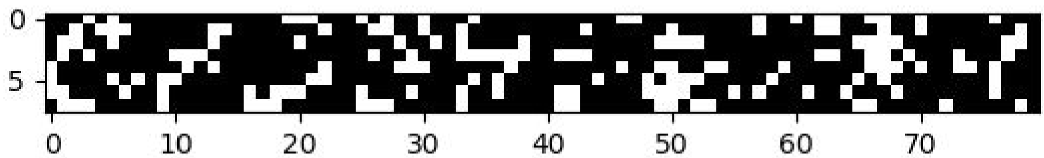

Figure 5: Generated digit images for each class in [0,9]

5. Average conditional log-likelihood for Train set: -0.9437538618
   Average conditional log-likelihood for Test set: -0.987270433725

6. Classification accuracy on Train set: 0.7741428571428571
   Classification accuracy on Test set: 0.76425

Part 2.4

In terms of accuracy, kNN and the Conditional Gaussian Classifier performed identically, with ~97% accurate classification of the test set, while Naive Bayes performed poorly in comparison, with only ~76% accuracy. This is in line with my expectations, as the assumption under Naive Bayes (independence between features) doesn't apply well to our data.

In terms of computation speed (without accurate time measurements, since none of the algorithms have been optimized), Naive Bayes was significantly faster than kNN and Conditional Gaussian. This is expected, as the assumption under Naive Bayes is a trade-off between accuracy and computation time (corroborated by the lower accuracy above). kNN and the Conditional Gaussian Classifier performed similarly in computation-time for our train and test sets. However, Conditional Gaussian Classifiers take a long time to train, but are quick to classify, while kNN needs no training, but classification is slow. As such, the choice between kNN and Conditional Gaussian Classifiers may be dictated by the size of the train and test sets.