ECE175 HW4 Report

Di Guan (A91041815)

Problem 1.

Consider the binomial random variable X with parameters *n* and *p*, i.e.

$$P_X(x) = \begin{bmatrix} n \\ x \end{bmatrix} p^x(1-p)^{n-x}$$

Assuming that the parameter n is known, and given a sample D={x1,.....,xN}, what is the maximum likelihood estimate of the parameter p?

Solution:

Given the probability distribution and the sample data D, we have the likelihood function shown below:

$$\begin{aligned} L(\theta; D) &= L(n, p; x_1, x_2, \ldots, x_N) = P_X(x_1)P_X(x_2)\ldots P_X(x_N) \\ &= \begin{bmatrix} n \\ x_1 \end{bmatrix} p^{x_1}(1-p)^{n-x_1} \quad \begin{bmatrix} n \\ x_2 \end{bmatrix} p^{x_2}(1-p)^{n-x_2} \quad \ldots\ldots \quad \begin{bmatrix} n \\ x_N \end{bmatrix} p^{x_N}(1-p)^{n-x_N} \\ &= \begin{bmatrix} n \\ x_1 \end{bmatrix}\begin{bmatrix} n \\ x_2 \end{bmatrix}\ldots\begin{bmatrix} n \\ x_N \end{bmatrix} p^{x_1+x_2+\ldots+x_N}(1-p)^{nN-(x_1+x_2+\ldots+x_N)} \\ &= \frac{n!}{(n-x_1)!x_1!}\frac{n!}{(n-x_2)!x_2!}\ldots\frac{n!}{(n-x_N)!x_N!}\, p^{\sum_{i=1}^N x_i}(1-p)^{nN-\sum_{i=1}^N x_i} \end{aligned}$$

To find the MLE of parameter *p*, we need to take the log of the likelihood function for computational convenience, then set the first derivative of log likelihood function to zero and check whether the second derivative is less than zero or not; if yes, then we obtain the MLE of parameter p; otherwise, we don't.

- obtain log-likelihood function

$$ln(L) = ln(\frac{n!}{(n-x_1)!x_1!}\ldots\frac{n!}{(n-x_N)!x_N!}) + ln(p)\sum_{i=1}^N x_i + ln(1-p)(nN - \sum_{i=1}^N x_i)$$

- set the first derivative to zero

$$\frac{d\,ln(L)}{dp} = \frac{d}{dp}[ln(p)\sum_{i=1}^N x_i + ln(1-p)(nN - \sum_{i=1}^N x_i)] = 0$$

$$\frac{d\,ln(L)}{dp} = \frac{1}{p}\sum_{i=1}^N x_i - \frac{1}{1-p}(nN - \sum_{i=1}^N x_i) = 0$$

$$(1-p)\sum_{i=1}^N x_i - p(nN - \sum_{i=1}^N x_i)) = 0$$

$$\sum_{i=1}^N x_i - p\sum_{i=1}^N x_i - pnN + p\sum_{i=1}^N x_i = 0$$

$$\sum_{i=1}^N x_i - pnN = 0$$

$$p = \frac{1}{nN}\sum_{i=1}^N x_i$$

- check the second derivative is less than zero

$$\begin{aligned}\frac{d\,ln(L)^2}{d^2p} &= \frac{d}{dp}[\frac{1}{p}\sum_{i=1}^{N}x_i - \frac{1}{1-p}(nN-\sum_{i=1}^{N}x_i)] \\ &= -\frac{1}{p^2}\sum_{i=1}^{N}x_i - \frac{1}{(1-p)^2}(nN-\sum_{i=1}^{N}x_i) \\ &= -\frac{1}{p^2}\sum_{i=1}^{N}x_i - \frac{1}{(1-p)^2}(nN-\sum_{i=1}^{N}x_i) < 0 \\ & x_1+\ldots+x_N < n+\ldots+n, since\ x_1,\ldots,x_N < n \\ & that\ is\ \sum_{i=1}^{N}x_i < nN \\ & every\ term\ is\ less\ than\ zero, thus\ \frac{d\,ln(L)^2}{d^2p} < 0\end{aligned}$$

$$Hence,\ the\ likelihood\ function\ achieves\ the\ maximum\ value\ when\ p = \frac{1}{nN}\sum_{i=1}^{N}x_i$$

Problem 2.

Consider a d-dimensional random variable Y

$$Y = A\mathrm{x} + \mathrm{n}$$

where **x** is an unknown, but deterministic, d-dimensional vector **n** is a d-dimensional Gaussian random vector of mean 0 and covariance Σ and **A** is a n*n positive definite matrix.

(a). what is the joint density for the random variable Y?

Solution:

Ax is constant since A is a positive definite matrix and x is an unknown but deterministic vector; plus n is a d-dimensional Gaussian random variable of mean 0 and covariance Σ, thus **Y is also a d-dimensional Gaussian random variable of mean Ax and covariance Σ**

(b) show that, given an observation y, the maximum likelihood estimates of parameter of x is

$$\hat{\mathrm{x}} = (A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}\mathrm{y}$$

Solution:

Given a sample, to obtain ML estimate, we need to solve

$$\hat{\theta}_{ML} = arg\ max\ P_D(D;\theta)$$
$$while\ P_D(D;\theta) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}e^{-\frac{1}{2}(y-\mu)^T\Sigma^{-1}}(y-\mu)$$

that is,

$$\begin{aligned}\hat{\theta}_{ML} &= arg\ max_x\ log[\frac{1}{\sqrt{(2\pi)^d|\Sigma|}}e^{-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)}]\\ &= arg\ max_x\ \ -\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)-\frac{1}{2}log(2\pi)^d|\Sigma|\\ &= arg\ min_x\ (y-\mu)^T\Sigma^{-1}(y-\mu)\\ &= arg\ min_x\ (y-Ax)^T\Sigma^{-1}(y-Ax)\\ &= arg\ min_x\ (y-Ax)^TQ^TQ(y-Ax)\quad Let\ \Sigma^{-1}=Q^TQ\\ &= arg\ min_x\ (Qy-QAx)^T(Qy-QAx)\\ &= arg\ min_x\ ||Qy-QAx||^2\end{aligned}$$

So, we are back to solve the least-square problem in linear algebra, which is to find the projection of Qy onto the range of QA, R(QA).

Suppose that vector w is the projection of Qy onto R(QA), where w=QAx then Qy-w should be orthogonal to R(QA). That is,

$$\begin{gathered}(QA)^T(Qy-\hat{w})=0\\ (QA)^TQy-(QA)^T\hat{w}=0\\ A^TQ^TQy=A^TQ^T\hat{w}\\ A^TQ^TQy=A^TQ^TQA\hat{x}\end{gathered}$$

$$\begin{gathered}then\ \hat{x}=(A^TQ^TQA)^{-1}A^TQ^TQy,\ if A^TQ^TQA\ or\ QA\ is\ invertible\\ since\ \Sigma\ is\ positive\ definite\ matrix, then\ there\ exists\ matrix\ P\\ s.t.\Sigma=P^TP, where\ P\ has\ independent\ columns,\ \ same\ thing\ for\ \Sigma^{-1}=Q^TQ\\ where\ Q\ have\ independent\ columns, thus\ rank(Q)=n;\\ Additionally, A\ is\ a\ positive\ definite\ matrix, then\ rank(A)=n;\\ due\ to\ the\ rule\ of\ product\ rank, rank(QA)<=min[rank(A),rank(Q)]=min(n,n)=n;\\ and\ rank(QA)>=rank(A)+rank(Q)-n=n+n-n=n;\\ that\ is\ n<=rank(QA)<=n,\ \ therefore, rank(QA)=n, which\ implies\ that\ QA\ is\ invertible\end{gathered}$$

$$\begin{gathered}Thus\ \hat{x}=(A^TQ^TQA)^{-1}A^TQ^TQy\\ which\ is\ \hat{x}=(A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}y\end{gathered}$$

(c). what is the least squares problem whose solution is equivalent to (b)? Assume Σ is diagonal. What is the role of this matrix, i.e. how does it change the canonical least squares problem?

Solution:

The least square problem whose solution is equivalent to (b) is

$$\begin{aligned}\hat{x}_{ML} &= arg\ min_x\ (y-Ax)^T\Sigma^{-1}(y-Ax)\\ &= arg\ min_x\ (y-Ax)^TQ^TQ(y-Ax)\ \ where\ Q^TQ=\Sigma^{-1}\\ &= arg\ min_x\ ||Qy-QAx||^2\end{aligned}$$

If Σ is diagonal matrix, then

$$\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots . \sigma_n^2,)$$
$$\Sigma^{-1} = diag(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \ldots . \frac{1}{\sigma_n^2},)$$
$$with\ \hat{x}_{ML} = arg\ min_x\ (y - Ax)^T \Sigma^{-1} (y - Ax)$$
$$= arg\ min_x \sum_{i=1}^{n} (y - Ax)_i \frac{1}{\sigma_i^2} (y - Ax)_i$$
$$= arg\ min_x \sum_{i=1}^{n} \frac{(y - Ax)_i^2}{\sigma_i^2}$$

Problem 3.

The digital scan of the digits in Fig.1 is often noisy and may vary a lot in terms of overall intensity. We want to classify these corrupted digits. We shall continue with the training data used in the previous experiment, but with a new set of test data testImagesNew, which is corrupted by noise and re-scaled in amplitude. Using the NN approach, we will find the training image that is nearest to the test image, but instead of the Euclidean distance, we will use the distance to the MLE of the uncorrupted test pattern. We assume the test image Y is of the form

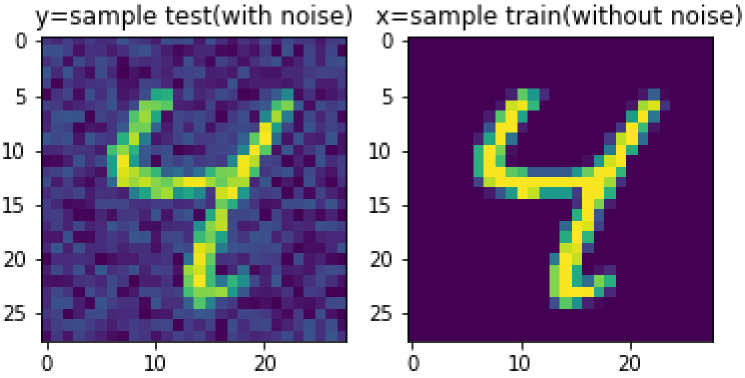
$$Y = a * \mathrm{x} + N$$

the result of corrupting the training image x through 1) amplitude re-scaling by the scalar a, and 2) the addition of independent zero mean Gaussian noise with variance v, i.e. N~G(0,vI). The scale factor a is the MLE given the observation y(the test image) and the known training image x. The Euclidean distance between the normalized test image and the training image then serves as the metric for the NN classifier.

$$a* = arg\ max_a P_Y(Y|X;\theta) = arg\ max_a P_Y(Y|X;a,v)$$

1. using the two sample images sampletest.png and sampletrain.png. Calculate the MLE of the scale parameter a.

Figure_1



The terms of a*x is fixed since scalar a and vector x, training image, is constant.

Y, testing image, is Gaussian distribution with mean a*x and covariance vI since Y=ax+N and N~G(0,vI).

$$\begin{aligned}\hat{a}_{ML} &= arg\ min_a\ (\mathrm{y}-\mathrm{x}a)^T\Sigma^{-1}(\mathrm{y}-\mathrm{x}a)\ where\ \Sigma^{-1}=diag(\frac{1}{v},\frac{1}{v},..,\frac{1}{v})\\ &= arg\ min_a\sum_{i=1}^{n}(\mathrm{y}-\mathrm{x}a)_i\frac{1}{v}(\mathrm{y}-\mathrm{x}a)_i\\ &= arg\ min_a\sum_{i=1}^{n}\frac{(\mathrm{y}-\mathrm{x}a)_i^2}{v}\\ &= arg\ min_a\ \frac{1}{v}\sum_{i=1}^{n}(\mathrm{y}-\mathrm{x}a)_i^2\\ &= arg\ min_a\ ||\mathrm{y}-\mathrm{x}a||^2\end{aligned}$$

$$with\ the\ gradient\ is\ equal\ to\ 0, we\ have\ \mathrm{x}^T(\mathrm{y}-\mathrm{x}a)=0, then\ \mathrm{x}^T\mathrm{y}=\mathrm{x}^T\mathrm{x}a,$$

$$thus\ we\ have\ a=\frac{\mathrm{x}^T\mathrm{y}}{\mathrm{x}^T\mathrm{x}}=0.6796\ where\ \mathrm{x}\ is\ the\ training\ image\ and\ y\ is\ testing\ image$$

2. Now for the new test set, testImagesNew, perform the task of classification using the least square distance metric. Compute and plot the error rates for each class, and the total error rate.

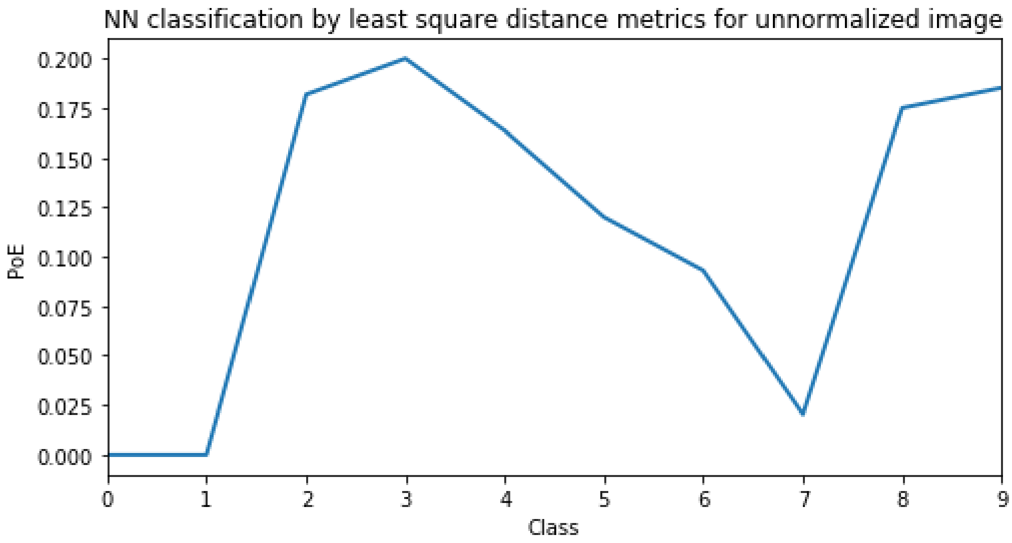
- In this case, our least square distance metric is a problem of the type

$$\begin{gathered}min\ ||\mathrm{y}-a\mathrm{x}||^2\\ where\ a=0.6796\ obtained\ from\ part\ (1)\\ that\ is\ min||\mathrm{y}-0.6796\mathrm{x}||^2\end{gathered}$$

- Basically, what we do is to calculate the least square distance described above in terms of the new testing image(with noise and amplification) with each training image from 5000 training set, and then return the minimum distance among those 5000 distances and classify this testing image to one of the classes that is corresponding to the training label based on the minimum distance obtained.

- The table and plot below demonstrate the error rates for each class

```
+---------+-----------------+----------+--------------+
|   Class |   Total Samples |   Errors |   Error Rate |
+=========+=================+==========+==============+
|       0 |              42 |        0 |        0     |
+---------+-----------------+----------+--------------+
|       1 |              67 |        0 |        0     |
+---------+-----------------+----------+--------------+
|       2 |              55 |       10 |        0.182 |
+---------+-----------------+----------+--------------+
|       3 |              45 |        9 |        0.2   |
+---------+-----------------+----------+--------------+
|       4 |              55 |        9 |        0.164 |
+---------+-----------------+----------+--------------+
|       5 |              50 |        6 |        0.12  |
+---------+-----------------+----------+--------------+
|       6 |              43 |        4 |        0.093 |
+---------+-----------------+----------+--------------+
|       7 |              49 |        1 |        0.02  |
+---------+-----------------+----------+--------------+
|       8 |              40 |        7 |        0.175 |
+---------+-----------------+----------+--------------+
|       9 |              54 |       10 |        0.185 |
+---------+-----------------+----------+--------------+
```



- The total error rate, **P(error)=0.112**

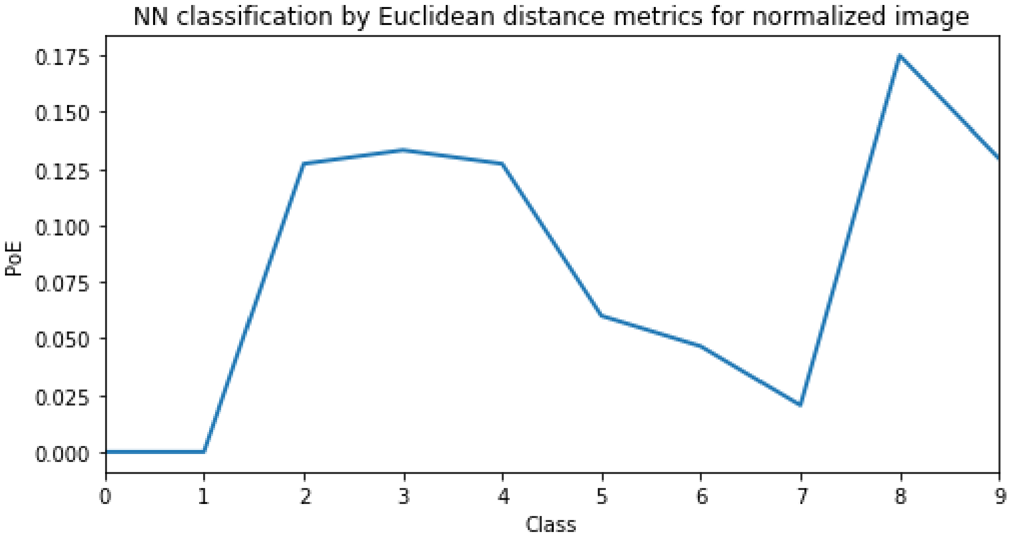
3. Perform a NN classification on the new test set, using the algorithm of Euclidean distance metric. Compare the results with the NN classification performed in part 2.

- In this case, after we normalized the given testing image and training image, the MLE of parameter 'a' is different from part 1 where the images are unnormalized. Here, **a=0.9218**
- To perform NN classification, we use the metrics of Euclidean distance between the normalized testing image and training image. Same procedure as part 2, compare testing image with each training image from 5000 training set. While the Euclidean distance is defined below:

$$\begin{gathered} min \; ||\mathrm{y} - \mathrm{x}||^2 \\ where \; \mathrm{y} \; and \; \mathrm{x} \; are \; normalized \end{gathered}$$

- The table and plot below demonstrate the error rates for each class

```
+---------+-----------------+----------+--------------+
|   Class |   Total Samples |   Errors |   Error Rate |
+=========+=================+==========+==============+
|       0 |              42 |        0 |        0     |
+---------+-----------------+----------+--------------+
|       1 |              67 |        0 |        0     |
+---------+-----------------+----------+--------------+
|       2 |              55 |        7 |        0.127 |
+---------+-----------------+----------+--------------+
|       3 |              45 |        6 |        0.133 |
+---------+-----------------+----------+--------------+
|       4 |              55 |        7 |        0.127 |
+---------+-----------------+----------+--------------+
|       5 |              50 |        3 |        0.06  |
+---------+-----------------+----------+--------------+
|       6 |              43 |        2 |        0.047 |
+---------+-----------------+----------+--------------+
|       7 |              49 |        1 |        0.02  |
+---------+-----------------+----------+--------------+
|       8 |              40 |        7 |        0.175 |
+---------+-----------------+----------+--------------+
|       9 |              54 |        7 |        0.13  |
+---------+-----------------+----------+--------------+
```



- The total error rate, **P(error)=0.08**