FUNDAMENTALS OF STATISTICAL LEARNING
PROJECT1 – REPORT
Dharmit Prajapati

Task 1:

We have a 28x28 array of pixels for each image
For python code,
We have flattened each image from 28x28 to 1x784
For each image I,
We have found out the mean(mi) and standard deviation from the pixels
Each image is now represented as [mi,si]

Then we calculate,

M1 = mean of all means(mi) of all images(i)
M2 = mean of all Standard deviations(si) of all images(i)
S1 = SD of all means(mi) of all images(i)
S2 = SD of all SDs(si) of all images(i)

Finally each image is now represented by normalized feature vector Yi

Yi = [y1i, y2i] = [(mi-M1)/S1, (si-M2)/S2]

Thus, we have done feature extraction and normalization which are one of the initial stages of Pattern Recognition.

Task 2:

Task2 represents the more general case of estimating θ where μ and ∑ are both unknowns (for Normal/Gaussian distribution)

Let θ1 and θ2 be the unknown parameters that constitute components of parameter vector θ

Considering it for univariate case with $\theta_1 = \mu$ & $\theta_2 = \sigma^2$

we know,

$$p(x_k|\theta) = \frac{1}{\sqrt{2\pi\theta_2}} \exp\left[\frac{-1}{2\theta_2}(x_k-\theta_1)^2\right]$$

$$\ln p(x_k|\theta) = \frac{-1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k-\theta_1)^2$$

$$\nabla_\theta l = \nabla_\theta \ln(p(x_k|\theta)) = \begin{bmatrix} \frac{1}{\theta_2}(x_k-\theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k-\theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

(Taking partial derivative w.r.t θ_1 & θ_2)

we know,

$$\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta \ln(p(x_k|\theta)) = 0 \quad \text{(for estimating } \hat{\theta}\text{)}$$

$$\therefore \sum_{k=1}^{n} \frac{1}{\theta_2}(x_k-\theta_1) = 0$$

$$\sum_{k=1}^{n} (x_k-\theta_1) = 0$$

$$\sum_{k=1}^{n} x_k = \theta_1(n)$$

$$\therefore \boxed{\theta_1 = \frac{1}{n}\sum_{k=1}^{n} x_k} \quad \boxed{\text{ie } \hat{\theta}_1 = \mu}$$

also

$$\sum_{k=1}^{n} \frac{-1}{2\theta_2} + \sum_{k=1}^{n} \frac{(x_k-\theta_1)^2}{2\theta_2^2} = 0$$

$$\boxed{\theta_2 = \frac{1}{n}\sum_{k=1}^{n}(x_k-\theta_1)^2}$$

$$\text{ie } \boxed{\hat{\theta}_2 = \sigma_2}$$

For a multivariate case,
We can again prove that the parameters are mean and covariance matrix.
The parameters for 2-D normal distribution will be sample mean and covariance matrix respectively. (same has been calculated in the python code)

Task 3:

In order to obtain probability of error we use:

$P(error) = \int P(error|x)p(x)\, dx$ ………(1)

We choose class 3 over class 7 (let w1=class 3 & w2 = class7)

$P(w1|x) > P(w2|x)$

Ie $p(x|w1)P(w1) > p(x|w2)P(w2)$

We know,

$P(error|x) = \min(P(w1|x), P(w2|x))$

$= \min(p(x|w1)P(w1)/p(x),\ p(x|w2)P(w2)/p(x))$ ….(2)

From (1) and (2)

$P(error) = \int \min(p(x|w1)P(w1),\ p(x|w2)P(w2))\, dx$ ……(3)

In the python code,

We have summed up all the error by using (3) and divided by the number of samples.

OUTPUT :

```
[Command: python -u 'D:\Dharmit\Masters\Fundamentals of Statistical Learning\Assignment_hw_project\Assignment1\
FSL_project1.py']
[M1 M2]: [32.50435107 76.44042397]
[S1 S2]: [ 9.34944945 10.50972539]
Mu3(Mean3) [0.37687996 0.31851855]
Mu7(Mean7) [-0.36900004 -0.31185886]
Sigma3 [[1.0491056  0.98717364]
 [0.98717364 0.96037982]]
Sigma7 [[0.67669136 0.74435619]
 [0.74435619 0.842203  ]]
Total error for training set for case1 is: 0.1584498396653004
Total error for training set for case2 is: 0.10538315677294711
Total error for test set for case1 is: 0.15370718669928896
Total error for test set for case2 is: 0.10223303694517855
[Finished in 6.244s]
```