Density Estimation and Classification of MNIST Data

Objective:

In this project, there are multiple objectives. They are,

- Parameter Estimation for MNIST dataset of digits 7 and 8 assuming them as a Normal distribution
- Extracting features from the data set aka mean and standard deviation for each digit.
- Performing Naive Bayes and Logistic Regression on the extracted features and compare corresponding accuracies.

Feature Extraction:

The data set given has training data and testing data. The input training data (trX) is of matrix 12116 X 784 and input testing data (tsX) is of matrix 2002 X 784. Here we have 784 features for each digit, from those we have to extract two features for each digit. They are the standard deviation of the 784 features and mean of 784 features. Calculated these two features for each digit and stored in a new matrix fX(12116 X 2). Similarly, features are extracted for the testing data set(tsX) and stored in a matrix fsX(2002 X 2). The formulae used for mean and standard deviation are given below.

$$Mean(\mu) = (\sum_{i=0}^{n-1} xi)/n \qquad \mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

Where $\overline{x} = mean(\mu)$

Parameter Estimation for Naive Bayes:

As given, assume that these two features are independent and each feature of the image represents a Normal Distribution. So in total, we have two 2-D Normal Distributions one for digit 7 and digit 8. From the matrix fX, created matrices fX7, fX8 which contain features of digit 7 and digit 8 respectively. Now we have a total of 4 Normal distributions two for digit 7 (for each feature) and two for digit 8 (for each feature). Now the goal is to find standard deviation and mean for each distribution using the formulae above. The calculated mean and standard deviation for each distribution are stored in fX7mean(1 X 2), fX7std (1 X 2), fX8mean (1 X 2), and fX8std(1 X 2).

The parameters obtained are as follows

	Mean	Standard Deviation
Digit 7	For Feature 1: 0.28755657 For Feature 2: 0.1145277	For Feature 1: 0.03820108 For Feature 2: 0.0306324
Digit 8	For Feature 1: 0.32047584 For Feature 2: 0.15015598	For Feature 1: 0.03996007 For Feature 2: 0.03863249

Implementing Naive Bayes:

As assumed that features are independent we can use the Naive Bayes technique for Classification. According to Naive Bayes Algorithm

- For classifying as digit 7 : $P(\,(Y = Digit\ 7) \mid X\,) = P(X \mid (Y = Digit\ 7)) * P(Y = Digit\ 7) / P(X)$
- For classifying as digit 8 : $P(\,(Y = Digit\ 8) \mid X\,) = P(X \mid (Y = Digit\ 8)) * P(Y = Digit\ 8) / P(X)$
- As all the features are independent,
- $P(X \mid (Y = Digit\ 7)) = P(X_1 \mid (Y = Digit\ 7)) * P(X_2 \mid (Y = Digit\ 7))$ where X_1 and X_2 are features of X
- $P(X_1 \mid (Y = Digit\ 7))$ is the probability of finding X_1 in the distribution X_1 where they are classified as digit 7. Which can be estimated by the Maximum Likelihood Equation for Normal Distribution which is given by
- $P(X_1 \mid (Y = Digit\ 7)) = (1/\sigma\sqrt{2\pi}\,)^2 * e^{-(X_i-\mu)^2/2\sigma^2}$, where μ is the mean of all the values in feature 1 in fX7(features where the digit is 7) and σ is the standard deviation of all the values in feature 2 in fX7. This is implemented in the code defining a function ***probabilityND***. Similarly, remaining probabilities for digit 8, feature X_2 and digit 7 and digit 8 can be calculated.
- $P(Y = Digit\ 7) = 6265/12116$ as there are 6265 7 digits out of all, this implemented in the code and declared as **p7 which is 0.517**
- $\therefore P(Y = Digit\ 8) = 1 - P(Y = Digit\ 7)$, i.e p8= 0.4829151535160119

Now we have all the essential terms for calculating the probability of a given digit being classified as 7 and being classified as 8. If the probability of given digit has a higher probability being digit 7 then we classify it as 7 (or 0) and if the digit has a higher probability being digit 8 we classify it as digit8 (or 1).

In the code, pr_7 calculates the probability of all the numbers in the testing data being digit 7 and similarly, pr_8 calculates the probability of all the numbers in the testing data being digit 8. Now we compare pr_7 with pr_8 element-wise. If any element of pr_7 is greater than the corresponding element in pr_8 then we classify it as 0(digit 7) similarly if any element of pr_7 is less than the corresponding element in pr_8 we classify it as 1(digit 8). This is implemented using a for-loop.

Accuracy of Naive Bayes (Result):

Accuracy of a classification problem is defined as the ratio of correct classification with total classification. To find this pr_7 is compared with the tsY element-wise(given classifications of testing data) using a for-loop in the code. **The overall accuracy** of **69.53** is obtained. Accuracy for Classifying **Digit 7** is **75.97** and accuracy of classifying **Digit 8** is **62.73**

Implementing Logistic Regression:

As already the features have been extracted, the features can be used to implement logistic regression. To implement logistic regression we use sigmoid/logistic function on the linear regression hypothesis to classify the inputs either 0(digit 7) or 1 (digit 8). The **linear regression** hypothesis is given by

$Y_{prediction} = W^T X$ where W is weight/coefficient matrix and X is the input feature matrix. Upon applying the sigmoid function to this we get the final hypothesis for the logistic regression. Where the sigmoid function is given by,

$\sigma(t) = 1/1 + e^{-t}$. Now applying sigmoid to out linear regression hypothesis i.e,

Logistic Regression $Hypothesis(h) = \sigma(W^T X) = 1/1 + e^{-W^T X}$

The Sigmoid function is implemented in the code defining the function *sigmoid* and to obtain $W^T X$ function *hyp* is defined.

So, now as the final Logistic Regression is obtained, the logistic regression can be implemented. But to implement the Logistic Regression the value of W is necessary. The value of W is such that it maximizes the Conditional Log-Likelihood function which is given by

$$W^* = argmax_W l(W)$$
$$= argmax_W \sum_{i=1}^{n} [y^{(i)} W^T X^{(i)} - log(1 + exp(W^T X^{(i)}))]$$

But solving the above equation does not produce a closed-form solution so gradient ascent is implemented such that we find the value of W which maximizes the above equation

Gradient Ascent:

The gradient Ascent equation is used to find W which is given by,

$$W^{(k+1)} = W^k + \eta \nabla_{W^k} l(W)$$

Where $\eta > 0$ is a constant called the learning rate. Which is an arbitrary value which generally taken in the order of 0.01 and $\nabla_{W^k} l(W)$ is given by

$$\nabla_{W^k} l(W) = \sum_{i=1}^{n} [Y^{(i)} X^{(i)} - e^{W^T X} X^{(i)} / 1 + e^{W^T X}]$$
$$= \sum_{i=1}^{n} [X^{(i)} (Y^{(i)} - e^{W^T X} / 1 + e^{W^T X})]$$

$$= \sum_{i=1}^{n} [X^{(i)}(Y^{(i)} - \sigma(W^T X))]$$

$$\nabla_{W^k} l(W) = \sum_{i=1}^{n} [X^{(i)}(Y^{(i)} - h^{(i)})]$$

Now substituting the above equation in the gradient ascent equation completes the gradient ascent equation and the gradient ascent is used for about iterations of 100000 until we get the best accuracy. To implement the gradient ascent in the code a function named *gradient* and a for-loop is used for iterations of 1000000 and the learning rate is taken to be 0.001. After completing all the iterations the final W is used in our logistic regression hypothesis with a threshold of 0.5, i,e if the output of the hypothesis is greater than or equal to 0.5 it is taken as 1(digit 8) else 0 (digit 7). This is implemented in the code using the function *predict* which returns the final prediction containing all the classifications.

The final value of W (parameters for Logistic Regression) obtained **=23.12908283, -181.23578248, 247.49770926**

Accuracy of Logistic Regression (Result) :

Now the final value W is obtained we used the *predict* function where we input the testing data with added feature one. The predict function is fed with the testing data and final value of W which returns the predictions based upon the threshold value that is stored in the variable *re*. Accuracy of a classification problem is defined as the ratio of correct classification with total classification. So the final predictions *re* is compared with testing classification i.e tsY. **The overall accuracy** of **81.66** is obtained, for **Digit 7 accuracy** is **78.59** and **Digit 8 accuracy** is **84.90,** which is a huge development from the Naive Bayes classification. But the time taken to compute Naive Bayes is much lesser than the time taken for logistic regression.