Density Estimation and Classification

CSE 575 – Statistical Machine Learning

Motthilal Baskaran
1215168292
mbaskar2@asu.edu
02/15/2020
Spring 2020

1. Introduction:

The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset [1]. The set of images in the MNIST database is a combination of two of NIST's databases: Special Database 1 and Special Database 3. Special Database 1 and Special Database 3 consist of digits written by high school students and employees of the United States Census Bureau, respectively. We use only images for digit "7" and digit "8" in this question.

2. Feature Extraction:

The extracted dataset contained training and testing samples for the digit '7' and '8'.

- Number of samples in the training set: "7": 6265; "8": 5851.
- Number of samples in the testing set: "7": 1028; "8": 974

Mean and Standard Deviation are the features extracted for each image (array) in the training dataset.
Estimated values for both classes are as below:
For class 7:

```
In [46]: print(mean7,std7)
0.1145276977510873 0.030634849714227527
```

For class 8:

```
In [47]: print(mean8,std8)
0.15015598189369683 0.03863579015494639
```

3. Naïve Bayes Classification:

A Gaussian distribution can be summarized using the mean and the standard deviation. Therefore, we can estimate the probability of a given value called a Gaussian Probability Distribution Function (or Gaussian PDF) and can be calculated as below:

f(x) = (1 / sqrt(2 * PI) * sigma) * exp(-((x-mean)^2 / (2 * sigma^2)))

Calculation of the Prior:

Prior 7 & 8:

```
In [48]: print(prior_7,prior_8)
0.5134865134865135 0.4865134865134865
```

It has been assumed that the features are independent, the calculation of final probability for each class for the given test dataset is as below:

Probability of Mean * Probability of Standard Deviation * prior (respective class)

The class with higher probability has been chosen as the predicted class.

Accuracy for class 7:

```
In [51]: print( correct7 / (correct7 + wrong7))
0.7558365758754864
```

Accuracy for class 8:

```
In [52]: print(correct8 / (correct8 + wrong8))
0.6273100616016427
```

Final Accuracy:

```
In [53]: print(correct / (correct + wrong))
0.6933066933066933
```

4. **Logistic Regression Classification:**
   We need to calculate the log-likelihood (ll) over the training data and can be represented mathematically as below:

$$ll = \sum_{i=1}^{N} y_i \beta^T x_i - log(1 + e^{\beta^T x_i})$$

   y – target class
   x – Data point
   β – weights vector

   Getting the derivative of the above equation and with some transformation, we calculate gradient as

$$\bigtriangledown ll = X^T(Y - Predictions)$$

Also, we use the sigmoid function as below:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Also, decaying learning rate (decreasing the learning rate over iterations) is used.
No. of Iterations: 10000
Learning rate: 3e-3

Final Accuracy: 0.983016983016983

Also find the relevant python script files attached with this submission that contains the original classification scripts and they are as below:

1. Naive_Bayes.py
2. Logistic_regression.py

References:

1. https://en.wikipedia.org/wiki/MNIST_database