

Density Estimation and Classification

Project 1
Yongbaek Cho

1. Introduction

The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. Two images of digits which are '7' and '8'. The datasets for the digit '7' and '8' are divided into training and testing and there are 6265 training data and 1028 testing data in the case of digit '7' and 5851 training data and 974 testing data for in the case of digit '8'. I try to obtain the accuracy of two typical classification algorithms, Naïve Bayes and Logistic Regression models, which have also been evaluated as performing well.

2. Naïve Bayes Classification

The Naïve Bayes is a conditional probability model and generative model. Also, it uses the Bayes theorem to find conditional probability of given x, y is a label that the algorithm predicts, and x is a dataset of independent features.

Bayes theorem:

$$P(Y | X) = \frac{P(Y)P(X | Y)}{P(X)}$$

As Bayes theorem terminology:

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{Evidence}}$$

I can plug in the Bayes Theorem with the features of digit 7 and 8 and two possible values of y which can be labels of 7 and 8. In Naïve Byes, features should be independent. Therefore, “*Two features, mean and standard deviation, should be independent*” is crucial assumption in Naïve Bayes.

Now, we can modify the Bayes Theorem by plugging in features.

Here are the formulas:

$$\text{Digit 7: } P(Y = 7/\text{mean}, \text{std}) = P(Y = 7) * P(\text{mean}/Y) * P(\text{std}|Y)$$

$$\text{Digit 8: } P(Y = 8/\text{mean}, \text{std}) = P(Y = 8) * P(\text{mean}/Y) * P(\text{std}|Y)$$

Prior probabilities are $P(Y = 7)$ and $P(Y = 8)$. These two prior probabilities can be calculated using the number of training data.

Python code:

#prior probability

priorprob_7 = len(train7)/(len(train7) + len(train8))

priorprob_8 = len(train8)/(len(train7) + len(train8))

$P(Y = 7) = 6265 / (6265 + 5851) = 0.51708$

$P(Y = 8) = 5851 / (6265 + 5851) = 0.48291$

The function states that:



$$p(\mu) = N(\mu_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right)$$

$$P(\text{mean}|Y=7) = \frac{1}{\sqrt{2\pi\sigma_{\text{mean}}^2}} \times \exp\left(-\frac{(\text{mean}_{\text{test}} - \mu_{\text{mean}})^2}{2(\sigma_{\text{mean}})^2}\right)$$

$$P(\text{std}|Y=7) = \frac{1}{\sqrt{2\pi\sigma_{\text{std}}^2}} \times \exp\left(-\frac{(\text{std}_{\text{test}} - \mu_{\text{std}})^2}{2(\sigma_{\text{std}})^2}\right)$$

$$P(\text{mean}|Y=8) = \frac{1}{\sqrt{2\pi\sigma_{\text{mean}}^2}} \times \exp\left(-\frac{(\text{mean}_{\text{test}} - \mu_{\text{mean}})^2}{2(\sigma_{\text{mean}})^2}\right)$$

$$P(\text{std}|Y=8) = \frac{1}{\sqrt{2\pi\sigma_{\text{std}}^2}} \times \exp\left(-\frac{(\text{std}_{\text{test}} - \mu_{\text{std}})^2}{2(\sigma_{\text{std}})^2}\right)$$

Parameters	Estimated Value	
$\text{mean}_{\text{test}}$ (Mean for test dataset)		
std_{test} (Standard deviation for test data)	Digit '7' 	Digit '8' 
$\sqrt{\sigma_{\text{mean}}^2}$ (Standard deviation of mean in training data of digit 7)	0.03063240469648835	0.03863248837395887
$\sqrt{\sigma_{\text{std}}^2}$ (Standard deviation of mean in training data of digit 8)	0.038201083694320306	0.03996007437065856
μ_{mean} (Mean for training data)	0.11452769775108769	0.1501559818936975
μ_{std} (Standard deviation for training data)	0.28755656517748474	0.3204758364888714

Using the test data, the digit 7 and 8 are calculated by using normal distribution equation for obtaining each posterior probability. (Posterior probability can be calculated by using normal distribution equation) And then, multiply posterior probability and prior probability. To classify between the numbers 7 and 8, we have to compare the probabilities which are multiplied by the posterior probability and the prior probability. If the probability of digit 7 is higher than 8, the Naïve Bayes Algorithm classifies it as digit 7. Otherwise, it classifies it as digit 8.

2.1 Result of Naïve Bayes Classification

The value predicted by the algorithm compares the actual value of Y and value of Y in test data to calculate the accuracy.

- Naïve Bayes Classifier – Accuracy for digit 7 is: 77.14007782101167%
- Naïve Bayes Classifier – Accuracy for digit 8 is: 61.49897330595483%
- Naïve Bayes Classifier – The Overall Accuracy is: 69.53046953046953%

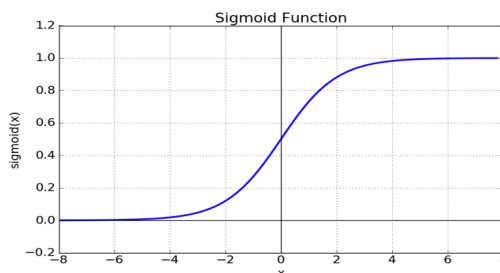
3. Logistic Regression

Logistic Regression uses regression to predict the probability of data with a category from zero to one and classify it into a more likely category depending on the probability. Also, it is a discriminative model and directly learn $P(Y|X)$. The logistic Regression which I used in this project uses the sigmoid function to obtain $W(\text{parameter})$ and calculate the decision boundary by using gradient ascent.

3.1 Sigmoid Function

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

```
def sigmoid(X, W): #sigmoid function
    return 1 / (1 + np.exp(-np.dot(X, W)))
```



According to training X data and weight value W, the sigmoid function returns a value between 0 and 1 as shown in the graph above. If the value is less than 0.5 it is considered as digit 7 and otherwise considered as digit 8.

3.2 Gradient Ascent

The gradient ascent is the first way to compute log likelihood and find the gradient. The difference from the gradient descent is to find the maximum value. The most important thing in the gradient ascent is to update parameter W(weight). This means the new weights which is updated based on the gradient ascent will increase the maximum likelihood.

Update the weights using this following equation:

$$\text{weights} += \text{learning rate} \times X^T(Y - \text{predictions})$$
$$(X^T(Y - \text{predictions}) = \log(\text{likelihood}))$$

Iterate n time steps until the log likelihood function is maximized. (Initially, the weight is zero)

The log likelihood function is maximized as the weight is updated through iteration. When the function obtains the weight to be maximized, the program can predict the value whether the digit 7 or 8.

Parameter W(weights)	Estimated Value
W1	56.88551675
W2	-24.73382512

(In the program, *learning rate* set as 0.001 and *iteration* is 3000)

3.3 Result of Logistic Regression

- Logistic Regression - Accuracy for digit 7: 79.6692607003891%
- Logistic Regression - Accuracy for digit 8: 67.96714579055441%
- Logistic Regression -The Overall Accuracy for both 7 and 8: 73.97602397602398%