Density Estimation and Classification

1. Introduction

The main objective of this project is to perform the parameter estimation and feature estimation of the dataset given and classify them into respective digits, here in our case it's 7 and 8 digits which are two different classes. We are using two different methods to classify the digits namely Logistic Regression and Naïve Bayes Classification. These are Discriminative and Generative respectively. Discriminative model calculates the posterior probability directly but whereas the Generative model calculates the posterior probability with the help of the conditional probability and the evidence provided. It is observed that when the training data size is large the performance of logistic regression exceeds the performance of Naïve Bayes Classifier.

2. Dataset:

In this project we are using MNSIT dataset. It is a large database` of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets. It contains 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. For our project we have taken only two class 0 and 1 and associated it with the digits 7 and 8.

Here are the details about the Training set and the Testing set

Training set:

1. "7" – 6265 samples.
2. "8" - 5851 samples.

Testing set:

1. "7" - 1028 samples.
2. "8" - 974 samples.

3. Feature Extraction:

In the given data set, we have extracted two main features for every image:

3.1 The mean of all the pixel values in an image(TrX1):

Since the distribution of all the pixel values in each image is assumed to be gaussian distribution, we find mean of all the pixel values after row flattening the (12116*784) matrix into single rows. Here is the formula used to calculate the average of all the pixels of a given image.

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

xi = The pixel values of a given image after row flattening the (12116*784) matrix .

3.2 The Standard deviation of all the pixel values in the image(TrX2)

On the similar lines as the first one, we assume the pixel values are normally distributed and we find the standard deviation value for all the pixels values in each image. Below is the formula we used to find the variance and the standard deviation. Variance (σ^2) and then square root of that value gives us the Standard Deviation (σ)

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}$$

4. Methodology:

As mentioned above the two approaches used are Logistic Regression and Naïve Bayes Classifier. Naïve Bayes Classifier assumes that all the features are independent of each other and hence the name. Logistic Regression directly calculates the posterior probabilities directly.

4.1.1 Naïve Bayes Classification:

Naïve Bayes is a Generative Model which calculates the posterior probability considering the fact that the features are independent and the calculates the posterior probabilities depending on the conditional probability and evidence provided. The features extracted in the given dataset are Mean and the Standard Deviation for the given set of the images. Later we classify the digits in to two different classes for supervised learning ; here 0- seven digit, 1- digit eight. We calculate the co-variance matrix for each digit and prior probability for each digit and compute what is the likelihood of the digit data to be in each of the classes. Once the probability of likelihood is calculated for each individual classes we classify the given input to a classes with higher probability .

Suppose 'y' denotes the label to be predicted and $\vec{x}$ is the feature vector containing the features x1 and x2, then probability that the given test-data is of label 'y' is given by:

$$P(y|x) = \frac{P(y) \cdot \prod_{i=1}^{2} P(x_i|y)}{P(x)}$$

Here since there are only 2 features mean and standard deviation, we multiply the conditional probabilities of' x1' given 'y' and 'x2' given 'y' with P(y) which is known as the 'prior probability'. Here, since P(x) is a constant value, we generally ignore the denominator. Also, since it is assumed that the images are drawn from a 2-D normal distribution, we find the mean (μ) and standard deviation (σ) of all values of x1 and x2 features.

Then, we get the conditional probability of xi using the below formula:

$$L = PDF(x_i, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\Pi}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

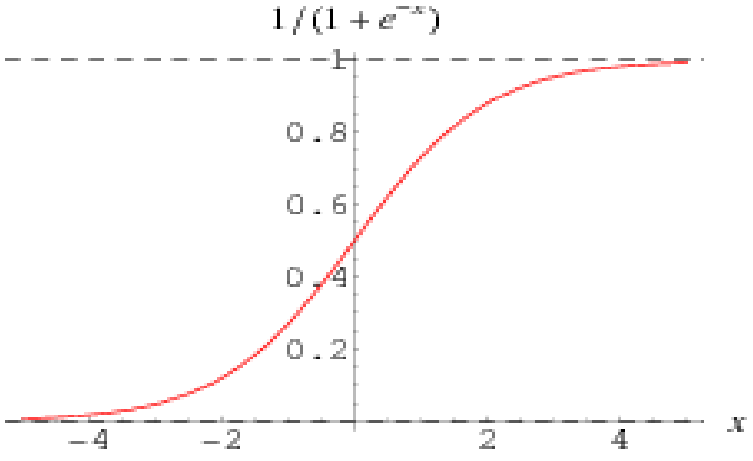
In this way, we get the probability of P(Y = 7|x) by multiplying P(Y=7), P(x1|Y=7) and P(x2|Y=7) . Similarly, we also get the P(Y=8) for the same test-data. Finally, we classify the test-data as digit '7' or digit '8' depending on whichever probability value is higher.

Here is the snippet of the results from the given dataset displaying the covariance matrix for individual digits and the prior probabilities for each digits and also the final accuracy of the model for classifying each digit 7 or 8.

```
Covariance metrix for digit 7  [[0.0009384940170133098, 0], [0, 0.0014614198211096577]]

Covariance metrix for digit 8  [[0.0014927242808970569, 0], [0, 0.0015991201963910897]]
Prior probability of digit seven : 0.5170848464839881

Prior probability of digit Eight : 0.4829151535160119

Accuracy of classification using Naive Bayes Classifier for digit 7 is :  74.31906614785993  %

Accuracy of classification using Naive Bayes Classifier for digit 8 is :  63.65503080082136  %

Accuracy of the entire DataSet is 69.13086913086913
```

4.1.2 Logistic Regression

Logistic Regression belongs to a class of Generative models which classifies the models based on the posterior probabilities calculated directly. It uses the sigmoid curve to convert the linearity in the data between 0 – 1. There assumption that each feature is conditionally independent is invalid here. We use a logistic function as mentioned here which is a sigmoid curve.



$$f(x) = \frac{1}{1 + e^{-(\vec{w} * \vec{x})}}$$

Here w is a parameter vector which helps to fit a curve with the independent variables.

In our case we have 3 values in the parameter vector w0,w1,w2. We find P(y=1|x) and P(y=0|x), and we check which amongst these two have more probability estimate and associate the given sample to that particular class. The optimization technique used here is Gradient Ascent.

We calculate the cost function and differentiate the cost function to maximize the probability and equate to zero and calculate the gradient for the cost function. We perform this process iteratively over a course of time until we find the optimal values for all the weights.

This maximizes the following log likelihood function

$$J(x, \theta, y) = \sum_{i=1}^{m} y_i \log(h(x_i)) + (1 - y_i)\log(1 - h(x_i))$$

where our hypothesis is a sigmoid function

$$h(x_i) = \frac{1}{1 + e^{\theta^T \bar{x}}}$$

We update the weights after calculation the gradient for each iteration and this process

Is calculated until the weights get optimized

$$\theta^+ = \theta^- + \alpha(y_i - h(x_i))\bar{x}$$

5. Results:

5.1 Naïve Bayes Classifier

The below are the results for the Naïve Bayes Classifier for digits 7 and 8.

Covariance matrix for digit 7 :
[[0.0009384940170133098, 0], [0, 0.0014614198211096577]]

Covariance matrix for digit 8 :
[[0.0014927242808970569, 0], [0, 0.0015991201963910897]]

Prior probability of digit seven : 0.5170848464839881

Prior probability of digit Eight : 0.4829151535160119

Accuracy of classification using Naive Bayes Classifier for digit 7 is : 74.319066147859 %

Accuracy of classification using Naive Bayes Classifier for digit 8 is : 63.655030800821 %

Accuracy of the entire Dataset is 69.13086913086913 %

5.2 Logistic Regression:

We check the accuracy for different values of alpha and with different number of iterations. If alpha – the learning rate is small, then it takes many iterations to converge to the maxima, if the learning rate is more we might miss out on the convergence. The decision boundary considered for each case here Is 0.5

S. No	Alpha	Iterations	Accuracy for digit "7"	Accuracy for digit "8"	Overall Accuracy
1.	0.00005	50000	76.94 %	66.63 %	71.92 %
2.	0.0001	70000	81.22 %	68.48%	75.02 %
3.	0.0009	50000	84.04 %	74.94 %	79.62 %
4.	0.00011	80000	82.19 %	68.78 %	75.6 %

6. Conclusion

In this project we have accomplished the task of classifying the two classes 7 and 8 with two methodology Logistic Regression and Naïve Bayes Classification. The main difference between the Logistic Regression and Naïve Bayes is that Logistic Regression uses the Logistic function and calculates the prior probability directly whereas the Naïve Bayes calculates the posterior probability using the conditional probability of each feature. Based on the observation Logistic Regression has a better classification accuracy than the Naïve Bayes algorithm. But it was observed that Naïve Bayes increases errors as the size of the data set increases. There are other parameters to be tuned to get more classification accuracy. We can increase the accuracy by choosing different decision boundary and different learning rate.