Fundamentals of Statistical Learning and Pattern Recognition

Part 1

1230029452 - Venkata Sai Rohit Bathi
Dr. Baoxin Li
October 30, 2023

Index

Introduction

Problem Statement

The presented problem is to classify an image that represents the number digit 5 or 6 as accurately as possible. We must do this using the MNIST dataset as our training data to classify the image having the number digit. We must use statistical methods to analyze the data and also to predict the new digit.

Dataset

The dataset I have used for this task is a subsection of the MNIST dataset. For training and testing image classification models, many people utilize the MNIST dataset, a well-liked collection of 28x28 pixel grayscale images with handwritten digits from 0 to 9. With the appropriate digit tagged on each, the dataset includes 10,000 testing samples and 60,000 training samples. All photos are centered and adjusted for user-friendliness in machine learning applications. Researchers and practitioners may reliably assess and compare the performance of various algorithms thanks to the MNIST dataset, which serves as a standard benchmark in the area. It is a fundamental dataset for machine learning research and education because of its accessibility and simplicity.

This implementation uses only the image data for digit 5and digit 6. The data extracted from the entire MNIST dataset is then split into testing and training dataset and provided to us. We have 4 files consisting of training and testing image data for each digit. The data we have is as follows:

```
training data shape:  (11339, 784)
testing data shape:  (1850, 784)
training labels shape:  (11339,)
testing labels shape:  (1850,)
```

```
labels sample:  [5. 5. 5. ... 6. 6. 6.]
```

	Digit 5	Digit 6
Number of samples in the training set	5421	5918
Number of samples in the testing set	892	958

Method

To perform the classification I have first loaded the dataset and then normalized the data for ease of calculations and eliminating domination of other features. After normalization, we perform Principal Component Analysis on the data for identifying principal components in the data and then reduce the number of dimensions of the data. Then this data is used to estimate ecach data class's (5 or 6) likelihood function parameters, where we assume that each data class is a normal distribution. This likelihood function is then used to estimate the probability of the input image belonging to that class, upon which we make the classification decision.

Data extraction and preprocessing

The data is in 4 different files: training_data_5, training_data_6, testing_data_5, testing_data_6. These .mat files are loaded using the spicy library load function. This data contains images data of handwritten digits 5 an 6. The loaded data is then processed to produce features for the image data, where each image is given by color value array of dimensions 28x28 (784 pixels). **Note:** This is done by taking each pixel color value as a feature. Each image data array is flattened hence resulting in each image having a feature length of 784. We will then concatenate the digit 5 and digit 6 training and testing data separately to form the training and testing datasets. We then create the labels separately for these datasets and concatenate to form training and testing label datasets. These are Numpy arrays having the following shapes:

Task 1 - Feature normalization

To normalize the data we follow the following formula:

$$\text{data}(i) = \frac{x_i - \text{mean}}{\text{standard deviation}}$$

Here each data(i) would be the each image vector from the training dataset and the mean and standard deviation would be vectors which are the mean and standard deviation for each feature in the training dataset. This is applied to the entire training dataset to obtain the normalized training data Numpy array. **Note:** Wherever the standard deviation of that certain

feature is 0, I have entered the normalization value as 0 because in the formula if standard deviation is 0 then the result becomes an NaN value in the normalized data. This is because standard deviation can be 0 only when $x_i = \text{mean}$. Hence I have entered 0 directly.

Task 2 - PCA on training samples

Then Principal Component analysis is performed on the normalized training data. Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving most of the original variance. It identifies "principal components" as linear combinations of features, ordered by their contribution to data variance.

To perform this we must first calculate the covariance matrix of the dataset. This covariance will be between features of the dataset. We will hence get a matrix of shape 784x784. This is then decomposed using Eigen decomposition to get the Eigen values and Eigen vectors of the covariance matrix. This eigen values and eigen vectors matrix represents how much each feature preserves the variance or discrimination in the image data. Each eigen value corresponds to the variance preserved by the latent feature in the image data. So, the eigen value and eigen vector matrices are sorted based on the eigen values. So we pick the first 2 principal components from these eigen values and vectors.

Task 3 - Dimension reduction using PCA

The above eigen vectors can be used to reduce the number of dimensions of the image data. We pick the top 2 principal components/eigen vectors and project our image data (dot product the 2 matrices) to get the image data on the reduced number of dimensions. The reason behind this is because we have noted that these eigen vectors preserve most of the variance in the data, hence we can omit most of the dimensions and just project our data on these 2 dimensions, while still preserving the variance in the data. Therefore, the 784 dimensions in our image data have been reduced to 2.

Task 4 - Density Estimation

The data distribution of the normalized data is then assumed to be a normal distribution. Then Maximum Likelihood Estimation is performed to get the parameters (mean, covariance) of the assumed distribution. We perform this for data of both the classes (digit 5 and digit 6). Therefore, we will get 4 parameters, mean and covariance of digit 5 and mean and covariance of digit 6.

To perform MLE on the distribution we first obtain the general equation of a normal data distribution.

$$f(\mathbf{x} \ \ \mu, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k \ \mathbf{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right)$$

Here we perform MLE to obtain $\hat{\mu}_{\text{MLE}}$ and $\hat{\Sigma}_{\text{MLE}}$ by partially differentiating the normal distribution function each time and equating it to 0. We get expressions for the parameters which are:

$$\hat{\boldsymbol{\mu}}_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$$

$$\hat{\Sigma}_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu}_{\text{MLE}})(x_i - \hat{\mu}_{\text{MLE}})^T$$

Here we notice that the mean and the standard deviation terms are in fact the mean and covariance of the data. Hence we just need to find the mean and covariance of the data to get the MLE parameters of the normal distributions of digit 5 and digit 6 likelihood functions. We obtain the parameters of the normal distribution functions for each class by doing the above for data from each digit class.

Task 5 - Bayesian Decision Theory for optimal classification

Bayesian theory states the following:

$$P(w_i \ \ x) = P(x \ \ w_i)P(w_i)$$

Now, the minimum error rate classification rule states that

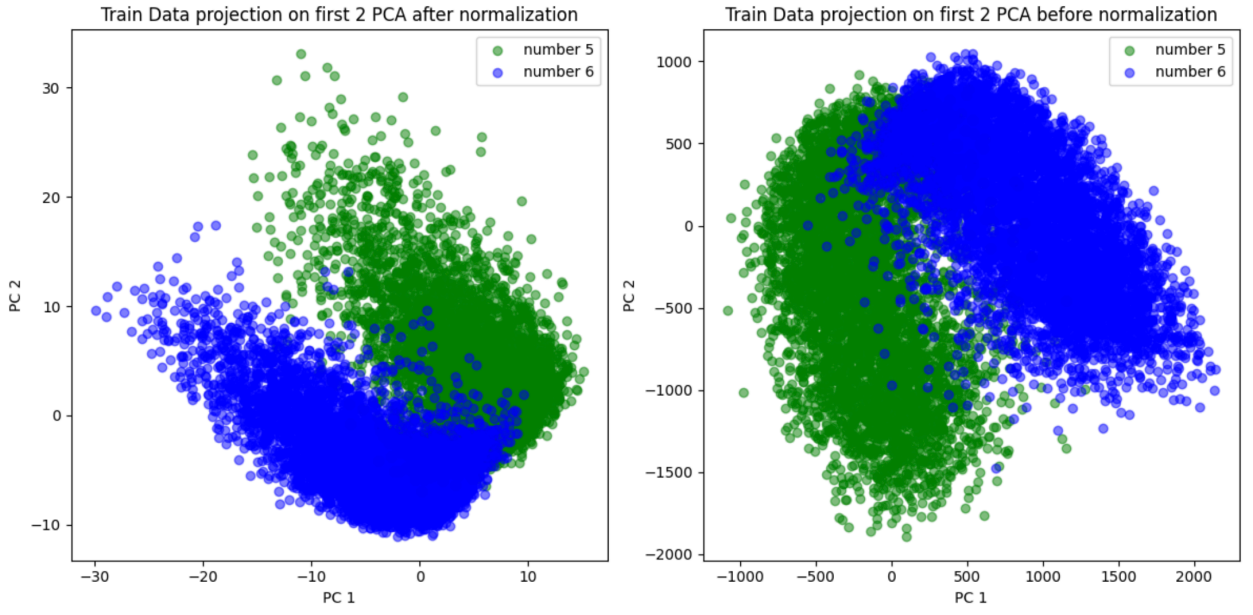
$$\text{Decide } w_1 \text{ if } P(x \ \ w_1)P(w_1) > P(x \ \ w_2)P(w_2)$$

Using this rule and the above likelihood function we calculated or each class we can find the probability of an image feature vector belonging to that particular class. We use this to predict the label for each image vector for both training and testing image data. Here, it is assumed that the prior probabilities $P(w_1) = P(w_2) = 0.5$. We then check the accuracy of such the decision rule by using the accuracy of the predicted labels when compared with actual labels of testing dataset.

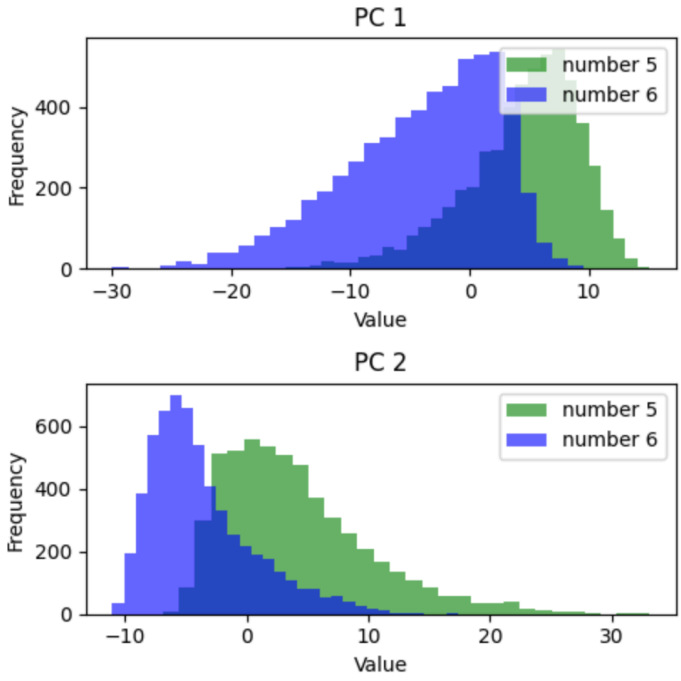
Results and observation

PCA results

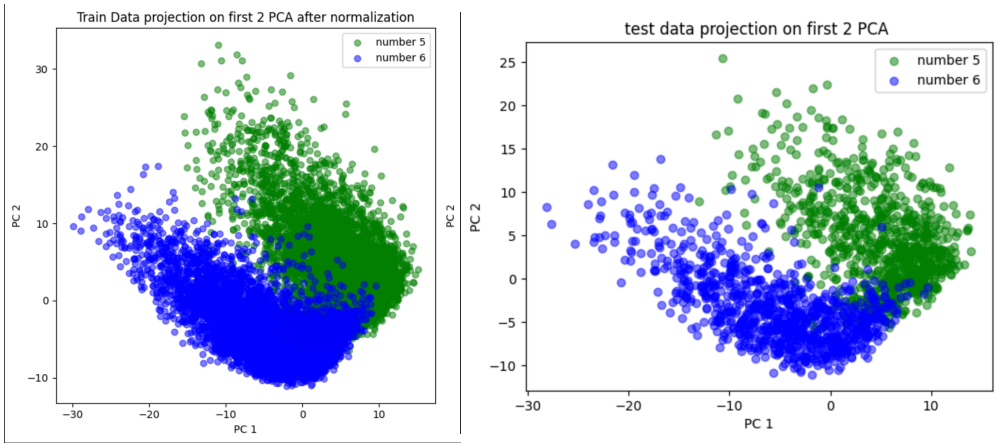
After performing PCA on the normalized training data, we can observe the effect normalization has on the data. Normalization scales down the data and distinguishes the class data from each other more clearly. It also distributes the data similar to a normal distribution (as we can see data is elliptical and clusters at mean and reduces when spread out). We can notice this difference in the following plot



The plot on the left has class clusters well separated when compared to the one on the right and the data is more normalized compared to the one on the right. This is more evident in the next plot



The above plot is a histogram of the PCA dimension reduced normalized training data. This shows that the data of each reduced component is well separated from each other, hence well clustered and the data is also normalized (similar to a normal distribution). We can compare this with test data normalized also. The following shows comparison between normalized training data and testing data reduced to 2 principal dimensions.



The testing data is projected on the training data's eigen vectors as we must not let the testing sample data leak out, also in real life scenario we have no access to testing data. We can observe here that both the training data and testing data after normalization and dimension reduction to 2 dimensions resemble each other, in another sense, they are similar clusters in similar ranges and patterns hence the classifier must give good accuracy.

Density parameter estimation results

As shown in the above histogram plot, the mean and covariance of each digit class will be as shown in the below output:

```
digit 5 parameters:
mean:
 [4.4531976  4.06950581]
cov:
 [[ 23.39752387 -15.13656603]
 [-15.13656603  36.44251424]]

digit 6 parameters:
mean:
 [-4.07921328 -3.72774434]
cov:
 [[ 42.26834766 -17.94685568]
 [-17.94685568  18.33380651]]
```

The mean and covariance here will be the estimated MLE parameters for the normal pdfs for each digit class because when we perform MLE on the multivariate normal density function as discussed in the previous methodology we get the mean an covariance of each digit class data.

Bayesian Decision classification results

Using the above discussed method for classification we can obtain the following results when performed on both training and testing data.

```
Training data :


Accuracy score:
 0.9427639121615663

Classification report:
              precision    recall  f1-score   support

         5.0       0.94      0.94      0.94      5421
         6.0       0.94      0.95      0.95      5918

    accuracy                           0.94     11339
   macro avg       0.94      0.94      0.94     11339
weighted avg       0.94      0.94      0.94     11339
```

```
Testing data:


Accuracy score:
 0.9383783783783783

Classification report:
              precision    recall  f1-score   support

         5.0       0.93      0.94      0.94       892
         6.0       0.94      0.94      0.94       958

    accuracy                           0.94      1850
   macro avg       0.94      0.94      0.94      1850
weighted avg       0.94      0.94      0.94      1850
```

From the above results we can see that the classification rule classifies the training data well with 94.2% accuracy and the testing data well with 93.8% accuracy. This aligns with our inference with the data plots where the clusters are we'll separated with distinguishable patterns.

Conclusion

Through this analysis, we applied multiple techniques to the MNIST subset focusing on digits "5" and "6".

- **Feature Normalization**: We converted training images into 784-dimensional vectors and normalized each feature based on its mean and standard deviation. This ensured consistency and improved the performance of subsequent PCA.
- **PCA Implementation**: Dimensionality reduction was achieved using PCA, reducing from 784 dimensions to 2. This transformation was essential for efficient data visualization and computation.
- **2-D Visualization**: By representing the data in a 2-D space, we observed the clustering patterns of the two digit classes, evaluating whether they followed a Gaussian distribution.

- **Density Estimation**: Assumptions were made regarding the Gaussian distribution of each class in the 2-D space. Parameters of these distributions were estimated using training data.
- **Bayesian Classification**: Using the derived Gaussian parameters, Bayesian decision theory was applied to classify the digits. The model yielded an accuracy of 0.93 for both training and test datasets.

The results indicate the efficacy of the chosen techniques in classifying the two digits. Future work can explore alternative dimensionality reduction methods and classifiers for comparison.