Multi-Classification with MNIST Dataset

Lucen Zhao
MATH4432 Monthly-Project 1
Student ID: 20256435
lzhaoaj@connect.ust.hk

Abstract

In this project, I applied all classification models we learned in this course to the feature-extracted MNIST database and evaluated the classification results. Among all the models, logistic regression, linear discriminant analysis and 3-nearest neighbour models achieved the best results. The results are evaluated with F1 scores, confusion matrices, p-values and ROC curves.

1 Introduction

The classification of hand-written digits is an interesting research topic with applicational potential. As the development of machine learning methods facilitate, abundant research works have been done on this topic, and various kinds of classification models have been proved to produce promising results. Most of the works are based on the MNIST database, a popular dataset of grayscale hand-written digits.

In this project, I examined some machine learning models on the MNIST database. In this report, I will explain the feature extraction process firstly. Then I will apply logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and k-nearest neighbour (KNN) models to the feature vectors for classification. The results of classification on the MNIST database will be shown, followed by in-detail analysis of the results.

2 The MNIST Database

The original MNIST database contains 60,000 training images and 10,000 testing images, each normalized into a 20x20 pixel grayscale image. Because a large number of well-known digit recognition and classification models have been applied to this dataset, the performance of my models can be evaluated by comparing my results to the existing works.

The experiments given in [1] can provide an overview on the performance of baseline models. By applying classification models to raw images of the original MNIST database, that is to say, without additional feature extraction, the test error rate for linear regression model and pairwise linear regression model are 8.4% and 7.6% respectively, while the performance of k-nearest neighbour classifier is much better, with a test error rate of 2.4% when k=3.

In this work, I will use another version of MNIST database, with 7291 training images and 2007 testing images, normalized into 16x16 pixel space. Some sample images are shown in Figure 1.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

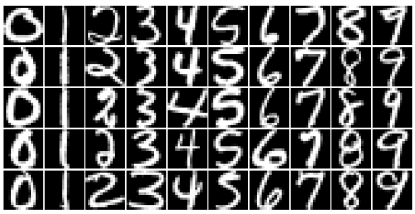

Figure 1: Samples from MNIST database

3 Methodology

3.1 Feature Extraction

To enhance the performance of my models, instead of using the whole image as input, I tried to compress the information from the image and extract feature vectors as input. several methods are used to extract feature vectors.

Histogram Histogram on the original image is taken as features, to show the distribution of pixels in different zones of the image. The whole image is divided either into 16 zones with 4x4 pixel, or into 9 zones with 6x6 pixel (with overlap). Because the original images are in grayscale, the histogram put all pixels into 2 bins, which respectively indicate the number of pixels in the digit and in the background. In practice, only the bin for the digit are sufficient to be used for features, since the sum of 2 bins is always the size of the zone. The histograms are also normalized by dividing the size of the zone.

Gradient Gradients of each pixel in horizontal and vertical directions are calculated to show whether there is a change in colour around the pixel, i.e. whether the pixel is on the "edge" or the hand-written digit. Compared with histogram and pooled image, the gradient can show more structural information of the image. After calculating the gradient, the original gradient values, pooled gradient values and histograms by zones of gradient values can all be used as feature vector. Some samples of gradient values are shown in Figure 2.

Figure 2: Gradients on horizontal and vertical directions

Chain Code The chain code is a chain of pixels, each with a direction, that depicts the outline of an object. In this project, 8-direction chain codes of each image are calculated to show the structure of each digit. Because the size of chain codes are different for different images, the histogram by 9 zones are calculated as feature. Some sample chain codes are shown in Figure 3.

Figure 3: Connected chain code pixels

Feature vectors extracted by different methods can reflect different structural information of the image. The features and the original image can be selectively combined together for better classification results.

3.2 Classification Models

Logistic regression, LDA, QDA and KNN models are used for classification.

Logistic Regression Logistic regression models are applied in this project. I tried both regression models for 2-class and multi-class. In the 2-class case, for each class of digit (0-9), it will decide the probability that the image belongs to current class or not. L2 penalty is used because it can achieve more stable performance compared with L1 penalty. Several algorithms for learning are used as well, including coordinate descent (CD) algorithm, Newton conjugate gradient algorithm and limited-memory BGFS (LBGFS).

Linear Discriminant Analysis LDA model can classify the input data based on a distinct linear function for each class. Compared with logistic regression, much time can be saved by using LDA.

Quadratic Discriminant Analysis QDA is similar to LDA, but the function is quadratic instead of linear.

K-Nearest Neighbour The KNN classifier can classify the input data by assigning it the same class as the majority of its neighbours. In our project, several k values are tried to find the most suitable model. The distance between data points are measured by their Euclidean distance, and the features are normalized to ensure that they make similar contribution to the final result.

3.3 Validation and Evaluation

I selected the k value of KNN algorithm and the solvers for logistic regression model with 3-fold cross validation. Moreover, the methods for feature extraction are also selected with 3-fold cross validation.

The F1 scores for each class and confusion matrices on test dataset are used to evaluation the results of selected models. F1 score is defined as follow:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Where precision is the percentage of true positive among all positive predictions, and recall is the percentage of true positive among the set of all true positive and false negative.

Bootstrap was not used in this project, because the dimension of features is very high in this work, resampling and retraining the model can be very time consuming, especially for logistic regression and k-NN models.

4 Experiment

4.1 Model Selection

Because I am trying logistic regression models with different solvers, and k-nearest neighbour algorithms with different k values, it is very time-consuming to run these models on all combinations of feature vectors. Hence, a 3-fold cross validation is used for model selection to select the best logistic regression and KNN models based on limited number of cases.

4.1.1 Model Selection based on original image

In this case, only the original image is used as input. By 3-fold cross validation on the training set, we obtained the accuracy values for the models shown in Table 1.

Table 1: Cross-validated accuracy

Model	Score
Logistic regression (CD)	0.9429
Logistic regression (Newton-CG)	0.9439
Logistic regression (LBGFS)	0.9439
Multiclass logistic regression (Newton-CG)	0.9460
Multiclass logistic regression (LBGFS)	0.9462
K-nearest neighbour (k=1)	0.9624
K-nearest neighbour (k=3)	0.9606
K-nearest neighbour (k=10)	0.9506

4.1.2 Model Selection based on feature vector

In this case, we combined the original image and all the kinds of feature we extracted to the feature vector except pooled image (because original image has already been used here) to check the performance of all there models. The performance are shown in Table 2.

Table 2: Cross-validated accuracy based on features

Model	Score
Logistic regression (CD)	0.9727
Logistic regression (Newton-CG)	0.9728
Logistic regression (LBGFS)	0.9727
Multiclass logistic regression (Newton-CG)	0.9716
Multiclass logistic regression (LBGFS)	0.9684
K-nearest neighbour (k=1)	0.9702
K-nearest neighbour (k=3)	0.9688
K-nearest neighbour (k=10)	0.9629

Based on the results above, finally I chose logistic regression model and multiclass logistic regression model with Newton-CG solver, and KNN models with k equals to 1 and 3.

4.2 Feature Selections

Because I am using many kinds of feature, to simplify the experiment, I combined the original image with each kind of feature separately to find the suitable features by 3-fold cross validation, with accuracy scores. The results are shown in Table 3.

From the table above, we can see that although histogram and the gradients on original image achieved good performance on some models, their performance on QDA is disaster-like. Even on other models, the results of these features are ok but not better than other kinds of features. Hence, finally we only chose the histogram of gradients, histogram of chain codes and the original image as our features.

4.3 Experiment Results

4.3.1 Baseline Results

The baseline results in Table 4 are achieved by testing the models with test set using the original image as input. In some cases, our baseline results are actually worse than the results achieved in [1], probably because we are using a light version of the original dataset, and the number of samples is much smaller.

4.3.2 Final Results

By combining the original image, histogram of gradients and histogram of chain codes, we achieved fair results with all classifiers. The details of the results are listed in Table 5.

Table 3: Average F1 scores with different types of features

Model		
Name	Feature	Score
LDA	Histogram (4x4)	0.8121
QDA	Histogram (4x4)	0.5686
Logistic regression	Histogram (4x4)	0.8505
Multiclass logistic regression	Histogram (4x4)	0.8701
KNN (k=1)	Histogram (4x4)	0.8826
KNN (k=3)	Histogram (4x4)	0.8897
LDA	Gradient histogram (4x4)	0.9041
QDA	Gradient histogram (4x4)	0.8083
Logistic regression	Gradient histogram (4x4)	0.9212
Multiclass logistic regression	Gradient histogram (4x4)	0.9288
KNN (k=1)	Gradient histogram (4x4)	0.9398
KNN (k=3)	Gradient histogram (4x4)	0.9450
LDA	Gradient image	0.9179
QDA	Gradient image	0.2021
Logistic regression	Gradient image	0.9340
Multiclass logistic regression	Gradient image	0.9294
KNN (k=1)	Gradient image	0.9356
KNN (k=3)	Gradient image	0.9281
LDA	Chain code	0.9428
QDA	Chain code	0.7373
Logistic regression	Chain code	0.9519
Multiclass logistic regression	Chain code	0.9558
KNN (k=1)	Chain code	0.9481
KNN (k=3)	Chain code	0.9521

Table 4: Baseline results

Model	Correct Predictions	Wrong Predictions	Precision	Recall	F1 Score
LDA	1777	230	0.89	0.89	0.89
QDA	1061	946	0.40	0.53	0.44
Logistic regression	1834	173	0.91	0.91	0.91
Multiclass logistic regression	1842	165	0.92	0.92	0.92
K-nearest neighbour (k=1)	1894	113	0.94	0.94	0.94
K-nearest neighbour (k=3)	1896	111	0.95	0.94	0.94

From the results above, we can see that the results of LDA, logistic regression, multiclass logistic regression and 3-nearest neighbour models are the best, with around 95% testing data classified correctly. Because the 2 logistic regression models are similar, in the following section, I will mainly focus on the 2-class case for evaluation.

5 Evaluation

In this section, I will mainly evaluate the models which achieved the best performance, namely, LDA, logistic regression and 3-nearest neighbour models. For other models, I will evaluate possible reasons that they did not achieve a good performance.

5.1 Discriminant Analysis Models

5.1.1 LDA

The precision, recall, F1 score of each class and the confusion matrix is shown in Figure 4.

Table 5: Final results

Model	Correct Predictions	Wrong Predictions	Precision	Recall	F1 Score
LDA	1904	103	0.95	0.95	0.95
QDA	1728	279	0.87	0.86	0.86
Logistic regression	1909	98	0.95	0.95	0.95
Multiclass logistic regression	1899	108	0.95	0.95	0.95
K-nearest neighbour (k=1)	1895	112	0.94	0.94	0.94
K-nearest neighbour (k=3)	1902	105	0.95	0.95	0.95

	precision	recall	f1-score	support
0	0.98	0.97	0.98	359
1	1.00	0.97	0.98	264
2	0.96	0.91	0.94	198
3	0.91	0.94	0.92	166
4	0.92	0.93	0.92	200
5	0.94	0.91	0.93	160
6	0.96	0.97	0.96	170
7	0.96	0.93	0.94	147
8	0.92	0.95	0.93	166
9	0.91	0.97	0.94	177
avg / total	0.95	0.95	0.95	2007

```
[[350   0   1   2   1   0   1   0   3   1]
 [  0 256   1   0   3   0   3   1   0   0]
 [  0   0 181   4   4   1   1   2   5   0]
 [  0   0   2 156   0   7   0   1   0   0]
 [  0   1   3   0 185   0   2   0   1   8]
 [  2   0   0   9   0 146   0   0   1   2]
 [  2   0   1   0   2   0 165   0   0   0]
 [  0   0   0   1   6   0   0 136   1   3]
 [  3   0   0   0   1   1   0   0 157   4]
 [  0   0   0   0   0   0   0   2   3 172]]
```

Figure 4: Results of LDA model

From the precision and recall above, we can see that the average test error rate for both type I and type II error are 0.05, which is satisfying for a linear model. Among all the classes, the error rate for digit 3, 4, 5 and 8 are relatively higher, and the error rates between class 3 and 5 are relatively higher. Taking these classes as example, the ROC curves are shown in Figure 5.

From the curves, it is clear that all those classes are quite distinguishable, with the true positive rate near to 1.0 while false positive rate is still at a very low level (although the curve for digit 3 is slightly under others). From the curves, we can conclude that in our model, data points belonging to different classes are highly distinctive.

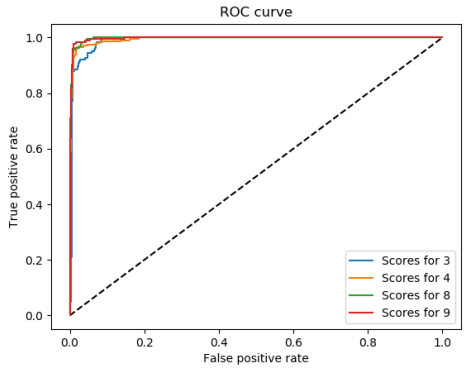

Figure 5: ROC curves of LDA model

5.1.2 QDA

The precision, recall, F1 score and confusion matrix of each class is shown in Figure 6.

The performance of QDA is much worse than LDA. By analysing the F1 scores of each classes, we found that its performance on class 0, 1 and 7 are still stable, while the performance greatly declined for other classes, especially for class 4, 8 and 9, where the F1 score are lower than 0.8.

This is very likely to be caused by overfitting. From the sample data in Figure 1, we can see that there is a large variance for digits such as 4, 8 and 9, while digits like 0, 1 and 7 are mostly in very

similar shape. Because of the large variance of data, QDA is likely to under perform on these classes, because its quadratic assumption makes it more likely to suffer from large variances.

	precision	recall	f1-score	support
0	0.92	0.94	0.93	359
1	0.98	0.92	0.95	264
2	0.73	0.96	0.83	198
3	0.88	0.87	0.88	166
4	0.85	0.66	0.74	200
5	0.87	0.90	0.89	160
6	0.94	0.85	0.89	170
7	0.98	0.86	0.91	147
8	0.82	0.65	0.72	166
9	0.68	0.89	0.77	177
avg / total	0.87	0.86	0.86	2007

```
[[339   0  13   2   1   0   0   0   3   1]
 [  0 243   2   0   3   0   8   0   5   3]
 [  3   0 190   1   2   1   0   0   0   1]
 [  2   0   8 145   0   7   0   0   3   1]
 [  1   1  13   1 132   1   1   1   1  48]
 [  3   0   2   4   0 144   0   0   4   3]
 [  5   0  12   0   2   6 144   0   1   0]
 [  0   1   4   5   6   0   0 126   1   4]
 [ 15   0  14   7   2   6   0   0 108  14]
 [  1   2   1   0   8   0   0   2   6 157]]
```

Figure 6: Results of QDA model

5.2 Logistic Regression

5.2.1 2-Class Logistic Regression

The precision, recall, F1 score and confusion matrix of each class is shown in Figure 7.

	precision	recall	f1-score	support
0	0.97	0.99	0.98	359
1	1.00	0.97	0.98	264
2	0.94	0.91	0.92	198
3	0.93	0.91	0.92	166
4	0.93	0.93	0.93	200
5	0.91	0.94	0.93	160
6	0.96	0.96	0.96	170
7	0.96	0.95	0.95	147
8	0.94	0.96	0.95	166
9	0.94	0.97	0.96	177
avg / total	0.95	0.95	0.95	2007

```
[[354   0   1   1   2   0   0   0   0   1]
 [  1 255   0   0   3   0   2   1   2   0]
 [  2   0 180   5   2   2   1   2   4   0]
 [  1   0   2 151   0   9   0   1   1   1]
 [  0   1   3   0 185   1   3   1   0   6]
 [  2   0   0   6   0 150   0   0   0   2]
 [  0   0   3   0   2   1 163   0   1   0]
 [  0   0   1   0   6   0   0 139   0   1]
 [  3   0   2   0   0   1   0   0 160   0]
 [  1   0   0   0   0   0   0   1   3 172]]
```

Figure 7: Results of logistic regression model

Its performance on class 2, 3, 4 and 5 is slightly worse than other classes, and the highest error rate occurred between class 3 and 5. This is understandable: by using the structural features we obtained, these 2 digits have similar distribution on their shape of curves.

To further evaluate the model, taking class 2 as example, the p-values of each feature are calculated. However, it is not sufficient for evaluation on high-dimensional data. Because no single feature made great contribution to classification, the p-values of all features are very large (all larger than 0.01). The reason behind this is very intuitive: no matter which single data point we dropped from the feature vector, it can still make a relatively good classification because of the overly high dimension of feature vectors.

5.2.2 Multi-Class Logistic Regression

The results of multi-class case is only slightly worse than the "one vs rest" classifier. Because they are intrinsically similar, this classifier also slightly under performed on class 2, 3, 4 and 5, and the highest error rate also occurred between class 3 and 5.

5.3 K-Nearest Neighbour

5.3.1 3-Nearest Neighbour

The precision, recall, F1 score and confusion matrix of each class is shown in Figure 8.

From the scores and matrix, it is clear that the performance for class 2, 4, 5, 8 and 9 are not as good as other classes, and between class 0 and 2 many errors occured.

Both 3-NN and 1-NN models did not improve much compared with their corresponding baseline results. This is possible related with the nature of k-NN model: because all features are valued with same weight, with a feature vector of shape 358, 256 dimensions of it are from the original image. Hence the original image actually contributed to most of the Euclidean distance value. However, by removing the original image from feature vector, or substituting it with max pooled image (window size 2x2; stride 2), the performance was not enhanced. From this we can conclude that the k-NN classification can be done sufficiently with the original images.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	359
1	0.99	0.98	0.98	264
2	0.94	0.91	0.92	198
3	0.94	0.94	0.94	166
4	0.93	0.91	0.92	200
5	0.94	0.91	0.93	160
6	0.97	0.97	0.97	170
7	0.95	0.94	0.94	147
8	0.94	0.92	0.93	166
9	0.90	0.96	0.93	177
avg / total	0.95	0.95	0.95	2007

```
[[355   0   3   0   0   0   0   0   0   1]
 [  0 258   0   0   3   0   2   1   0   0]
 [  9   0 180   0   0   2   0   2   5   0]
 [  2   0   1 156   1   3   0   2   0   1]
 [  1   2   2   0 182   1   2   2   1   7]
 [  3   0   1   5   0 146   1   0   1   3]
 [  0   0   2   0   1   2 165   0   0   0]
 [  0   0   1   0   5   0   0 138   1   2]
 [  2   0   2   5   0   1   0   0 152   4]
 [  1   0   0   0   4   0   0   1   1 170]]
```

Figure 8: Results of 3-NN model

5.3.2 1-Nearest Neighbour

The results of 1-NN model is slightly worse than 3-NN model mostly because of the unstable nature of 1-NN model. Because in 1-NN model, the decision is solely based on the single nearest data point to the testing data, compared with 3-NN, it is more easily to be affected by noise.

5.4 Wrong Classifications

To further evaluate the model, I chose to look back into the dataset. Taking our best model, logistic regression model as example, I randomly chose some samples of wrongly classified images shown in Figure 9. Comparing the samples with Figure 1, most of the wrongly classified samples are not very distinguishable by nature. Hence I think it is understandable to classify those images into wrong classes.

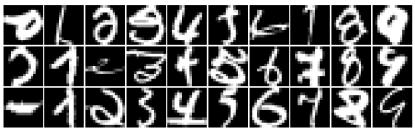

Figure 9: Wrongly classified images

6 Conclusion

To conclude, this project tried LDA, QDA, logistic regression (based on 2 classes), multi-class logistic regression, 1-NN and 3-NN models to classify hand-written digits in the MNIST dataset. Models and features are selected with 3-fold cross validation. The combination of original image, 16-zone histogram of gradients and 9-zone histogram of chain code is used as the feature vector. The logistic regression model achieved best result of classification with the F1 score of 0.95, and the LDA and 3-NN models achieved similar results in shorter running time.

Reference

[1] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard & V. Vapnik. (1995) Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, pp. 53–60. Paris, 1995. EC2 & Cie.