STAT 139 Project
Handwritten Digit Classification Using Image Data

Ming Long Wu, Qin Lyu, Hengte Lin

1. Introduction

Image recognition has been a challenging problem that involves statistics, mathematics and computation. However, recent developments in statistics and in machine learning start to enable image recognition which draw lots of interests in related fields. In this project, we aim at identifying hand written digit numbers by analyzing grayscale raw images. Mathematically, identifying hand written digits can be formulated as a classification problem, with each digit from 0 to 9 being a class. In this project, we compare statistical models (e.g., dual-class and multiclass Logistic Regression) for better classification of input images. In addition, dimension reduction and variable selection are implemented in order to gain insights of data and to efficiently reduce model complexity. Furthermore, we use cross validation to confirm the stability of the selected model. Results demonstrate that our trained model is capable of classifying hand written digits with typical accuracy of 0.97.

2. Materials and Methods

2.1 Image datasets

Datasets from MNIST (Mixed National Institute of Standards and Technology database) are used in this project [1]. Figure 1 shows example images of hand written digits used in this project. For each digit, from 0 to 9, the subset contains around 5,000 grayscale images with dimensions of 28x28, summing up to a total of 784 pixels per image. The grayscale values of 784 pixels in each image are used as input variables to classification models. In this project, a total of 50,000 images are included and are randomly separated as training and testing sets. All data preprocessing and modeling are performed using R and python. Specific methods and settings are detailed below.

Figure 1. Example images of hand written digits used for classification. Datasets are downloaded from MNIST (Mixed National Institute of Standards and Technology database).

2.2 Binary Logistic Regression for initial tests

The first classification model we use is Logistic Regression. The Logistic Regression model works by performing a sigmoid e^y/(1+e^y) transformation on predicted value. As a result, the output value will be scaled into [0,1] range to work as classification probability. Figure 2 plots a sigmoid function for demonstration [2].

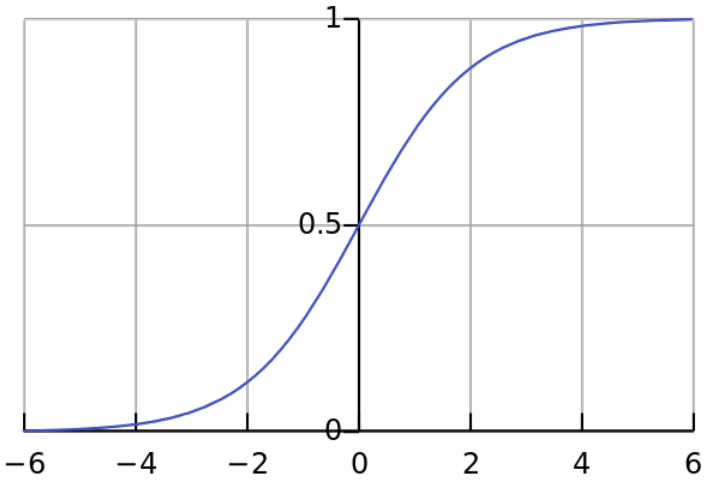

Figure 2. A sigmoid function used in the Logistic Regression model.

In our project, mlogit package of R is initially used to perform multiclass Logistic Regression with grayscale pixel data. However, from our tests, it is found that mlogit cannot process input data due to its large matrix size. Therefore, we perform a simpler binary classification between 0 and 1 using R built-in glm function. Based on our finding in this baseline test, we decide to use python for most of our subsequent data analysis and data modeling.

2.3 Dimension reduction of input data

One of the reasons why image recognition is challenging is because image data are frequently large and, therefore, are usually computationally expensive. Taking datasets used in our project as an example, using grayscale values of all pixels as learning features can be impractical or even infeasible. Training a model with 784 variables (i.e, pixels) can be slow and expensive. In addition, inclusion of too many variables (or predictors) can result in overfitting of models. Our finding of R not being able to handle data size of 50,000 x 784 is a vivid example demonstrating the importance of data reduction before actually fitting any model.

2.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a method commonly used for reducing data dimension. To reduce data dimension, PCA projects data onto orthogonal directions in the feature space with the goal of maximizing variability in the projected data. In practice, Principal Components (PC)

(i.e., projection directions) are ranked by percentages of variance they explained. Therefore, dimension reduction can be achieved by discarding PCs that explain little variance. In this project, PCA is implemented both in R and in Python. Figure 3(a) shows variance explained from original data by all PCs. Cumulative explained variance in data is plotted in Figure 3(b), which is used for selecting number of PCs to be included in the reduced data.

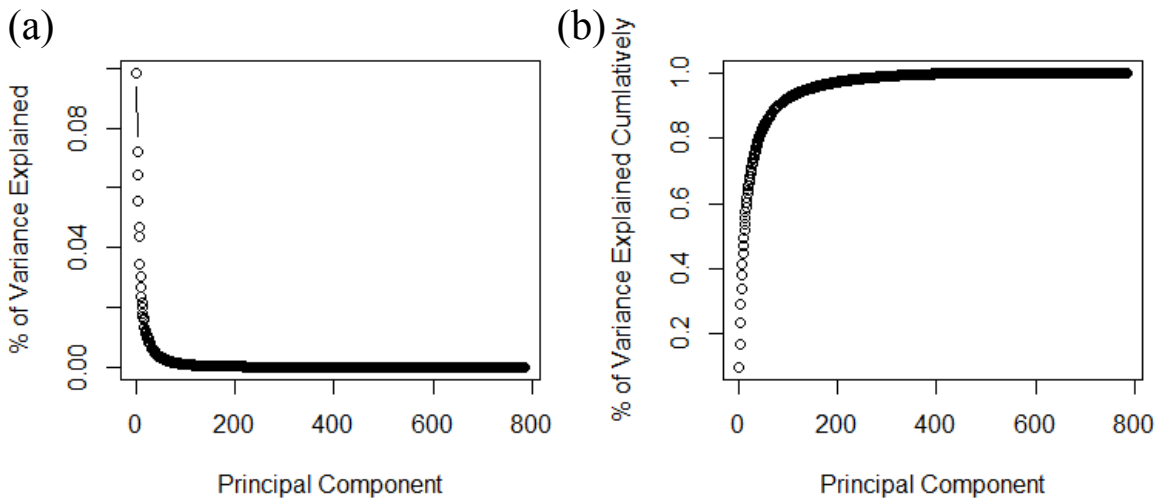

Figure 3. Principal Component Analysis (PCA) for data reduction. (a) Percentage of variance explained from original data by all Principal Components (PCs). (b) Cumulative plot of explained variance for selecting number of PCs to be included.

From cumulative plot in Figure 3(b), we select top 150 PCs, which explain 95.035% of variance from original data, to construct the reduced data. The reduced 150 features are then used for training classification models.

2.3.2 Alternative feature construction

In addition to PCA, we propose an alternative method for constructing reduced feature space based on key characteristics of image data. Instead of using grayscale values of all image pixels, features are calculated from all images to provide good summary of entire images. Two features are calculated from all images: (1) the percentage of pixels that contains ink and (2) average grayscale value of each individual image. The proposed dimension reduction method efficiently reduces dimension of feature space from 784 to 2 and it is computationally inexpensive. The alternative feature construction method is implemented in R and is then tested both in the binary Logistic Regression and in the multiclass Logistic Regression.

2.4 Variable selection

According to results from PCA, we recognize that there is great potential for reducing the dimensionality of the original data. In this section, we explore the possibility of variable selection, in which way we can avoid using all 784 dimensions of features.
In the image identification problem of this project, features are grayscale values of raw pixels. As shown in Figure 1, in hand writing images, ink presents only in certain areas of images. Therefore, intuitively, different local areas in the images should not be weighted equally in our classification task and we even speculate that some local areas are not helpful for classification. To identify the importance of different areas in raw images, we analyze whether all local areas in the images are statistically significant for classification (i.e., discriminant among digit classes).
First, we divide each image into a 7 by 7 local areas, where each area contains a 4 by 4 patch of image. Then, we calculate the average grayscale values from each 4 by 4 patch followed by ANOVA in R to analyze whether average grayscale values of all local areas are different among groups (i.e., digit classes). Log(F-value) and significance maps of local areas are shown to demonstrate variable selection.

2.5 Multiclass Logistic Regression

To include all digit classes in a model, we implemented multiclass Logistic regression using functions provided by python sklearn package. PCA is used for dimension reduction by including top 150 PCs for multiclass Logistic regression (i.e., reduced data dimension 50,000x150). The reduced dataset is then randomly separated to a training set (50%) and a testing set (50%).

2.6 Quadratic Discriminant Analysis (QDA)

In this project, we also include Quadratic Discriminant Analysis (QDA) for comparison. Both Linear Discriminant Analysis (LDA) and QDA assume input data as mixture of Gaussian distributions (i.e., classes). By examining reduced dataset from PCA, we observed that most predictors are symmetric and do not deviate too far from Gaussian distribution (as shown in Figure 6). Therefore, we consider that LDA and QDA may be suitable for our classification problem. In LDA, classes are modeled with equal variance while, in QDA, different variances are estimated for all classes [3]. In this project, LDA and QDA are implemented using python sklearn functions.

2.7 Cross validation

In this project, image classes in our dataset are almost equally distributed and random split is used to generate a training dataset and a testing dataset. Therefore, we expect that accuracies calculated by fitting models once should be reliable (and reproducible). However, to make sure that the model we built is reliable, we implement cross validation (n_fold = 5) in python and examine the stability of accuracies both in training and in testing datasets.

3. Results

3.1 Binary Logistic Regression and alternative features

Binary Logistic Regression in R using all pixels (784 features)

As an initial test, the following R output shows accuracies of > 0.87 both in training and testing datasets in binary logistic regression of class 0 and 1.

Null deviance: 7.6382e+02 on 599 degrees of freedom
Residual deviance: 3.4809e-09 on 225 degrees of freedom
AIC: 750
Accuracy in training data set: 0.991725
Accuracy in test data set: 0.872473

With 784 variables, it seems that the model is a little overfitting.

Binary Logistic Regression in R using alternative features (2 features)

Alternative features perform well in binary classification (accuracy > 0.99) as shown in the following R output. In addition, fitting a logistic regression model with alternative image features takes less computation time compared to fitting both original and PCA reduced data.

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.51091	-0.00167	-0.00012	0.00195	1.96771

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	31.5576	6.1441	5.136	2.80e-07	***
binary_percent_col_pixel	-358.6068	67.8482	-5.285	1.25e-07	***
binary_average_col	0.9578	0.2142	4.472	7.74e-06	***

Signif. codes: 0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Null deviance: 1265.506 on 999 degrees of freedom
Residual deviance: 35.523 on 997 degrees of freedom
AIC: 41.523

Accuracy in train data set: 0.995428
Accuracy in test data set: 0.990049

Multiclass Logistic Regression in R using alternative features (2 features)

From the following R output, we observe underfitting of the multiclass Logistic Regression model (accuracy ~ 0.32). This implies that the two constructed variables from raw images are insufficient for explaining difference in hand written digits in the multiclass classification problem.

Coefficients:

	(Intercept)	percent_col_pixel	average_col
1	23.898296	-301.283406	0.88414228
2	5.174183	-39.703003	0.08058412
3	5.296316	-37.962848	0.06800333
4	9.649015	-80.188513	0.18551817
5	7.673253	-51.577586	0.07838684
6	8.174729	-75.601469	0.20270305
7	12.702857	-130.910698	0.37793875
8	2.528163	-3.694433	-0.05633352
9	10.196398	-90.298256	0.22782807

Std. Errors:

	(Intercept)	percent_col_pixel	average_col
1	0.5305465	6.842271	0.02969759
2	0.4746553	5.673406	0.02420695
3	0.4717959	5.638659	0.02414164
4	0.4561167	5.644514	0.02470142
5	0.4691953	5.767911	0.02530527
6	0.4566408	5.591953	0.02401545
7	0.4435461	5.557798	0.02419227
8	0.4922251	5.690000	0.02429198
9	0.4493997	5.547624	0.02418503

Residual Deviance: 12717.74
AIC: 12771.74
Accuracy in train data set: 0.318235
Accuracy in test data set: 0.316683

3.2 Principal Component Analysis

Figure 4 shows top 10 PCs of the original data from PCA. Note that PCs have information at the center of images but not at the margins. Figure 5 shows 3D scatter plot by projecting the original data onto the feature space of the top 3 PCs. It can be seen that 10 digit classes can be identified and be visually separated. Figure 6 shows histograms and Normal Quantile-Quantile plots (QQ-plot) of 5th predictor (in (a) and (b)) and 100th predictor (in (c) and (d)) in the PCA-reduced dataset. Most predictors in PCA-reduced data are symmetric and largely follow Normal distribution (data not shown). Variance in the 5th predictor is larger than that in the 100th predictor (150287 vs 3230). The fact that predictors in reduced dataset present Normal distribution implies that LDA and QDA may be used for classification in this project.

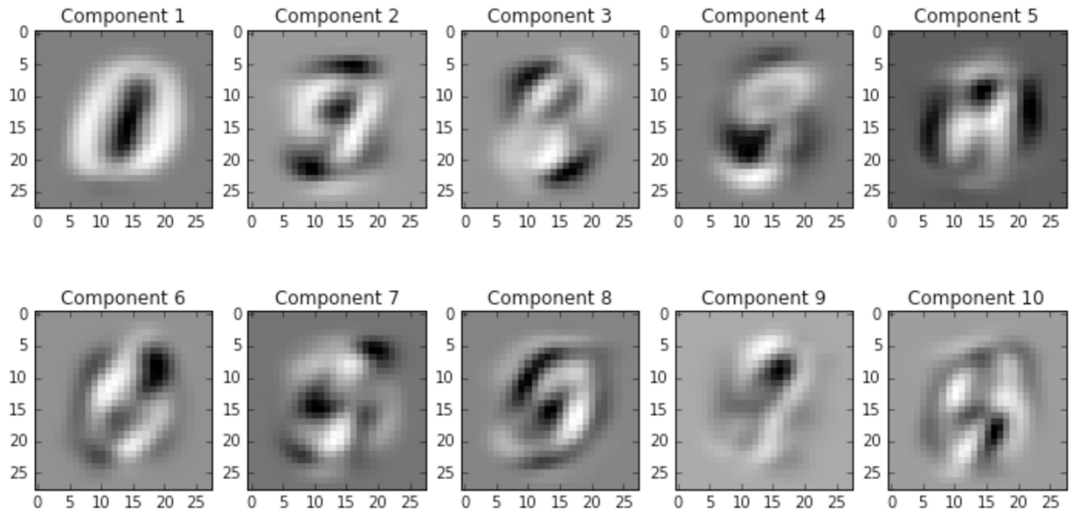

Figure 4. Top 10 Principal Components of the original data. Note that PCs have information at the center of images but not at the margins.

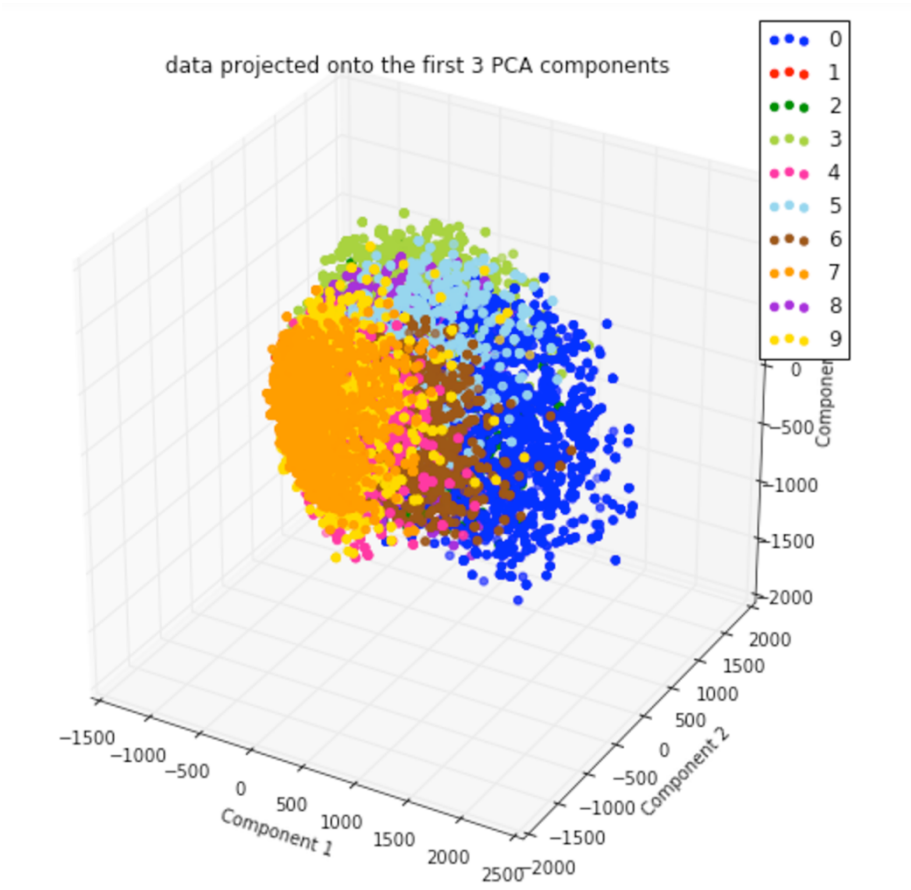

Figure 5. Projection of the original data onto feature space of the top 3 PCs. Note that 10 digit classes can be identified and be visually separated.

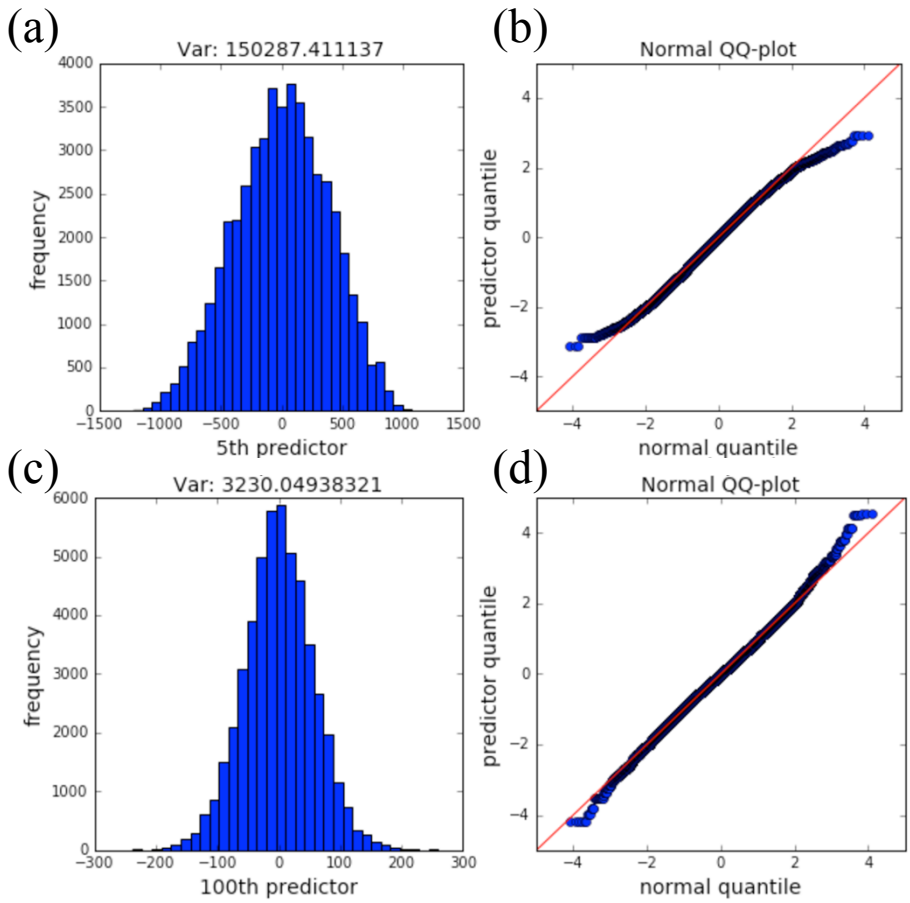

Figure 6. Histogram and QQ-plot of reduced predictors from PCA. (a) and (b) show histogram and QQ-plot of the 5th predictor (variance 150287). (c) and (d) show histogram and QQ-plot of the 100th predictor (variance 3230). Both predictors largely follow Normal distribution.

3.3 Variable selection

Figure 7(a) shows log(F-value) of 7 by 7 local areas (each containing a 4x4 patch) from ANOVA. As can be seen, local areas near image corners have low F-value and, hence, show less statistical significance of representing difference among the average grayscale values of 10 digit classes. On the other hand, local areas at the center have the highest F-value, indicating that the average grayscale values in these areas might be significantly different among digit classes.

Figure 7(b) shows significant local area in red with alphas set to 0.01. According to Figure 7(b), we conclude that local areas at image corners are not significantly different among digit classes. More importantly, we can discard those insignificant local areas for performing classification. The statistical results agree well with our intuition because image corners frequently do not contain important information (i.e., digits are centered in MNIST data) and, thus, can be excluded for image classification task.

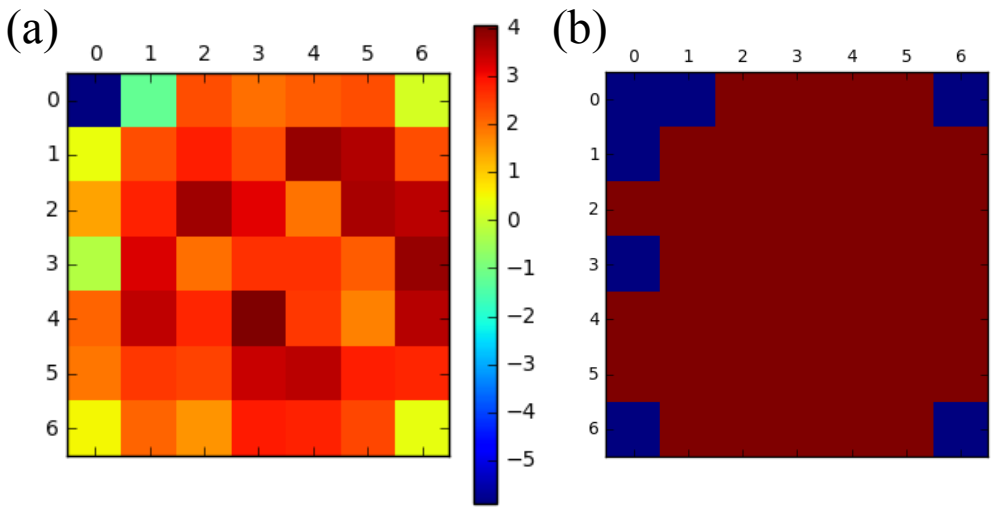

Figure 7. Variable selection by analyzing 7 by 7 local areas (each containing a 4x4 patch) of all images. (a) log(F-value) of all local areas from ANOVA. (b) Significant local areas are shown in red (alpha = 0.01).

3.4 Regression models

Multiclass Logistic Regression, LDA and QDA are compared regarding accuracy and efficiency. Table 1 shows accuracies of the compared methods. It can be seen that QDA gives highest accuracy (> 0.97) among three methods. Although Logistic Regression also achieves high accuracy (> 0.93), its computation time is 1412 folds longer than that of QDA. Therefore, we select QDA among three methods based on test results.

	Logistic Regression	**LDA**	**QDA**
Training accuracy	0.938	0.890	**0.984**
Testing accuracy	0.932	0.881	**0.975**
Computation time (s)	228.789	0.204	**0.162**

Table 1. Comparison of accuracy and efficiency in multiclass Logistic Regression, LDA and QDA.

3.5 Cross validation and fine tuning model

Five-fold cross validation of QDA gives consistently high accuracies in all iterations (> 0.97 in training and testing). In addition, fine tuning of QDA shows that similarly high accuracies can be achieved when its regularization parameter is set in the interval of (1e-6, 1e-1).

4. Discussion

In this project, we investigate the problem of identifying hand written digits. The task of image recognition is modeled as a statistical classification problem. By using image data as input, we inevitably face the challenges of large datasets. On one hand, the large number of predictors

(each consisting of a pixel value) makes fitting statistical models computationally expensive. On the other hand, generally speaking, fitting models with (unnecessarily) high complexity increase risk of overfitting. From initial tests, we realize that we need to switch from R to python to accommodate image data. In real life scenario of image recognition, challenges arise from large datasets can be even more severe due to the fast advancement of image capture devices such as smartphone cameras.

We, therefore, make many efforts to explore dimension reduction and variable selection, with both aiming at reducing model complexity. Among methods that are explored, PCA has the advantage of being universally applicable to most data type (i.e., not limited to image data). The degree of dimension reduction can be controlled by number of PCs included in the reduced data with the tradeoff of explained variance from the original dataset. However, the PCs included cannot be interpreted easily (i.e., directly relate to data) as seen in Figure 4. On the other hand, the proposed alternative feature method empirically extracts image features from the original dataset and can be readily be interpreted. In fact, in the binary Logistic Regression model, high accuracies (> 0.99) and high dimension reduction (784/2 = 392 folds) are achieved at the same time. It has to be noted that although only two alternative features are used in this project, more features can be calculated based on understandings of the original dataset and based on the goal of classification. Lastly, a simple yet statistically rigorous method is applied to achieve variable selection by performing ANOVA on averaged grayscale values of local image areas. Insignificant local areas (image corners in our case) can be safely excluded for fitting classification models because they likely do not contain discriminant information for the task. More important, from the perspective of information content, PCs in PCA present information only at the center of images, which agrees well with results from ANOVA.

Among compared classification methods, QDA excels both in accuracy (> 0.97) and in computation efficiency (~0.16 sec). Examination of normality in predictor distribution (Figure 6) implies that QDA can be a good candidate model. We conjecture that the higher accuracy achieved by QDA compared to LDA can be because variances in digit classes are unequal. In addition, although multiclass Logistic Regression also achieves high accuracy (> 0.93), it is more than 1,000 folds slower than LDA/QDA. Lastly, cross validation results confirm that the fitted QDA model is stable for MNIST data. For real world data, we expect that cross validation and fine tuning the QDA regularization parameter will be even more important.

Base on the above results, we conclude that successful results are achieved for image recognition from raw images of hand written digits. The techniques used in this project, including preprocessing and modeling methods, should be able to be generalized to real world digit number identification with careful fine tuning.

Reference

[1] https://www.tensorflow.org/versions/r0.10/tutorials/mnist/beginners/index.html

[2] https://en.wikipedia.org/wiki/Sigmoid_function

[3] G. James et al., An Introduction to Statistical Learning, Springer 2013