HANDWRITTEN DIGIT DIMENSIONALITY REDUCTION AND CLASSIFICATION

RYAN MOUW

Applied Mathematics Department, University of Washington, Seattle, WA
rmouw@uw.edu

Abstract. The MNIST dataset contains 16 x 16 images of handwritten digits. We examine a subset of 2000 training images, and 500 test images, with the goal of training a classifier that can predict the label of handwritten digits. Principle Component Analysis is used to reduce the dimensionality of the images. Pairs of integers are then projected onto the reduced dimensional space and a binary ridge classifier is trained with the intention of distinguishing between the integer pairs.

1. Introduction and Overview

The original MNIST dataset (Figure 1) is a collection of 60,000 handwritten training digits (0-9), and 10,000 testing images.[1] We examine 2000 of the training images (X_{Train}), and 500 of the test images (X_{Test}), in order to train a classifier. Principal Component Analysis is used to reduce the dimensionality of the data. To investigate the amount of dimension reduction that is possible, we compare the Frobenius norm of the singular values of all the dimensions, to the Frobenius norm of the lower dimension approximation, with certain rational goals (e.g. 90%). The reduction in dimensionality lowers the computational complexity, and thus the computational cost, of the problem.

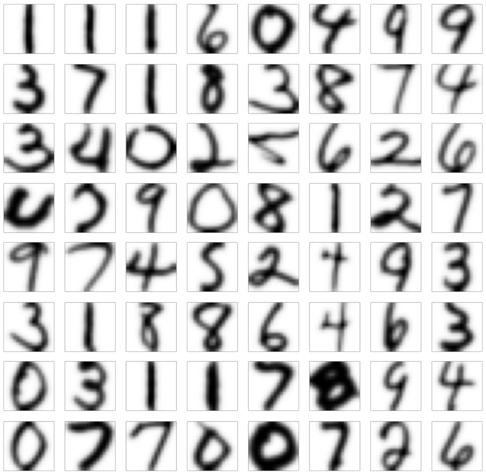

Figure 1. Sample of Handwritten Digits Using 256 PC Modes From the MNIST Dataset

Date: February 11, 2022.

2. Theoretical Background

The initial problem is to understand the dimensionality of the data. Because the images are 16x16, this results in a 2000 x 256 training dataset and a 500 x 256 test dataset. In the context of these images, each pixel is a "feature", resulting in a high-dimensional problem. It is natural, then, to ask whether the images can be distinguished using fewer than 256 pixels. Principle Component Analysis (PCA) will help us understand the answer to this question. PCA is an application of matrix Singular Value Decomposition (SVD).[2] SVD decomposes matrices into three parts. For any real matrix A, SVD is a decomposition of A into;

$$A_{m\times n} = U_{m\times m}\ \Sigma_{m\times n}\ V^T_{n\times n}$$

Where $\Sigma_{m\times n}$ is a diagonal matrix with elements equal to the root of the positive eigenvalues of AA^T and A^TA. The diagonal elements of Σ are known as the singular values. This decomposition is not unique, and it is always possible to choose it in a way such that the diagonal entries of Σ are in descending order, and thus reorders the columns of U and V^T. The columns of V^T are eigenvectors of A^TA and form a basis for $\mathbb{R}^n$. The columns of U are the eigenvectors of AA^T and form a basis for $\mathbb{R}^m$. The columns of U are also referred to as the Principal Components of A. In this problem, matrix A is used to represent image data. If the Σ values are decomposed in such a way as to be in descending order, then the corresponding columns in U signify the direction of variance, in descending order, of the image data.

As mentioned before, the singular values are the diagonal entries of $\Sigma_{m\times n}$. The singular values can be interpreted as scalars describing the amount of variance in the data along a certain axis (The correlated eigenvectors in U). Thus, to get a sense of the total variance in the data, we use the Frobenius norm of the singular values. The square of the Frobenius norm for a matrix B is defined as;

$$||B||_F^2 = \sum_{j=1}^{min\{m,n\}} \sigma_j(B)^2, \quad B \in \mathbb{R}^{m\times n}$$

Where $\sigma_j(B)$ is the j^{th} diagonal entry in Σ_B. Because the singular values (σ_j) are associated with the variance in the data, the Frobenius norm can give us a scalar value of the total variance in the data set, discretized by axis. As the dimensionality of the images are reduced, we can calculate the Frobenius norm to determine how much variance remains in the data when it is projected to the lower dimensions.

Ridge regression is often used when divvying up a space in which classification is being done.[4] Simply put, Ridge regression is a least squares fit, but with an added L2 regularization parameter that penalizes models that produce large coefficients in their approximation of functions. This is beneficial because large coefficients tend to create high variance in predictions which leads to inaccuracy. The added L2 penalization of the coefficients forces the model to keep some level of bias, which lowers prediction variance, and thus increases accuracy. The L2 penalization parameter itself can be modified. This is done via a process called Cross-Validation, which helps select the optimal amount of L2 penalization needed.

3. Algorithm Implementation and Development

Sci-Kit Learn was used for both the PCA and Ridge Classification in this problem.[3] The general steps were as follows;

(1) Extract 16 PC modes of highest variance from X_{Train}
(2) Project a pair of handwritten integers (e.g. 1 and 8) from X_{Train} onto the 16 PC modes

(3) Train a Ridge Classifier using Cross Validation on the projected integers (Assigning labels of -1 and 1 to the digits being compared. This normalizes the output of the classifier.)
(4) From X_{Test}, project the same pair of handwritten digits on the 16 PC modes
(5) Using the Ridge classifier trained in Step 3, predict the labels of the handwritten digits from both the Training set (X_{Train}) and the Test set (X_{Test})
(6) Calculate the Mean Square Error (MSE) of both the Training and Test sets

4. Computational Results

Figure 2 shows that the variance drops roughly 3 orders of magnitude among the (roughly) first 25 Principal Components. After this steep decline, the variance decreases slowly, dropping about 2 orders of magnitude over roughly the next 125 Principal Components. Due to this difference, it seems there is room for dimensionality reduction amongst the first 25 (or so) Principal Components.

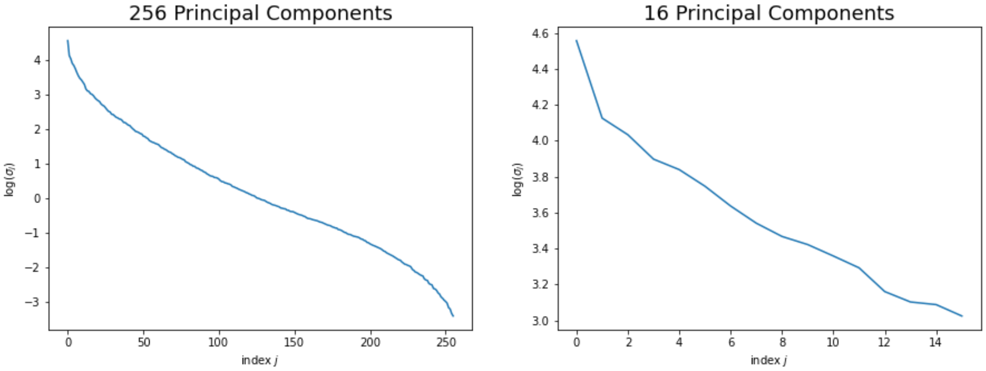

Figure 2. Log of Singular Values

Figure 3 shows the 16 PC modes that are associated with the most variance, as 16 × 16 images, in decreasing order. These are the 16 PC modes that the training data will be projected onto, in order to fit the Regression Classifier.

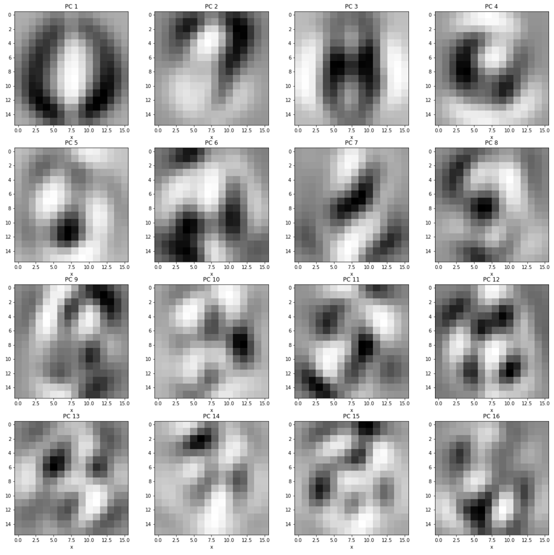

Figure 3. 16 × 16 Images of 16 Principle Components with Largest Associated Singular Values

Figure 4 shows the reconstructed handwritten images after the dimension reduction using 16 PC modes. The images are still quite readable, albeit more blurry than the images in Figure 1 which use all 256 PC modes.

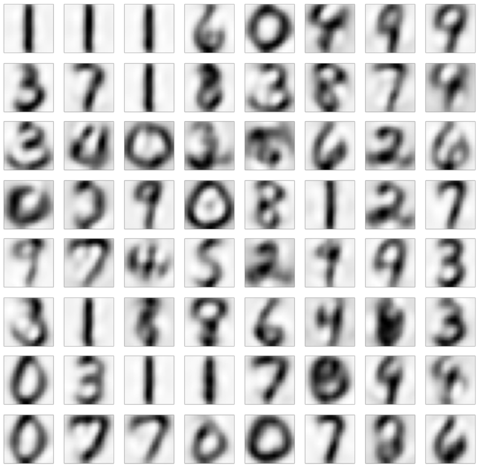

Figure 4. Handwriting samples using 16 Principal Components

Figure 5 reveals how the Frobenius norm changes as the data is projected onto different numbers of Principal Components. Both the number of PC modes and the log of the PC modes are shown to give a different perspective on the rate of change. Figure 5 shows that the number of PC modes needed to approximate X_{train} to a minimum of 60%, 80%, and 90% of the Frobenius norm, are 3, 7, and 14, respectively. This illustrates that reducing the original 16 × 16 image down to a select 14 pixels will be sufficient to maintain 90% of the variance in the dataset.

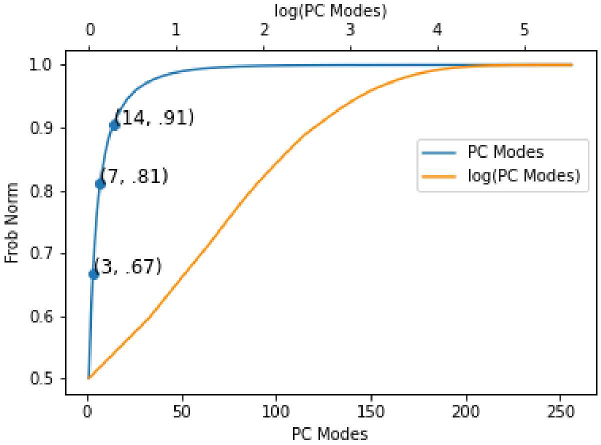

Figure 5. Frobenius Norm vs PC modes and Log(PC Modes)

16 PC modes are used to train the Ridgle Classifier between pairs of handwritten digits. Table 1 displays the Mean Square Error (MSE) of both the training and testing datasets between the pairs of handwritten digits (1,8), (3,8), and (2,7).

Digit Pairs	MSE Train	MSE Test
(1,8)	0.0881	0.0856
(3,8)	0.1903	0.2295
(2,7)	0.0981	0.1292

Table 1. Train and Test set MSE for Ridge Classifiers Trained on Different Pairs of Handwritten Integers

The performance difference in the classifiers can be understood rather naturally from a visual perspective (i.e. 3's and 8's look alike). A more formal way to explain the performance difference is to look at the Euclidean distance between the handwritten digits once they are projected onto 16 PC modes. Figure 6 shows the average L2 norm between each unique pair of digits and the associated Test MSE. The line of best fit shows that the average Euclidean distance between the the digit pair and the Test MSE are inversely related. This means that as the digits become more alike visually, their corresponding data points become closer and potentially overlap in the 16 dimensional space to which they were projected. The classifier is then able to distinguish between the two groups with less accuracy.

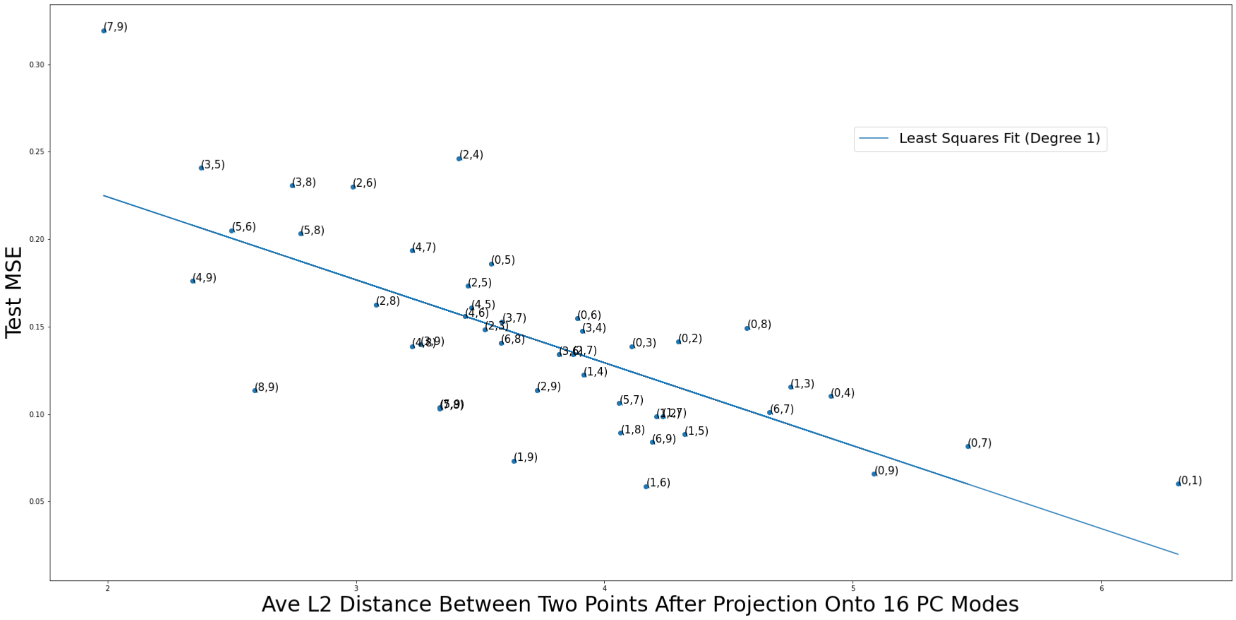

Figure 6. Average L2 Norm vs MSE of Pairs of Integers From MNIST Dataset

5. Summary and Conclusions

This problem involved reducing the dimensionality of the MNIST dataset and training a classifier on the reduced dataset to distinguish between pairs of handwritten digits. PCA was used to investigate the data dimensionality and Ridge Regression was used to train the classifier.

The investigation into the dimensionality of the MNIST dataset revealed that significant reduction was possible in the context of classification. The Ridge Classifier also showed that the penalization parameter played an insignificant role in determining Test MSE, with lambda values ranging form 10 to .01 changing the test MSE only slightly. A likely reason for this is the high number of samples relative to the number of features in the data.

Particular pairs of digits that had higher MSE, indicating that they are more difficult to differentiate, were (7,9), (3,8), (3,5), and (2,4). Digits pairs with relatively low MSE, indicating a more accurate differentiation, were (1,6), (0,9), (0,1), and (1,9).

An interesting observation is that certain pairs of digits have larger differences in their respective training and testing MSE. This could suggest that certain handwritten digits are more prone to overfitting than others. Future studies could investigate this suggestion, and if true, explore reasons as to why.

Acknowledgements

I would like to thank Nalu Zou for suggesting the inverse relationship between Euclidean distance and MSE. I believe connection and the resulting visual in Figure 6 provided needed context to this problem. I would also like to thank Katherine Owens for providing guidance surrounding the computation of the Mean Square Error.

References

[1] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[2] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] V. Vovk. Kernel ridge regression. In *Empirical inference*, pages 105–116. Springer, 2013.