AMATH 582: HOME WORK 3

SUKHJIT KAUR

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

Abstract. This project explores the classification of handwritten digits from the well-known MNIST dataset using dimensionality reduction and machine learning classifiers. Leveraging Principal Component Analysis (PCA) the high-dimension space is reduced and key modes are kept to represent the majority of the data variance. Binary classification is conducted for three digit pairs using Ridge regression, with cross-validation used to assess model performance. For multi-class classification, Ridge regression, K-Nearest Neighbors (KNN), and Linear Discriminant Analysis (LDA) are compared in terms of accuracy and computational efficiency. The results show that KNN achieves the highest accuracy and fastest training time, while Ridge and LDA provide comparable but less accurate results. The study highlights how PCA aids in dimensionality reduction without significant loss of information, and how classifier choice impacts accuracy.

1. Introduction and Overview

Provided an image set of handwritten digits from the well-known Modified National Institute of Standards and Technology (MNIST) dataset, we attempt to train a classifier to recognize the digits. The dataset is split - consisting of 60,000 images in a training set used to train the classifier, and 10,000 images in a test set used for validation of the classifier method. The training and test sets, X_{train} and X_{test} are made up of 28px x 28px sized greyscale images. Each pixel in the 28x28 image (784 total pixels) represents a *feature*. The digit assigned to each image in the training or test set, y_{train}, y_{test} are referred to as a *label* (Figure 2).

Applying Principal Component Analysis (PCA) on X_{train} we can reduce the dimensionality of the data and determine which principal component modes (k) are necessary to represent the highest amount of variance of the data (Figure 1). This increases computational efficiency, and reduces any noise that may be present in the data.

Using these number of (k) PC modes, the image can be reconstructed with many less pixels - using only those that correspond to the most prevalent features in each image (Figure 4b).

With the processed and dimension-reduced data, a classifier is trained to differentiate two specific digits from the set, repeated for 3 pairs. This is done by extracting the features and labels of a given digit-pair (e.g. 1,8) and

Date: February 28, 2025.

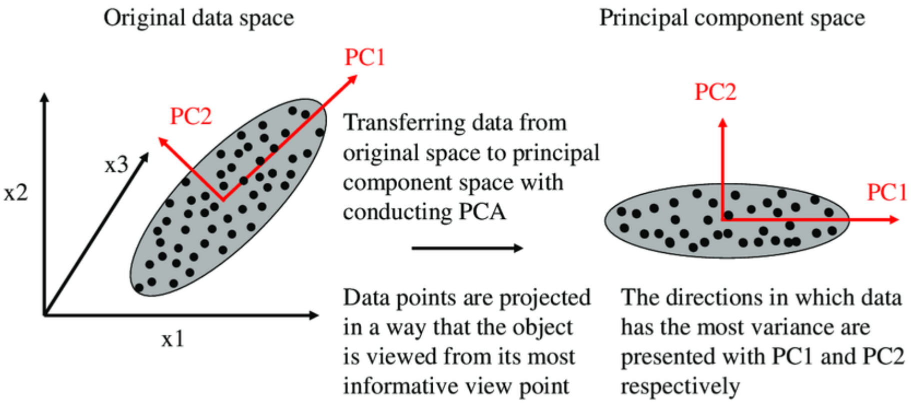

Original data space
Principal component space
PC1
PC2
x3
x2
x1
Transferring data from original space to principal component space with conducting PCA
PC2
PC1
Data points are projected in a way that the object is viewed from its most informative view point
The directions in which data has the most variance are presented with PC1 and PC2 respectively

Figure 1. A PCA transformation, where the original data (left) is transformed/projected to the lower dimension PC space. Full image credit to Ghasemi et al. [1][2]

Figure 2. The first 64 images from the MNIST dataset

projecting those on to the PC modes of the training image set. Ridge classifier is used to classify the test set and cross validation is performed. The training and testing accuracies and mean-squared error is compared.

Finally, multi-class classification is completed using Ridge classifier and is compared to k-Nearest Neighbors (kNN) and Linear Discriminant Analysis (LDA) classifiers to determine the most accurate method.

2. Theoretical Background

Classification is made easier using **Principal Component Analysis** (PCA). PCA is a mathematical method for dimensionality reduction and used to represent higher-dimensional data using only a number of coefficients. PCA is closely related to Singular Value Decomposition (SVD) and involves the covariance of the matrix X,

$$C_x = \frac{1}{N-1} X X^T \tag{1}$$

where,

$$X = U \Sigma V^T \tag{2}$$

resulting in the following, where U^T is the PC basis, and the columns of U are the eigenvectors of C_x and known as the PC modes. The eigenvectors tell us the optimal direction that represents the most variance of the dataset.

$$C_x \approx \frac{1}{N-1} U \Sigma^2 U^T \tag{3}$$

The projection of the data into the PC basis can also be done with truncated (k) PC modes. The number of modes, k, corresponding to the largest singular values, σ are kept in the truncation. [4]

Using the Frobenius norm, the energy can be defined as, $E = ||A||_F^2$, where $||A||_F = \sqrt{\Sigma_{j=1}^{min(m,n)} \sigma_j^2}$, resulting in,

$$E = \Sigma_{j=1}^{min(m,n)} \sigma_j^2 \tag{4}$$

The cumulative energy of the singular values can be used to determine the number of PC modes that are necessary to retain some percentage of the variance in the dataset (e.g. 85%).

To perform supervised learning, specifically classification, linear regression can be used,

$$f(X) = \beta_0 + \Sigma_{j=1}^{d} \beta_j x_j \tag{5}$$

Further, the **maximum likelihood estimation** (MLE) to 5 is given by,

$$f_{MLE}(X) = \beta_{MLE} = argmin_{\beta \in IR} \frac{1}{2\sigma^2} ||A\beta - Y||^2 \tag{6}$$

changing the problem to finding the least-squares solution for $A\beta = Y$. However, since A could be higher-dimensional, the chances of it being linearly dependent increases, and thus also the likelihood that $A^T A$ is singular.

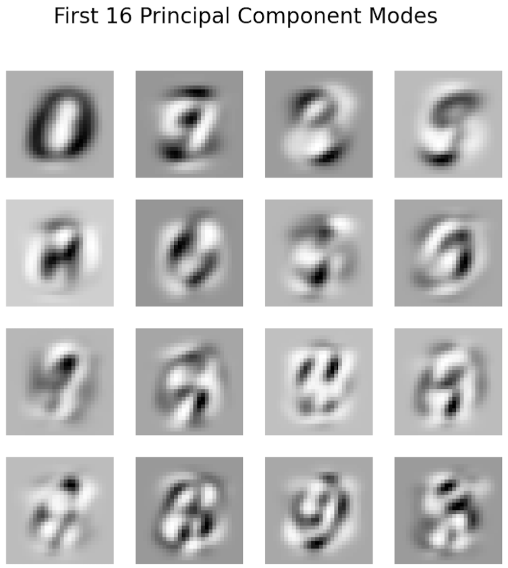
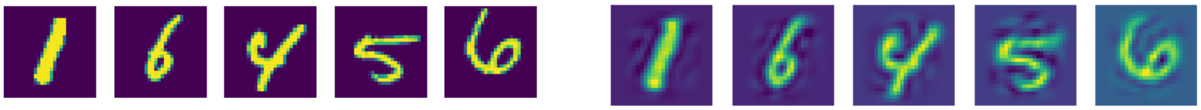


Figure 3. PCA modes of the training set



(A) Original

(B) Reconstructed with 59 modes

Figure 4. A sample of images from the MNIST data set

By adding a regularization term, $\frac{\lambda}{2}||\beta||_p^p$, with $p = 2$, the equation becomes,

$$\beta_{MLE} = argmin_{\beta \in IR} \frac{1}{2\sigma^2}||A\beta - Y||^2 + \frac{\lambda}{2}||\beta||_2^2 \tag{7}$$

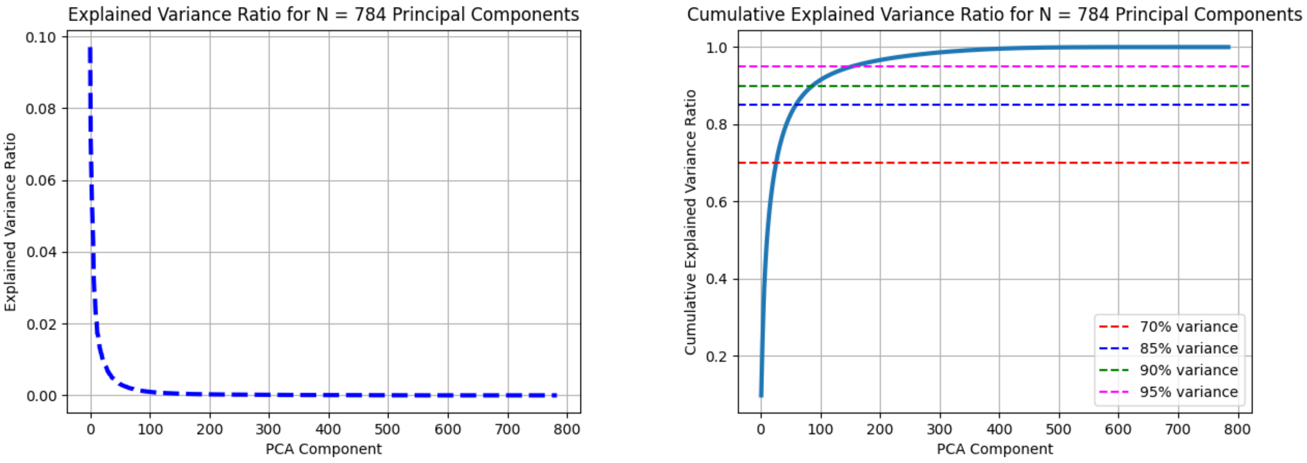
This regularization essentially introduces a term to β: $\sigma^2 \lambda I$, and thus ensures that the matrix is always invertible, alleviating any issues due to singularity. Regularization also effectively reduces the variance of the estimates. This is called **Ridge Regression**.

3. Algorithm Implementation and Development

To create a classifer for the MNIST dataset, PCA was implemented on the entire training image set using `.PCA()` from scikit-learn [5]. Performing PCA reduces the high dimensionality of the matrix and identifies all the principal components of the data - in particular sorting them from most to least variance captured. The first 16 principal component modes are visualized as 28 × 28 images, giving some insight into which features characterize the images (Figure 3).

Next, the cumulative energy of the singular values from the PCA analysis is found to determine the optimal number of principal components, k, needed to retain 85% of the total variance (Table 1). Matplotlib is used to plot the variance ratios (Figure 5b) [3]. Through reconstruction of some images using k truncated PC modes, the quality of the approximation is visually verified before further use later (Figure 4b).

Specific subsets of data from X_{train} and X_{test} are extracted using a newly defined function, `select_digits` and then projected onto the reduced-dimensionality PCA space. The selected digits (e.g. 1 and 8) are classified using `RidgeClassifierCV` for both the training and test data. The subset accuracy (`accuracy_score`) is found, and cross-validation is conducted (`cross_validate`) to evaluate the model performance. This is visualized using `confusion_matrix, ConfusionMatrixDisplay`.



(A)

(B)

Figure 5. Explained variance ratio used to determine principal components necessary for variance retention (See Table 1)

Finally, three supervised classification methods (Ridge, K-Nearest Neighbors (KNN), and Linear Discriminant Analysis (LDA)) were implemented on the entire dataset projected onto PCA space. Each classifier implemented on the test data is compared via accuracy and computational time. Accuracy is determined using `.score`, and time to fit each classifier is found using `tic(), toc()` from the `ttictoc` library.

4. Computational Results

Using PCA, 59 PC modes were determined to be sufficient to approximate 85% of the cumulative explained variance. This was deemed reasonable by visual comparison of a few reconstructed images in the k = 59 PC space (Figure 4b) with the original counterparts (Figure 4a). Likewise, 26 modes are needed to approximate 70% of the cumulative variance and 87 modes are needed to achieve 90%.

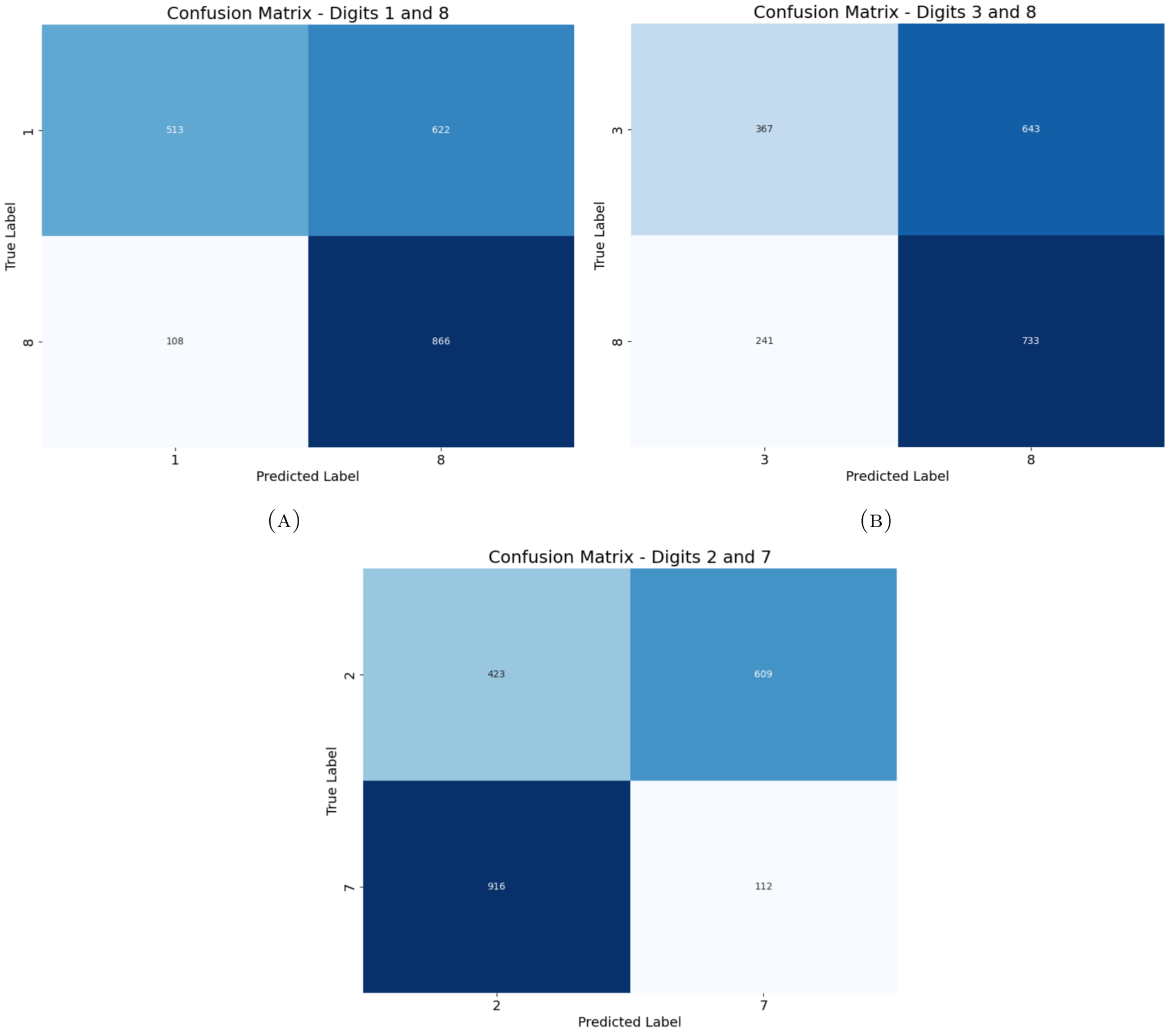
% variance	PC-modes
70	26
85	59
90	87
95	154

Table 1. PCA modes necessary for retention of different explained variances

Ridge classification of the subset of digits 1, 8 from the PC-projected training data performed with an accuracy of 61%, 55% for the subset of digits 2, 7, and 53% for digits 3, 8. For the same digit subsets of the PC-projected test set, performance was slightly higher than the training set for digits 1, 8 and 3, 8, but significantly worse for digits 2, 7 (26%) (Table 3). This suggests that digit likeness affects performance more than choice of classifier. As can be seen in the confusion matrix (Figure 6c), the classifier, more often than not, incorrectly predicted the digit in the 2, 7 pair. There is a similar behavior for digits 3 and 8, which also closely resemble each other (Figure 6b).

The values for the test mean-squared-errors (MSE) were larger than the training MSE's for each digit pair. For digits 2, 7 the test MSE was almost 3. However, expected MSE should be as low as possible, thus this is something that should be investigated further.

Although Ridge classifier achieves decent training accuracy (84.54% in 0.0667s), it is evident that KNN outperforms with the highest accuracy and shortest computational time (98.93% in 0.0160s) . LDA performs comparatively to Ridge (86.65%), but is almost 4 times as compuationally time-intensive (0.2278s) when fitting to the training data. 3



Confusion Matrix - Digits 1 and 8

True Label \ Predicted Label	1	8
1	513	622
8	108	866

(A)

Confusion Matrix - Digits 3 and 8

True Label \ Predicted Label	3	8
3	367	643
8	241	733

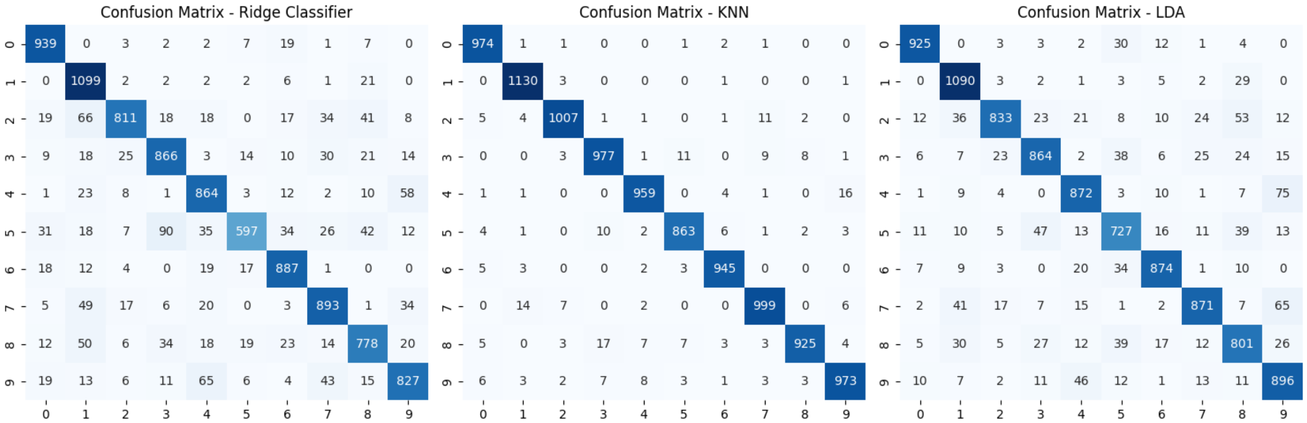
(B)

Confusion Matrix - Digits 2 and 7

True Label \ Predicted Label	2	7
2	423	609
7	916	112

(C)

Figure 6. Confusion matrices for each digit pair subset classification using Ridge Classifier



Confusion Matrix - Ridge Classifier

	0	1	2	3	4	5	6	7	8	9
0	939	0	3	2	2	7	19	1	7	0
1	0	1099	2	2	2	2	6	1	21	0
2	19	66	811	18	18	0	17	34	41	8
3	9	18	25	866	3	14	10	30	21	14
4	1	23	8	1	864	3	12	2	10	58
5	31	18	7	90	35	597	34	26	42	12
6	18	12	4	0	19	17	887	1	0	0
7	5	49	17	6	20	0	3	893	1	34
8	12	50	6	34	18	19	23	14	778	20
9	19	13	6	11	65	6	4	43	15	827

Confusion Matrix - KNN

	0	1	2	3	4	5	6	7	8	9
0	974	1	1	0	0	1	2	1	0	0
1	0	1130	3	0	0	0	1	0	0	1
2	5	4	1007	1	1	0	1	11	2	0
3	0	0	3	977	1	11	0	9	8	1
4	1	1	0	0	959	0	4	1	0	16
5	4	1	0	10	2	863	6	1	2	3
6	5	3	0	0	2	3	945	0	0	0
7	0	14	7	0	2	0	0	999	0	6
8	5	0	3	17	7	7	3	3	925	4
9	6	3	2	7	8	3	1	3	3	973

Confusion Matrix - LDA

	0	1	2	3	4	5	6	7	8	9
0	925	0	3	3	2	30	12	1	4	0
1	0	1090	3	2	1	3	5	2	29	0
2	12	36	833	23	21	8	10	24	53	12
3	6	7	23	864	2	38	6	25	24	15
4	1	9	4	0	872	3	10	1	7	75
5	11	10	5	47	13	727	16	11	39	13
6	7	9	3	0	20	34	874	1	10	0
7	2	41	17	7	15	1	2	871	7	65
8	5	30	5	27	12	39	17	12	801	26
9	10	7	2	11	46	12	1	13	11	896

Figure 7. Confusion matrices for each classifier method on entire MNIST dataset

digit subset	% training-accuracy	% test-accuracy	test MSE	training MSE	cross-validation
1,8	61.11	65.39	1.5558	1.3845	0.6020 +/- 0.0082
3,8	52.71	55.44	1.8915	1.7823	0.5113 +/- 0.0053
2,7	55.28	25.97	1.7888	2.9612	0.5317 +/- 0.0146

Table 2. Ridge classifier with k = 59 PC modes, performed on different subsets

classifier	% training-accuracy	% test-accuracy	time to train classifier (s)
Ridge ($\alpha = 1$)	84.54	85.61	0.0667
KNN (k = 3)	98.93	97.52	0.0160
LDA	86.65	87.53	0.2278

Table 3. Comparison of different classifiers on entire dataset

5. Summary and Conclusions

Principal Component Analysis was successfully implemented to reduce the dimensionality of the large MNIST dataset, demonstrating the most dominant features can be retained by using much fewer principal components. With just 59 PC modes, approximately 85% of the dataset's variance was reserved, allowing for a compressed and informative representation of the digit images.

Binary classification using Ridge regression showed the challenges of distinguishing similar digit pairs, particularly for digits like 2, 7, where test accuracy dropped to 26%. This emphasizes the limitations of linear classifiers in such cases. The multi-class classification comparison showed that K-Nearest Neighbors (KNN) significantly outperformed Ridge and LDA in both accuracy and computational time, achieving a test accuracy of 97.52% with minimal training time.

Overall, this study demonstrates the importance of dimensionality reduction for efficient data analysis and highlights how classifier selection influences performance. While Ridge regression proved useful for regularized linear classification, its struggles with visually similar digits suggest the need for more sophisticated methods. Fine-tuning hyper-parameters could provide even more robust results for digit classification tasks. Future work could expand on this by incorporating non-linear classifiers such as neural networks to further improve classification accuracy.

Acknowledgements

The author acknowledges the use of the Modified National Institute of Standards and Technology dataset derived from Yann Lecun. The author also appreciates the use of Large Language Models for code debugging resolution and further study on topics covered in this report.

References

[1] Principal component neural networks for modeling, prediction, and optimization of hot mix asphalt dynamics modulus - scientific figure on researchgate.

[2] P. Ghasemi, M. Aslani, D. Rollins, and R. Williams. Principal component neural networks for modeling, prediction, and optimization of hot mix asphalt dynamics modulus. *Infrastructures*, 4:53, 08 2019.

[3] J. Hunter, D. Dale, E. Firing, M. Droettboom, and et al. Using matplotlib.

[4] J. Kutz. *Methods for Integrating Dynamics of Complex Systems and Big Data.* Oxford, 2013.

[5] scikit learn. User guide - 2.5. decomposing signals in components (matrix factorization problems).