AMATH 482: HOMEWORK 3

NICHOLAS NUGRAHA

Applied Math Department, University of Washington, Seattle, WA
nickjn@uw.edu

Abstract. This report applies Principal Component Analysis (PCA) to classify hand-written digits. Using PCA, the dataset is reduced while also preserving key patterns in the data. Projecting the data into a different dimensional PCA space allows for classification via linear Ridge and k-Nearest Neighbors classifier methods. Results show that a the KNN classifier method with a k-value of 3 is better for classifying the digits with a cross-validation score of 97%, compared to the 85% the Ridge classifier produced.

1. Introduction and Overview

In this assignment, I work with data from the famous MNIST data set in order to train classifiers to distinguish images of handwritten digits. The data set is split into training and test sets where I will be using the training set to train my classifiers and the test set to validate them. The data itself is from Yann Lecun. We will be utilizing projection onto different dimensions in order to optimize the classification algorithm.

2. Theoretical Background

Since we are in the context of dimensionality reduction, there are two mathematical concepts we must first understand: Singular Value Decomposition (SVD) and Principal Component Analysis (PCA). With these tools, we can transform high-dimensional data like our digit data into a more compact and interpretable form. The data we have contain redundant features that make it difficult to interpret so reducing the dimensionality retains the most significant structures, essentially ignoring the redundant information.

SVD is a matrix factorization technique that decomposes any $m \times n$ matrix A into three matrices:

$$A = U\Sigma V^T$$

where U is an $m \times m$ orthogonal matrix containing left singular vectors, Σ is an $m \times n$ diagonal matrix with singular values arranged in descending order, and V^T is an $n \times n$ orthogonal matrix containing right singular values. The singular values in Σ indicate the significance of each corresponding dimension, with large values representing more important features in the data.

PCA is a statistical method used for transforming high-dimensional data into lower-dimensional space, preserving variance as much as possible. This method identifies the directions in the data that exhibit the greatest variability. These are known as principal components which capture the most significant patterns in the data. PCA is also often performed using SVD. Our dataset X is decomposed into $X = U\Sigma V^T$ where the columns of V correspond to the principal components. The number of principal components k we choose also has an effect on how much information was retained in the process, as well as how well classification tasks, like identifying the digit, can be performed.

Date: March 3, 2025.

Beyond this, we will be using two specific classification methods: the Ridge classifier and the k-Nearest Neighbors (KNN) classifier. The Ridge classifier is a linear model that extends ordinary least squares regression by using L2 regularization in order to prevent overfitting. It minimizes the sum of squared errors while penalizing large coefficient values. This method is useful for high-dimensional data and linearly separable problems because it provides a stable decision boundary while simultaneously reducing noise sensitivity.

The KNN classifier is a supervised nonparametric classifier that uses the distance to its k closest neighbors to predict the value of a new data point in a training set. The choice of k also affects the performance of the algorithm. Smaller k-values risk overfitting because the decision boundary closely following the training data. Large k-values make the decision boundary more generalized, thus making the model less likely to overfit because it considers a broader neighborhood instead of the one or two points you might get with a smaller k-value.

3. Algorithm Implementation and Development

Using `sklearn`'s implementation of PCA, we can easily break down our matrix into components that we can then analyze. In this PCA algorithm, the data is automatically centered, however we do have to transpose our matrix to abide with the row and column conventions of `sklearn`. Along with this, we use `numpy`'s tools to perform various mathematical operations and `matplotlib` to visualize our findings. We also use `sklearn`'s implementation of the linear Ridge classifier and KNN classifier, as well as the cross-validation method to test the accuracy of the two methods.

4. Computational Results

By performing PCA analysis of the digit images in the training set, we can plot the first 16 PC modes as 28 x 28 images.

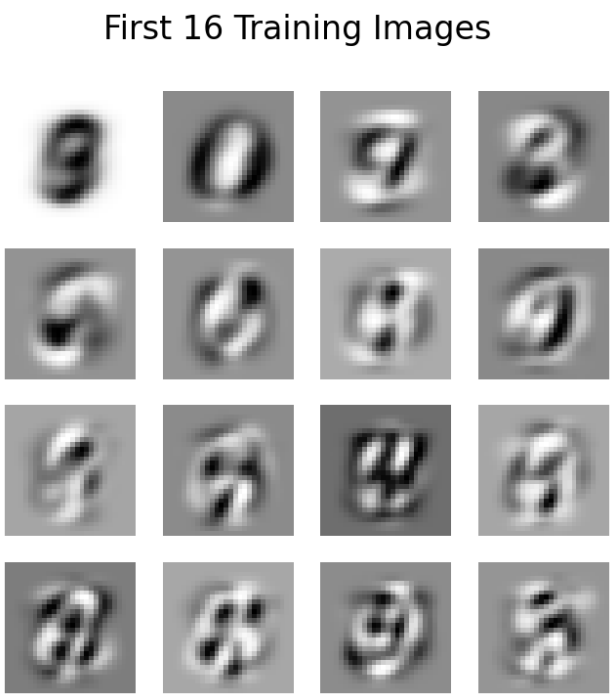

Figure 1. First 16 PC modes of the digit images in the training set

In Figure 1 we can see that as the number of modes increase, the images become more fragmented, indicating that finer details are being captured at the expense of more noise. This is why the first mode (top left image) is somewhat clearer compared to the others.

In order to see how much variance in the data is preserved as we increase the number of principal components, we can look at the cumulative energy which tells us how many components are required to retain a certain percentage of the original dataset's variance. In order to approximate 85% of the energy, we would need to use 42 PC modes.

By constructing a function that selects a subset of particular digits from all samples, we can see how the Ridge classifier performs when trying to distinguish between the two chosen digits using cross-validation

Digits Selected	**CV Score (accuracy)**	**Standard Deviation**
1, 8	0.9617	0.0038
3, 8	0.9567	0.0061
2, 7	0.9818	0.0027

TABLE 1. Cross-validation score and standard deviation for the selected digit pairs using a linear Ridge classifier

From Table 1, we can see that the Ridge classifier classified digits 3 and 8 with the least amount of accuracy. This makes sense because if we were to look at the digits 3 and 8, they are the most similar in terms of the structure of the digit. Because they are so similar, the classifier is more unsure of its prediction and it gets a lower accuracy

Now, instead of choosing a subset of digits, we will use all the digits (0-9) and perform the same classification procedure using both the Ridge and KNN classifiers.

Classifier	**CV Score (accuracy)**	**Standard Deviation**
Ridge	0.8488	0.0086
KNN	0.9701	0.0012

TABLE 2. Cross-validation score and standard deviation for all digits using a linear Ridge classifier and KNN classifier

Looking at Table 2, it can be concluded that the KNN classifier performs better with a 97% accuracy, while the Ridge classifier performs at 85% accuracy. This is because the MNIST dataset is highly non-linear, meaning the digits do not separate well with a simple linear decision boundary. It is better represented in the non-parametric KNN classifier which is much more flexible with its decision boundaries, making it more adaptable to this dataset.

The number of k-neighbors chosen was determined using cross-validation. After running the algorithm with different values of k, we found that $k = 3$ provided the best results for the KNN classifier.

5. Summary and Conclusions

In this assignment, PCA was used to reduce the dimensionality of the MNIST dataset. The transformed data was then classified using two different approaches: the linear Ridge classifier and the k-Nearest Neighbors classifier. The objective was to evaluate the effectiveness of PCA-based dimensionality reduction in combination with these classification methods.

Our results demonstrated that KNN significantly outperforms Ridge classifier in classifying the handwritten digits. The Ridge classifier achieved an accuracy of 85%, while KNN reached 97%, highlighting its effectiveness for this dataset. The higher accuracy of KNN can be attributed to its

ability to capture non-linear decision boundaries, whereas Ridge, being a linear classifier, struggled with digit classes that are not linearly separable.

Future work could explore alternative classification models such as SVM or neural networks to further improve classification accuracy.

Acknowledgements

The author is thankful to Professor Natalie Frank for useful lectures about PCA and classification methods. I also thank TA Rohin Gilman for assisting in office hours. I acknowledge that artificial intelligence was used to make edits to this document, fix minor bugs, and gain a better understanding of the concepts.

References

[1] T. N. Community. *NumPy Documentation*, 2025. Accessed: 2025-01-25.
[2] N. Frank. Amath 482: Computational methods for data analysis. Lecture notes, University of Washington, 2025.
[3] J. N. Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. OUP Oxford, 2013.
[4] N. Nugraha. Principal component analysis for robotic movement classification. Homework Report AMATH 482: Homework 2, Applied Math Department, University of Washington, Seattle, WA, February 2025.
[5] OpenAI. Chatgpt. `https://chat.openai.com/`, 2025.