AMATH 482/582: HOMEWORK 3

HAI VAN LE

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

Abstract. For this project, we are interested in training classifiers to distinguish images of handwritten digits from the famous MNIST data set which has been used extensively in training machine learning models. Here, Principle Component Analysis (PCA), Ridge, K-Nearest Neighbors (KNN) and Linear Discriminant Analysis (LDA) Classifiers are used to differentiate and identify handwritten digits.

1. Introduction and Overview

The MNIST[2] database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems in machine learning. The MNIST database contains 60,000 training images and 10,000 testing images. We downloaded the data set from the MNIST database which is split into training and test sets. We trained our classifiers using the training set while the test set is only used for validation/evaluation of our classifiers.
Using PCA on the training dataset, we were able to identify the main features to be used as the dimensions for the classifiers....

2. Theoretical Background

This project mostly deploys Principal Component Analysis (PCA) [1] and classifiers such as Ridge, K-Nearest Neighbors (KNN) and Linear Discriminant Analysis (LDA) Classifiers, all of which are explained briefly below.

2.1. **Principal Component Analysis.** This project represents a key application of the SVD, or alternatively a variant of Principal Component Analysis (PCA). PCA is a dimensionality reduction technique employed to discover patterns in high-dimensional data. It is used to transform the initial variables into a new set of variables called principal components, which are linear combinations of the initial variables. The order of the principal components is that the first component tells us the maximum variance in the data, the second component tells us the maximum remaining variance, etc. With these principal components, the dimension of the dataset can be reduced significantly yet still preserve most of its important information.
We can measure the variance of the data in each axis that is centralized with a mean of 0 by:
$C_x = cov(x) \sim \frac{1}{N-1} XX^T$: approximation of covariance matrix
$C_x \sim \frac{1}{N-1} U\Sigma V^T \cdot V\Sigma U^T = \frac{1}{N-1} U\Sigma^2 U^T$
PCA modes are eigenvectors of C_x and G^2 are eigenvectors of C_x. Thus, PCA modes indicate the optimal directions to represent variance of the data.

Date: February 22, 2024.

2.2. **Classifiers.**

- Ridge Classifier first converts the target values into -1, 1 and then treats the problem as a regression task (multi-output regression in the multiclass case). The classifier is built upon Ridge regression which is a statistical regularization technique, correcting for overfitting on training data in machine learning models.
- The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity (which explains its name - nearest neighbor) to make classifications or predictions about the grouping of an individual data point.
- Linear discriminant analysis (LDA) is an approach used in supervised machine learning to solve multi-class classification problems. It separates multiple classes with multiple features through data dimensionality reduction in order to optimize the model.

3. Algorithm Implementation and Development

This project deploys different library interfaces such as *NumPy* to load the data, *matplotlib.pyplot* - a plotting library for the Python programming language and its numerical mathematics extension NumPy. Most importantly, it uses *sklearn* for PCA, RidgeClassififer, k-NN, and LDA.
Specifically, the data is provided in two sets: training and testing. We first investigated the dimensionality of X_{train} and X_{test} using PCA function and plotted the first 16 modes as 28x28 images. Then, we determined the number of modes to approximate 85% of the energy using the explain-variance-ratio attribute of PCA. For task 3, we defined a function that selects a subset of particular digits (all samples of them) from X_{train}, y_{train}, X_{test}, y_test and returns the subset as new matrices $X_{subtrain}$, $y_{subtrain}$, $X_{subtest}$, and $y_{subtest}$. Then, I made use of the provided Jupyter notebooks on Ridge, KNN and LDA Classifiers. While applying the classifiers, I remembered to use k = 59 as the n-components for PCA and transform and/or fit-transform the data.

4. Computational Results

The first 16 PCA modes are shown in 1. The cumulative energy of singular values is plotted in 2. We also inspected several approximated digit images reconstructed from k truncated PC modes and plotted them as in 3 to make sure that the image reconstruction using truncated modes is reasonable. For task 2, we also found that the number of PC modes needed to approximate 85% of the energy is 59.
For task 4 and 5, we found that generally, the Ridge classifier performs well on all digit pairs, but the performance varies depending on the pair. The [1,8] pair exhibits the best performance with high accuracy and low variability in cross-validation scores. One explanation might be that digits 1 and 8 have distinct shapes, making them relatively easy for the classifier to distinguish. This leads to high accuracy in both training and testing. The [3,8] pair shows lower performance and potential overfitting, while the [2,7] pair also performs well but with a slightly higher risk of overfitting. The reason is that, unlike 1 and 8, digits 3 and 8 may have some visual similarities ("half" of 8 is 3), making it more challenging for the classifier to distinguish between them accurately. This could contribute to the higher likelihood of overfitting as the model tries to capture subtle differences between these digits. Similarly, though digits 2 and 7 have relatively distinct shapes, they may have some complexity in certain handwritten variations, leading to a slightly higher risk of overfitting compared to the [1,8] pair. In particular,

- Digit pair [1,8]:
  - Training Score: 0.9652981815294212
  - Testing Score: 0.979611190137506
  - Ridge Cross-validation mean: 0.9635513859560115
  - Cross-validation standard deviation: 0.002550624887080633
- Digit pair [3,8]:

  - Training Score: 0.9544316474712068
  - Testing Score: 0.8155241935483871
  - Ridge Cross-validation mean: 0.9591894918731889
  - Ridge Cross-validation standard deviation: 0.006079529909477922
- Digit pair [2,7]:
  - Training Score: 0.9799558209932095
  - Testing Score: 0.9160194174757281
  - Ridge Cross-validation mean: 0.9811014160968476
  - Cross-validation standard deviation: 0.0023612708353201242

For task 6, I found that the Ridge classifier shows moderate performance with similar scores between training and testing. The cross-validation mean is also consistent with the training score, indicating stable performance across different subsets of the training data. The standard deviation of cross-validation scores suggests moderate variability. The LDA classifier shows moderate to good performance with slightly higher training than testing scores. The cross-validation mean is consistent with the training score, indicating stable performance across different subsets of the training data. The standard deviation of cross-validation scores suggests moderate variability. The best performer is the KNN classifier with high training and testing scores. The cross-validation mean is very close to the training score, indicating consistent performance across different subsets of the training data. Its very small standard deviation suggests low variability in cross-validation scores, highlighting the stability of the model. In particular,

- Ridge Classifier:
  - Training Score: 0.8440166666666666
  - Testing Score: 0.8565
  - Cross-validation mean: 0.8440333333333333
  - Cross-validation standard deviation: 0.009741748645221095
- K-Nearest Neighbors Classifier:
  - Training Score: 0.9828666666666667
  - Testing Score: 0.9618
  - Cross-validation mean: 0.9753999999999
  - Cross-validation standard deviation: 0.0011406625754845651
- LDA Classifier:
  - Training Score: 0.8666666666666667
  - Testing Score: 0.8751
  - Cross-validation mean: 0.8655333333333333
  - Cross-validation standard deviation: 0.008728083663923285

5. Summary and Conclusions

This project attempted to build digit classifier to differentiate images of handwritten digits from the MNIST dataset. This project made use of PCA to examine the dimensionality of the training dataset so that we can retain major features and reduce dimension. Then, it calculated the cross-validation mean and standard deviation for each classifier: Ridge, KNN and LDA using 5-fold cross-validation, which give us a better understanding of the stability and performance of each classifier during training.

Acknowledgements

The author is thankful to Prof. Eli for useful discussions about the SVD algorithm and Python libraries. She is also thankful to the TAs who answered all questions regarding class materials and resources.

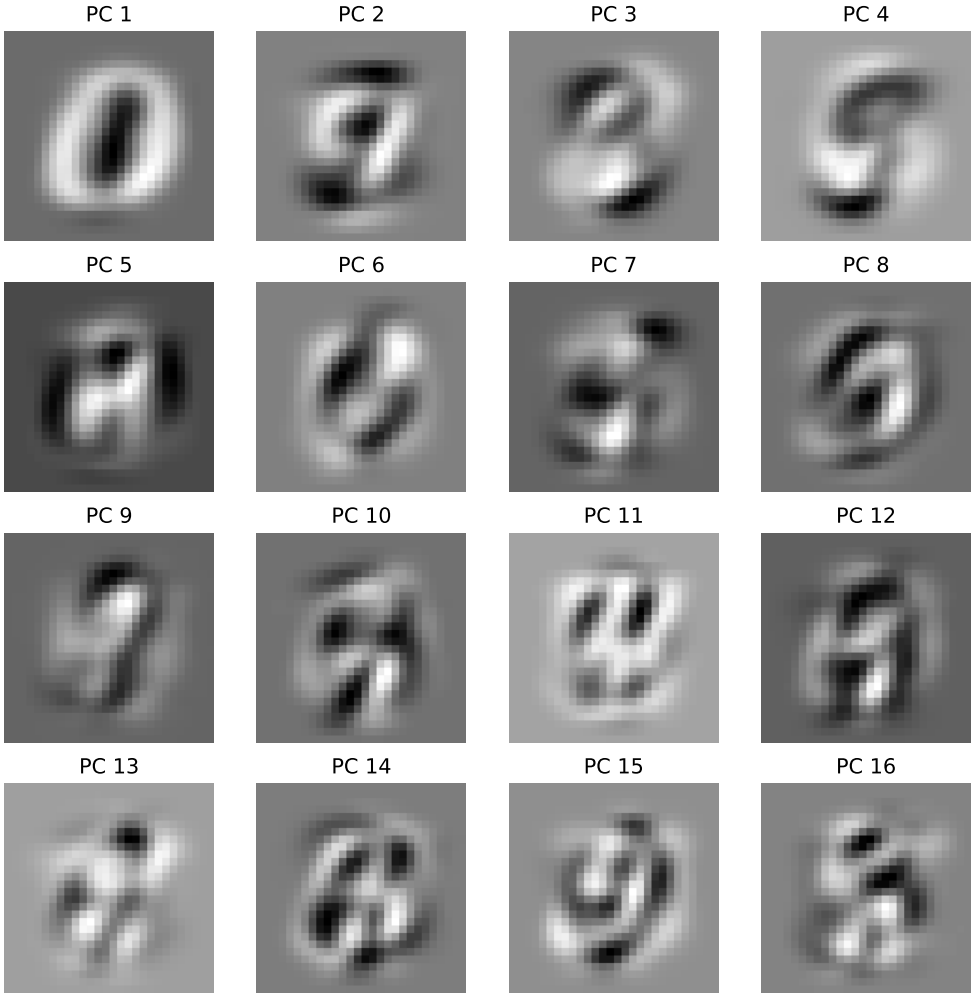

FIGURE 1. First 16 PCA modes

REFERENCES

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

[2] Y. LeCun, C. Cortes, and C. Burges. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

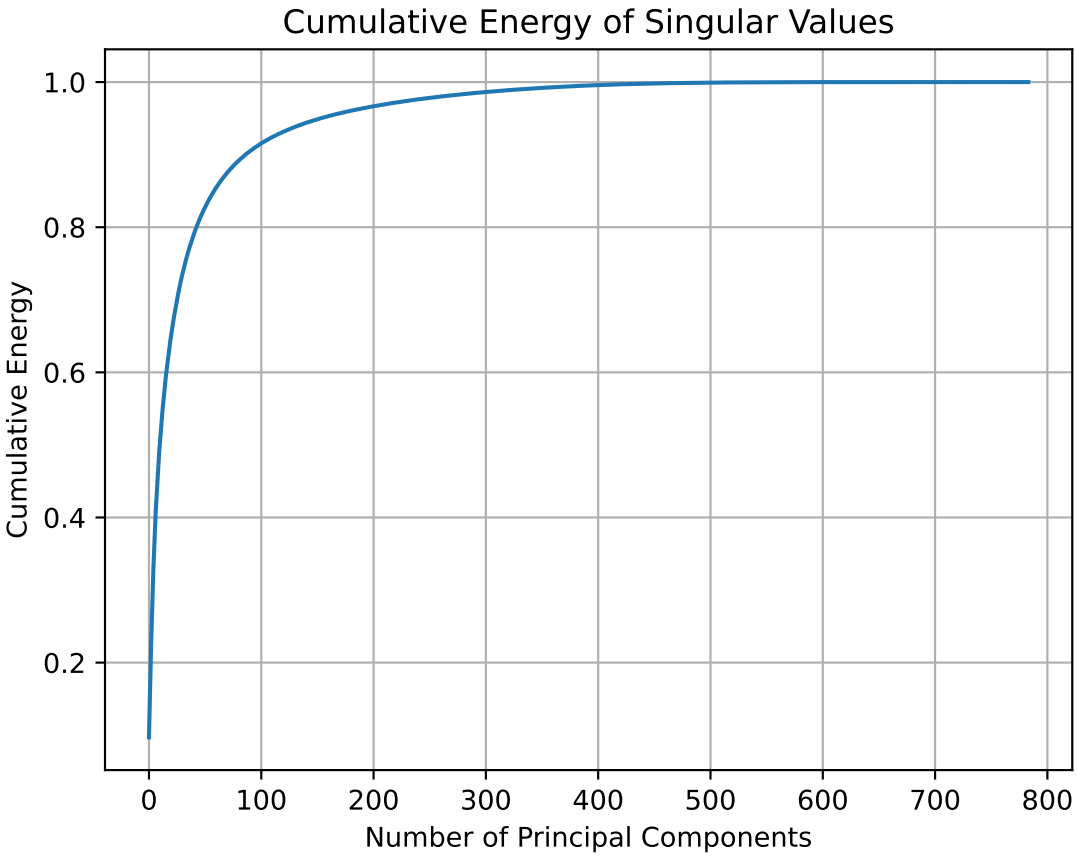

FIGURE 2. Cumulative energy of singular values

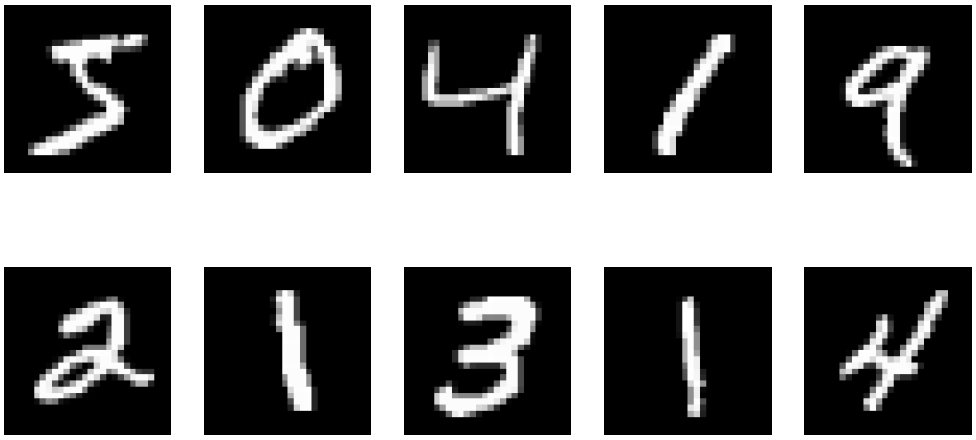

FIGURE 3. Plot reconstructed images