MNIST DIGIT CLASSIFICATION

SAMIP KARKI

Applied Mathematics, University of Washington, Seattle, WA
karki1@uw.edu

Abstract. Classifying the MNIST dataset is a canonical problem in machine learning. In this work, I investigate different classifying algorithms. I see that ridge regression is able to do binary classification very well with various pairs of digits from the MIST dataset. In addition, I test ridge regression, linear discriminant analysis, and K-nearest neighbors in multi-class classification and see that K-nearest neighbors was the best classifier for this dataset.

1. Introduction and Overview

Classification of the MNIST dataset of 28×28 pixel digits is canonical benchmark task used in machine learning to evaluate the performance of different classification methods. In this project, I will use the MNIST dataset to contrast the performance of methods such as ridge regression, linear discriminant analysis, and K-nearest neighbors.

2. Theoretical Background

Here, I will give a short introduction to some of the data methods used in this work.

Dimension reduction via SVD and principal component analysis is used in this work. Consider a matrix A which represents a dataset. The SVD of A is given by $A = U\Sigma V^T$. The columns of U

Date: September 2, 2024.

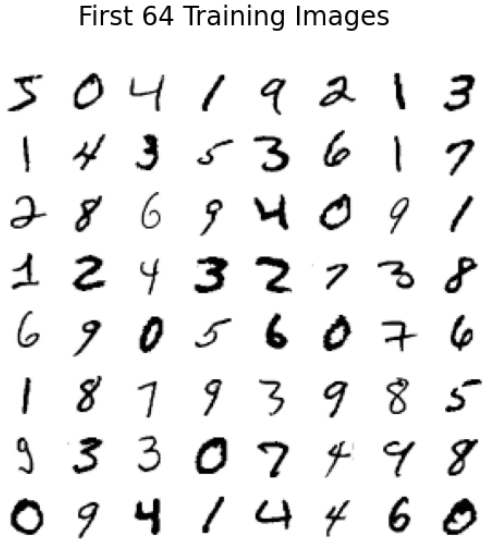

Figure 1. Example of handwritten digits in the MNIST dataset

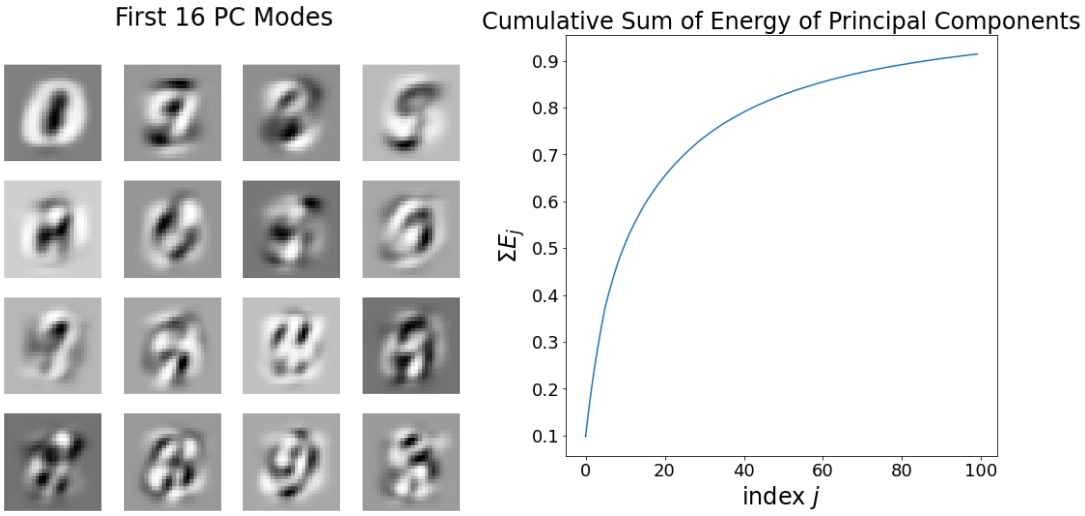

FIGURE 2. (**Left**) First 16 PC modes of the MNIST training data set. (**Right**) Cumulative sum of the energy of the the PC modes.

represent the principal component (PC) modes of the data. Dimension reduction can be done by projecting a dataset onto the first k PC modes instead of using all the features of a given dataset.

A classifier is some model which maps an input vector $\vec{x}$ to a label y. In this work, I will be using three different models: ridge regression, linear discriminant analysis, and K-nearest neighbors. Linear ridge regression is given by

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \tag{1}$$

where $\beta = (A^T A + \lambda I)^{-}1 A^T y$, for some hyper parameter λ.

Linear discriminant analysis (LDA) is a classification method that finds a linear combination of features that best separates two or more classes. LDA seeks to maximize the ratio between-class variance to within-class variance, providing the best discrimination between the classes. The linear discriminant function is given by

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{2}$$

where $\mathbf{S}_W$ is within-class scatter matrix, and $\mathbf{m}_1$ and $\mathbf{m}_1$ are the mean vectors of the two classes. This projection maximizes the separation between the projected class means.

k-nearest neighbors is a classification algorithm that assigns a class to a sample based on the majority class among its k nearest neighbors in the feature space. The distance between points is usually measured using Euclidean distance.

3. Algorithm Implementation and Development

All the computation in this project was done using python with use of the canonical packages of `numpy, matplotlib, sklearn, and scipy`. For principal component analysis, the SVD was computed using `scipy`. Ridge regression, K-nearest neighbors, and linear discriminant analysis classifiers were implemented using the `sklearn` package.

4. Computational Results

The MNIST datasets used includes 60,000 digits in the training dataset and 10,000 testing dataset. The columns $\hat{X}_{train}$ and $\hat{X}_{test}$ are the flattened vectors of length 784 for each 28×28 pixel

digit in the training and testing datasets, respectively. Principle component analysis is done using the $\hat{X}_{train}$. X_{train} and X_{test} represent $\hat{X}_{train}$ and $\hat{X}_{test}$ projected onto the PC modes of $\hat{X}_{train}$. The first 16 PC modes are plotted on the left of figure2. The energy of the n-th PC mode is given by:

$$E_n = \frac{\sigma_n^2}{\sum_{i=1}^{784} \sigma_i^2}.$$

where σ_i is the i-th singular value. On the right of figure2, the cumulative sum of the first j PC modes is plotted. It can be seen that the first 59 PC modes account for 85% of the total energy. From here on, analysis of different classifiers will be done with X_{train} and X_{test} truncated to the first 59 features.

Let $X_{train(i,j)}$ and $X_{test(i,j)}$ be subsets of the X_{train} and X_{test}, respectively, which only include the digits i and j. I investigate how the Ridge classifier performs at binary classification of different pairs of digits: 1 and 3, 3 and 8, 2 and 7. For each pair of digits (i, j), the Ridge regression model is trained on only $X_{train(i,j)}$ and its accuracy in labeling $X_{test(i,j)}$ is reported along with 5-fold cross validation score. It can be seen in table1 that for the three pairs of digits chosen, the ridge regression classifier was adequate for the binary classification problem, producing very high accuracy when tested on the testing dataset and in cross validation.

Digits	Accuracy on Testing Set	5-Fold Cross Validation
1 and 3	0.9801	0.9643 ± 0.0027
3 and 8	0.9642	0.9588 ± 0.0061
2 and 7	0.9748	0.9797 ± 0.0014

TABLE 1. Binary Ridge Classification on various pairs of digits. The $\pm$ represents the standard deviation the cross validation.

In addition to binary classification, I compare how different classifiers perform at multi-class classification. I use un-truncated dataset X_{train} to train the following models: ridge regression, k-nearest neighbors, and linear discriminant analysis and see how they perform at classifying X_{test}. Multi-class classification is summarized in table2. It can be seen that out of the three classifier algorithms, k-nearest neighbors out performs linear ridge regression and linear discriminant analysis, both of which had incorrect labels about four to five times more often than k-nearest neighbors. Interestingly, although ridge regression was able to distinguish between binary digits with very high accuracy, the accuracy was much lower in multi-class classification, which suggests that multi-class classification is a much harder problem then binary classification.

Classifier	Accuracy on Testing Set
Ridge Regression	0.8556
k-nearest neighbors	0.9730
Linear Discriminant Analysis	0.8747

TABLE 2. Multi-class classification

5. Summary and Conclusions

This project was an instructive survey of some different algorithms used in machine learning. I find that it is interesting how easy these algorithms can be used, even without rigorous knowledge of how the algorithms actually work. Packages like `sklearn` make it very eary to incorporate them into your own work.