Amath482HW3

Tong Wu

March 2025

1 Introduction

In this assignment, we used different methodologies to explore the MNIST digit classification, such as Principal Component Analysis. By applying PCA, we reduce the number of features while retaining 85 percent of variance. Three other machine learning classifiers (Ridge Classifier, K-Nearest Neighbors Classifier, and Support Vector Classifier) are used to distinguish images of handwritten digits from the MNIST data set.

In this report, the data set consists of :
Xtrain - training data, the feature matrix that has the shape (784, 60000), where each column represents the digit image of 784 pixel values.
ytrain - training label, a vector of shape (60000,), corresponding digit labels of 0-9 for each training image.
Xtest - a matrix that has the shape (784, 10000), with 10000 test samples.
ytest - testing labels, a vector of shape (10000,), containing the true labels for Xtest data.

2 Theoretical Background

2.1 Principal Component Analysis
Principal Component Analysis is one of the techniques that can take high-dimensional data and present it in a lower-dimensional form without losing too much information. By transforming correlated variables into linearly uncorrelated components and diagonalizing the covariance matrix using SVD, PCA isolates the most significant modes of variation[1].

2.2 Ridge Regression
Ridge regression is a regularized version of linear regression.
The standard linear regression is represented by :

$$y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

However, to solve the overfitting problems when the predictors are highly correlated, we add an L2 penalty to the cost function. We then have:

$$\min_w \|Xw - y\|_2^2 + \alpha \|w\|_2^2$$

[2] where

$$\hat{\beta}_{ridge} = (X'X + \lambda I_p)^{-1} X'Y$$

2.3 K-Nearest Neighbors Classification
K - Nearest Neighbors is a supervised machine learning algorithm that makes predictions based on the proximity of the data points. The KNN assumes that similar data points are close together and can be used for both classification and regression. In this classification, it assigns a new data point for the majority class of its nearest neighbors. [3]

2.4 SVM
Support Vector Machines is a supervised learning algorithm that is used for classification, regression tasks, and outlier detection. It works for finding an optimal hyperplane that separates data points into different classes with maximum margin making it highly effective in high-dimensional spaces, and it is very memory efficient [4].

3 Algorithm Implementation and Development

- (Task 1)
- Loaded and Transposed Data: convert the Xtrain and Xtest from shape (784, 60000) and (784, 10000) to (60000, 784) and (10000, 784).
- Initialized PCA model and fitted it into Xtrain.
- Extracted the first 16 principal components (PC modes) which represent the significant patterns in the data
- Display the plot the images for the top 16 modes
- (Task 2)
- Used np.cumsum(pca.explained_variance_ratio_) to calculate the cumulative variance explained by PCA components.
- Found the smallest number of principal component (k_85)
- Initialized PCA with the smallest number of the principal component (k_85) and fitted it to Xtrain

- Transformed Xtrain to Xtrain_reduced with shape (60000, K_85)
- Reconstructed Xtrain_ recon by applying the inverse PCA transform.
- Used plot_digits( ) to display the first 64 reconstructed images.
- (Task3)
- Created a function named digit that isolates only the samples that belong to a particular digit
- Inside, created a boolean mask for the training labels to identify which rows correspond to the desired digits
- Used this mask to slice out only the relevant rows of X train and Y train data
- Repeated the same process for x test and y test
- (Task 4 and 5)
- Created a function that uses a linear classifier (RidgeClassifier in sklearn.linear_model) to distinguish between digits such as 1 and 8
- Projected onto PCA, use the trained PCA to transform the subdataset. (PCA) is used here to reduce the dimensionality of the dataset while preserving 85% of the variance.
- Trained a ridge classifier on the training subsets
- Launched a cross validation, which reduces the overfitting problem and gives a more reliable accuracy estimate
- After choosing the models, refit on the entire training subset and measure the accuracy
- (Task 6)
- Initialized the Ridge Classifier and KNN Classifier from KNeighborsClassifier and RidgeClassifier
- Performed cross validation on the entire training set
- Refitted on the entire training data
- Computed the predictions on the entire test and test accuarcy
- (Task 7)
- Initialize SVM classifier from sklearn.svm
- Performed CV by using cross_val_score and refit
- Computed predictions and test accuracy

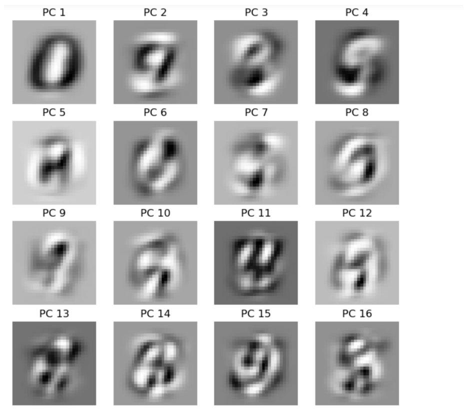

Figure 1: First 16 PC modes

Figure 2: First 64 images using K= 59 PC modes

4 Computational Results

1. By capturing the first 16 PC modes, the result is shown in Figure 1.

2. By initializing PCA with the smallest number of the principal component (k_85) and fitting it to Xtrain, the result is shown in Figure 2.

3. By putting the digit pairs into the two_digit function, the results are shown below:

pairs — mean ± sd —test accuracy
1 vs 8 — 0.9642 ± 0.0025 — 0.9801
3 vs 8 — 0.9589 ± 0.0062 — 0.9642
2 vs 7 — 0.9799 ± 0.0019 — 0.9743

4. By performing both Ridge and KNN classifiers, the results are shown below:
Ridge Classifier test accuracy: 0.8560
Knn Classifier test accuracy: 0.9758
Ridge multi-class CV accuracy: 0.8439 ± 0.0096
KNN multi-class CV accuracy: 0.9747 ± 0.0013

5. By performing SVM classifiers, the results are shown below:
SVC Classifier test accuracy: 0.9841
SVC multi-class CV accuracy: 0.9812 ± 0.0016

5 Summary and Conclusions

In this report, by applying Principal Component Analysis and other three machine learning classifiers, we evaluated the efficiency and accuracy for these classifiers in recognizing different digits from the MNIST dataset. We compared their performance based on cross-validation scores and test accuracy for each model to compare which is the best model we can use.

References

[1] Shalizi, C. R. (2012). *Advanced data analysis from an elementary point of view* (Lecture 18). Retrieved from https://www.stat.cmu.edu/ cshalizi/uADA/12/lectures/ch18.pdf

[2] ScikitLearn. (2011). *Scikit-learn: Machine Learning in Python.* Retrieved from https://scikit-learn.org/stable/modules/linear$_model.htmlridge-regression$

[3] Elastic. (n.d.). *What is k-nearest neighbors (kNN)?* Retrieved from https://www.elastic.co/what-is/knn

[4] ScikitLearn. (n.d.). *Support Vector Machines.* Retrieved from https://scikit-learn.org/stable/modules/svm.html