CLASSIFYING DIGITS WITH PCA AND RIDGE REGRESSION

XUECHENG LIU

Applied Mathematics Department, University of Washington, Seattle, WA
xl0306@uw.edu

Abstract. Principal component analysis (PCA) is a useful technique for visualizing high dimensional data by reducing it to human perceivable dimensions. By applying PCA to MNIST data set, we could significantly reduce the measurements needed for each digit and achieve a relatively low mean square error in both training and testing process.

1. Introduction and Overview

The goal of this project is to developing a classifier using ridge regression to classify each digit in MNIST data set. Since each digit is a 16×16 matrix, the dimension of each digit is 256 after we flatten it, which is computationally expensive for machine learning tasks. Here, we first apply PCA to investigate the dimension of the training set. Later we will use Frobenius norm as the criteria to decide the amount of principal components to keep. Finally, we project the original data to a lower dimension and conduct ridge regression for digit classification and ends in a decent mean square error.

2. Theoretical Background

According to [Brunton and Kutz, 2019], PCA was conducted on flattened image set using Singular Value Decomposition (SVD). With SVD, the data matrix $\mathbf{X}$ can be expressed into a product of $\mathbf{X} = \mathbf{U\Sigma V^T}$, where $\mathbf{U}$ and $\mathbf{V}$ are left singular matrix and right singular matrix respectively. $\mathbf{\Sigma}$ is a diagonal matrix of singular values. A transformed data matrix $\mathbf{T} = \mathbf{U\Sigma V^T V}$ can be viewed as the projection of the original data into orthogonal basis.V is called the loadings and each column is a principal component listed in order. A dimension-reduced data can be obtained by $\mathbf{T_r} = \mathbf{XV_r}$ where $\mathbf{V_r}$ is the first r columns of $\mathbf{V}$.

The Frobenius norm is defined as

$$\|B\|_F^2 = \sum_{j=1}^{min(m,n)} \sigma_j^2$$

where σ_j is the sigular value of B. We could take the cumulative sum of each singular value to check how many principal component we need to keep for a certain amount of variation in the data set.

According to [Brunton and Kutz, 2019], in ridge regression, $\hat{\beta}$ is defined as

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^J}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|A\beta - Y\|^2 - \frac{\lambda}{2} \|\beta\|_p^p$$

λ is called the penalty parameter and $\hat{\beta}$ can be solved as

$$\hat{\beta} = (A^T A + \sigma^2 \lambda I)^{-1} A^T Y$$

Date: February 12, 2022.

Mean square error (MSE) of the classifier is defined as

$$MSE = \frac{1}{N} \times \|X\beta \hat{-} Y\|_2^2$$

which could be used as a measurement of the performance of our classifier.

3. Algorithm Implementation and Development

The core packages used in this project is **Numpy** and **Sklearn**. The following algorithm illustrates how to find the principal component of the data matrix.

Algorithm 1 Principal Component Finding Algorithm

import data
$X \leftarrow data$
$U, \Sigma, V^T = SVD(X)$
return V

The next algorithm illustrates how data is reduced in dimension and the training of ridge regression.

Algorithm 2 Dimension Reduction and Ridge Training

$T_r \leftarrow XV_r$
$\hat{\beta} \leftarrow (T_r^T T_r + \sigma^2 \lambda I)^{-1} T_r^T Y$
$\hat{Y} \leftarrow T_r \hat{\beta}$
$\hat{Y}[i] = -1 \; if \; \hat{Y}[i] < 0 \; else \; 1$
$MSE \leftarrow \frac{1}{N} \times \|\hat{Y} - Y\|_2^2$

4. Computational Results

Figure 1 shows how the dimension of X_train is investigated using PCA. The left figure shows the latter principal components explain very few variation in the data set. The right figure illustrates the percentage of variation explained by adding more and more principal components. It is obvious that we could truncate the measurements in our data set by preserving most of the variations.

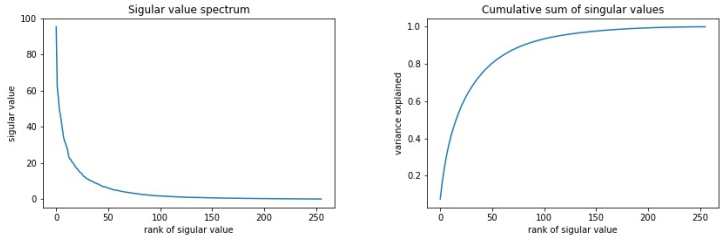

Figure 1. Investigation of Principal Components

Figure 2 illustrates the first 16 PCA modes in loading matrix **V**. Each sub-figure is plotted by reshaping one column of **V**,which is an eigenvector.

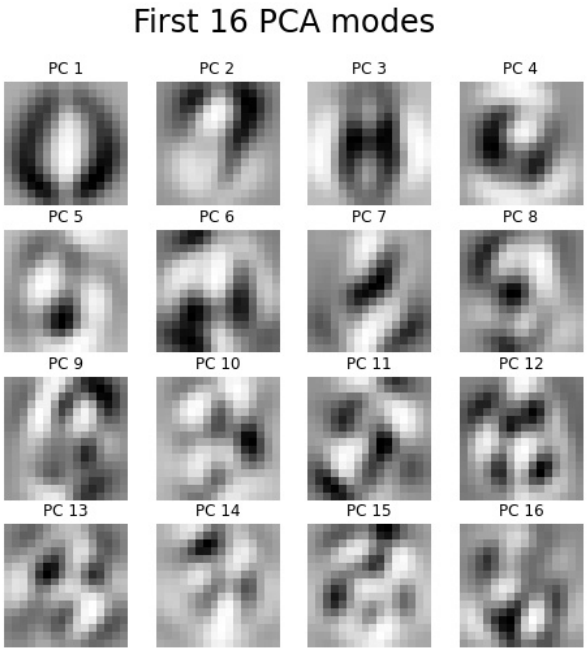

FIGURE 2. First 16 PCA modes

Figure 3 illustrates the number of PCA modes to keep to approximate the variance in Frobenius norm. We found we need to use 3,7,14 PCA modes to approximate 60%, 70% and 80% in the Frobenius norm.

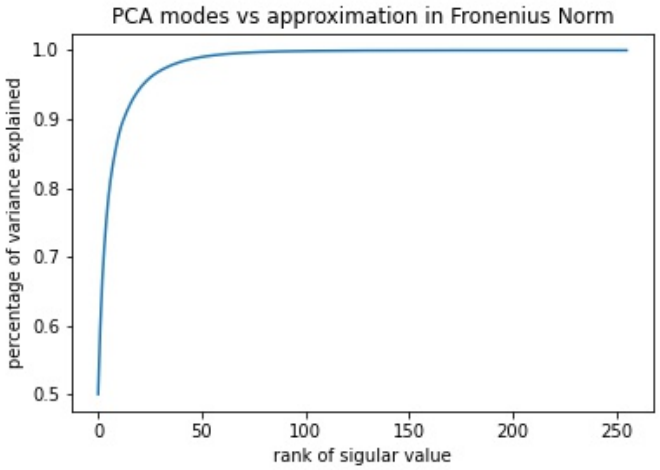

FIGURE 3. PCA modes vs approximation in Fronenius Norm

We can also reconstruct the figure using low rank approximation with different numbers of PCA modes used. Figure 4 illustrates the how recognizable the image is using 14 and all modes. It is obvious that using only 14 PCA as a low rank approximation gives highly recognizable figures.

FIGURE 4. Reconstruction of data using less principal components

Applying Ridge Regression to the data set of digit 1 and digit 8, the best choice of alpha via cross validation is 9.71 for all alpha in $2^n, n \in [-4...20]$. Figure 5 shows how MSE changes for different values of α in both training and testing set. We could conclude that a large α makes the model under fit and inflates the MSE.

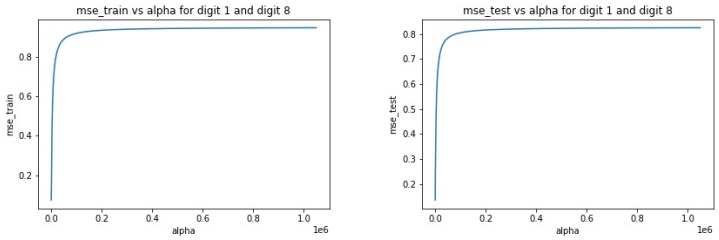

FIGURE 5. MSE vs α for digit 1,8 in training and testing set

We could repeat the same experiment on digits(3,8) and digits(2,7) and the MSE vs α plots follow the same trend. The best α for digits(3,8) is 14.72 and that for digits(2,7) is 7.5. In addition, MSE of testing set for digits(3,8) is 0.26, which is much higher than digits(1,8) where MSE = 0.0826 and digits(2,7) where MSE = 0.1325.

5. Summary and Conclusions

In summary, PCA is a powerful tool in dimension reduction as well as preserving the variations to speed up machine learning tasks. One thing worth further investigating is to find the reason why MSE of testing set for digits(3,8) is much higher than that of digits(1,8) and digits(2,7). One reasonable approach is to visualize their position in the PCA space, but due to time limit this work is not conducted.

Acknowledgements

The author is thankful to Prof. Bamdad Hosseini and Teaching Assistant, Katherine Grace Owens for useful discussions and helper visualizing function about the problem.

References

[Brunton and Kutz, 2019] Brunton, S. and Kutz, J. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.