Machine Learning Assignment 4 Report

110062219

1 Problems Encountered

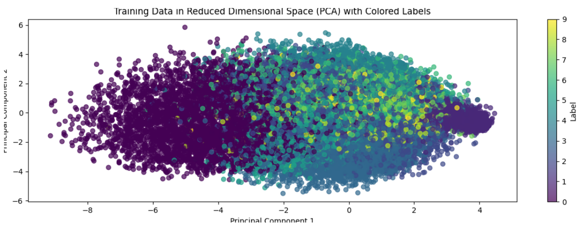
In addition to the problems that have been discussed on eeclass, I stumbled on the problem that for sparse PCA, the given definition of *sum of squared difference (SSD)* $\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{x_{proj}}_i)^2$ seemed awful since it's clear that $\mathbf{x}$, $\mathbf{x_{proj}}$ must have different shape — the former would be $(n, 784)$ and the latter would be (n, k). Other students and I coincidentally transformed $\mathbf{x_{proj}}$ back before computing *SSD*. However, though we obtained results identical to expected output, it turned out that the convergence didn't achieved within 1000 iterations. So I would like to doubt whether they have any thing to do with?

Meanwhile, I also have reservations about the iterative method we're asked to realize sparse PCA. I lack confidence in the effect of only applying soft threshold and normalizing during each iteration. I would appreciate if you could provide some reference since I could hardly find any hint or clue.

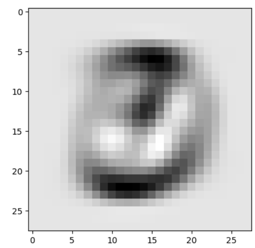
2 Implementation of Basic PCA

First, the covariance matrix Σ could be obtained by $\frac{1}{n}{X_{cent}}^T X_{cent}$. Then, we should compute the eigenvalues and eigenvectors to Σ, sort them in descending order of eigenvalues and cast them to real floats. Finally, we could pick first k eigenvectors as our principal components.

3 Eigenvectors and Reconstructed Image



(a) Eigenvectors



(b) Reconstructed Image

Figure 1: Eigenvectors and Reconstructed Image – Training Data

4 Distribution of Eigenvalues

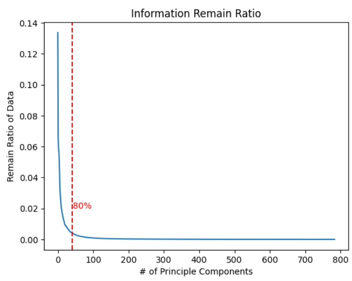


Figure 2: Distribution of Eigenvalues

5 Implementation of Advanced Part

For advanced part, I normalized the data based on concatenation of training and testing data and randomly shuffled them. I adopted ordinary PCA and took $k = 256$.