Applications of Principal Component Analysis

Zhaokai Huang

Part 1 - Visualisation

In this part, I use basic implementation of PCA (pca1.m), which is the function of PCA derived from the co-variance matrix and then I apply it to the IRIS data in order to obtain results by performing proposed requirements. The results are as followings:

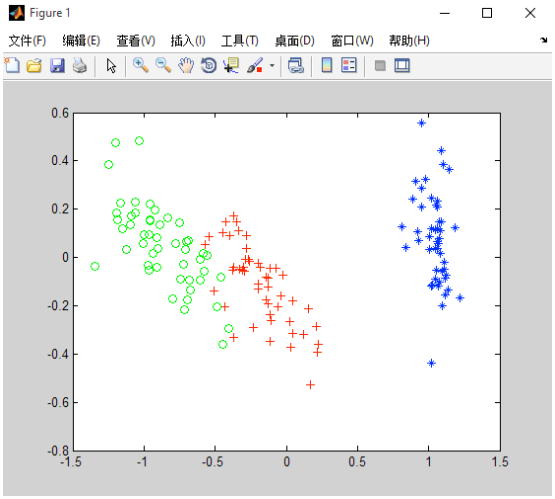

Figure 1: Display of results in PC_1-PC_2

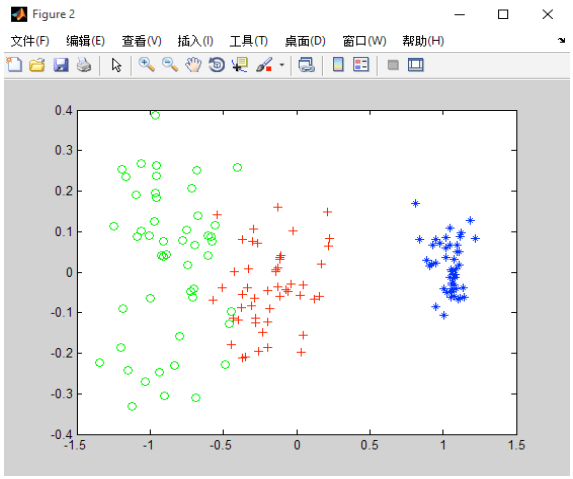

Figure 2: Display of results in PC_1-PC_3

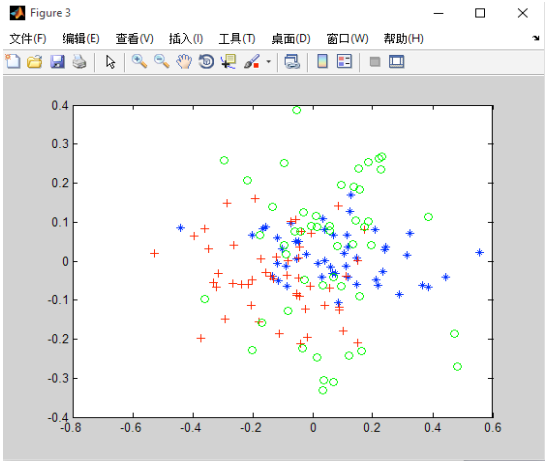

Figure 3: Display of results in PC_2-PC_3

With respects to the overview distribution of these three images, the data in PC_1-PC_2 subspace is relatively more organized or regular than that in PC_1-PC_3, and the data in PC_1-PC_3 is also relatively more organized or regular than that in PC_2-PC_3, which means the rank of the performance of subspace are PC_1-PC_2, PC_1-PC_3 PC_2-PC_3 (descending). The conclusion is drew from the above observation that PC_1 is the most important principal component in the IRIS data set and PC_2 is the second one, which can be explained by the fact PC_1 has largest corresponding eigenvalue 0.6608 and PC_2 has second largest corresponding eigenvalue 0.0368 (i.e., eigenvalue means variance here). From other perspectives, the data belong to

class 2 and 3 have similar performance due to their characteristics, which can be seen from the image that data points belong to class 2 and 3 are very close to each other compared with class 1 in PC_1-PC_2 and PC_1-PC_3. Besides, the data points in PC_2-PC_3 are so disordered to some extent highlight the importance of PC_1.

Part 2 - Image compression

In this part, I design the encoding system in Dual Algorithm which is divided into five processes. At first, make data centralization by subtracting off the mean vector for each dimension. Secondly, construct the matrix Y and apply the SVD to it. Theses two steps are achieved together by function pca2. Thirdly, select first M columns of V by proportion of variance (PoV) when it is larger than 90% to form a project matrix. Then I encode data point (here is test data) by U_M to get M-dimensional vector which is the low-dimensional representation of data point. Finally, I reconstruct the data point and calculate the test error.

And below are specific answers to the questions:

1. PCA2 would be more appropriate for this application as the given data now is 874*300 now (i.e., 847 and 300 are the numbers of dimension and instance respectively). To illustrate it more, the detailed explanation is that SVD allows us to deal with high dimensional data without using co-variance matrix directly like basic algorithm. When data comes to very high dimension, calculating co-variance matrix is computationally expensive and time-consuming. Here I use function tic and toc to record the time each algorithm use to achieve first two processes (PCA2 uses 0.0652 seconds, PCA1 uses 0.212 seconds). The difference is not big due to 874 is not very high.

2. When PoV comes to over 90%, it covers most variation of the given data set while the number of chosen principal components is just 47, which means removing other dimensions not very meaningful or noises and choosing the ones that has top largest variation.

3. For 10 images in the test set, the low-dimensional representations are Z, whose size is 47*10 (due to it covers too large of the page, the data can be checked in the lab session).

4. After encoding data point, reconstructing test examples back to original space based on selected top principal components (eigenvectors) with portions of data are removed in some dimensions.

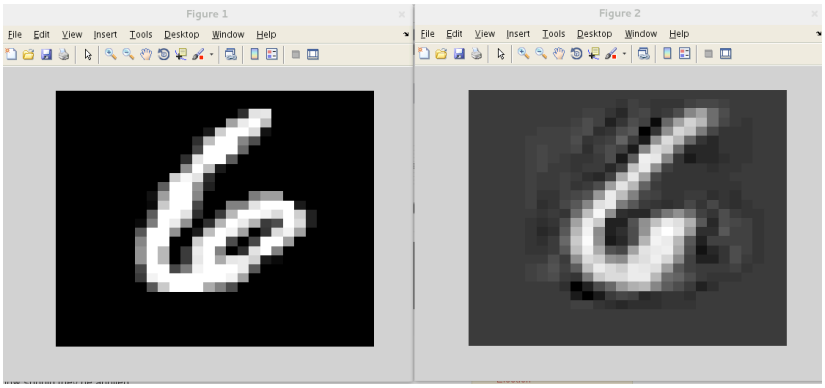

Figure 4: Original image (left) and its corresponding reconstructed image (right)

Reconstructed image is not more precise than compared with its original image. Considering the pixels are just 28*28 for one image and 784 dimensions at all, it is low-quality images especially when only choose top principal components (i.e., the information preserved is not very much).

5. As I reconstruct the test data by mapping the encoded data point back to original space, the reconstruction error here is the sum squared length between original data point and its reconstructed data point. Besides, I apply squared length to each original vector and its reconstructed vector and calculate their mean value. The reconstruction error is 5.2685 now.