COEN 140L Lab 9 Intro

Xuyang Wu

Dataset

- A face image database of 10 subjects, (in the att_faces_10.zip file).
- Each subject has 10 gray-scale images of 112×92 pixels
- use face images 1,3,4,5,7,9 as the training images, and face images 2,6,8,10 as the test images.

PCA for face recognition

- C subjects/Classes
- Training set (each subject has N training images)
  - Subject 1: $\mathbf{x}_{11}, \ldots, \mathbf{x}_{1N}$ (i.e. $\mathbf{x}_{1n}, n = 1, \cdots, N$)
  - Subject 2: $\mathbf{x}_{21}, \ldots, \mathbf{x}_{2N}$
  - …
  - Subject C: $\mathbf{x}_{C1}, \ldots, \mathbf{x}_{CN}$

PCA for face recognition

- SVD for dimensionality reduction
  - $\mathbf{X}=[\mathbf{x}_{11},\dots,\mathbf{x}_{1N},\mathbf{x}_{21},\dots,\mathbf{x}_{2N},\dots,\mathbf{x}_{C1},\dots,\mathbf{x}_{CN}]$
    $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$
  - Find the top-d left singular vectors
    $$\mathbf{U}_{:,[1:d]}=[\mathbf{u}_1,\cdots,\mathbf{u}_d\,]$$
  - Project each training image onto these vectors
    - The nth training image of the cth subject: $\mathbf{x}_{cn}$

$$\mathbf{y}_{cn}=\begin{bmatrix} y_{cn,1}\\ \vdots\\ y_{cn,d}\end{bmatrix}=\mathbf{U}_{:,[1:d]}{}^{T}\mathbf{x}_{cn}=\begin{bmatrix}\mathbf{u}_1^T\mathbf{x}_{cn}\\ \vdots\\ \mathbf{u}_d^T\mathbf{x}_{cn}\end{bmatrix}$$

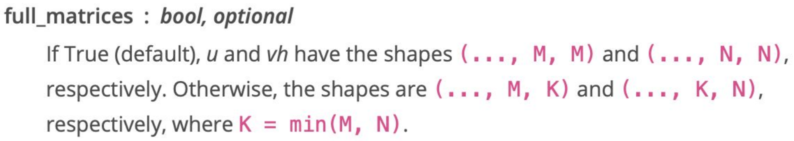
- Nearest-Neighbor Classifier
  - For a test image $\mathbf{x}$
  - Project the test image onto vectors

$$\mathbf{y}=\begin{bmatrix} y_1\\ \vdots\\ y_d\end{bmatrix}=\begin{bmatrix}\mathbf{u}_1^T\mathbf{x}\\ \vdots\\ \mathbf{u}_d^T\mathbf{x}\end{bmatrix}=\mathbf{U}_{:,[1:d]}{}^{T}\,\mathbf{x}$$

  - Determine its class label by

$$\hat{c}=\arg\min_{\substack{c=1,2,\dots,C\\ n=1,2,\dots,N}}\|\mathbf{y}-\mathbf{y}_{cn}\|_2$$

SVD Algorithm



full_matrices : *bool, optional*

If True (default), *u* and *vh* have the shapes `(..., M, M)` and `(..., N, N)`, respectively. Otherwise, the shapes are `(..., M, K)` and `(..., K, N)`, respectively, where `K = min(M, N)`.

U,Sigma,V = numpy.linalg.svd(L, full_matrices=False) # must set full_matrices=False

# use this link to check how to use numpy.linalg.svd
https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html

# check whether the singular values in Sigma are sorted from the largest to smallest. If not, need to sort them from the largest to smallest, and also sort the left singular vectors in U correspondingly.

Basic idea of this assignment

# loading images as train and test dataset

- Convert each image to a vector of length D=112×92=10304. Stack 6 training images of all 10 subjects to form a matrix of size 10304×60.

# Apply SVD by numpy.linalg.svd(L,full_matrices=False)

- Apply singular-value decomposition (SVD) for dimensionality reduction. Find the top-K left singular vectors (K=1,2,3,6,10,20, 30, and 50) corresponding to the K largest singular values of the data matrix.

# calculate the distance between the test image and each of the training image

- Project the face images to the top-K left singular vectors and apply the nearest-neighbor classifier in the reduced dimensional space.

Include in the report

- Plot the recognition accuracy rate, versus different K values.
    - ○ *number of correct classificationtotal /number of test images*)
- Analyze the results you observe.

Lab Tasks

- **Need demo** for week 9 assignment(10% points).
- Submit to Camino a pdf report with answers(60% points), the report contains some **results** which required by lab document, you also need to add some **observations** for the questions.
- Submit all the source code needed to generate these answers to Camino(30% points).