ES – 637 MATHEMATICAL FOUNDATIONS FOR COMPUTER VISION AND GRAPHICS ASSIGNMENT 1

Face Recognition using PCA

Introduction

Principal component analysis is a statistical tool used to reduce the dimensionality of a data set. The idea is to convert a set of correlated variables into a set of uncorrelated variables. The transformation takes place in such a way such that the variance along the principal components decreases with the first component showing the maximum variance.

For facial recognition, an image in high dimension is transformed to a lower dimension, where classification becomes easy and computationally less expensive. The image is brought to a lower dimensional subspace using principal component analysis, identifying components with maximum variance and throwing away components with redundant data. This approach is known as eigenfaces.

Algorithm

Principal component analysis

1. Construct the data matrix X_{mxn}, where m is the number of features and n is the number of data instances.

$$X = \begin{pmatrix} x11 & x12 & \dots & \dots & x1n \\ x21 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ xm1 & xm2 & \cdots & \cdots & xmn \end{pmatrix}$$

2. Find the mean column vector μ by averaging all the column vectors and subtract it from each of the column vectors to from the mean shifted data matrix D

$$\mu = \sum_{i=1}^{n} \begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ \vdots \\ x_{3i} \end{pmatrix}$$

$$D_i = X_i - \mu_i$$

3. If the number of dimensions is less than the number of data points,

- compute the covariance matrix, $cov = (\frac{1}{n})DD'$. Then find the eigenvectors, eigenvalues of the covariance matrix by invoking the eig function in MATLAB.

$$cov = \begin{pmatrix} var(d1) & Cov(d1,d2) & \dots & \dots & Cov(dm,d1) \\ Cov(d1,d2) & var(d2) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ Cov(dm,d1) & Cov(dm,2) & \dots & \cdots & Var(dm) \end{pmatrix}$$

Eigenvectors of matrix A are computed using the following equation:

$$VA = \lambda V$$

where V and λ represent the eigenvector and eigenvalue of matrix A

4. If the number of dimensions is more than the number of data points as in case of images then computing the covariance matrix would be computationally expensive.so the following trick is used:
   - Compute the eigenvector, eigenvalues of $\left(\frac{1}{n}\right) D'D$. The required eigenvector matrix is obtained by $E' = DE$ and E' is then normalised
5. But since in case of images the number of dimensions is much greater than the number of images therefore the trick has to be utilised to reduce computation.
6. Eigenvalues are sorted in descending order and the corresponding eigenvectors are arranged in the same order. It is done because the eigenvectors with large eigenvalues give high variation of data along its direction. So we need to omit those eigenvectors along which variation is insignificant to consider.

7. So we take first k eigenvectors which we call the principal components and take the projection of the mean shifted data matrix D along those components.

$$Y = P^T X = \sum_{i=1}^{n} P^T (X_i - \mu)$$

Where Y is the projected matrix and P is the principal component matrix

Face recognition

1. Construct the image vector, S containing images as column vectors.
2. Construct the projection matrix by computing the principal components by PCA and selecting k of them
3. Given a set of test images, construct the test image vector and obtain the corresponding mean shifted image vector
4. Project both the training images and test images into the principal components subspace.
5. For each projected test image, compare by taking the L2 norm with all the training image and select the face with least distance, which is taken as the matched face.

Implementation Details

- The dataset used for training and test is AT&T ”The database of Faces”.
- 70% of the images from the databases are used for training and the rest used for testing.
- The number of principal components used is 100.
- Size of Training image vector is 10304x280 and that of test image vector is 10304x120
- Resulting size of principal component matrix is 10304x100

Results

Accuracy obtained = 95

Reference

- https://github.com/bytefish/facerecognition_guide/raw/master/facerec_python.pdf
- https://docs.opencv.org/trunk/dc/dc3/tutorial_py_matcher.html
- https://docs.scipy.org/doc/numpy-1.13.0/reference/index.html
- http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html