Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition

Chandni Shrivastava, Sachin Sundar Pungampalayam Shanmugasundaram

Abstract—In this paper, a new technique coined two-dimensional principal component analysis (2DPCA) is implemented for image representation. As opposed to PCA, 2DPCA is based on 2D image matrices rather than 1D vectors so the image matrix does not need to be transformed into a vector prior to feature extraction. Instead, an image covariance matrix is constructed directly using the original image matrices, and its eigenvectors are derived for image feature extraction. To test 2DPCA and evaluate its performance, experiments were performed on the face image database: Extended Yale face database B. The recognition rate observed was higher using 2DPCA than PCA. The experimental results also indicated that the extraction of image features is computationally more efficient using 2DPCA than PCA.

Index Terms—Principal Component Analysis (PCA), Eigenfaces, feature extraction, image representation, face recognition.

1 Introduction

Principal Component Analysis (PCA) is a classical technique for feature extraction and data representation which is widely used in the areas of pattern recognition and computer vision such as stock market predictions, the analysis of gene expression data, and many more. Since face recognition technology can be applied in a wide range of fields, PCA has become the most successful approaches till date. However, PCA could not represent the arbitrary effects of illumination very well.

There have been other methods related to PCA that were adopted recently to do this – Independent Component Analysis (ICA) and Kernel Principal Component Analysis (Kernel PCA). ICA performed better than PCA when cosines were used as a similarity measure, but not significantly different if used with Euclidean distance. Kernel PCA also performed better than the traditional Eigenfaces method used in PCA. However, both of them proved to be computationally expensive than PCA.

PCA uses the concept of dimensionality reduction in the dataset. 2D face image matrices are first transformed into 1D image vectors. The resulting image vectors of faces usually lead to a high dimensional image vector space, where it is difficult to evaluate the covariance matrix accurately due to its large size and the relatively small number of training samples. The eigenvectors can be calculated efficiently using SVD techniques, but evaluation accuracy of the eigenvectors is not assured.

In this paper, we will be implementing an enhanced technique for image representation – Two Dimensional Principal Component Analysis (2DPCA). Unlike native PCA, 2DPCA is based on 2D matrices rather than 1D vectors, i.e., image matrices are not transformed into image vectors. An image covariance matrix is constructed directly using the original image matrices. Size of the image covariance matrix is smaller as compared to that of PCA. Hence it becomes easier to evaluate the covariance matrix accurately. Also, determination of the eigenvectors takes less time.

In the following sections, we will be discussing about the algorithm, approach, experiment, results and conclusion.

2 Two-Dimensional Principal Component Analysis

2.1 Algorithm

Suppose we have an image **A**. We want to project its (m x n) matrix onto a n-dimensional unit vector **X** by the linear transformation:

$$\mathbf{Y} = \mathbf{AX} \quad (1)$$

We will get an m-dimensional projected vector Y. This is the projected feature vector of image A. We need to determine the total scatter of the projected samples. The following criterion was adopted:

$$\mathbf{J(X) = tr(S_x)} \quad (2)$$

Here, $\mathbf{S_x}$ denotes the projected covariance matrix of the projected feature vectors of the training samples and $\mathbf{tr(S_x)}$ denotes the trace of $\mathbf{S_x}$.

The covariance matrix $\mathbf{S_x}$ can be denoted by:

$$\begin{aligned}\mathbf{S_x} &= \mathbf{E\ (Y - EY)\ (Y\ EY)^T} \\ &= \mathbf{E\ [AX - E(AX)]\ [AX - E(AX)]^T} \\ &= \mathbf{E\ [(A - EA)\ X][(A - EA)X]^T}\end{aligned}$$

$$\mathbf{tr(S_x) = X^T[E(A - EA)^T(A - EA)]X} \quad (3)$$

Another matrix, image covariance (scatter) matrix, was defined:

$$\mathbf{G_t = E\ [(A - EA)^T(A - EA)]} \quad (4)$$

For M training samples, $\mathbf{G_t}$ can be evaluated by:

$$\mathbf{G}_t = \frac{1}{M}\sum_{j=1}^{M}(\mathbf{A}_j - \bar{\mathbf{A}})^T(\mathbf{A}_j - \bar{\mathbf{A}}) \quad (5)$$

where $\bar{\mathbf{A}}$ is the average image of all the training samples.

From eq. (2), we can re-write:

$$\mathbf{J(X) = X^TG_tX} \quad (6)$$

The unitary vector **X** that maximizes the criterion is called the optimal projection axis. The optimal projection axis $\mathbf{X_{opt}}$ is the unitary vector that maximizes $\mathbf{J(X)}$, i.e., the eigenvector of $\mathbf{G_t}$ corresponding to the largest eigenvalue. Hence we select the orthonormal eigenvectors of $\mathbf{G_t}$ corresponding to the first d largest eigenvalues.

2.2 Feature Extraction

The optimal projection vectors of 2DPCA, $\mathbf{X_1, \ldots X_d}$, are used for feature extraction. We project a group of projected feature vectors $\mathbf{Y_1, \ldots Y_d}$, which are called the principal component vectors of the sample image **A**, from eq. (1)

$$\mathbf{Y_k = AX_k}, \quad k = 1, 2, \ldots, d \quad (7)$$

As we see here, each principal component of 2DPCA is a vector, unlike PCA where each of them were scalar.

These principal component vectors are used to form the feature (m x d) matrix or feature image, $\mathbf{B = [Y_1, \ldots Y_d]}$, of the image sample **A**.

2.3 Classification

Once all the image covariance matrix is decomposed into *d* eigen vectors using 2D-PCA, the feature matrix is obtained using dot product of the eigen vector matrix with the data matrix. Using the feature matrix of all the images and nearest neighbor classifier, we classify all the images so that similar person's will be classified into same group. For nearest neighbor classification, find the Euclidean distance between the feature matrices, $\mathbf{B_i = [Y_1, Y_2, \ldots, Y_d]}$ and $\mathbf{B_j = [Y_1, Y_2, \ldots, Y_d]}$ and classify the matrices that are near to each other.

$$\textbf{Distance, d} = \sum \| \mathbf{Y^{(i)} - Y^{(j)}} \|$$

where $\mathbf{Y^{(i)}}$ and $\mathbf{Y^{(j)}}$ are the feature matrices, the distance measure used is Euclidean distance.

3 Reconstruction of Images

To reconstruct the image from the decomposed feature matrices, combine the principal components and eigen vectors of the 2D-PCA of the image matrix. Select the first *d* largest eigen vectors of the image covariance matrix. Project them on to the principle component vectors. The result will be in the same dimensions of the original image matrix and represents the reconstructed image.

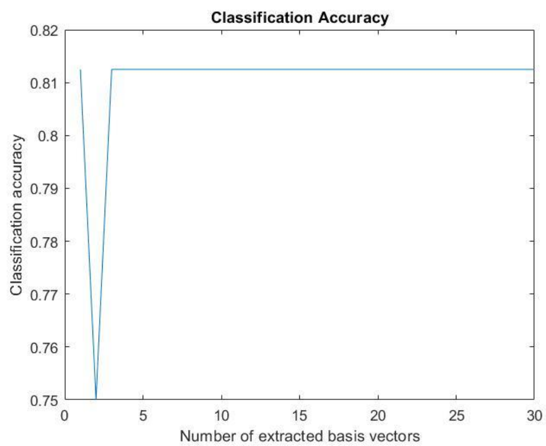
$$\bar{\mathbf{A}}' = \mathbf{VU^T}$$

where **V** is the principal component matrix (or) the feature matrix, $\mathbf{V = [Y_1, Y_2, \ldots, Y_d]}$ and **U** is the eigen vector matrix of the *d* largest eigen vectors, $\mathbf{U = [X_1, X_2, \ldots, X_d]}$.

$\bar{\mathbf{A}}'$ is the reconstructed sub-image image of **A**. To get the maximum accuracy, increase the ***d*** value. If ***d*** = total number of eigen vectors, the image will be completely reconstructed. As ***d*** reduces, reconstruction accuracy decreases, and the reconstructed image will approximate to the original image.

4 Experiment

This method is tested on the Yale face data set and measure the classification accuracy by nearest neighbor classifier and the reconstruction accuracy by 2D-PCA. The data set contains 15 subjects and each subject has 11 samples. Yale data set measures the system performance when both facial expressions and illumination are varied. Figure 1 shows that the classification accuracy is saturated after the first 5 largest eigen vectors. This shows that most of the data is over the first 5 eigen vectors.



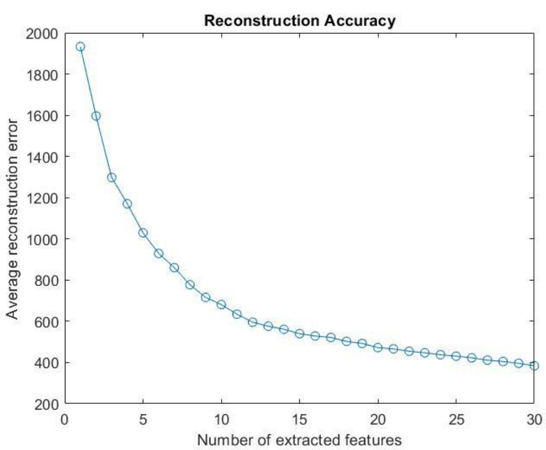


Fig. 1: Classification accuracy using k-NN Classifier
Fig. 2: Reconstruction accuracy using 2D-PCA

Figure 2 shows that as the number of eigen vectors increases, reconstruction error decreases. Even though data is concentrated on first 5 eigen vectors, still data is over other eigen vectors and that helps is reducing the reconstruction error.

5 Conclusion

2D-PCA has advantages over other PCA methods. It is based on the image matrix and its 2-dimensional covariance matrix which simpler and easier to use for image feature extraction. 2D-PCA also has a better recognition accuracy than other PCA methods. One disadvantage of 2D-PCA is it constructs a 2-dimensional covariance matrix and for large data set that can take more memory than PCA as it constructs a 1D vector.

6 References

[1] Jian Yang, David Zhang, Alejandro F. Frangi, Jing-yu Yang, *"Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition".*

[2] R.M. Mutelo, S.S. Dlay, W.L. Woo, *"Two Dimensional Incremental Reduction PCA: A Novel appearance based technique for image representation and recognition".*