Principal Component Analysis and K-means Clustering in Biometrics

Suyog Daga, *Graduate Student, University of Florida*

***Abstract*— Principal component analysis (PCA) is a technique that has various applications like dimensionality reduction, lossy data compression, feature extraction, data visualization and biometrics. It gives an orthogonal projection of data in lower dimensional space, known as the principal subspace such that the variance of data is maximized in that subspace. PCA also is a tool to reduce multidimensional data to lower dimensions while retaining most of the information. K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori In this paper, we check the application of PCA and K-means clustering in biometrics.**

***Index Terms*— PCA, K-means, soft biometrics, facial recognition.**

I. Introduction

Biometrics can be referred as verification or identification of a person's physical characteristics. Many techniques have been devised for verification and identification based on biometrics using various unique aspects of human body. Facial recognition is one widely known technique in the field of biometrics. Some facial recognition techniques and algorithms identify face related features by extracting features, from an image of the subject's face. For example, an algorithm may check the relative orientation position, size and shape of the eyes, nose related to face. These features are then used to search for other images with matching features. There are two categories of recognition geometric, which looks at distinguishing features, or photometric, which is a statistical method that distils an image into values and then compares the values with different templates to eliminate variance.

Principal component analysis (PCA), also known as Karhunen-Loeve expansion, is a classical feature extraction and data representation technique widely used in the areas of pattern recognition and computer vision such as face recognition [3].

Principal component analysis reduces dataset in high dimension space to low dimension space still maintaining the variance. PCA is used along with Eigenfaces method which consists of extracting the characteristic features from the face and then representing the face as a linear combination of the so-called Eigenfaces which are obtained from the feature extraction process. Principal components for data in training is then calculated. We then project the training data on the space formed by Eigenfaces. Once this projection is done, we used a distance metric like Euclidean distance for comparing the Eigenvectors of the Eigenfaces and Eigenface of the data image. If the metric distance is small enough, the person is said to be recognized. On the other hand, if the distance is very large, the image is said to belong to an individual for which the system needs to be trained.

From PCA, we do dimensionality reduction, but still maintaining maximum variance in lower dimensional space. We can select as many principal components as we need and then can apply K-means unsupervised learning algorithm to classify data.

II. K –means Method

K-means is an unsupervised algorithm that solves clustering problems. Given a dataset with, it classifies them in K clusters which are fixed beforehand. The main idea for defining K centers is one for each cluster. These centers should be placed in a smart way because of different location causes a different result. A good strategy would be placing all these centers as far as possible. Then we take a point from the dataset and associate it with the respective center. When no point is pending, the first step is completed and an early group age is done. Now we again calculate the new centroids. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center [4]. A loop has been generated. As a result of this loop, we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move anymore. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

Where,

||x_i-v_j|| is the Euclidean distance
'c_i' is the number of data points in i^{th} cluster
'c' is the number of clusters.

Following are the algorithmic steps for K-means algorithm:

Let X = { $x_1, x_2, x_3 \ldots x_n$} be set of data points and V = { $v_1, v_2, v_3 \ldots v_c$} be the set of centers.

1. Randomly select *'c'* cluster centers.
2. Calculate the distance between each data point and cluster centers.
3. Assign the data point to the cluster center whose distance from the cluster center is a minimum of all the cluster centers.
4. Recalculate the new cluster center using:

$$\boldsymbol{v}_i = (1 / \boldsymbol{c}_i) \sum_{j=1}^{\boldsymbol{c}_i} \boldsymbol{x}_i$$

5. Recalculate the distance between each data point and newly obtained cluster centers.
6. If no data point was reassigned then stop, otherwise repeat from step 3).

III. PRINCIPAL COMPONENT ANALYSIS AND EIGEN FACE METHOD

A naive approach for extracting the information contained in an image of a face is to capture the variation in a collection of face images, use this information to process images and then for comparison. Therefore, we wish to find the principal components for faces, or the eigenvectors of the covariance matrix of the data images, assuming images as vectors in a very high dimensional space. Each of the eigenvectors accounts for different variation in images. All the eigenvectors are like features which represent variances between face images. Each image location contributes more or less to each eigenvector so that we can display the eigenvector as a sort of ghostly face which we call an eigenface.

All images can be represented exactly in terms of a linear combination of the eigenfaces. Each face can also be approximated using only the good eigenfaces-those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images [1].

A very exciting application of PCA that was first considered by Sirovich and Kirby [2] and implemented by Turk and Pentland [1] is to the field of human facial recognition by computers. The basic idea of this "Eigenface" method is to form a dataset of images of faces of a wide variety of people (n people black and white images). Append second column vector below first, third below second and son on so we get a huge vector. Each entry in the respective vector represents brightness in pixel (assume 0 for white and 1 for black). Running PCA on this data set gives us principal components, which can again be converted back into face images. These are eigenfaces, each of which is a linear combination of faces from the data set. 100 eigenfaces can give us a good approximation of images [1][2].

Below is the approach for face recognition involving the following initialization operations, as specified in Turk and Pentland [1].

1. Acquire an initial set of face images (the training set).
2. Calculate the eigenfaces from the training set, keeping only the M images that correspond to the highest eigenvalues. These M images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.
3. Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their face images onto the "face space."

These operations can also be performed from time to time whenever there is free excess computational capacity.

Having initialized the system, the following steps are then used to recognize new face images:

1. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.
2. Determine if the image is a face at all (whether known or unknown) by checking to see if the image is sufficiently close to "face space."
3. If it is a face, classify the weight pattern as either a known person or as unknown.
4. If needed, update the eigenfaces and/or weight patterns.
5. Again if needed, if the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into the known faces.

Following is the mathematical representation of the above algorithm to calculate the Eigenfaces [1]:

1. We assume our image as N*N image which we first need to convert into N^2*1 vector.

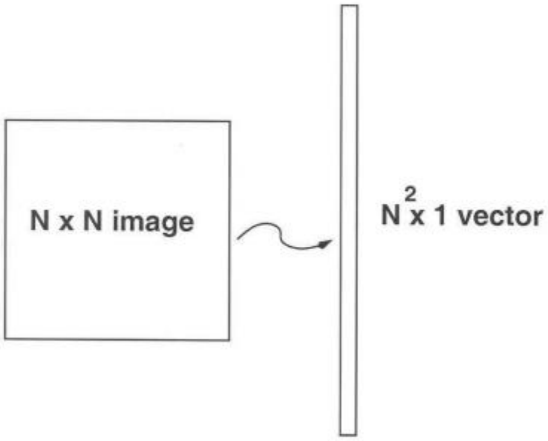


Fig 2.1 N*N image to N^2*1 vector transformation.

2. Our overall objective is to represent this N dimension image in some K dimension image where K<<N^2.

Suppose λ is an N^2*1 vector corresponding to N*N face image I. As per our objective , we need to transform λ into smaller dimension space, where,

$$\Gamma - \text{meanface} = w_1 u_1 + w_2 u_2 + .. + w_k u_k$$

For the above-mentioned transformation we follow the below steps:

1. Obtain the data set of images $I_1, I_2, I_3 .. I_m$.
2. Represent every image as I_i as Γ_i
3. Compute the average face vector ψ :

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$$

4. Subtract the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

5. Compute the covariance matrix C:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \quad (N^2 \text{x} N^2 \text{ matrix})$$

$$\text{where } A = [\Phi_1 \ \Phi_2 \cdots \Phi_M] \quad (N^2 \text{x} M \text{ matrix})$$

6. Compute the eigenvectors of u_i of AA^T.
   Here AA^T can have up to N^2 eigenvalues and vectors and A^TA can have up to M eigen values and vectors, therefore we keep M best eigen vectors.
7. We keep only k eigenvectors corresponding to K largest eigenvalues as per our requirements.

Once we get the K largest Eigen Faces, we can facial recognition using following steps:

1. Get an unknown image Γ , which is similar size as the training faces and which is centered.
2. Normalize Γ: Φ = Γ – Ψ
3. Project on Eigen space:

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$$

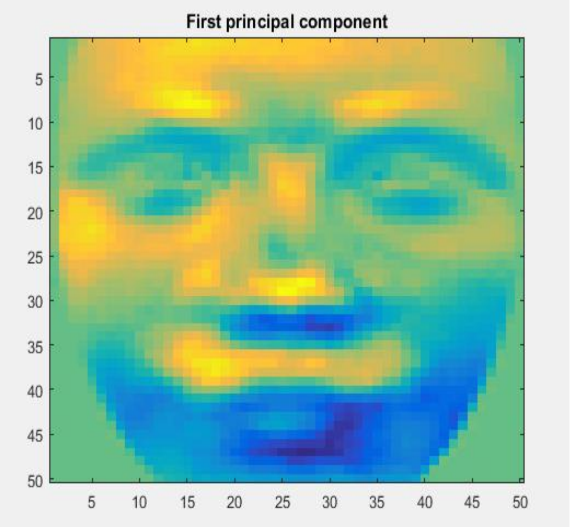
4. Represent Φ as: Ω = $[w_1, w, w_3 .. w_k]$' .
5. Find the Euclidean distance between Ω and the original training data set images which have been transformed to K- dimension space, the one with minimum distance represents a match.

Here is the overall summary of Eigenface recognition procedure:

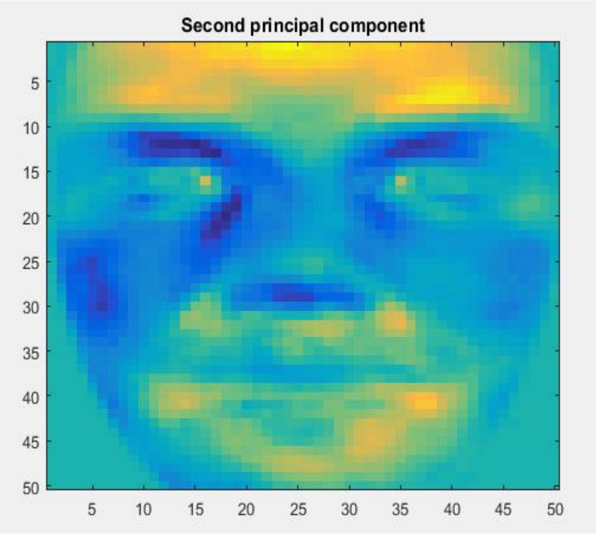
1. Collect a set of characteristic face images of the known individuals. This set should include a number of images for each person, with some variation in expression and in the lighting.
2. Calculate the matrix, find its eigenvectors and eigenvalues, and choose the M eigenvectors with the highest associated eigenvalues.
3. Combine the normalized training set of images
4. For each known individual, calculate the class vector by averaging the eigenface pattern vectors IR
5. For each new face image to be identified, calculate its pattern vector, the distances to each known class, and the distance E to face space.
6. If the new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated (steps 1-4). This gives the opportunity to modify the face space as the system encounters more instances of known faces.

III. SIMULATION AND RESULTS

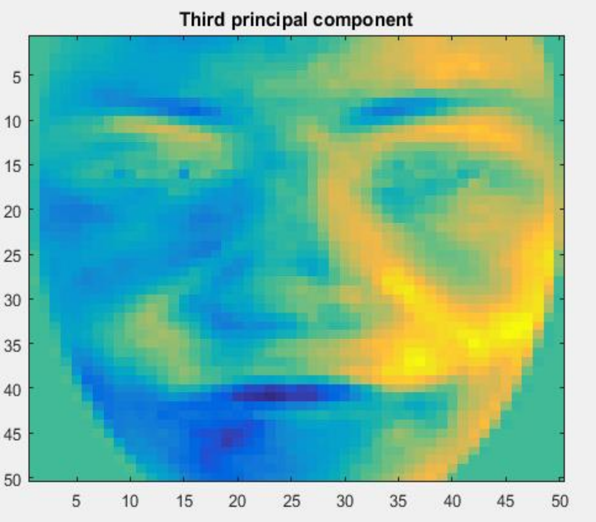
- We have used to datasets in this paper, first is GallerySet which contains facial images of 100 individual(one image per person). The second dataset is ProbeSet which contains facial images of same 100 individuals but has 2 images per person. We also have been provided with gender data for 100 persons in the dataset as male/female.
- Objective for this project is to implement and evaluate the performance of Principal Component Analysis and K-means algorithms as they apply to the problems of biometric recognition and soft biometric (gender classification).
- Firstly, I have explored the GallerySet dataset as it would be used to training . There are 100 images in it and each image is represented using 50*50 pixels.
- Each image needs to be converted from 50*50 needs to be converted into 2500*1.
- So overall we represent GallerySet as 2500*100 matrix ,giving us an idea that each image is represented as 2500 dimensions.
- We then apply principal component analysis to it and find out the first three principal components for the data.
- Below are the images represented by first three principal components:



3.1 Image of first principal component

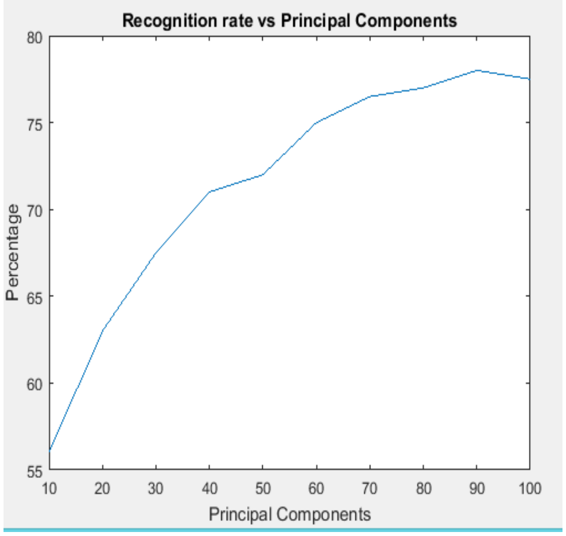


3.2 Image of second principal component



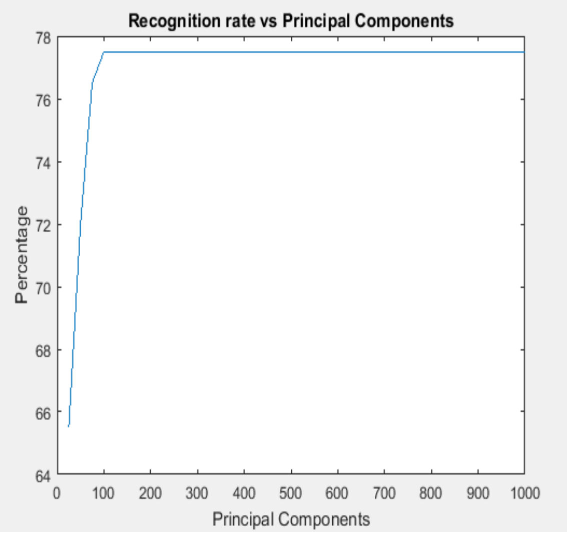
3.3 Image of second principal component

- The eigenfaces have the same dimensionality as the original pictures, and hence, can be displayed as visual images. The first three eigenfaces are shown above The homogeneous structure of human faces (two eyes above a central nose, above a central mouth) means that the values of corresponding pixels in different faces are not random. Consequently, the eigenfaces have a face-like quality; although from their murky appearance, it is not immediately apparent what characteristics each eigenface is coding[5][6]. All that we can derive from a visual inspection of these images is that areas of light and dark (areas that deviate from the uniform gray level) indicate features that differ across subsets of faces; although, the actual sign (light/dark) of these areas is essentially arbitrary. On the above basis, we can see that first eigenface or largest principal component may represent fringe length[6].
- We then plot recognition rate which ratio of actual persons recognized correctly, as a function of principal components.



3.4 Recognition rate vs Principal Components

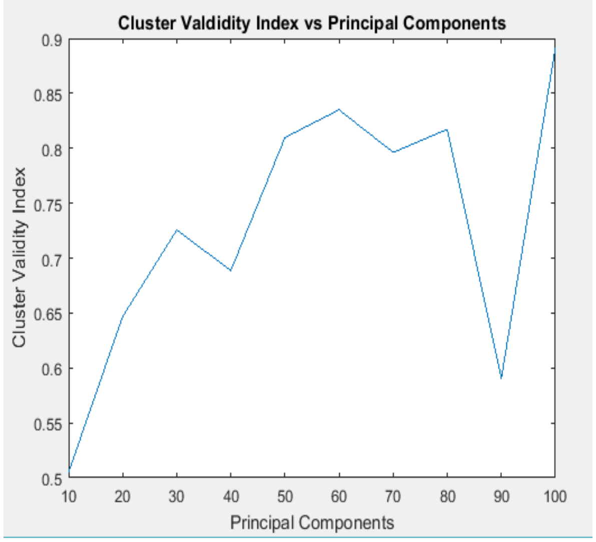
- When principal components are varied from k= 10 to 100 we get recognition rate in the range of 53 % to 77.5 %, as shown in figure (3.4).
- If we further increase the principal components, say k > 100 we still get the same recognition rate or our recognition rate remains constant, here is a plot where the principal component is varied between k=25 to k=1000.



3.4 Recognition rate vs Principal Components(25-1000)

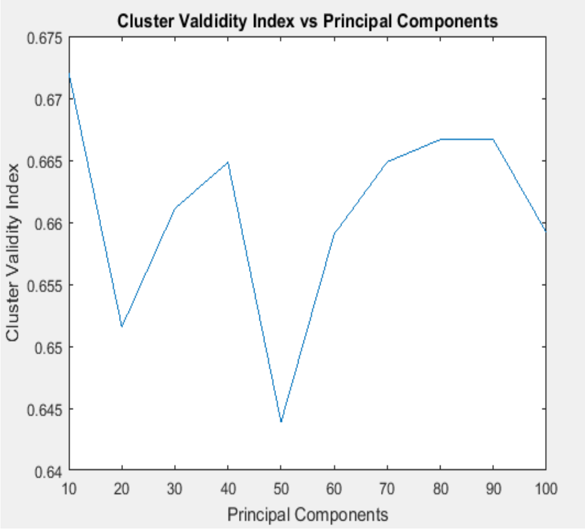
- Even when using the original image vectors and Euclidean distance as a distant measure, we get similar recognition ratio as compared to what we get using 100 principal components. This, specifically shows PCA can reduce the dimensions, and still maintain proper performance.
- For cluster evaluation I have evaluated the clustering using two different criteria, first is Silhouette

coefficient which is an Internal cluster validity index, that contrasts the average distance to elements in the same cluster with an average distance to elements in another cluster. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering and is also used to determine the optimal number of clusters. I have checked with K=2 and varied principal components k from 10 to 100. We get best results on k= 100 , where the value is maximum. Here is the plot related to it:



3.5 Cluster Validity Index vs Principal Components

- F- measure is an external validation criterion, which can be used to balance the contribution of false negatives by weighting recall through a parameter , its value is almost constant for K=2 , and varying the principal components from k=10 to K=100. So we can see that false negatives have been balanced properly. Below is the plot related to it, we see that cluster validity values are in the range of 0.64 to 0.67.



3.6 Cluster Validity Index vs Principal Components

IV CONCLUSION

We have seen the applications of Principal Component Analysis and K- means clustering in biometrics. Principal Component Analysis is a powerful technique, which can be used for dimensionality reduction ,converting from a higher space to a lower dimensional space. We produced approximately same results for facial recognition, using 100 principal components as compared to 2500 dimensions. This definitely helps us to save computational time. Also, we saw that largest principal component may represent the fringe width for a face. PCA's key advantages are its low noise sensitivity, the decreased requirements for capacity and memory, and increased efficiency given the processes taking place in smaller dimensions. K-means helped us to cluster the data in 2 clusters, we checked how reduced dimension-set by PCA can be useful in still finding appropriate clusters for data. Cluster validity indices further clarified the significance with respect to principal components.

V. REFERENCES

[1] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." Journal of cognitive neuroscience 3, no. 1 (1991): 71-86.

[2] Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A, 4(3), 519-524

[3] C. Li, Y. Diao, H. Ma, and Y. Li, "A Statistical PCA Method for Face Recognition," in Intelligent Information Technology Application, 2008, pp. 376-380.

[4]https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm.

[5] Calder, Andrew J., A. Mike Burton, Paul Miller, Andrew W. Young, and Shigeru Akamatsu. "A principal component analysis of facial expressions." Vision research 41, no. 9 (2001): 1179-1208.

[6] Hancock PJ, Burton AM, Bruce V. Face processing: Human perception and principal components analysis. Memory & Cognition. 1996 Jan 1;24(1):26-40.