Principal Components Analysis for Compression (Part 1 of 2)

Objectives

1. Derive eigenvectors and their eigenvalues from image and spectral data.
2. Convert a digital camera into a spectral imager using principal components analysis

Principal Components Analysis

Principal Components Analysis (PCA) allows the compression of a high dimensional feature vector to a fewer number of coefficients. In this activity, we employ PCA to find the few basis functions that can represent our data ensemble through a linear superposition.

The Karhunen Leove Expansion (KL) is an orthonormal expansion of a vector **X** using the eigenvectors of the covariance matrix of the data set where **X** belongs. This representation of a data set by select eigenvectors is known in statistical literature as ***principal components analysis***. The eigenvectors possessing the highest eigenvalues are used for the reconstruction of data while those with zero and low eigenvalues are discarded thus reducing the dimensionality of the feature vector space.

Let $\boldsymbol{X}$ be an n-dimensional random vector which can be represented by an expansion of the form

$$X = \sum_{i=1}^{n} y_i \Phi_i = \Phi Y. \tag{1}$$

The components of Φ span the n-dimensional space containing $\boldsymbol{X}$ and are called basis vectors.

If the components of Φ form an orthonormal set

$$\mathbf{\Phi_i^T \Phi_j} = \begin{Bmatrix} 1 & \text{if} & i = j \\ 0 & \text{if} & i \neq j \end{Bmatrix} \tag{2}$$

then the components of $\boldsymbol{Y}$ are given by

$$y_i = \Phi_i^T X. \tag{3}$$

$\boldsymbol{Y}$ then is an orthonormal transformation of the random vector and is also a random vector.

Each component of $\boldsymbol{Y}$ is a feature which contributes to representing the observed vector $\boldsymbol{X}$.

We now want to use only m (<n) components of $\boldsymbol{Y}$ to at least estimate $\boldsymbol{X}$. Replacing the uncalculated components of $\boldsymbol{Y}$ by constants, $\boldsymbol{b}$, we will form the following estimate

$$\tilde{X}(m) = \sum_{i=1}^{m} y_i \Phi_i + \sum_{i=m+1}^{n} b_i \Phi_i. \tag{4}$$

This estimate introduces an error given by

$$\Delta\tilde{X}(m) = X - \tilde{X}(m) = X - \sum_{i=1}^{m} y_i \Phi_i - \sum_{i=m+1}^{n} b_i \Phi_i \tag{5}$$

Substituting eq.(1) for $\boldsymbol{X}$ we obtain

$$\Delta\tilde{X}(m) = \sum_{i=m+1}^{n} (y_i - b_i)\Phi_i. \tag{6}$$

Our objective now is to determine b such that we minimize the mean square error

$$\begin{aligned} \bar{\varepsilon^2}(m) &= E\{\left\|\Delta\tilde{X}(m)\right\|^2\} \\ &= E\{\textstyle\sum_{i=m+1}^{n}\sum_{j=m+1}^{n}(y_j - b_j)(y_i - b_i)\Phi_i^T\Phi_j\} \\ &= \textstyle\sum_{i=m+1}^{n} E\{(y_i - b_i)^2\} \end{aligned} \tag{7}$$

Applying the derivative test to get b, we have

$$\frac{\partial\bar{\varepsilon^2}(m)}{\partial b_i} = \frac{\partial E\{(y_i - b_i)^2\}}{\partial b_i} = 2[E\{y_i\} - b_i] = 0 \tag{8}$$

This implies that

$$\begin{aligned} b_i &= E\{y_i\} \\ &= \Phi_i^T E\{X\}. \end{aligned} \tag{9}$$

Now the mean-square error can be written as

$$\begin{aligned}\bar{\varepsilon^2}(m) &= \sum_{i=m+1}^{n} E[(y_i - E\{y_i\})^2] \\ &= \sum_{i=m+1}^{n} E[(\Phi_i^T X - \Phi_i^T E\{X\})^2] \\ &= \sum_{i=m+1}^{n} \Phi_i^T E[(X - E\{X\})(X - E\{X\})^T]\Phi_i \\ &= \sum_{i=m+1}^{n} \Phi_i^T R_x \Phi_i\end{aligned} \tag{10}$$

where $\boldsymbol{R_x}$ is by definition the covariance matrix of $\boldsymbol{X}$.

The optimum choice for the basis vectors are those which satisfy the eigenvalue equation

$$R\Phi_i = \lambda_i \Phi_i \tag{11}$$

Thus the minimum mean-square error is

$$\bar{\varepsilon^2}(m)_{opt} = \sum_{i=m+1}^{n} \lambda_i \tag{12}$$

The effectiveness of each feature is determined by its eigenvalue. The higher the eigenvalue, the greater the contribution of the eigenvector (principal component) to the variance of the set. To minimize dimensionality, the smallest eigenvalues should be deleted first. This means that if the eigenvalues are indexed in descending order, that is,

$$\lambda_1 > \lambda_2 > \cdots > \lambda_n \tag{13}$$

then the features should be ordered in the same manner and the (m+1)th up to the nth eigenvalue may be discarded.

Procedure

1. First – a group work. Capture 5 images of your classmate's face (there's 17 of you e so 17 people × 5 captures = 85 faces). We only need low resolution and black and white images so we just need to line up and take 5 class photos. Remove glasses for now.
2. Using GIMP or Photoshop transform the faces as follows:
   a) Crop the faces to exclude hair.
   b) Resize face image to 50×50pixels only.
   c) Facial features such as eyes, nose and mouth are aligned.

3. Either arrange face data into a mosaic or concatenate image data into a 85×2500 matrix $X = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_{2500}^1 \\ X_1^2 & X_2^2 & \cdots & X_{2500}^2 \\ \vdots & \cdots & \ddots & \vdots \\ X_1^{85} & X_2^{85} & \cdots & X_{2500}^{85} \end{bmatrix}$ where each row are the concatenated pixel values of the face and each column are the pixels.
4. Obtain the covariance matrix of $\boldsymbol{X}$.
5. Apply PCA on the covariance matrix of $\boldsymbol{X}$. You may either solve the eigenvalue equation on cov($\boldsymbol{X}$) or use standard PCA packages in your scientific software. In Matlab use **pcacov**.
6. Alternatively instead of steps 4-5, you may straight away use **pca** no need to compute the covariance matrix but do select "Centered" in one of its input parameters and include the mean in the output.
7. Find how many principal components out of the 2500 possible ones can represent the face up to 99% (or with 1% error only). Take the cumulative sum of the normalized eigenvalues to find out the m principal components.
8. Reconstruct faces using linear superpositions of weighted eigenfaces and comment on the quality as you further reduce the number of eigenfaces used. You will need to reshape the vector if it was previously concatenated. Comment on the quality.

Reference

1. Jolliffe, Ian. Principal component analysis. John Wiley & Sons, Ltd, 2002.

Due in 2 week (Mar 23, 2022)