CHAPTER

Appendix 3: Principal components analysis

12

CHAPTER OUTLINE HEAD

12.1 Principal components analysis

This appendix introduces PCA. This technique is also known as the *Karhunen Loeve* transform or as the *Hotelling transform*. It is based on factorization techniques developed in linear algebra. Factorization is commonly used to diagonalize a matrix, so its inverse can be easily obtained. PCA uses factorization to transform data according to its statistical properties. The data transformation is particularly useful for classification and compression.

Here, we will give an introduction to the mathematical concepts and give examples and simple implementations, so you should be able to understand the basic ideas of PCA, to develop your own implementation and to apply the technique to your own data. We use simple matrix notations to develop the main ideas of PCA. If you want to have a more rigorous mathematical understanding of the technique, you should review concepts of eigenvalues and eigenvectors in more detail (Anton, 2005).

You can think of PCA as a technique that takes a collection of data and transforms it such that the new data has given statistical properties. The statistical properties are chosen such that the transformation highlights the importance of data elements. Thus, the transformed data can be used for classification by observing important components of the data. Also data can be reduced or compressed by eliminating (filtering out) the less important elements. The data elements can be seen as features, but in mathematical sense, they define the axes in the coordinate system.

Before defining the data transformation process defined by PCA, we need to understand how data is represented and also have a clear understanding of the statistical measure known as the covariance.

12.2 Data

Generally, data is represented by a set of m vectors

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \tag{12.1}$$

Each vector $\mathbf{x}_i$ has n elements or features, i.e.,

$$\mathbf{x}_i = [\mathrm{x}_{i,1}, \mathrm{x}_{i,2}, \ldots, \mathrm{x}_{i,n}] \tag{12.2}$$

The way you interpret each vector $\mathbf{x}_i$ depends on your application. For example, in pattern classification, each vector can represent a measure and each component of the vector a feature such as color, size, or edge magnitude.

We can group features by taking the elements of each vector. That is, the feature column vector k for the set $\mathbf{X}$ can be defined as

$$\mathbf{c}_{\mathbf{X},k} = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ \vdots \\ x_{m,k} \end{bmatrix} \tag{12.3}$$

for k ranging from 1 to n. The subindex $\mathbf{X}$ may seem unnecessary now; however, this will help us to distinguish features of the original set and of the transformed data.

We can group all the features in the feature matrix by considering each vector $\mathbf{c}_{\mathbf{X},k}$ to be a column in a matrix, i.e.,

$$\mathbf{c}_{\mathbf{X}} = \begin{bmatrix} \mathbf{c}_{\mathbf{X},1} & \mathbf{c}_{\mathbf{X},2} & \cdots & \mathbf{c}_{\mathbf{X},n} \end{bmatrix} \tag{12.4}$$

The PCA technique transforms the feature vectors $\mathbf{c}_{\mathbf{X},k}$ to define new vectors defining components with better classification capabilities. Thus, the new vectors can be grouped by clustering according to distance criteria on the more important elements, i.e., the elements that define important variations in the data. PCA ensures that we highlight the data that accounts for the maxima variation measured by the covariance.

12.3 Covariance

Broadly speaking, the covariance measures the linear dependence between two random variables (DeGroot and Schervish, 2001). So by computing the covariance, we can determine if there is a relationship between two sets of data. If we consider that the data defined in the previous section has only two components, then the covariance between features can be defined by considering the component of each vector. That is, if $\mathbf{x}_i = \{x_{i,1}, x_{i,2}\}$, then the covariance is

$$\sigma_{\mathbf{X},1,2} = E[(\mathbf{c}_{\mathbf{X},1} - \boldsymbol{\mu}_{\mathbf{X},1})(\mathbf{c}_{\mathbf{X},2} - \boldsymbol{\mu}_{\mathbf{X},2})] \tag{12.5}$$

Here, the multiplication is assumed to be element by element and $E[\ ]$ denotes the expectation which is loosely the average value of the elements of the vector. We denote $\boldsymbol{\mu}_{\mathbf{X},k}$ as a column vector obtained by multiplying the scalar value $E\lfloor \mathbf{c}_{x,k} \rfloor$ by a unitary vector. That is, $\boldsymbol{\mu}_{\mathbf{X},k}$ is a vector that has the mean value on each element. Thus, according to Eq. (12.5), we first subtract the mean value for each feature and then we compute the mean of the multiplication of each element.

The definition of covariance can be expressed in matrix form as

$$\sigma_{\mathbf{X},1,2} = \frac{1}{m}((\mathbf{c}_{\mathbf{X},1} - \boldsymbol{\mu}_{\mathbf{X},1})^{\mathrm{T}}(\mathbf{c}_{\mathbf{X},2} - \boldsymbol{\mu}_{\mathbf{X},2})) \tag{12.6}$$

where T denotes the matrix transpose. Sometimes, features are represented as rows, so you can find the transpose operating on the second factor rather than on the first. Note that the covariance is symmetric and thus $\sigma_{\mathbf{X},1,2} = \sigma_{\mathbf{X},2,1}$.

In addition to Eqs (12.5) and (12.6), there is a third alternative definition of covariance that is obtained by developing the products in Eq. (12.6), i.e.,

$$\sigma_{\mathbf{X},1,2} = \frac{1}{m}(\mathbf{c}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{c}_{\mathbf{X},2} - \boldsymbol{\mu}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{c}_{\mathbf{X},2} - \mathbf{c}_{\mathbf{X},1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X},2} + \boldsymbol{\mu}_{\mathbf{X},1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X},2}) \tag{12.7}$$

Since

$$\boldsymbol{\mu}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{c}_{\mathbf{X},2} = \mathbf{c}_{\mathbf{X},1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X},2} = \boldsymbol{\mu}_{\mathbf{X},1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X},2} \tag{12.8}$$

we have

$$\sigma_{\mathbf{X},1,2} = \frac{1}{m}(\mathbf{c}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{c}_{\mathbf{X},2} - \boldsymbol{\mu}_{\mathbf{X},1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X},2}) \tag{12.9}$$

This can be written in short form as

$$\sigma_{\mathbf{X},1,2} = E[\mathbf{c}_{\mathbf{X},1}, \mathbf{c}_{\mathbf{X},2}] - E[\mathbf{c}_{\mathbf{X},1}]E[\mathbf{c}_{\mathbf{X},2}] \tag{12.10}$$

for

$$E\left[\mathbf{c}_{\mathbf{X},1}, \mathbf{c}_{\mathbf{X},2}\right] = \frac{1}{m}(\mathbf{c}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{c}_{\mathbf{X},2}) \tag{12.11}$$

Equations (12.5), (12.6), and (12.11) are alternative ways to compute the covariance. They are obtained by expressing products and averages in algebraic equivalent definitions.

As a simple example of the covariance, you can think of one variable representing the value of a spectral band of an aerial image, while the other the amount of vegetation in the ground region covered by the pixel. If you measure the covariance and get a positive value, then for new data, you should expect that an increase in the pixel intensity means an increase in vegetation. If the covariance value is negative, then you should expect that an increase in the pixel intensity means a decrease in vegetation. When the values are zero or very small, then the values are uncorrelated and the pixel intensity and vegetation are independent, and we cannot tell, if the change in intensity is related to any change in vegetation. Let us recall that the probability of two independent events happening together is equal to the product of the probability of each event. Thus, $E[\mathbf{c}_{\mathbf{X},1},\mathbf{c}_{\mathbf{X},2}] = E[\mathbf{c}_{\mathbf{X},1}]E[\mathbf{c}_{\mathbf{X},2}]$ is characteristic of independent events. That is, Eq. (12.10) is zero.

The covariance value ranges from zero (indicating no relationship) to large positive and negative values that reflect strong dependencies. The maximum and minimum values are obtained by using the Cauchy–Schwarz inequality and they are given by

$$|\sigma_{\mathbf{X},1,2}| \leq \sigma_{\mathbf{X},1}\sigma_{\mathbf{X},2} \tag{12.12}$$

Here, $|\;|$ denotes the absolute value and $\sigma^2_{\mathbf{X},1} = E[\mathbf{c}_{\mathbf{X},1}, \mathbf{c}_{\mathbf{X},1}] - E[\mathbf{c}_{\mathbf{X},1}]E[\mathbf{c}_{\mathbf{X},1}]$ defines the variance of $\mathbf{c}_{\mathbf{X},1}$. Remember that the variance is a measure of dispersion; thus, this inequality indicates that the covariance will be large if the data has large ranges. When the sets are totally dependent, then $|\sigma_{\mathbf{X},1,2}| = \sigma_{\mathbf{X},1}\sigma_{\mathbf{X},2}$.

It is important to stress that the covariance measures a linear relationship. In general, data can be related to each other in different ways. For example, the color of a pixel can increase exponentially as heat of a surface or the area of a region increases in square proportion to its radius. However, the covariance only measures the degree of linear dependence. If features are related by other relationship, for example quadratic, then the covariance will produce a low value, even if there is perfect relationship. Linearity is generally considered to be the main limitation of PCA; however, PCA has proved to give a simple and effective solution in many applications; linear modeling is a very common model for many data, and covariance is particularly good if you are using some form of linear classification.

To understand the linearity in the covariance definition, we can consider that features $\mathbf{c}_{\mathbf{X},2}$ are a linear function of $\mathbf{c}_{\mathbf{X},1}$, i.e., $\mathbf{c}_{\mathbf{X},2} = A\mathbf{c}_{\mathbf{X},1} + \mathbf{B}$ for A an arbitrary constant and $\mathbf{B}$ an arbitrary column vector. Thus, according to Eq. (12.11)

$$E[\mathbf{c}_{\mathbf{X},1}, \mathbf{c}_{\mathbf{X},2}] = E[A\mathbf{c}^{\mathrm{T}}_{\mathbf{X},1}\mathbf{c}_{\mathbf{X},1} + \mathbf{c}^{\mathrm{T}}_{\mathbf{X},1}\mathbf{B}] \tag{12.13}$$

We also have i.e.,

$$E[\mathbf{c}_{\mathbf{X},1}]E[\mathbf{c}_{\mathbf{X},2}] = AE[\mathbf{c}_{\mathbf{X},1}]^2 + E[\mathbf{B}]E[\mathbf{c}_{\mathbf{X},1}] \tag{12.14}$$

By substituting these equations in the definition of covariance in Eq. (12.10), we have i.e.,

$$\sigma_{\mathbf{X},1,2} = A\left(E[\mathbf{c}^{\mathrm{T}}_{\mathbf{X},1}\mathbf{c}_{\mathbf{X},1}] - E[\mathbf{c}_{\mathbf{X},1}]^2\right) \tag{12.15}$$

i.e.,

$$\sigma_{\mathbf{X},1,2} = A\sigma^2_{\mathbf{X},1} \tag{12.16}$$

As such, when features are related by a linear function, the covariance is a scaled value of the variance. We can follow a similar development to find the covariance as a function of $\sigma^2_{\mathbf{X},2}$. If we consider that $\mathbf{c}_{\mathbf{X},1} = \frac{1}{A}\mathbf{c}_{\mathbf{X},2} - \frac{\mathbf{B}}{A}$, then

$$\sigma_{\mathbf{X},1,2} = \frac{1}{A}\sigma^2_{\mathbf{X},2} \tag{12.17}$$

Thus, we can use Eqs (12.16) and (12.17) to solve A, i.e.,

$$A = \frac{\sigma_{\mathbf{X},2}}{\sigma_{\mathbf{X},1}} \tag{12.18}$$

By substituting in Eq. (12.16), we get

$$\sigma_{\mathbf{X},1,2} = \sigma_{\mathbf{X},1}\sigma_{\mathbf{X},2} \tag{12.19}$$

That is, the covariance value takes its maximum value given in Eq. (12.12) when the features are related by a linear relationship.

12.4 Covariance matrix

When data has more than two dimensions, the covariance can be defined by considering every pair of components. These components are generally represented in matrix that is called the covariance matrix. This matrix is defined as

$$\boldsymbol{\Sigma}_{\mathbf{X}} = \begin{bmatrix} \sigma_{\mathbf{X},1,1} & \sigma_{\mathbf{X},1,2} & \cdots & \sigma_{\mathbf{X},1,n} \\ \sigma_{\mathbf{X},2,1} & \sigma_{\mathbf{X},2,2} & \cdots & \sigma_{\mathbf{X},2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{\mathbf{X},n,1} & \sigma_{\mathbf{X},n,2} & \cdots & \sigma_{\mathbf{X},n,n} \end{bmatrix} \tag{12.20}$$

According to Eq. (12.5), the element (i,j) in the covariance matrix is given by

$$\sigma_{\mathbf{X},i,j} = E[(\mathbf{c}_{\mathbf{X},i} - \boldsymbol{\mu}_{\mathbf{X},i})(\mathbf{c}_{\mathbf{X},j} - \boldsymbol{\mu}_{\mathbf{X},j})] \tag{12.21}$$

By generalizing this equation to the elements of the feature matrix and by considering the notation used in Eq. (12.6), the covariance matrix can be expressed as

$$\Sigma_{\mathbf{X}} = \frac{1}{m}((\mathbf{c}_{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{X}})^{\mathrm{T}}(\mathbf{c}_{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{X}})) \tag{12.22}$$

Here, $\boldsymbol{\mu}_{\mathbf{X}}$ is the matrix that has columns $\boldsymbol{\mu}_{\mathbf{X},i}$. If you observe the definition of the covariance given in the previous section, you will note that the diagonal of the covariance matrix defines the variance of a feature and that given the symmetry in the definition of the covariance, the covariance matrix is symmetric.

A third way of defining the covariance matrix is by using the definition in Eq. (12.10), i.e.,

$$\mathbf{\Sigma_X} = \frac{1}{m}(\mathbf{c_X^T c_X}) - \mathbf{\mu_X^T \mu_X} \tag{12.23}$$

The covariance matrix gives important information about the data. For example, by observing values close to zero, we can highlight independent features useful for classification. Very high or low values indicate dependent features that will not give any new information useful to distinguish groups in your data. PCA exploits this type of observation by defining a method to transform data in a way that the covariance matrix becomes diagonal. That is, all the values, but the diagonal, are zero. In this case, the data has not dependences, so features can be used to form groups. Imagine you have a feature that is not dependent on others, then by choosing a threshold you can clearly distinguish between two groups independently of the values of other features. Additionally, PCA provides information about the importance of elements in the new data. So you can distinguish between important data for classification or for compression.

12.5 Data transformation

We are looking for a transformation **W** that maps each feature vector defined in the set **X** into another feature vector for the set **Y**, such that the covariance matrix of the elements in **Y** is diagonal. The transformation is linear and it is defined as

$$\mathbf{c_Y} = \mathbf{c_X W^T} \tag{12.24}$$

or more explicitly

$$\begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,n} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ y_{m,1} & y_{m,2} & \cdots & y_{m,n} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{2,1} & \cdots & w_{n,1} \\ w_{1,2} & w_{2,2} & \cdots & w_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ w_{1,n} & w_{2,n} & \cdots & w_{n,n} \end{bmatrix} \tag{12.25}$$

Note that

$$\mathbf{c_Y^T} = \mathbf{W c_X^T} \tag{12.26}$$

or more explicitly

$$\begin{bmatrix} y_{1,1} & y_{2,1} & \cdots & y_{m,1} \\ y_{1,2} & y_{2,2} & \cdots & y_{m,2} \\ \vdots & \vdots & \cdots & \vdots \\ y_{1,n} & y_{2,n} & \cdots & y_{m,n} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{2,1} & \ldots & x_{m,1} \\ x_{1,2} & x_{2,2} & \ldots & x_{m,2} \\ \vdots & \vdots & \ldots & \vdots \\ x_{1,n} & x_{2,n} & \ldots & x_{m,n} \end{bmatrix} \tag{12.27}$$

To obtain the covariance of the features in $\mathbf{Y}$ based on the features in $\mathbf{X}$, we can substitute $\mathbf{c_Y}$ and $\mathbf{c_Y^T}$ in the definition of the covariance matrix as

$$\mathbf{\Sigma_Y} = \frac{1}{m}\left[(\mathbf{W}\mathbf{c_X^T} - E\left[\mathbf{W}\mathbf{c_X^T}\right])\ (\mathbf{c_X}\mathbf{W}^T - E\left[\mathbf{c_X}\mathbf{W}^T\right])\right] \tag{12.28}$$

By factorizing $\mathbf{W}$, we get

$$\mathbf{\Sigma_Y} = \frac{1}{m}\left[\mathbf{W}(\mathbf{c_X} - \boldsymbol{\mu}_\mathbf{X})^T(\mathbf{c_X} - \boldsymbol{\mu}_\mathbf{X})\mathbf{W}^T\right] \tag{12.29}$$

or

$$\mathbf{\Sigma_Y} = \mathbf{W}\mathbf{\Sigma_x}\mathbf{W}^T \tag{12.30}$$

Thus, we can use this equation to find the matrix $\mathbf{W}$ such that $\mathbf{\Sigma_Y}$ is diagonal. This problem is known in matrix algebra as matrix diagonalization.

12.6 Inverse transformation

In the previous section, we define a transformation from the features in $\mathbf{X}$ into a new set $\mathbf{Y}$ whose covariance matrix is diagonal. To map $\mathbf{Y}$ into $\mathbf{X}$, we should use the inverse of the transformation. However, this is greatly simplified since the inverse of the transformation is equal to its transpose, i.e.,

$$\mathbf{W}^{-1} = \mathbf{W}^T \tag{12.31}$$

This definition can been proven by considering that according to Eq. (12.30), we have that

$$\mathbf{\Sigma_X} = \mathbf{W}^{-1}\mathbf{\Sigma_Y}(\mathbf{W}^T)^{-1} \tag{12.32}$$

But since the covariance is symmetric, $\Sigma_\mathbf{x} = \Sigma_\mathbf{x}^T$ and

$$\mathbf{W}^{-1}\mathbf{\Sigma_Y}(\mathbf{W}^T)^{-1} = (\mathbf{W}^{-1})^T\mathbf{\Sigma_Y}((\mathbf{W}^T)^{-1})^T \tag{12.33}$$

which implies that

$$\mathbf{W}^{-1} = (\mathbf{W}^{-1})^T \ \text{ and } \ (\mathbf{W}^T)^{-1} = ((\mathbf{W}^T)^{-1})^T \tag{12.34}$$

These equations can only be true if the inverse of $\mathbf{W}$ is equal to its transpose. Thus, Eq. (12.26) can be written as

$$\mathbf{W}^{-1}\mathbf{c_Y^T} = \mathbf{W}^{-1}\mathbf{W}\mathbf{c_X^T} \tag{12.35}$$

i.e.,

$$\mathbf{W}^T\mathbf{c_Y^T} = \mathbf{c_X^T} \tag{12.36}$$

This equation is important for reconstructing data in compression applications. In compression, the data $\mathbf{c_X}$ is approximated by using this equation by considering only the most important components of $\mathbf{c_Y}$.

12.7 Eigenproblem

By considering that $\mathbf{W}^{-1} = \mathbf{W}^{\mathrm{T}}$, we can write Eq. (12.30) as

$$\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^{\mathrm{T}} = \mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{Y}} \tag{12.37}$$

We can write the right side in more explicit form as

$$\begin{aligned}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{Y}} &= \begin{bmatrix} w_{1,1} & w_{2,1} & \cdots & w_{n,1} \\ w_{1,2} & w_{2,2} & \cdots & w_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ w_{1,n} & w_{2,n} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \\ &= \lambda_1 \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \lambda_2 \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \cdots + \lambda_n \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix}\end{aligned} \tag{12.38}$$

Here, diagonal elements of the covariance have been named as λ using the notation used in matrix algebra.

Similarly, for the left side we have

$$\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^{\mathrm{T}} = \boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \cdots + \boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix} \tag{12.39}$$

i.e.,

$$\begin{aligned}&\boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \cdots + \boldsymbol{\Sigma}_{\mathbf{X}} \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix} \\ &= \lambda_1 \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \lambda_2 \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \cdots + \lambda_n \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix}\end{aligned} \tag{12.40}$$

Thus, we obtain that $\mathbf{W}$ can be found by solving the following equation:

$$\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{w}_i = \lambda_i\mathbf{w}_i \tag{12.41}$$

for $\mathbf{w}_i$ is the ith row of $\mathbf{W}$. λ_i defines the eigenvalues and $\mathbf{w}_i$ defines the eigenvectors. "Eigen" is actually a German word meaning "hidden" and there are alternative names such as characteristic values and characteristic vectors.

12.8 Solving the eigenproblem

In the eigenproblem formulated in the previous section, we know $\boldsymbol{\Sigma}_{\mathbf{x}}$, and we want to determine $\mathbf{w}_i$ and λ_i. To find them, first you should note that $\lambda_i\ \mathbf{w}_i = \lambda_i\ \mathbf{I}\mathbf{w}_i$, where $\mathbf{I}$ is the identity matrix. Thus, we can write the eigenproblem as

$$\lambda_i\mathbf{I}\mathbf{w}_i - \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{w}_i = 0 \tag{12.42}$$

or

$$(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_{\mathbf{X}})\mathbf{w}_i = 0 \tag{12.43}$$

A trivial solution is obtained for $\mathbf{w}_i$ equal to zero. Other solutions exist when the determinant det is given by

$$\det\,(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_{\mathbf{X}}) = 0 \tag{12.44}$$

This is known as the characteristic equation and it is used to solve the values of λ_i. Once the values of λ_i are known, they can be used to obtain the values of $\mathbf{w}_i$. According to the previous formulations, each λ_i is related to one in $\mathbf{w}_i$. However, several λ_i can have the same value. Thus, when a value λ_i is replaced in $(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_{\mathbf{X}})$ $\mathbf{w}_i = 0$, the solution should be determined by combining all the independent vectors obtained for all λ_i. According to the formulation in the previous section, once the eigenvectors $\mathbf{w}_i$ are known, the transformation $\mathbf{W}$ is simply obtained by considering $\mathbf{w}_i$ as its columns.

12.9 PCA method summary

The mathematics of PCA can be summarized in the following eight steps:

1. Obtain the feature matrix $\mathbf{c}_{\mathrm{x}}$ from the data. Each column of the matrix defines a feature vector.
2. Compute the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$. This matrix gives information about the linear independence between the features.
3. Obtain the eigenvalues by solving the characteristic equation $\det(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_{\mathbf{X}}) = 0$. These values form the diagonal covariance matrix $\boldsymbol{\Sigma}_{\mathbf{Y}}$. Since the matrix is diagonal, each element is actually the variance of the transformed data.

4. Obtain the eigenvectors by solving $\mathbf{w}_i$ in $(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_\mathbf{X})\mathbf{w}_i = 0$ for each eigenvalue. Eigenvectors should be normalized and linearly independent.
5. The transformation $\mathbf{W}$ is obtained by considering the eigenvectors as their columns.
6. Obtain the transform features by computing $\mathbf{c}_\mathbf{Y} = \mathbf{c}_\mathbf{X}\mathbf{W}^\mathrm{T}$. The new features are linearly independent.
7. For classification applications, select the features with large values of λ_i. Remember that λ_i measures the variance, and features that have large range of values will have large variance. For example, two classification classes can be obtained by finding the mean value of the feature with largest λ_i.
8. For compression, reduce the dimensionality of the new feature vectors by setting to zero components with low λ_i values. Features in the original data space can be obtained by $\mathbf{c}_\mathbf{X}^\mathrm{T} = \mathbf{W}^\mathrm{T}\mathbf{c}_\mathbf{Y}^\mathrm{T}$.

12.10 Example

Code 12.1 is a Matlab implementation of PCA, illustrating the method by a simple example with two features in the matrix `cx`.

In the example code, the covariance matrix is called `CovX` and it is computed by the Matlab function `cov`. The code also computes the covariance by evaluating the two alternative definitions given by Eqs (12.22) and (12.23). Note that the implementation of these equations divides the matrix multiplication by $m-1$ instead of m. In statistics, this is called an unbiased estimator and it is the estimator used by Matlab in the function `cov`. Thus, we use $m-1$ to obtain the same covariance values than the Matlab function.

To solve the eigenproblem, we use the Matlab function `eig`. This function solves the characteristic equation $\det(\lambda_i\mathbf{I} - \boldsymbol{\Sigma}_\mathbf{X}) = 0$ to obtain the eigenvalues and to find the eigenvectors. In the code, the result of this function is stored in the matrices $\mathbf{L}$ and $\mathbf{W}$, respectively. In general, the characteristic equation defines a polynomial of higher degree requiring elaborate numerical methods to find its solution. In our example, we have only two features, thus the characteristic equation defines the quadratic form

$$\lambda_i^2 - 1.208\lambda_i + 0.039 = 0 \tag{12.45}$$

for which the eigenvalues are $\lambda_1 = 0.0331$ and $\lambda_1 = 1.175$. The eigenvectors can be obtained by substitution of these values in the eigenproblem. For example, for the first eigenvector, we have

$$\begin{bmatrix} 0.033 - 0.543 & -0.568 \\ -0.568 & 0.033 - 0.665 \end{bmatrix} \mathbf{w}_1 = 0 \tag{12.46}$$

```
%PCA

%Feature Matrix cx. Each column represents a feature and
%each row a sample data
cx= [1.4000 1.55000
     3.0000 3.2000
     0.6000 0.7000
     2.2000 2.3000
     1.8000 2.1000
     2.0000 1.6000
     1.0000 1.1000
     2.5000 2.4000
     1.5000 1.6000
     1.2000 0.8000
     2.1000 2.5000 ];
[m,n]=size(cx);

%Data Graph
figure(1);
plot(cx(:,1),cx(:,2),'k+');     hold on;  %Data
plot(([0,0]),([-1,4]),'k-');    hold on;  %X axis
plot(([-1,4]),([0,0]),'k-');              %Y axis
axis([-1,4,-1,4]);
xlabel('Feature 1');
ylabel('Feature 2');
title('Original Data');

%Covariance Matrix
covX=cov(cx)

%Covariance Matrix using the matrix definition
meanX=mean(cx)        %mean of all elements of each row

cx1=cx(:,1)-meanX(1);    %substract mean of first row in cx
cx2=cx(:,2)-meanX(2);    %substract mean of second row in cx

Mcx=[cx1 cx2];
covX=(transpose(Mcx)*(Mcx))/(m-1) %definition of covariance

%Covariance Matrix using alternative definition
meanX=mean(cx);          %mean of all elements of each row

cx1=cx(:,1);    %substract mean of first row in cx
cx2=cx(:,2);    %substract mean of second row in cx

covX=((transpose(cx)*(cx))/(m-1) )-
((transpose(meanX)*meanX)*(m/(m-1)))

%Compute Eigenvalues and Eigenvector
[W,L]=eig(covX)     %W=Eigenvalues  L=Eigenvector
```

CODE 12.1

Matlab PCA implementation.

```
%Eigenvector Graph
figure(2);
plot(cx(:,1),cx(:,2),'k+');                          hold on;
plot(([0,W(1,1)*4]),([0,W(1,2)*4]),'k-');   hold on;
plot(([0,W(2,1)*4]),([0,W(2,2)*4]),'k-');
axis([-4,4,-4,4]);
xlabel('Feature 1');
ylabel('Feature 2');
title('Eigenvectors');

%Transform Data
cy=cx*transpose(W)

%Graph Transformed Data
figure(3);
plot(cy(:,1),cy(:,2),'k+');     hold on;
plot(([0,0]),([-1,5]),'k-');    hold on;
plot(([-1,5]),([0,0]),'k-');
axis([-1,5,-1,5]);
xlabel('Feature 1');
ylabel('Feature 2');
title('Transformed Data');

%Classification example
meanY=mean(cy);

%Graph of classification example
figure(4);
plot(([-5,5]),([meanY(2),meanY(2)]),'k:');  hold on;
plot(([0,0]),([-5,5]),'k-');                hold on;
plot(([-1,5]),([0,0]),'k-');                hold on;
plot(cy(:,1),cy(:,2),'k+');                 hold on;
axis([-1,5,-1,5]);
xlabel('Feature 1');
ylabel('Feature 2');
title('Classification Example');
legend('Mean',2);

%Compression example
cy(:,1)=zeros;
xr=transpose(transpose(W)*transpose(cy));

%Graph of compression example
figure(5);
plot(xr(:,1),xr(:,2),'k+');     hold on;
plot(([0,0]),([-1,4]),'k-');    hold on;
plot(([-1,4]),([0,0]),'k-');
axis([-1,4,-1,4]);
xlabel('Feature 1');
ylabel('Feature 2');
title('Compression Example');
```

CODE 12.1

(Continued)

Thus,

$$\mathbf{w}_1 = \begin{bmatrix} -1.11s \\ s \end{bmatrix} \tag{12.47}$$

where s is an arbitrary constant. After normalizing this vector, we obtain the first eigenvector

$$\mathbf{w}_1 = \begin{bmatrix} -0.74 \\ 0.66 \end{bmatrix} \tag{12.48}$$

Similarly, the second eigenvector is obtained as

$$\mathbf{w}_2 = \begin{bmatrix} 0.66 \\ 0.74 \end{bmatrix} \tag{12.49}$$

Figure 12.1 shows the original data and the eigenvectors. The eigenvector with the largest eigenvalue defines a line that goes through the points. This is the direction of the largest variance of the data.

Figure 12.2 shows the results obtained by transforming the features $\mathbf{c_Y} = \mathbf{c_X}\mathbf{W}^{\mathrm{T}}$. Basically, the eigenvectors become our main axes. The second feature has points more spread along the axis; this is related to a higher value in the eigenvector. Remember that for the transformed data, the covariance matrix is diagonal, thus there is not any linear dependence between the features.

If we want to classify our data in two classes, we should consider the variation along the second transformed feature. Since we are using the axis with the highest eigenvalue, the classification is performed along the axis with highest variation in the data. In Figure 12.3, we divide the points by the line defined by the mean value.

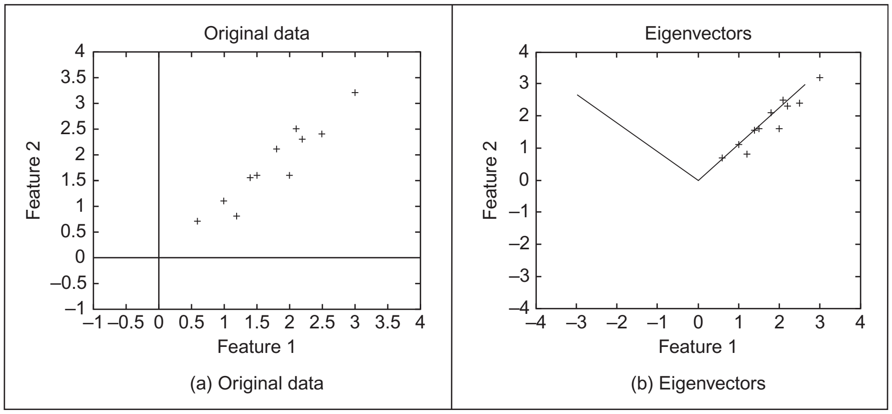

FIGURE 12.1

Data samples and the eigenvectors.

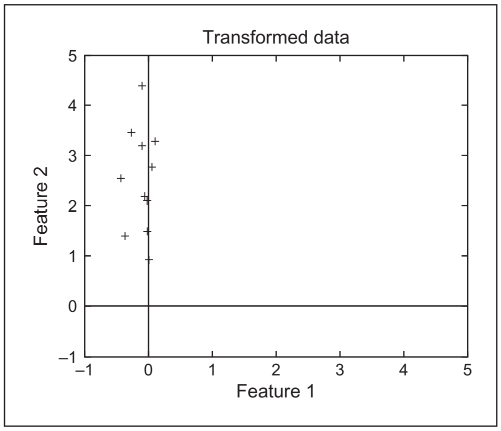

FIGURE 12.2

Transformed data.

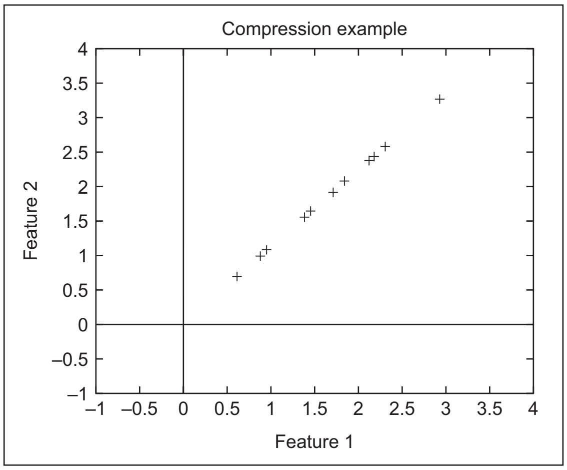

FIGURE 12.3

Classification via PCA.

For compression, we want to eliminate the components that have less variation; so in our example, we eliminate the first feature. In the last part of the Matlab implementation, data is reconstructed by setting to zero the values of the first feature in the matrix `cy`. The result is shown in Figure 12.4. Note that losing one dimension in the transformed set produces data aligned in the original space. So some variation in the data has been lost. However, the variation along the first eigenvector is maintained.

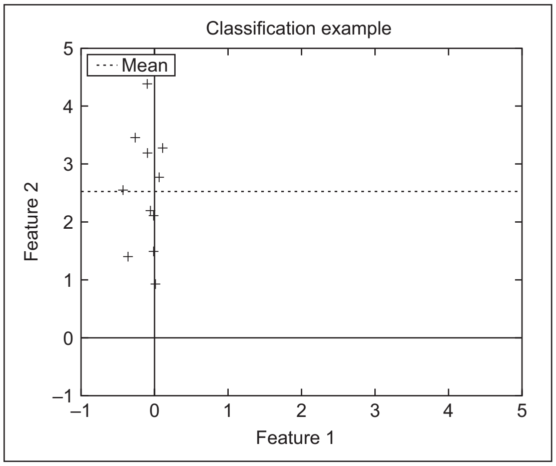

FIGURE 12.4

Compression via PCA.

Data with two features, as shown in this example, may be useful in some application such as reducing a stereo signal into a single channel. Other low-dimensional data such as three features can be used to reduce color images to gray level. However, in general, PCA is applied to data with many features. In these cases, the implementation is practically the same, but it should compute the eigenvalues by solving a characteristic equation defining a polynomial of high degree.

Data with many features are generally used for image classification wherein features are related to image metrics or to pixels. For example, face classification has been done by representing pixels in an image as features. Pixels are arranged in a vector and a set of eigenfaces is obtained by PCA. For classification, a new face is compared to the others by computing a new image according to the transformation obtained by PCA. The advantage is that PCA has independent features.

Another area that has extensively used PCA is image compression. In this case, pixels with the same position are used for the vectors. That is the first feature vector is formed by grouping all the values of the first pixel in all the images. Thus, when PCA is applied, the pixel value on each image can be obtained by reconstructing data with a reduced set of eigenvalues. As the number of eigenvalues is reduced, most information is lost. However, if you chose low eigenvalues, then the information lost represents low data variations.

Although classification and compression are perhaps the most important areas of application for PCA, this technique can be used to analyze any kind of data. You can find that PCA applications are continuously being developed in many research. For example, you can find that PCA has been used in applications as diverse as to compress animation of 3D models and to analyze data in spectroscopy. The difference in each application is how data is interpreted, but the fundamentals of PCA are the same.

12.11 References

Anton, H., 2005. Elementary Linear Algebra: With Applications. Wiley.

DeGroot, M.H., Schervish, M.J., 2001. Probability and Statistics, third ed. Addison Wesley.