# 1 Principal Component Analysis

*Principal component analysis* (PCA) was developed in 1901 by Karl Pearson and is now mainly used for data visualisation and making predictive models. Its mechanism can be understood as follows: if a multivariate dataset is visualised as a set of coordinates in a high-dimensional dataspace (1 axis per variable), PCA supplies the user with a low dimensional picture, a "shadow" of the dataset when viewed from its exact most informative viewpoint.

Thus, in plain words, it is a transformation of the variables in a given system. The goal is to concentrate the variability of the sample set as much as possible in each new variable. These new variables are called *principal components*. The first principal component accounts for as much variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

As a mathematical procedure (which is detailed below), PCA involves the calculation of the eigenvalue decomposition of the data covariance matrix, usually after some kind of data normalization.

## 1.1 Mathematical Background

Here we describe all mathematical tools used for PCA execution. In the next subsection we make a step-to-step outline of the PCA procedure.

### 1.1.1 Statistics - Mean, Variance and Standard Deviation

The entire subject of statistics is based around the ideas of collection, organization and interpretation of data. The prototype of a statistical task is to analyse a dataset in terms of the relationships between the individual elements.

There are some common, important and somewhat simple measures employed for achieving these goals. As far as PCA is concerned, we need to describe the mean, variance and the standard deviation.

$$mean(X) = m_x = \mu_x = \overline{x} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{1}$$

$$variance(X) = var_x = \sigma_x^2 = s_x^2 = \frac{\sum_{i=1}^{n} (X_i - mean)^2}{n} \tag{2}$$

$$standard\ deviation(X) = std(X) = \sigma_x = s_x = \sqrt{\frac{\sum_{i=1}^{n}(X_i - mean)^2}{n}} \quad (3)$$

As one can read out of the formulas, the mean is the average value, calculated by the division between the sum of all values and the number of values. The variance is a measure of the deviation from the mean, i.e. the more the values deviates from the mean, the bigger the variance is. The variance is always a positive real number (or zero). The standard deviation is simply the square root of the variance.

Note that we can find different notations for mean, variance and standard variation. Within the formulas above, we listed some of which are the most common.

Another peculiarity is that some approaches defines variance (and standard deviation) with a division by $n - 1$, not by $n$ as we did. The main reason is that empirically, if we are using a subset of our study field[1], our *real* variance is better approximated using $n - 1$.

As an illustration, one can note that the sets $A = \{13, 14, 15, 16\}$ and $B = \{8, 11, 18, 21\}$ have the same mean but different variance (and standard variation)[2]. One can notice another property of these measures: they do not depend on the order in which individual samples are presented.

### 1.1.2 Covariance and the Covariance Matrix

These three measures ($\mu$, $\sigma^2$ and $\sigma$) are all one-dimensional. It is useful to know how a variable varies with respect to another one. Covariance is such a measure and it is the variance concept applied to a 2-dimensional fit. It is calculated as follows:

$$covariance(X, Y) = cov(X, Y) = c_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \mu_x)(Y_i - \mu_y)}{n} \quad (4)$$

A covariance is always a measure between two dimensions. Note that if you calculate the covariance between a dimension and itself you get the variance. Another important property of the covariance is that $cov(X, Y) =$

[1] E.g. we are doing statistic of a city's preferences based on a sampling of two thousand citizens

[2] $\mu_A = \mu_B = 14.5$, $\sigma_A \sim 1.118$ and $\sigma_B \sim 5.22$

$cov(Y, X)$, $\forall\ X,\ Y$ . If you have more dimensions, you can get the covariances two-by-two. Thus, if one is dealing with X,Y and Z, it should be able to find and use $cov(X, Y)$, $cov(X, Z)$ and $cov(Y, Z)$. This leads us to the covariance matrix.

If we have $n$ different dimensions, there are $\frac{n!}{(n-2)!*2}$ covariance values. A convenient way to organize all the covariance values between all the studied dimensions is by calculating all of them and putting them on a matrix. The covariance matrix for a set of data with n dimensions is:

$$C^{n\times n} = \Big(c_{i,j}\Big) = \Big(cov(Dim_i, Dim_j)\Big) \tag{5}$$

Where $C^{n\times n}$ is a matrix with $n$ rows and $n$ columns. $Dim_x$ is the $x$th dimension. Therefore the covariance matrix of an $n$-dimensional dataset has $n$ rows and $n$ columns and each entry in the matrix is the result of calculating the covariance between two separate dimensions. Note that in the main diagonal one encounters the variances. Here is an example of a $3 \times 3$ covariance matrix:

$$\begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \tag{6}$$

### 1.1.3 Eigenvectors and Eigenvalues

Eigenvectors and eigenvalues are properties of a matrix. In general, a matrix acts on a vector by changing both its magnitude and its direction. However, a matrix may act on certain vectors by changing only their magnitude, and leaving their direction unchanged (or possibly reversing it). These vectors are the eigenvectors of the matrix. A matrix acts on an eigenvector by multiplying its magnitude by a factor, which is positive if its direction is unchanged and negative if its direction is reversed. This factor is the eigenvalue associated with that eigenvector.[3]

Formally, if A is a linear transformation, a non-null vector x is an eigenvector of A if there is a scalar $\lambda$ such that:

$$A\mathbf{x} = \lambda\mathbf{x}\,. \tag{7}$$

[3]An eigenspace is the set of all eigenvectors that have the same eigenvalue, together with the zero vector.

The scalar $\lambda$ is said to be an eigenvalue of $A$ corresponding to the eigenvector $\mathbf{x}$.

There are a number of different methods for finding eigenvectors and eigenvalues. A classical method is as follows:

$$\begin{gathered} A\mathbf{x} = \lambda\mathbf{x} \\ A\mathbf{x} - \lambda I\mathbf{x} = \mathbf{0} \\ (A - \lambda I)\mathbf{x} = \mathbf{0} \end{gathered}$$

If $(A - \lambda I)$ has an inverse, we can multiply both sides by $(A - \lambda I)^{-1}$ and find a trivial $\mathbf{x} = 0$. Therefore - so that $(A - \lambda I)$ doesn't have an inverse:

$$\det(A - \lambda I) = 0\,. \tag{8}$$

By noticing that the matrices $A$ and $I$ are given, different values for $\lambda$ can be found, which are the eigenvalues. Now all that is needed to do is to get back to equation (7) and find the eigenvectors ($\mathbf{x}$) related to the eigenvalues found.

It is quite important to know that for each eigenvalue there is a corresponding eigenvector. The number of different pairs is always equal to the order of the - square - matrix (excluding the trivial $\lambda = 0$ and $\mathbf{x} = 0$).

There are numerous computer implementations for finding eigenvalues and eigenvectors. These implementations are usually the way by which they are calculated, i.e. eigenfields are most frequently found by the use of a computer library or a standard routine present in a linear algebra package.

## 1.2 PCA - Mathematical Description

### 1.2.1 Definition

For a data matrix $X$ with zero mean in each data dimension (usually achieved by subtracting the mean), where each row represents a different object and each column gives the result for a particular variable, the PCA transformation is given by:

$$Y = XW \tag{9}$$

That is, the PCA transformation is a weighted linear sum of the original data matrix. The rest of this section is dedicated to describe a PCA implementation.

### 1.2.2 Data Normalization

It is mandatory that each dimension is centered at zero (this is usually achieved by subtracting the mean). If that is not fulfilled, the first principal component will not describe de direction of the maximum variance, but will correspond to the mean of the data.

$$\mathbf{X} = X - \overline{x} \times h \tag{10}$$

Where $h$ is a $1 \times n$ vector with every entry with value 1.
It is quite common to divide the data by its standard deviation.

$$\mathbf{X}' = \frac{\mathbf{X}}{\sigma} \tag{11}$$

This last procedure is not mandatory. It can lead to better results, but the best is to check both ways (dividing and not dividing by the standard deviation). In the following, one can substitute $\mathbf{X}$ with $\mathbf{X}'$.

### 1.2.3 Eigenvectors and Eigenvalues of the Covariance Matrix

Now we should find the covariance matrix of $\mathbf{X}$.

$$\begin{pmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_n) \\ cov(x_2, x_1) & cov(x_2, x_2) & \dots & cov(x_2, x_n) \\ \vdots & \vdots & \vdots & \vdots \\ cov(x_n, x_1) & cov(x_n, x_2) & \dots & cov(x_n, x_n) \end{pmatrix} \tag{12}$$

Where $x_i$ is the *ith* dimension of the normalized data matrix $\mathbf{X}$.

The eigenvalues and related eigenvectors of the covariance matrix are then calculated. The eigenvalues are ordered in inverse fashion. The related eigenvectors are ordered correspondingly.

### 1.2.4 Feature Vector

Feature vector is just a fancy name for a matrix of vectors. It is constructed by taking the eigenvectors that you want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns. Typically one takes the first two or three of these eigenvectors (which corresponds to the two or three largest eigenvalues) to form the feature vector.

$$Feature\ Vector = \Big( eigenvector_1\ eigenvector_2\ \ldots\ eigenvector_m \Big) \quad (13)$$

Note that $m \leq n$. And that the feature vector is a $n \times m$ matrix.

### 1.2.5 Final Data

The final data is just the multiplication of the original data with the feature vector, both transposed.

$$Final\ Data = Feature\ Vector^T \times \mathbf{X}^T \quad (14)$$

And this finishes the PCA procedure. Interpreting the results is concentrated around three observations. First, visualisation of the resulting data. Second, checking the eigenvalues and the proportion between the sum of all eigenvalues and the sum of the (usually) few eigenvalues related to the elected eigenvectors of the feature vector. At last, it is also useful to look at the eigenvectors used, to see what measures of the original data matrix contributes the most.

### 1.2.6 Visualization and Data Compression

Now the final data can be easily visualized in two or three dimensions (which is very useful, for example, as a hint as to how much a pattern recognition method will be able to succeed.

Another common use for PCA is data compression. Empirically, one can state that 80% (or more) of the variance is accounted in the first three new dimensions found by means of the PCA method in 90% of the cases.

One other, more conceptual use of the PCA, is to find what concepts of constructs are behind the observed measures. As a simple example, one can apply a PCA to a dataset concerning height and weight of a population. The PCA will concentrate almost all the variance in a new axis that can be understood as size or volume of each person.

## 1.3 Computer Implementation

Here we present some simple and direct computer implementations. Namely, we give a Python implementation and a Scilab implementation (and a C/C++ implementation?).

### 1.3.1 Python PCA implementation

```
#######
# NUM = number of retained principal components
# i.e. the final number of dimensions
NUM=2

import numpy

# The covariance matrix
c_m=numpy.cov(X)

# Eigenvalues end eigenvectors
# of the covariance matrix
eig_values, eig_vectors = numpy.linalg.eig(c_m)

# Ordering eigenvalues and eigenvectors
args=numpy.argsort(eig_values)[::-1]
eig_values=eig_values[args]
eig_vectors=eig_vectors[:,args]

# retaining only a selected number of eigenvectors
feature_vec=eig_vectors[:,:NUM]

# computing the final data
final_data=numpy.dot(feature_vec.T,X.T)
```

### 1.3.2 Scilab PCA implementation