Bruxelles, November 2018

Notes on Principal Component Analysis

camillejr.github.io/science-docs

Preface

These are notes on **Principal Component Analysis** (PCA), a dimensionality reduction technique in which the data set is reduced to maintain only the directions of the largest variance.

The deep and intuitive understanding of PCA requires the deep and intuitive understanding of basic linear algebra. I highly recommend a very pleasant to watch course by 3Blue1Brown YouTube channel, which, to my belief, will be everything you need to master about linear algebra to gain a great intuition behind PCA.

This document is still in preparation. Please feel free to contact me with any suggestions, corrections or comments.

Contents

1 Motivation for data reduction

There are several questions which stimulated the development of data reduction techniques and reduced-order modeling:

1. Can we send less data but preserve maximum information contained by the data (data compression)?
2. If the data were measurements from an experiment, can we predict what the outcome of another measurement will be?
3. Can we build a model that will make predictions about a system?
4. Can we make sense of a large data set with high-dimensionality (which cannot be plotted graphically)?

2 Datasets for PCA

For the rest of this document, we assume that the dataset for performing PCA is structured as follows: each column represends one variable and the rows correspond to variable observations (e.g. at different positions in space, or in time).

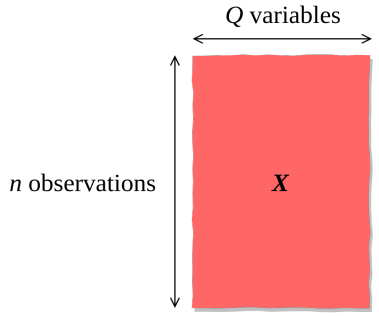

Figure 1: Data matrix for PCA.

This is also the data format that is needed for the MATLAB command `pca`. From the MATLAB documentation:

> `pca(X)`
> Rows of X correspond to observations and columns correspond to variables.

3 Covariance matrix

We start our discussion of PCA with explaining the concept of a *covariance matrix*, since computing this matrix is a starting point for performing PCA.

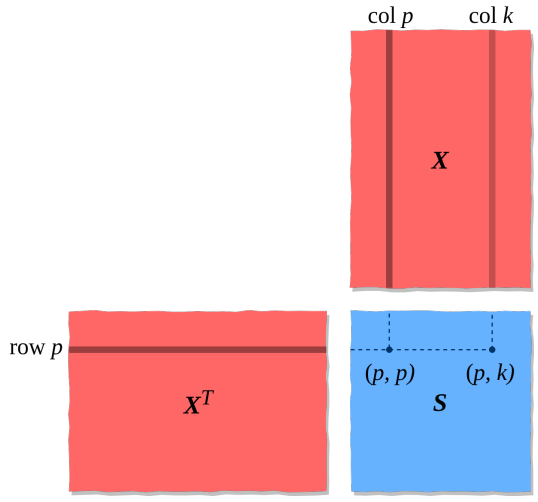

Figure 2: Covariance matrix $\boldsymbol{S}$ graphical interpretation.

$$\boldsymbol{S} = \frac{1}{n-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} \tag{1}$$

4 PCA workflow

The next step in PCA is to perform the eigendecomposition of the covariance matrix:

$$\text{eig}(\boldsymbol{S}) = [\boldsymbol{A}, \boldsymbol{\Lambda}] \tag{2}$$

The matrix $\boldsymbol{A}$ is a matrix of eigenvectors and it is size $(Q \times Q)$. Each eigenvector is called a *principal component* in the PCA world. The diagonal matrix $\boldsymbol{\Lambda}$ of size $(Q \times Q)$ is a matrix of corresponding eigenvalues.
The principal components now form a new basis in which we can represent our data set. We therefore perform a transformation of the original data matrix $\boldsymbol{X}$ from the original space to the new space represented by the eigenvectors of the covariance matrix. This transformation is achieved by the following multiplication:

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{A} \tag{3}$$

The new matrix $\boldsymbol{Z}$ is still our dataset $\boldsymbol{X}$ but represented in the basis associated with the matrix $\boldsymbol{A}$. It is also called the PC-*scores* since one may think of every element in this matrix as the "score" that the corresponding element in the original data matrix would get when represented in the new coordinate system after transformation.
We now approach the dimensionality reduction but first let's obtain the original data set back given that we know the PC-scores and the transformation matrix:

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A}^T \tag{4}$$

The above equation is our route back to obtain the original data set in which the PC-scores are projected on the basis associated with a transposed eigenvectors matrix A.
Suppose that we would like to find the approximation of the data matrix $\boldsymbol{X}$ with only q principal components (we project the PC-scores onto only q out of Q principal components).
We shrink the transformation matrix $\boldsymbol{A}$ to be of size $(Q \times q)$ (we only keep q principal components). To match the matrix sizes we also need to shrink in size the PC-scores matrix which originally is size $(n \times Q)$ - the same size as the data matrix $\boldsymbol{X}$. We will denote these truncated matrices $\boldsymbol{A_q}$ and $\boldsymbol{Z_q}$ respectively.
Projecting $\boldsymbol{Z_q}$ onto the basis $\boldsymbol{A_q^T}$ will result in an an approximation of the original data set:

$$\boldsymbol{X_q} = \boldsymbol{Z_q}\boldsymbol{A_q^T} \tag{5}$$

5 Why does it need to be eigenvectors?

In this section we come back to equation (2) and answer the question: why are principal components the eigenvectors of a covariance matrix? If you are new to PCA, this indeed might seem like a not-related-to-anything thing to do. So why is it eigenvectors? And why does it need to be eigenvectors of this special matrix called covariance matrix?

6 PCA example

7 Linear vs. non-linear PCA

The linear-PCA is adequate for data sets that can be approximated by a linear combination:

$$\text{Data} \approx A_1\vec{PC}1 + A_2\vec{PC}2 + \cdots + A_n\vec{PC}n \tag{6}$$

where $\vec{PC}1, \vec{PC}2, \ldots, \vec{PC}n$ are consecutive principal components (PCs).
The example of a data set that is suited for linear-PCA is presented in Figure 3. What is unique about such data set is that it has got a preferred direction in a lower dimension (in this case 1D) along which the data points are aligned. The first PC will be associated with this direction. There will also be a second PC which is perpendicular to the first one, however the weight of this second PC is much lower - there is much less variance of data along this other direction.
The data reduction that we can perform in our heads is that this data set almost aligns with a linear function $f(x) = x$ for $x \in \langle 1;3 \rangle$.

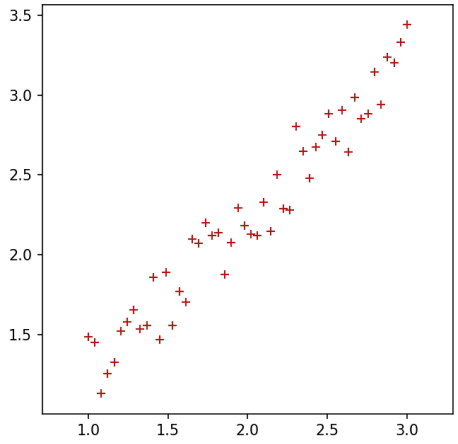

Figure 3: Data set for linear-PCA.

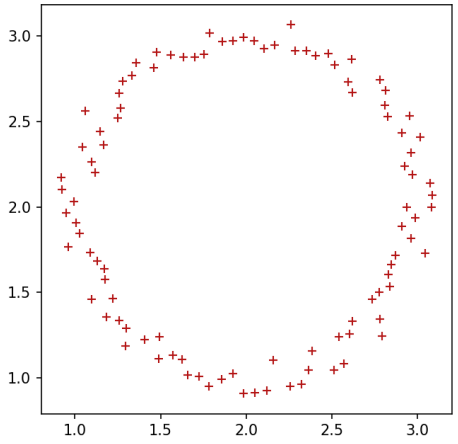

Figure 4: Data set for non-linear-PCA.

We may approximate the above data set simply by:

$$\text{Data} \approx A_1 P\vec{C}1 \tag{7}$$

When we take a look at a data set in Figure 4, we can see that the data is spread almost equally along two dimensions. The two PCs of such data set will have almost the same weight (the length of the two eigenvectors will be almost the same). It seems that there is no preferred direction in this data. So, does that mean that it cannot be reduced?

The underlying pattern in this data set is very clearly visible. Intuitively, we can say that the data points are almost aligned with a circle centered at $(2;2)$ with radius $r = 1$. If we knew a function $f(x)$ describing that circle, perhaps we might be able to write it in terms of just the first PC and approximate:

$$\text{Data} \approx f(P\vec{C}1) \tag{8}$$

where $f(P\vec{C}1)$ is essentially non-linear.

One idea to find the function f might be to associate the first PC with the radius of the circle (that is find the relation $r(PC1)$) and write:

$$f(x, P\vec{C}1)^2 = x^2 + r(P\vec{C}1)^2 \tag{9}$$

This will give us a centered approximation to the original data.

To conclude, the non-linear-PCA is helpful at times when there is no preferred direction in a data set, however there is an underlying non-linear relation that describes the data. The linear-PCA would in such case not give a satisfactory approximation after truncating the number of PCs.

A APP1

B APP2

References

[1] Ian T. Jolliffe *Principal Component Analysis*, Second Edition, 1986

[2] Gilbert Strang *Introduction to Linear Algebra*, Fifth Edition, 2016

[3] Jonathon Shlens, *A Tutorial on Principal Component Analysis*, 2016, `https://arxiv.org/abs/1404.1100`