Principal Component Analysis

April 15th, 2021

Technological University Dublin
Ireland
B00139079@mytudublin.ie

Abstract

This paper focuses on dimensionality reduction, reviewing the theoretical foundation of principal component analysis, and one of its variation which improves the computational part, the singular value decomposition. Finally, is presented a practical case applied to a small dataset to show the calculation step by step.

keywords: Principal component analysis, eigen values, singular value decomposition

1 Introduction

Principal Component Analysis (PCA) is one of the most popular techniques for dimensionality reduction, indeed it has been independently reinvented over the last years. So, the literature goes by a lot of different names depending on the field of study.The main idea behind PCA is trying to identify a line that best fits the data [1].
This method can be helpful for: compression, data visualization, dimension reduction for supervised learning. Considering a set of n observations with m variables, it is possible to use a number $k < m$ to explain the variance of the dataset. Furthermore, PCA can be used to identify relationship among the observations and the new variables, allowing to have a different and deeper interpretation of them. As many other data analysis techniques, PCA has a price which is the loss of the initial information [1]. Nevertheless, it is a method extensively used in several sectors because of its ability to limit the loss to an acceptable level. The trade off between loosing information and simplifying the analysis is usually still convenient [2].

2 Eigen Vectors, Eigen Values and Covariance

In order to explain PCA we need to introduce concepts that will be functional for the next sections, such as: eigen values, eigen vectors and covariance.
Considering an arbitrary matrix A and an arbitrary vector w

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} \qquad w = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

And multiplying them, we obtain:

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} \text{ x } \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 0 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

After matrix multiplication we find a vector which is 3 times the vector w we started with. So, after matrix multiplication we get the same vector out with a kind of scalar value being multiplied by it. This particular vector is known as eigen vector and the number 3, the scale value, is known as eigen value.
In general, eigen vectors satisfy the following equations:

$$\text{A w} = \lambda \text{ w}$$

Where λ is the eigen value and w the eigen vector. So, after matrix multiplication we get the same vector out that we started with, kind of scaled by a value λ.
After introducing the eigen values and eigen vectoris, we can further proceed with the concept of covariance.
Covariance is an index of joint variability of two random variables X and Y. It measures how much two variables change together.
Given two random variables X = [x_1, ...,x_n] and Y = [y_1, ..., y_n] respectively with average μ_X and μ_Y, the covariance is:

$$Cov(X,Y) = \frac{1}{n} \sum_{i=1}^{n} (X_i - \mu_X)(Y_i - \mu_Y)$$

In case Cov(X, Y) = 0, the variables are not correlated.

3 Principal Component Analysis (PCA)

After introducing the definition of eigen values and vectors, we can analyse PCA in more details. In order to reduce the dimensionality, PCA, tries to minimise the orthogonal distances from the data points to the line we want to fit the data. So, the data points are projected orthogonally on the line such as the distance is minimised and the spread of the data is maximised in the dimensional subspace. This is one of the difference with linear regression, where the distance taken into account is the distance on the y-axis. As shown in Figure 1.

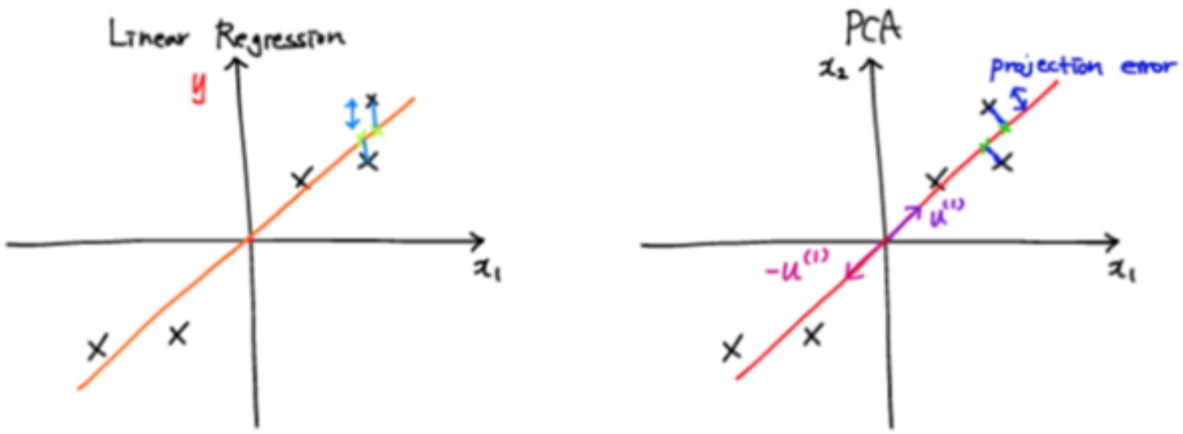

Figure 1. Comparison between linear regression and PCA about distances from data points and best fit line [3]

Given a data matrix X, composed by n rows and m attributes, so that the X matrix has n x m dimension. When n is a number higher than 3, visualizing those data becomes complicated [4]. So, we are interested in methods which allow to reduce this kind of big data into something that can be explained in fewer dimension, so that can be better visualized and gain an understanding of what is actually going on. The idea is that not all of the attributes are independent but there is actually correlation and patterns in the dataset, which is what we want to find out.
PCA is defined as the eigen decomposition of the covariance matrix:

$$\underset{m\times n}{\mathrm{X}^{\mathrm{t}}}\ \underset{n\times m}{\mathrm{X}} = \underset{m\times m}{\mathrm{W}}$$

So after computing the multiplication, it is obtained a square matrix *m x m* , as result of *(m x n) x (n x m)* multiplication. The outcome is a set of eigen vectors and eigen values, called "loadings". The loadings are the weights of each original variable when calculating the principal component. We can use them to describe our data, namely multiplying our initial matrix X by W we get the matrix T called "scores". The size of W is *m x m* matrix, so after multiplying, T is a *n x m* matrix too.

$$\underset{n\times m}{\mathrm{T}} = \underset{n\times m}{\mathrm{X}}\ \underset{m\times m}{\mathrm{W}}$$

Furthermore, each column of *W* is a "principal component", and they are ordered by their corresponding eigen values. So, the largest eigen value will be the first column and so on. The matrix *T* has the same size of our initial matrix because it is actually a transformation of our initial data. The structure of the data haven't been changed, they are just seen in a different way. We can observe that, so far, the transformed matrix didn't get any smaller. So, there is still the initial problem related to too high dimensionality. The fact that W columns are ordered, let us to truncate them at some point. Assuming that the first *Z* columns account for more the variance the data than the rest, we can truncate and keep only the first *Z* columns [5].

$$\underbrace{\begin{pmatrix} w1 & \dots & wr & wn \end{pmatrix}}_{w_z}$$

Then, we can multiply the initial matrix X by the new $w_z = w_1, ..., w_r$. The result will be a truncated representation of the data with dimension $n\ x\ r$, resulting from the following multiplication:

$$\underset{n\times r}{\mathrm{T}} = \underset{n\times m}{\mathrm{X}}\ \underset{m\times r}{\mathrm{W_Z}} \qquad (a)$$

3.1 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) method is mathematically identical to the PCA, but it is usually implemented differently in terms of the numerical linear algebra. Namely, how the single value decomposition is computed. The two methods can be considered interchangeable [6]. SVD also starts with the matrix X, which can be expressed as the product among U, Σ and V^*

$$X = U\,\Sigma\,V^* \qquad (b)$$

The * means *conjugate transpose*, if X has not complex values, which is usually the case for real data, then V^* is going to be just the transpose of V (V^T). The U matrix is known as the left singular vector, V is known as the right singular vector. The Σ matrix contains the singular values on its diagonal, more specifically Σ matrix looks like below

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{pmatrix}$$

The V matrix is equal to the W matrix we computed in the previous section, as prove that the two methos (PCA and SVD) are interchangeable. The formula (b) has specific properties such as:

(1) The values on the diagonal of Σ are proportional to the eigen values λ. They are ordered so that $\sigma_1 > \sigma_2 > \dots > \sigma_n$

(2) As W is identical to V, then first column of V correspond to the largest singular value, the second column correspond to the next smaller value and so on

(3) U and V are unitary matrices, namely U* U = I (where I is the identity matrix) and V* V = I

We saw in (a) that $T = X\ W$, we said that V can be interpreted as W, so

$$T = X\ W = X\ V$$

Using (b) we derive:

$$X\ V = U\,\Sigma\,V^*\ V$$

Using the property number (3), V* V = I, then

$$X\ V = U\,\Sigma \qquad (c)$$

We get the scores projection of our data after computing the singular value decomposition. The multiplication (c) is easy as Σ has only not null values on its diagonal, so we basically multiply every column of U by a single number [6]. If we want to consider only a specific subset, as

before, we can truncate Σ at some specific value. Then, we can execute the product between the U columns and the selected values of Σ.

$$T_r = U_r \, \Sigma_r$$

The advantage of SVD is that the computation is fast, thanks to the properties of U, Σ and V. SVD is more efficient especially when the number of columns of X is large [6].
The next important step is how to choose r, where the truncation is executed. Looking at the eigen values, there is a set of things that we can do in order to inform ourselves about how many dimension of the data can be truncated. Looking at the σ values, we can plot the cumulative values and choose the number of values where the "elbow" is shown. Namely, the points that are sufficients to explain the variance, and truncating the rest because there is not more variance in the data to be explained.

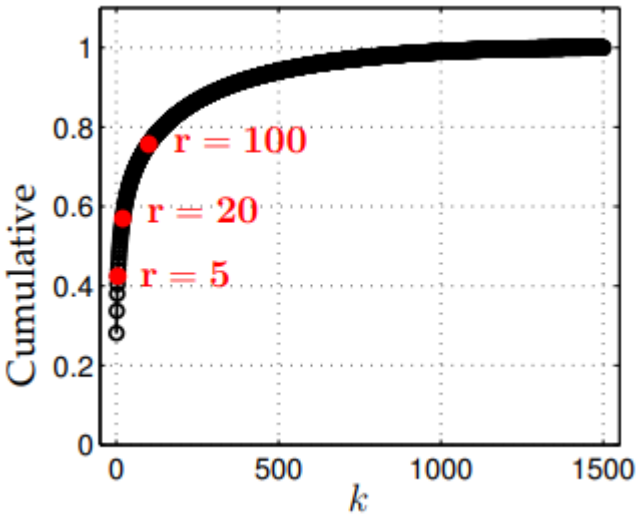

Figure 2. Representation of cumulative values of σ to choose where to truncate the data [6].

Of course, it will not be always easy to choose where to truncate the data [6]. In case of sharp curve it will be evident where the breaking point is. However, sometimes the data might produce a shallow curve which is more difficult to interpret, as every component actually does help to improve to explain the variance. In these cases, there is no clear breaking point and the decision is more complicate to make. A different approach might be setting the level of variance to be explained, e.g. setting this value equal to 95%. Wherever this threshold cut our data, that is our number of component we are going to choose. This method is more arbitrary than the previous one, but it does help when shallow curve does not help in the decision making.

4 Application of PCA

In the previous section, has been presented the theoretical foundation of dimensionality reduction methods (PCA and SVD). In this section, we are going to see a practical example of PCA. Considering a small students dataset, structured as follows:

Studente	Matematica	Scienze	Italiano
Carlo	6	7	8
Federico	5	7	6
Giuseppe	7	8	6
Michele	9	6	5
Alberto	7	7	7

Figure 3. Student dataset [7]

The dataset shows the final grades for five different students in three subjects: math, science and italian. The data can be represented as a matrix A,

$$A = \begin{bmatrix} 6 & 7 & 8 \\ 5 & 7 & 6 \\ 7 & 8 & 6 \\ 9 & 6 & 5 \\ 7 & 7 & 7 \end{bmatrix}$$

4.1 Calculation of covariance matrix

We can calculate the covariance matrix, as shown before in section 2, applying the following formula:

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_X)(Y_i - \mu_Y)$$

The result is the following covariance matrix, which shows on its diagonal the variance of the variables, namely the three different subjects

$$Cov = \begin{bmatrix} 2{,}2 & -0{,}5 & -0{,}9 \\ -0{,}5 & 0{,}5 & 0{,}25 \\ -0{,}9 & 0{,}25 & 1{,}3 \end{bmatrix}$$

The values that are not on the diagonal are the *covariance values*. If positive, it means the variables increase together. If negative, it means that when one variable increases, the other one decreases.

4.2 Calculation of eigen values and eigen vectors

The next step consists in calculating the eigen values λ_1, λ_2, λ_3 solving the following equation:

$$det(A - \lambda I) = 0$$

In our case it will be:

$$\det\left[\begin{bmatrix} 2{,}2 & -0{,}5 & -0{,}9 \\ -0{,}5 & 0{,}5 & 0{,}25 \\ -0{,}9 & 0{,}25 & 1{,}3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right]$$

which will lead to the solution: $\lambda_1 = \frac{3}{4}$, $\lambda_2 = \frac{-\sqrt{2545}+65}{40}$ and $\lambda_3 = \frac{\sqrt{2545}+65}{40}$.

Now, applying the same formula, we can calculate the eigen vector. As result of $det(A-\lambda I)v = 0$. The final outcome will be:

$$v = \begin{bmatrix} 0{,}5714 & -2{,}6862 & -1{,}6380 \\ -0{,}1429 & -100{,}4479 & 0{,}4480 \\ 1 & 1 & 1 \end{bmatrix}$$

4.3 Ordering the eigen vectors and choosing the principal components

We started with the aim of reducing the dimensionality, by projecting the initial space in a subset space where the eigen vectors are the new axes. In order to consider only the eigen vectors that are relevant in explaning the data, we have to look at the corresponding eigen values. The eigen vectors associated to the lowest eigen values will be deleted, because they provide the lowest amount of information about the data.
In our case $\lambda_1 = 0.75$, $\lambda_2 = 0.3638$ and $\lambda_3 = 2.8862$. So, we will choose the first and third eigen vectors, as they are associated to the highest eigen values. Passing from a three-dimensional space to a two-dimensional space. The W matrix will be:

$$W = \begin{bmatrix} -1{,}6380 & 0{,}5714 \\ 0{,}4480 & -0{,}1429 \\ 1 & 1 \end{bmatrix}$$

4.4 Generating the principal components

This is the last step, where the principal components are generated by multiplying the transpose of W and the transpose of the initial matrix A,

$$PC = W^t A^t = \begin{bmatrix} -1{,}6380 & 0{,}4480 & 1 \\ 0{,}5714 & -0{,}1429 & 1 \end{bmatrix} \begin{bmatrix} 6 & 5 & 7 & 9 & 7 \\ 7 & 7 & 8 & 6 & 7 \\ 8 & 6 & 6 & 5 & 7 \end{bmatrix} =$$

$$= \begin{bmatrix} 2{,}9104 & 2{,}5484 & -0{,}2796 & -5{,}4516 & 0{,}2724 \\ 1{,}1429 & -1{,}4286 & -0{,}4286 & 0{,}00 & 0{,}7143 \end{bmatrix}$$

Given the new data points, we can project them in the new subset space (Figure 4). So, in

the end we obtained the original data points but represented considering the eigen vectors as new axes. The new selected axes represent the principal components.

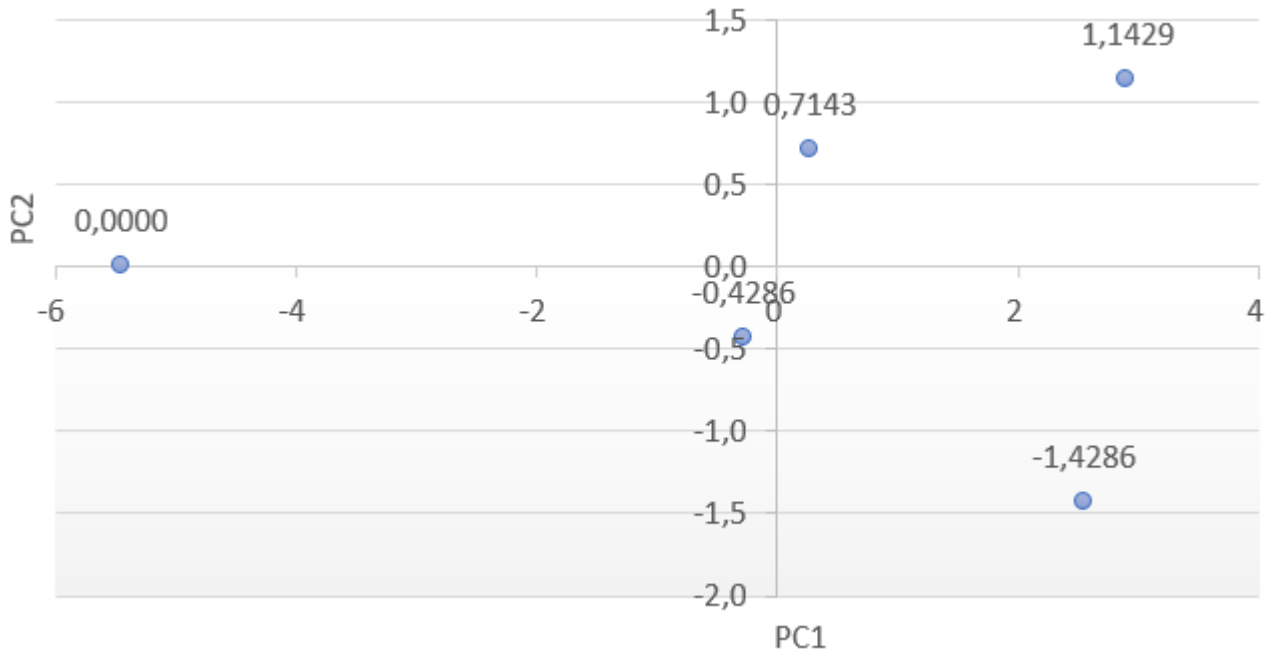

Figure 4. Principal components [7]

5 Conclusion

The dimensionality reduction has an important role in machine learning, especially when dealing with large datasets. This technique helps in semplifying the choice of the most relevant variables and, furthermore, it helps in improving classification and prediction algorithm. PCA is just one way to make sense of data. There are other methods which can be more complicated to implement, like: heatmaps, t-SNE plots, and multidimensional scaling plots (MDS).

References

[1] Steven L. Brunton, J. Nathan Kutz. *Data Driven Science & Engineering Machine Learning, Dynamical Systems, and Control*, 2019

[2] Richard Cangelosi, Alain Goriely. *Component retention in principal component analysis with application to cDNA microarray data*, 2007

[3] Principal Component Analysis the Machine Learning Perspective. *https://towardsdatascience.com/principal-component-analysis-the-machine-learning-perspective-part-2-a2630fa3b89e*

[4] Yunzhen Yao, Liangzu Peng, Manolis C. Tsakiris. *Unlabeled Principal Component Analysis*, 2021

[5] L. Paul, A. A. Suman, Nahid Sultan. *Methodological analysis of Principal Component Analysis* , 2014

[6] X Cao, Y He. *Singular vector decomposition based adaptive transform for motion compensation residuals*, 2013

[7] Bruno Farabegoli. *Analisi delle componenti principali: algoritmi e applicazioni, Master's thesis*, 2016