Principal Component Analysis Unveiled: Decoding the Mathematics Behind Dimensionality Reduction

Umberto Michelucci
TOELT LLC, Artificial Intelligence Research and Development, Switzerland
Lucerne University of Applied Sciences, Lucerne, Switzerland
{umberto.michelucci@toelt.ai}

January 15, 2024

Abstract

Principal Component Analysis (PCA) is a widely used method to extract relevant and sometime hidden information in complex datasets. This short paper describes briefly what PCA means, what its assumptions are, and shows one analytical approach to solve the problem.

1 Introduction

Principal Component Analysis (PCA) is a widely used method to extract relevant and sometimes hidden information in complex datasets. Very often, depending on how data collection is done (we will give an example later), we end up with a lot more variables and information than what is really relevant to the problem. PCA is called a *dimensionality reduction* technique, because by extracting only relevant information, it effectively reduces the number of dimensions of a problem ignoring redundant and irrelevant information.

To better understand what is meant, suppose that we want to study the movement of an object attached to a spring[1]. Assuming the right initial conditions, we know from physics that the movement will be approximately (the spring may not be ideal, the surface on which the experiment is done not completely flat, the initial condition not perfect, etc.) along a line. Let us suppose that we do not know much about the problem, and we decide to study it with three cameras positioned randomly around the system (spring plus object). Let us also assume that each camera measures the position of the object at regular intervals. We already know that three cameras are too much, and one would be more than enough. But by redundantly setting up our measurement system, we generate data with a much higher dimensionality than necessary. Very often, as

[1]Example adapted from [1].

experimenters, we often have no idea about which measurement setup reflect in the best possible way how a specific system works.

The question we are trying to address is *can we remove the redundant information and extract the really physical relevant data from our measurements?*

An additional complications that make our life more difficult is that in real life, systems and measurement systems are imperfect. This means that in our example, we will not measure a perfect line of points. The spring may be imperfect, the object not completely symmetric, friction on the surface may be irregular and make the spring oscillates in multiple directions, the movement will slow down with time, and so on. This means that we will not observe a perfect line of points, but a cloud (more or less spread) depending on a variety of factors that we cannot always control.

2 Basis of a Vector Space

Let us first briefly discuss some necessary mathematical formalism. First of all let us consider a **vector space** V. Intuitively, a vector space V is a set of vectors. For example the entire set of all bi-dimensional vectors (v_1, v_2) (in other words vectors with two components). The formal definition is more complex but is not necessary for our discussion (I give it in the next section, which can be safely skipped without compromising understanding). Secondly let us define what a basis of a vector space is.

Definition 2.1 (Definition of a basis)**.** *A basis of a vector space V, is a set B of vectors such that every vector in V can be written in a unique way as a linear combination of the vectors in B.*

In vector notation, a **basis** is described by a set of unit vectors (in other words, each having a length of one). An example of a basis in $\mathbb{R}^2$ is given by the vectors $(1, 0)$ and $(0, 1)$. Note that the vectors of a basis do not need to be orthonormal. For example, the vectors $(1, 0)$ and $(1, 1)$ form a basis for $\mathbb{R}^2$ but they are not orthonormal. We must determine how a transformation of basis can be represented using matrix notation. Note that we will consider here only a *linear* basis transformation by considering only a linear combination of the existing unit vectors that describe our original basis.

2.1 ★ Definition of a Vector Space

To define a vector space, we first need to define a field.

Definition 2.2. *A field $\mathbb{F}$ is a set of numbers, such that if $a, b \in \mathbb{F}$ then $a + b, a - b, a/b \in \mathbb{F}$. Assuming $b \neq 0$ in the division.*

Examples of fields are $\mathbb{R}, \mathbb{N}, \mathbb{Z}$, etc. An example of a set that is **not** a field is $\{1, 5\}$, as 1-5 is not in the set. Now we can define a vector space V.

Definition 2.3. *A vector space V, consists in a set of elements called vectors, a field $\mathbb{F}$ of elements called scalars, and two operations.*

- *Addition: that takes two vectors $v, w \in V$ and produce a vector $v + w \in V$.*
- *Scalar Multiplication: that takes one vector $v \in V$, and a scalar $a \in \mathbb{F}$ and produces a vector $av \in V$.*

The operations must satisfy the following axioms.

- *Associativity of addition: $(u + v) + w = u + (v + w)$, with $u, v, w \in V$.*
- *Zero vector: in V exist a **zero** vector $\mathbf{0}$ such that $v + \mathbf{0} = v \ \forall v \in V$.*
- *Negative vector: in V exist, $\forall v \in V$, a vector $-v$ such that $v - v = \mathbf{0}$.*
- *Associativity of multiplication: $(ab)u = a(bu) \ \forall a, b \in \mathbb{F}, u \in V$.*
- *Distributivity: $(a + b)u = au + bu \ \forall a, b \in \mathbb{F}, u \in V$.*
- *Unitarity: $1u = u \ \forall u \in V$.*

As you may notice, the definition is quite lengthy, but not necessary for a first understanding of PCA.

2.2 ★ Linear Transformartions (maps)

In general a linear transformation (also called *map*) f from a vector space V into itself $f : V \to V$, is one that satisfies the two properties

1. **Additivity**: $f(x + y) = f(x) + f(y)$
2. **Homogeneity**: $f(cx) = cf(x)$

There is an important theorem on linear transformations that is quite relevant to our discussion.

Theorem 1. *(a) Every linear transformation f between finite-dimensional vector spaces can be obtained by multiplication with a unique matrix. (b) Matrix multiplications are linear transformations.*

Proof. Let us first prove part (a) for a linear transformation between $V = \mathbb{R}^n$ and $V = \mathbb{R}^n$. V have, by hypothesis, a basis that we can indicate with $(\mathbf{v}_1, ..., \mathbf{v}_n)$. Now consider a linear transformation $f : V \to V$. We can write for a generic vector $\mathbf{v} \in V$

$$\mathbf{v} = \alpha_1 \mathbf{v}_1 + \cdots + \alpha_n \mathbf{v}_n \tag{1}$$

and that, assuming we are using the basis given by the vectors $\mathbf{v}_i$, $\mathbf{v}$ will be simply indicated by its components $\mathbf{v} = (\alpha_1, \cdots, \alpha_n)$. Now we can write for $f(v)$

$$\mathbf{u} \equiv f(\mathbf{v}) = \alpha_1 f(\mathbf{v}_1) + \cdots \alpha_n f(\mathbf{v}_n) \tag{2}$$

or in matrix notation

$$\mathbf{v} = (f(\mathbf{v}_1) \cdots f(\mathbf{v}_n)) \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \tag{3}$$

where $(f(\mathbf{v}_1), \cdots, f(\mathbf{v}_n))$ is a matrix where the i^{th} column is given by the vector $f(\mathbf{v}_i)$. This concludes the proof of (a). The proof of (b) is trivial, as matrix multiplication is a linear transformation (and if you are not convinced you can easily verify that). □

This is highly relevant to our discussion since **PCA does nothing else than expressing the original data on a new basis that has been obtained by a linear combination of the original basis vectors.**

In general, to make a *linear* change of basis, it suffices to multiply our identity matrix (our original basis) by a **transformation matrix** T. Let us give an example, to make the discussion more concrete. Let us consider the Euclidean space $\mathbb{R}^2$ with basis vectors $v_1 = (1, 0)$ and $v_2 = (0, 1)$. Now let us suppose that we want to rotate the axis of an angle α. It is easy to see, with some trigonometry, that the new vectors u_1 and u_2, obtained by rotating v_1 and v_2, are

$$u_1 = (\cos\alpha, \sin\alpha) \tag{4}$$
$$v_1 = (-\sin\alpha, \cos\alpha) \tag{5}$$

The formula can be easily understood by looking at Figure 1. It is then easy to

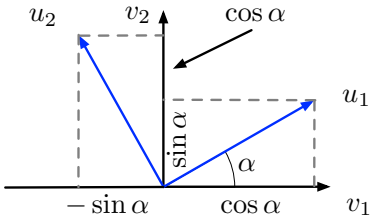

Figure 1: A change of basis from (v_2, v_2) to (u_1, u_2) obtained by rotating the axis by an angle α.

express the transformation as a matrix operation. Let us define the basis as a matrix (with each vector v_1 and v_2 as column vectors).

$$V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{6}$$

we can define the transformation matrix T as

$$T = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \tag{7}$$

Then the new basis matrix U (where we have indicated with $u_{1,x}$ the component of the vector u_1 along v_1, etc.) with the new basis vectors as columns

$$U = \begin{pmatrix} u_{1,x} & u_{2,x} \\ u_{1,y} & u_{2,y} \end{pmatrix} \tag{8}$$

can be calculated with

$$U = TV \tag{9}$$

It is easy to check that this is correct. Let's now consider a point $P = p_1 v_1 + p_2 v_2$ (we will consider the basis vectors in matrix formalism, as column vectors), that can be written in vector form as

$$P = p_1 v_1 + p_2 v_2 = p_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + p_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \tag{10}$$

How can we express P in the new basis (that we will indicate with P_V)? This can now be done easily using the transformation matrix T. In fact we have

$$P_V = TP = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} p_1 \cos\alpha - p_2 \sin\alpha \\ p_1 \sin\alpha + p_2 \cos\alpha \end{pmatrix} \tag{11}$$

This can be easily verified by drawing a diagram similar to Figure 1. In general, given a dataset X where each column is a single sample of our data, when we have a linear transformation T, the new representation Y in the new basis will be given by

$$Y = TX \tag{12}$$

that is the natural expansion of Equation 11. In fact in Equation 11 we just calculated the new coordinates of a single point, in Equation 12 we changed the coordinates of **all** points at the same time (since each is a column in the matrix X).

Equation 11 represents a change of basis. In general T is a rotation and a stretch of the space. The rows of T represents the new basis vectors. Note that the matrix T can have a higher dimension that 2×2 as in our example. In fact X is our datasets, and that means that the vertical dimension is given by the number of features that we have and that can be very high. This can be easily seen by writing the equation in the following form (assuming the matrix T has m rows and our dataset has n data sampels, or number of columns).

$$TX = \begin{pmatrix} — \, \boldsymbol{t}_1 \, — \\ \vdots \\ — \, \boldsymbol{t}_m \, — \end{pmatrix} \begin{pmatrix} | & & | \\ \boldsymbol{x_1} & \cdots & \boldsymbol{x_n} \\ | & & | \end{pmatrix} = \begin{pmatrix} \boldsymbol{t}_1 \cdot \boldsymbol{x}_1 & \cdots & \boldsymbol{t}_1 \cdot \boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{t}_m \cdot \boldsymbol{x}_1 & \cdots & \boldsymbol{t}_m \cdot \boldsymbol{x}_n \end{pmatrix} \tag{13}$$

And as you can see, the first row of the matrix TX will be the projection of the the dataset along the vector $\boldsymbol{t}_1$, the second along $\boldsymbol{t}_2$ and so on.

In general, any linear transformation can be interpreted, in geometric terms, as a rotation, a stretch or some other geometrical modification [2] (for example reflection with respect to some line). In Table 1 a list of possible transformations [2] and the respective transformation matrices in two dimensions is reported.

3 PCA

Now that we have the formalism out of the way, let us go back to the original question. With this formalism we still have to answer two main questions.

Geometrical Transformation	**Transformation Matrix**
Rotation of an angle α	$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$
Reflection through the x-axis	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Reflection through the y-axis	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
Reflection through the origin	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Table 1: List of transformation matrices for specific geometrical modifications [2].

- What is the best way to re-write our dataset X? Or in other words, how can we decide what is a good and what is a bad representation (or in other words a good or bad basis)?
- What is a good choice for T?

To answer the first question, PCA relies on two key assumptions:

1. The directions along which the data show the largest variance are the ones that contain the most relevant and interesting information.
2. Features that are correlated to each other are *redundant*, or in other words not useful in describing the phenomena we are studying.

Hypothesi 1 can be grasped by considering that if there is minimal variance along a certain direction, it implies that the data remain largely unchanged in that direction, suggesting a lack of significant information. Take, for instance, our spring example: the direction orthogonal to the spring's extension bears little relevance to describing the phenomenon. In this case, the movement of objects attached to the spring in the perpendicular direction would be negligible (or even non-existent), resulting in very low variance along this axis. Therefore, our objective is to identify a new basis on which the variance is maximized along some (or all) of its directions.

To summarize again our goal is twofold: firstly, we need to identify the directions along which the variance is largest, and secondly identify any redundant features (so that we can safely ignore them). In the next section I will describe one way of achieving this.

4 Covariance Matrix

One problem that we have not discussed, is how tp determine if one variable is redundant in relation to another. This is essentially about assessing whether the

two variables are correlated. For instance, consider a dataset with two features represented as $D = \{x_i, x_i\}_{i=1}^{N}$. It's clear that in this case, we don't require both features to describe D, as they are identical. One feature would certainly be adequate, meaning the second is *redundant*.

To explain this in more general terms let us consider two set of measurements: $X = \{x_i\}_{i=1}^{N}$ and $Y = \{y_i\}_{i=1}^{N}$. To simplify the equations, let us suppose that they both have zero mean. In this case the *covariance* of X and Y, given by the equation

$$\sigma_{XY}^2 = \frac{1}{N}\sum_{i=1}^{N} x_i y_i \tag{14}$$

measures the degree of relationship between the two variables. A large positive value indicates positively correlated data (if one grows, so does the other). A negative value indicates negatively correlated data (if one grows, the other decreases). By writing X and Y as matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ with dimensions $(1, N)$ we can re-write the covariance as a matrix product

$$\sigma_{XY}^2 = \frac{1}{N}\boldsymbol{X}\boldsymbol{Y}^T \tag{15}$$

In general if we have a large dataset we can generalize the definition. Let us consider a dataset $\boldsymbol{X}$

$$\boldsymbol{X} = \begin{pmatrix} — \ \boldsymbol{x}_1 \ — \\ \vdots \\ — \ \boldsymbol{x}_m \ — \end{pmatrix} \tag{16}$$

where each row contains a complete measurement (all the fatures), while each column all measurements of a specific feature[2]. In this case we can write the **covariance matrix $\boldsymbol{C}$** as

$$\boldsymbol{C} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T \tag{17}$$

Note that $\boldsymbol{C}$ is a square matrix, its diagonal terms are the variances of the features and the off-diagonal terms the covariance between pairs of different features.

Remembering our hypothesis, that what is interesting happens along directions that have large variance and small co-variance (low redundancy), we can make the following statements:

1. Large diagonal elements indicate interesting features.
2. large off-diagonal terms indicate large redundancy.

You may, at this point, see where we are going with this. Our goal is to **find a new basis (diagonalise the co-variance matrix) to (1) maximise the variance (so the diagonal elements) and (2) minimise redundancy (the**

[2]This is done in the opposite way as before, since in this case we are interested in the covariance between features and not measurements.

off-diagonal elements). This is accomplished by diagonalizing the covariance matrix, which is essentially the entirety of what PCA entails.

To make PCA easy to use and calculate, PCA assumes that the new basis will be an **orthonormal matrix**, or in other words that the vectors of the new basis are orthogonal to each other.

4.1 Overview of Assumptions

Let us review all the assumptions PCA makes.

1. *Linearity*: the change of basis we are searching for is a linear transformation (there are other approaches that lift this assumption, like t-SNE (t-distributed Stochastic Neighbour Embedding) [3]).

2. *Large variance is important*: this assumption is often safe to make, but it is a strong assumption that is not always correct. For example, if you are studying the transverse oscillations of a spring due to imperfections in your system, PCA may simply ignore those effects since the variance is minimal along the transverse direction.

3. *The new basis is orthogonal*: this makes using PCA fast and easy, but it does not work every time. It may well be that there are directions where the interesting phenomena occur that are not orthogonal to each other.

5 PCA with Eigenvectors and Eigenvalues

The problem that PCA solves can be stated as follows.

Problem 5.1 (PCA Problem Statement)**.** *Find an orthonormal matrix* $\boldsymbol{T}$*, such that, given a dataset* $\boldsymbol{X}$ *and* $\boldsymbol{Y} = \boldsymbol{TX}$ *the matrix* $\boldsymbol{C} = (1/N)\boldsymbol{YY}^T$ *is a diagonal matrix.*

Let us start by writing $\boldsymbol{C}$.

$$\begin{aligned}\boldsymbol{C} &= \frac{1}{N}\boldsymbol{YY}^T \\ &= \frac{1}{N}(\boldsymbol{TX})(\boldsymbol{TX})^T \\ &= \frac{1}{N}\boldsymbol{TXX^TTT} \\ &= \boldsymbol{T}\left(\frac{1}{N}\boldsymbol{XX}^T\right)\boldsymbol{T}^T \\ &= \boldsymbol{TC_XT}^T\end{aligned} \tag{18}$$

where with $\boldsymbol{C_X}$ we have indicated the covariance matrix of $\boldsymbol{X}$. Now you should know that any symmetric matrix $\mathbf{M}$ is diagonalized by a matrix composed of its

eigenvectors organized as columns. The trick now is to choose $\boldsymbol{T}$ to be a matrix where each row is an eigenvector of $\left(\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\right)$. Let us indicate this matrix with $\boldsymbol{E}^T$, and let us indicated with $\boldsymbol{D}$ the diagonalised version of $\left(\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\right)$.

$$\begin{aligned}
\boldsymbol{C} &= \boldsymbol{T}\boldsymbol{C}_{\boldsymbol{X}}\boldsymbol{T}^T \\
&= \boldsymbol{T}(\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T)\boldsymbol{T}^T \\
&= \{\text{Remember } \boldsymbol{E}^T = \boldsymbol{T}\} \\
&= (\boldsymbol{T}\boldsymbol{T}^T)\boldsymbol{D}(\boldsymbol{T}\boldsymbol{T}^T) \\
&= \{\text{Since the matrix } \boldsymbol{T} \text{ is orthonormal then } \boldsymbol{T}^T = \boldsymbol{T}^{-1}\} \\
&= (\boldsymbol{T}\boldsymbol{T}^{-1})\boldsymbol{D}(\boldsymbol{T}\boldsymbol{T}^{-1}) \\
&= \boldsymbol{D}
\end{aligned} \tag{19}$$

So this choice of $\boldsymbol{T}$ diagonalizes $\boldsymbol{C}$. This is how to calculate the transformation matrix $\boldsymbol{T}$, simply calculating the eigevectors of the covariance matrix of our data $\boldsymbol{X}$. There is actually another way of finding $\boldsymbol{T}$, and that is by using singular value decomposition (SVD), but that goes beyond the scope of this paper. This is what python and R libraries typically use, and they typically return the dataset in the new basis, already prepared for you.

5.1 One Implementation Limitation

All we discussed sound good, but one limitation of PCA is that when you want to diagonalise numerically the matrix $\left(\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\right)$, this has to fit completely in memory. With large dataset, this may be a problem (often is). This is why you may encunter difficulties in doing this. One variant of PCA that you may consider, is IPCA (Incremental Principal Component), that goes beyond the scope of this paper, but that is available in the Python library scikt-learn [4].

6 Conclusion

This short paper describes briefly what PCA is, what its assumptions are, and shows one analytical approach to solve Problem 5.1. The reader should be aware that there are different ways of solving Problem 5.1, the most notable one being single value decomposition (and that is how it is solved in the Python implementation of scikit-learn [5], for example).

References

[1] Lindsay I Smith. A tutorial on Principal Components Analysis.

[2] John Gilbert. Linear Transformations. `https://web.ma.utexas.edu/users/gilbert/`. [Last accessed 1st Oct. 2023].

[3] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[4] Incremental PCA. `https://scikit-learn/stable/auto_examples/decomposition/plot_incremental_pca.html`. [Last accessed 1st Oct. 2023].

[5] sklearn.decomposition.PCA. `https://scikit-learn/stable/modules/generated/sklearn.decomposition.PCA.html`. [Last accessed 1st Oct. 2023].