

[Get started](#)[Open in app](#)[Follow](#)

569K Followers



ESSENTIAL MATH FOR DATA SCIENCE

Essential Math for Data Science: Eigenvectors and application to PCA

Understand eigenvectors and eigenvalues and how they relate to Principal Component Analysis (PCA)



Hadrien Jean · Feb 23 · 9 min read



[Get started](#)[Open in app](#)

Matrix decomposition, also called *matrix factorization* is the process of splitting a matrix into multiple pieces. In the context of data science, you can for instance use it to select parts of the data, aimed at reducing dimensionality without losing much information (as for instance in Principal Component Analysis, as you'll later in this post). Some operations are also more easily computed on the matrices resulting from the decomposition.

In this article, you'll learn about the eigendecomposition of a matrix. One way to understand it is to consider it as a special change of basis (more details about change of basis in [my last post](#)). You'll first learn about eigenvectors and eigenvalues and then you'll see how they can be applied to Principal Component Analysis (PCA). The main idea is to consider the eigendecomposition of a matrix A as a change of basis where the new basis vectors are the eigenvectors.

Eigenvectors and Eigenvalues

As you can see in Chapter 7 of [Essential Math for Data Science](#) you can consider matrices as linear transformations. This means that if you take any vector u and apply the matrix A to it, you obtain a transformed vector v .

Take the example of:

$$u = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}$$

and

$$A = \begin{bmatrix} 1.2 & 0.9 \\ 0 & -0.4 \end{bmatrix}$$

If you apply A to the vector u (with the matrix-vector product), you get a new vector:

[Get started](#)[Open in app](#)

Let's draw the initial and transformed vectors:

[Get started](#)[Open in app](#)

[Get started](#)[Open in app](#)

Figure 1: Transformation of the vector u by the matrix A into the vector v .

Note that, as you can expect, the transformed vector v doesn't run in the same direction as the initial vector u . This change of direction characterizes most of the vectors you can transform by A .

However, take the following vector:



[Get started](#)[Open in app](#)

Figure 2: Transformation of the special vector x by the matrix A .

You can see in Figure 2 that the vector x has a special relationship with the matrix A : it is rescaled (with a negative value), but both the initial vector x and the transformed vector y are on the same line.

The vector x is an *eigenvector* of A . It is only scaled by a value, which is called an *eigenvalue* of the matrix A . An eigenvector of the matrix A is a vector that is contracted or elongated when transformed by the matrix. The eigenvalue is the scaling factor by which the vector is contracted or elongated.

Mathematically, the vector x is an eigenvector of A if:

[Get started](#)[Open in app](#)

Eigenvectors

Eigenvectors of a matrix are nonzero vectors that are only rescaled when the matrix is applied to them. If the scaling factor is positive, the directions of the initial and the transformed vectors are the same, if it is negative, their directions are reversed.

Number of eigenvectors

An n -by- n matrix has, at most, n linearly independent eigenvectors. However, each eigenvector multiplied by a nonzero scalar is also an eigenvector. If you have:

[REDACTED]

Then:

[REDACTED]

with c any nonzero value.

This excludes the zero vector as eigenvector, since you would have

[REDACTED]

In this case, every scalar would be an eigenvalue and thus would be undefined.

Hands-On Project: Principal Component Analysis

Principal Component Analysis, or PCA, is an algorithm that you can use to reduce the dimensionality of a dataset. It is useful, for instance, to reduce computation time, compress data, or avoid what is called *the curse of dimensionality*. It is also useful for visualization purposes: high dimensional data is hard to visualize and it can be useful to decrease the number of dimensions to plot your data.

In this hands-on project, you'll use various concepts that you can learn in the book [Essential Math for Data Science](#), as change of basis (Sections 7.5 and 9.2, some samples

[Get started](#)[Open in app](#)

In the first part, you'll learn about the relationship between projections, explained variance and error minimization, first with a bit of theory, and then by coding a PCA on the beer dataset (consumption of beer as a function of temperature). In the second part, you'll use Sklearn to use PCA on audio data to visualize audio samples according to their category, and then to compress these audio samples.

Under the Hood

Theoretical context

The goal of PCA is to project data onto a lower dimensional space while keeping as much of the information contained in the data as possible. The problem can be seen as a *perpendicular least squares* problem also called *orthogonal regression*.

You'll see here that the error of the orthogonal projections is minimized when the projection line corresponds to the direction where the variance of the data is maximal.

Variance and Projections

It is first important to understand that, when the features of your dataset are not completely uncorrelated, some directions are associated with a larger variance than others.

[Get started](#)[Open in app](#)

Figure 3: The variance of the data in the direction of the vector u (red) is larger than in the direction of the vector v (green). (Image by author)

Projecting data to a lower-dimensional space means that you might lose some information. In Figure 3, if you project two-dimensional data onto a line, the variance of the projected data tells you how much information you lose. For instance, if the variance of the projected data is near zero, it means that the data points will be projected to very close positions: you lose a lot of information.

For this reason, the goal of the PCA is to change the basis of the data matrix such that the direction with the maximum variance (u in Figure 3) becomes the first *principal component*. The second component is the direction with the maximum variance which is orthogonal to the first one, and so on.

When you have found the components of the PCA, you change the basis of your data such that the components are the new basis vectors. This transformed dataset has new features, which are the components and which are linear combinations of the initial features. Reducing the dimensionality is done by selecting some of the components only.

[Get started](#)[Open in app](#)

As an illustration, Figure 4 shows the data after a change of basis. The maximum variance is now associated with the x -axis. You can for instance keep only this first dimension.

In other words, expressing the PCA in terms of change of basis, its goal is to find a new basis (which is a linear combination of the initial basis) in which the variance of the data is maximized along the first dimensions.

Minimizing the Error

Finding the directions that maximize the variance is similar as minimizing the error between the data and its projection.



Figure 5: The direction that maximizes the variance is also the one associated with the smallest error (represented in gray). (Image by author)

You can see in Figure 5 that lower errors are shown in the left figure. Since projections are orthogonal, the variance associated with the direction of the line on which you project doesn't impact the error.

Finding the Best Directions

After changing the basis of the dataset, you should have a covariance between features close to zero (as for instance in Figure 4). In other terms, you want that the transformed dataset has a diagonal covariance matrix: the covariance between each pair of principal components is equal to zero.

[Get started](#)[Open in app](#)

calculate the eigenvectors of the covariance matrix of the dataset. They will give you the directions of the new basis in which the covariance matrix is diagonal.

To summarize, the principal components are calculated as the eigenvectors of the covariance matrix of the dataset. In addition, the eigenvalues give you the explained variance of the corresponding eigenvector. Thus, by sorting the eigenvectors in the decreasing order according to their eigenvalues, you can sort the principal components by importance order, and eventually remove the ones associated with a small variance.

Calculating the PCA

Dataset

Let's illustrate how PCA is working with the beer dataset showing the beer consumption and the temperature in São Paulo, Brazil for the year 2015.

Let's load the data and plot the consumption as a function of the temperature:

[Get started](#)[Open in app](#)

Figure 6: Consumption of beer as a function of temperature.

Now, let's create the data matrix X with the two variables: temperatures and consumption.

[Get started](#)[Open in app](#)

(365, 2)

The matrix X has 365 rows and two columns (the two variables).

Eigendecomposition of the Covariance Matrix

As you saw, the first step is to compute the covariance matrix of the dataset:

[Get started](#)[Open in app](#)

```
array([[18.63964745, 12.20609082], [12.20609082, 19.35245652]])
```

Remember that you can read it as follows: the diagonal values are respectively the variances of the first and the second variable. The covariance between the two variables is around 12.2.

Now, you will calculate the eigenvectors and eigenvalues of this covariance matrix:

[Get started](#)[Open in app](#)

```
(array([ 6.78475896, 31.20734501]), array([[ -0.71735154,  
-0.69671139], [ 0.69671139, -0.71735154]]))
```

You can store the eigenvectors as two vectors u and v .

[Get started](#)[Open in app](#)

Let's plot the eigenvectors with the data (note that you should use centered data because it is the data used to calculate the covariance matrix).

You can scale the eigenvectors by their corresponding eigenvalues, which is the explained variance. For visualization purpose, let's use a vector length of three standard deviations (equal to three times the square root of the explained variance):

[Get started](#)[Open in app](#)

Figure 7: The eigenvectors u (in gray) and v (in red) scaled according to the explained variance.

You can see in Figure 7 that the eigenvectors of the covariance matrix give you the important directions of the data. The vector v in red is associated with the largest eigenvalue and thus corresponds to the direction with the largest variance. The vector u in gray is orthogonal to v and is the second principal component.

Then, you just need to change the basis of the data using the eigenvectors as the new basis vectors. But first, you can sort the eigenvectors with respect to the eigenvalues in decreasing order:

[Get started](#)[Open in app](#)

```
array([[-0.69671139, -0.71735154], [-0.71735154, 0.69671139]])
```

Now that your eigenvectors are sorted, let's change the basis of the data:

[Get started](#)[Open in app](#)

You can plot the transformed data to check that the principal components are now uncorrelated:

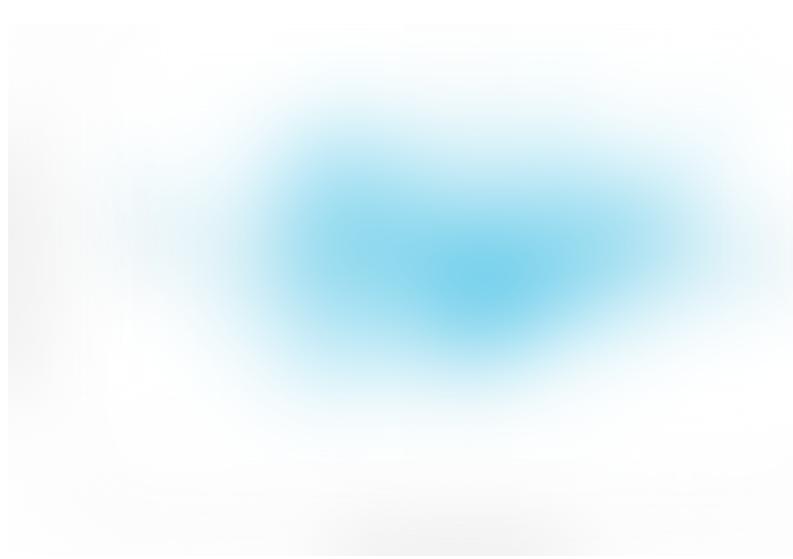
[Get started](#)[Open in app](#)

Figure 8: The dataset in the new basis.

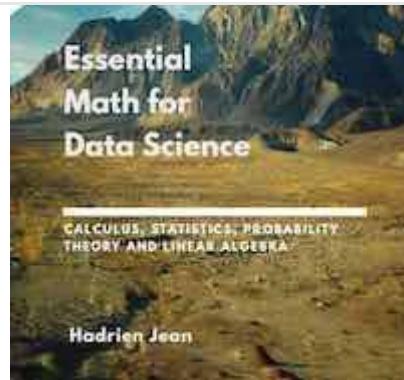
Figure 8 shows the data samples in the new basis. You can see that the first dimension (the x -axis) corresponds to the direction with the largest variance.

You can keep only the first component of the data in this new basis without losing too much information.

Covariance matrix or Singular Value Decomposition?

One caveat of using the covariance matrix to calculate the PCA is that it can be hard to compute when there are many features (as with audio data, like in the second part of this hands-on). For this reason, it is usually preferred to use the Singular Value Decomposition (SVD) to calculate the PCA.

This post is a sample of my book *Essential Math for Data Science*!

[Get started](#)[Open in app](#)

Get the book here: <https://bit.ly/2WVf4CR!>

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

[Get this newsletter](#)

[Essential Math](#) [Data Science](#) [Machine Learning](#) [Deep Learning](#) [Mathematics](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

