

The purpose of this paper is to provide a complete and simplified explanation of principal component analysis (PCA). I’ll cover how it works step by step, so everyone can understand it and make use of it, even those without a strong mathematical background.

PCA is a widely covered method on the web, and there are some great articles about it, but many spend too much time in the mathematics on the topic, when most of us just want to know how it works in a simplified way.

Principal component analysis can be broken down into five steps. I’ll go through each step, providing logical explanations of what PCA is doing and simplifying mathematical concepts such as standardization, covariance, eigenvectors and eigenvalues without focusing on how to compute them.

**HOW DO YOU DO A PRINCIPAL COMPONENT ANALYSIS?**

1. Standardize the range of continuous (numeric and non-discrete) initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast the data along the principal components axes

First, some basic (and brief) background is necessary for context.

**What Is Principal Component Analysis?**

Principal component analysis, or PCA, is a [dimensionality-reduction](https://builtin.com/data-science/dimensionality-reduction-python) method that is often used to reduce the dimensionality of large [data sets](https://builtin.com/data-science), by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a dataset naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for [machine learning algorithms](https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies) without extraneous variables to process.

So, to sum up, the idea of PCA is simple — reduce the number of variables of a dataset, while preserving as much information as possible.

**Step-by-Step Explanation of PCA**

**STEP 1: STANDARDIZATION**

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (for example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

or

**STEP 2: COVARIANCE MATRIX COMPUTATION**

The aim of this step is to understand how the variables of the input dataset are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × p symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional dataset with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from:

Dataset Covariance matrix

|  |  |  |
| --- | --- | --- |
| *x* | *y* | *z* |
|  |  |  |
|  |  |  |
|  |  |  |
| .  .  . |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Remember, if you are preparing your dataset for machine learning then, you should ignore your target attribute!

Since the covariance of a variable with itself is its variance (Cov(a,a)=Var(a)), in the matrix main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative (*Cov(a,b)=Cov(b*,a)), the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.

What do the covariances that we have as entries of the matrix tell us about the correlations between the variables? It says how the variance of one *correlated* to the second one.

It’s actually the sign of the covariance that matters:

1. *If positive then: the two variables increase or decrease together (correlated)*
2. *If negative then: one increases when the other decreases (Inversely correlated)*

Now that we know that the covariance matrix is not more than a table that summarizes the correlations between all the possible pairs of variables, let’s move to the next step.

**STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS**

*Eigenvectors* and *eigenvalues* are the linear algebra concepts that we need to compute from the covariance matrix to determine the principal components of the data. Before getting to the explanation of these concepts, let’s first understand what we mean by principal components.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the plot below.

Chart, bar chart, histogram

Description automatically generated

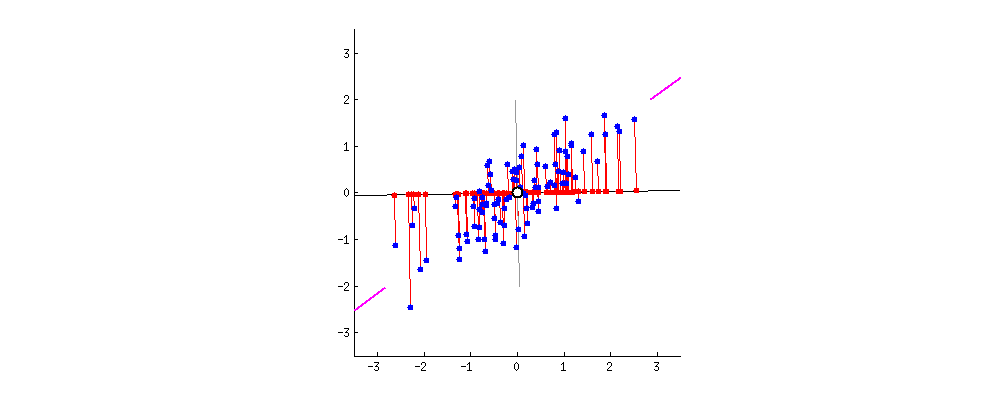
Organizing information in principal components this way, will allow us to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

An important thing to realize here is that the principal components are less interpretable and don’t have any real meaning since they are constructed as linear combinations of the initial variables.

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance the lines that capture most information of the data. The relationship between variance and information here, is that the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

***How PCA Constructs the Principal Components***

As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the dataset. For example, let’s assume that the scatter plot of our dataset is as shown below, can we guess the first principal component? Yes, it’s approximately the line that matches the purple marks because it goes through the origin and it’s the line in which the projection of the points (red dots) is the most spread out. Or mathematically speaking, it’s the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).



The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

This continues until a total of p principal components have been calculated, equal to the original number of variables.

Now that we understand what we mean by principal components, let’s go back to eigenvectors and eigenvalues. What you first need to know about them is that they always come in pairs, so that every eigenvector has an eigenvalue. And their number is equal to the number of dimensions of the data. For example, for a 3-dimensional dataset, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

Without further argument, it is eigenvectors and eigenvalues who are behind all the magic explained above, because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance(most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

### **How to calculate Eigenvalues and Eigenvectors**

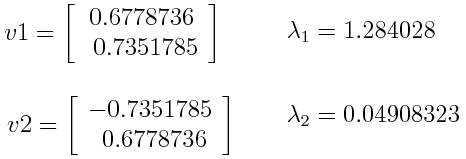
Given covariance matrix

1. Multiply the 2x2 identity matrix by an scalar
2. Subtract the identity matrix I multiple from the *Cov* matrix
3. Find the determinant of the matrix (from step 2) and the difference
4. Solve for value of that satisfy the equation
5. Solve the corresponding vector to each
7. =
8. det =

1. Simplify to This equation give us two answers for these are our eigenvalues
2. To get the eigenvectors we replace with their from 4 and solve for where is the eigenvector

Solving this give us our Eigenvector related to . We do the same think for to get the eigenvector for

***Example:***

Let’s suppose that our dataset is 2-dimensional with 2 variables x, y and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

|  |  |
| --- | --- |
| *x* | *y* |
|  |  |
|  |  |
|  |  |
| .  .  . |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

If we rank the eigenvalues in descending order, we get λ1>λ2, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) isv2.

After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

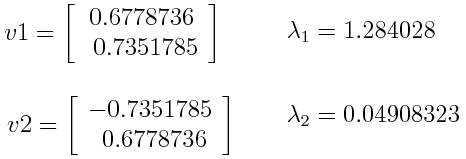
**STEP 4: FEATURE VECTOR**

As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature vector.

So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

***Example:***

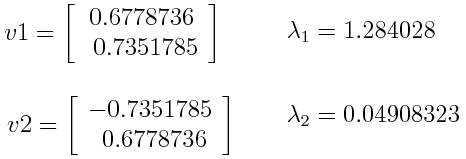
Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors v1 and v2:

Principal Component Analysis eigen vectors

Which gives us

|  |  |  |
| --- | --- | --- |
| v1 or PC1 = | 1.284028 | 0.96318144 or 96% |
| v2 or PC2 = | 0.04908323 | 0.03681856 or 36% |
| PC1+PC2 = | 1.33311123 |  |

Or discard the eigenvector v2, which is the one of lesser significance, and form a feature vector with v1 only:

Principal Component Analysis eigen vectors 2

Which is 96%

Discarding the eigenvector *v2*will reduce dimensionality by 1, and will consequently cause a loss of information in the final dataset. But given that *v*2 was carrying only 4% of the information, the loss will be therefore not important and we will still have 96% of the information that is carried by *v*1.

So, as we saw in the example, it’s up to you to choose whether to keep all the components or discard the ones of lesser significance, depending on what you are looking for. Because if you just want to describe your data in terms of new variables (principal components) that are uncorrelated without seeking to reduce dimensionality, leaving out lesser significant components is not needed.

**LAST STEP: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES**

In the previous steps, apart from standardization, you do not make any changes on the data, you just select the principal components and form the feature vector, but the input dataset remains always in terms of the original columns (i.e, in terms of the initial variables).

In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original columns to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original dataset by the transpose of the feature vector.

Additional Note:

**The transpose of a matrix**

The transpose of a matrix is simply a flipped version of the original matrix. We can transpose a matrix by switching its rows with its columns. We denote the transpose of matrix M by MT. For example, if

Then the transpose of M is