Principal Component Analysis - Part I

Biswajit Sahoo

2019-02-03

In this post, we will discuss about Principal Component Analysis (PCA), one of the most popular dimensionality reduction techniques used in machine learning. Applications of PCA and its variants are ubiquitous. Thus, a through understanding of PCA is considered essential to start one's journey into machine learning. In this and subsequent posts, we will first briefly discuss relevant theory of PCA. Then we will implement PCA from scratch without using any built-in function. This will give us an idea as to what happens under the hood when a built-in function is called in any software environment. Simultaneously, we will also show how to use built-in commands to obtain results. Finally, we will reproduce the results of a popular paper on PCA. Including all this in a single post will make it very very long. Therefore, the post has been divided into three parts. Readers totally familiar with PCA should read none and leave this page immediately to save their precious time. Other readers, who have a passing knowledge of PCA and want to see different implementations, should pick and choose material from different parts as per their need. Absolute beginners should start with Part-I and work their way through gradually. Beginners are also encouraged to explore the references at the end of this post for further information. Here is the outline of different parts:

- Part-I: Basic Theory of PCA
- Part-II: PCA Implementation with and without using built-in functions
- Part-III: Reproducing results of a published paper on PCA

For Part-II, Python, R, and MATLAB code are available to reproduce all the results. Part-III contains both R and Python code to reproduce results of the paper. In this post, we will discuss the theory behind PCA in brief.

Principal Component Analysis

Theory:

Given a data matrix, we apply PCA to transform it in a way such that the transformed data reveals maximum information. So we have to first get the data on which we want to perform PCA. The usual convention in storing data is to place variables as columns and different observations as rows (Data frames in R follow this convention by default). For example, let's suppose we are collecting data about daily weather for a year. Our variables of interest may include maximum temperature in a day, minimum temperature, humidity, max. wind speed, etc. Everyday we collect observations for each of these variables. In vector form, our data point for one day will contain number of observations equal to the number of variables under study and this becomes one row of our data matrix. Assuming that we are observing 10 variables everyday, our data matrix for one year (assuming it's not a leap year) will contain 365 rows and 10 columns. Once data matrix is obtained, further analysis is done on this data matrix to obtain important hidden information regarding the data. We will use notations from matrix theory to simplify our analysis.

Let $\mathbf{X}$ be the data matrix of size $n \times p$, where n is the number of data points and p is the number of variables. We can assume without any loss of generality that data is centered, meaning its column means are zero.

This only shifts the data towards the origin without changing their relative orientation. So if originally not centered, it is first centered before doing PCA. From now onward we will assume that data matrix is always centered.

Variance of a variable (a column) in $\mathbf{X}$ is equal to sum of squares of entries (because the column is centered) of that column divided by (n - 1) (to make it unbiased). So sum of variance of all variables is $\frac{1}{n-1}$ times sum of squares of all elements of the matrix . Readers who are familiar with matrix norms would instantly recognize that total variance is $\frac{1}{n-1}$ times the square of **Frobenius norm** of $\mathbf{X}$. Frobenius norm is nothing but square root of sum of squares of all elements of a matrix.

$$\|\mathbf{X}\|_F = (\sum_{i,j} x_{ij}^2)^{\frac{1}{2}} = \sqrt{trace(\mathbf{X}^T\mathbf{X})} = \sqrt{trace(\mathbf{X}\mathbf{X}^T)}$$

Using this definition, total variance before transformation =

$$\frac{1}{n-1}\sum_{i,j} x_{ij}^2 = trace(\frac{1}{n-1}\mathbf{X}^T\mathbf{X}) = \frac{1}{n-1}\|\mathbf{X}\|_F^2$$

Where, trace of a matrix is the sum of its diagonal entries and $\|\mathbf{X}\|_F^2$ is the square of **Frobenius norm**.

The aim of PCA is to transform the data in such a way that along first principal direction, variance of transformed data is maximum. It subsequently finds second principal direction orthogonal to the first one in such a way that it explains maximum of the remaining variance among all possible direction in the orthogonal subspace.

In matrix form the transformation can be written as

$$\mathbf{Y}_{n\times p} = \mathbf{X}_{n\times p}\mathbf{P}_{p\times p}$$

Where $\mathbf{Y}$ is the transformed data matrix. The columns of $\mathbf{Y}$ are called principal components and $\mathbf{P}$ is usually called loading matrix. Our aim is to find matrix $\mathbf{P}$. Once we find $\mathbf{P}$ we can then find $\mathbf{Y}$ just by a matrix multiplication. We will show in the next section that matrix $\mathbf{P}$ is the eigenvector matrix of the covariance matrix. Before that, let's first define the covariance matrix.

Given a data matrix $\mathbf{X}$(centered), its covariance matrix ($\mathbf{S}$) is defined as

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

Now we will show how to compute the loading vectors (columns of $\mathbf{P}$) and consequently the principal components (columns of $\mathbf{Y}$) from the given centered data matrix $\mathbf{X}$.

Sketch of the Proof

We call it a sketch because we will not be giving the full proof. Rather, we will give the proof only for the first principal component and then give a commentary as to how it can be extended for other principal components.

The first principal component is the result obtained by transforming original data matrix $\mathbf{X}_{n\times p}$ in such a way that variance of data along first principal component is the highest. The transformation is a linear transformation that is obtained by taking linear combination of the columns of $\mathbf{X}_{n\times p}$. The coefficients of the linear combination are called loading scores corresponding to original variables of $\mathbf{X}_{n\times p}$.

Assuming $\boldsymbol{\alpha}_{p\times 1} = [\alpha_1, \alpha_2, ..., \alpha_p]^T$, where α_1, α_2, ... , α_p are scalars, to be the loading vector (we don't know, as of now, from where to get $\alpha_{p\times 1}$. We will find that out shortly.), first principal component is obtained by the the product $\mathbf{X}_{n\times p}\boldsymbol{\alpha}_{p\times 1}$. This product can be written as

$$\mathbf{X}_{n\times p}\boldsymbol{\alpha}_{p\times 1} = \alpha_1\mathbf{X}_{[:,1]} + \alpha_2\mathbf{X}_{[:,2]} + ... + \alpha_p\mathbf{X}_{[:,p]}$$

Where, $\mathbf{X}_{[:,1]}$ is the first column of $\mathbf{X}_{n\times p}$. Similarly for other columns. The above equation makes it clear as to why first principal component is a linear combination of variables of original data matrix. In the original data matrix, each column corresponds to a variable.

Variance of first principal component is given by $\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}$ (As the columns are already centered. We have also ignored the factor $(\frac{1}{n-1})$ as it is just a scaling factor.). Now our goal is to find an $\boldsymbol{\alpha}_{p\times 1}$ that maximizes $\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}$. As $\boldsymbol{\alpha}_{p\times 1}$ is arbitrary, we can choose its entries in such a way that variance increases as much as we please. So to get any meaningful solution, we have to apply some constraints on $\boldsymbol{\alpha}_{p\times 1}$. The conventional condition is $\|\boldsymbol{\alpha}_{p\times 1}\|^2 = 1$. The optimization problem becomes

$$maximize \quad \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}$$

$$s.t. \|\boldsymbol{\alpha}\|^2 = 1$$

Using Lagrange multipliers, this problem can be written as

$$maximize \quad \mathcal{L}(\boldsymbol{\alpha}, \lambda) = \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} + \lambda(1 - \boldsymbol{\alpha}^T\boldsymbol{\alpha})$$

Taking gradient of $\mathcal{L}(\boldsymbol{\alpha}, \lambda)$ with respect to $\boldsymbol{\alpha}$ we get, $\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$. So $\boldsymbol{\alpha}$ is the eigenvector of $\mathbf{X}^T\mathbf{X}$. It turns out that for first principal component, $\boldsymbol{\alpha}$ is the eigenvector corresponding to the largest eigenvalue.

Loading vector for second principal component is computed with the added condition that second loading vector is orthogonal to the first one. With little bit of more work it can be shown that loading vectors for successive principal components are obtained from eigenvectors corresponding to eigenvalues in decreasing order. More details can be found in reference [1].

Now, it is straightforward to first form the covariance matrix and by placing its eigenvectors as columns, we can find matrix $\mathbf{P}$ and consequently the principal components. The eigenvectors are arranged in such a way that first column is the eigenvector corresponding to largest eigenvector, second column (second eigenvector) corresponds to second largest eigenvalue and so on. Here we have assumed that we will always be able to find all the p orthogonal eigenvectors. In fact, we will always be able to find p orthogonal eigenvectors as the matrix is symmetric. It can also be shown that the transformed matrix $\mathbf{Y}$ is centered and more remarkably, total variance of columns of $\mathbf{Y}$ is same as total variance of columns of $\mathbf{X}$. We will prove these two propositions as the proofs are short.

Properties of PCA Transformation

1. Principal components are centered.

   Proof: Let $\mathbf{1}$ be a column vector of all ones of size $(n \times 1)$. To prove that columns of $\mathbf{Y}$ are centered, just premultiply it by $\mathbf{1}^T$ (this finds column sum for each column). So

   $$\mathbf{1}^T\mathbf{Y} = \mathbf{1}^T\mathbf{X}\mathbf{P}$$

   But columns of $\mathbf{X}$ are already centered, so $\mathbf{1}^T\mathbf{X} = \mathbf{0}$. Thus $\mathbf{1}^T\mathbf{Y} = \mathbf{0}$. Hence columns of $\mathbf{Y}$ are centered.

2. Sum of variance of principal components is equal to sum of variance of variables before transformation.

   Proof: To prove that total variance of $\mathbf{Y}$ also remains same, observe that

   total covariance of $\mathbf{Y}$ =

   $$trace(\frac{1}{n-1}\mathbf{Y}^T\mathbf{Y}) = \frac{1}{n-1}trace((\mathbf{P}^T\mathbf{X}^T\mathbf{X})\mathbf{P}) = \frac{1}{n-1}trace((\mathbf{P}\mathbf{P}^T)\mathbf{X}^T\mathbf{X}) = trace(\frac{1}{n-1}\mathbf{X}^T\mathbf{X})$$

   The previous equation uses the fact that trace is commutative(i.e.$trace(\mathbf{AB}) = trace(\mathbf{BA})$) and $\mathbf{P}$ is orthogonal (i.e. $\mathbf{P}\mathbf{P}^T = \mathbf{I}$).

3. Principal components are orthogonal.

   Proof: To prove the above claim, it is sufficient to show that the matrix $\mathbf{Y}^T\mathbf{Y}$ is diagonal. Remember that columns of $\mathbf{Y}$ are principal components. So if we can somehow show $\mathbf{Y}^T\mathbf{Y}$ to be diagonal, it would automatically mean that principal components are orthogonal. We know, $\mathbf{Y} = \mathbf{XP}$. So $\mathbf{Y}^T\mathbf{Y} = \mathbf{P}^T\mathbf{X}^T\mathbf{XP}$. From sketch of the proof, we know that $\mathbf{P}$ is orthogonal as we have required successive loading vectors to be orthogonal to previous ones. We also know that $\mathbf{P}$ is the eigenvector matrix of $\mathbf{X}^T\mathbf{X}$. So from Eigen Decomposition Theorem, it follows that $\mathbf{P}^T(\mathbf{X}^T\mathbf{X})\mathbf{P}$ is diagonal as $\mathbf{P}$ is the eigenvector matrix of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{P}$ is orthogonal (so $\mathbf{P}^{-1} = \mathbf{P}^T$).

Link between total variance and eigenvalues

Total variance is sum of eigenvalues of covariance matrix ($\mathbf{S}$). This follows from the fact that *trace of a matrix is sum of its eigenvalues*. Total variance of original data matrix is $\frac{1}{n-1}trace(\mathbf{X}^T\mathbf{X}) = trace(\frac{1}{n-1}\mathbf{X}^T\mathbf{X}) = trace(\mathbf{S})$. We will show these calculations using a publicly available dataset in Part-II.

Variations of PCA

Sometimes our data matrix contains variables that are measured in different units. So we might have to scale the centered matrix to reduce the effect of variables with large variation. So depending on the matrix on which PCA is performed, it is divided into two types.

- Covariance PCA (Data matrix is centered but **not** scaled)
- Correlation PCA (Data matrix is centered and scaled)

Examples of these two types can be found in Part-II. Please note that the above two variations are just two among many variations. There are **Sparse PCA**, **Kernel PCA**, **Robust PCA**, **Non-negative PCA** and many others. We have mentioned the two that are most widely used.

Some common terminologies associated with PCA

In literature, there is no standard terminology for different terms in PCA. Different people use different (often contradictory) terminology thus confusing newcomers. Therefore, it is better to stick to one set of terminologies and notations and use those consistently. We will stick to the terminology used in reference [2].

- **Factor scores** corresponding to a principal component: Values of that column of $\mathbf{Y}$ that corresponds to the desired principal component.
- **Loading score**: Values corresponding to a column of $\mathbf{P}$. For example,loading scores of variables corresponding to first principal component are the values of the first column of $\mathbf{P}$.
- **Inertia**: Square of Frobenius norm of the matrix.

How actually are principal components computed?

The previously stated method of finding eigenvectors of covariance matrix is not computationally efficient. In practice, singular value decomposition (SVD) is used to compute the matrix $\mathbf{P}$. SVD theorem tells that any real matrix $\mathbf{X}$ can be decomposed into three matrices such that

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$$

Where, $\mathbf{X}$ is of size $n \times p$. $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of size $n \times n$ and $p \times p$ respectively. Σ is a diagonal matrix of size $n \times p$.

Given the SVD decomposition of a matrix $\mathbf{X}$,

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\Sigma^2\mathbf{V}^T$$

This is the eigen-decomposition of $\mathbf{X}^T\mathbf{X}$. So $\mathbf{V}$ is the eigenvector matrix of $\mathbf{X}^T\mathbf{X}$. For PCA we need eigenvector matrix of covariance matrix. So converting the equation into convenient form, we get

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \mathbf{V}(\frac{1}{n-1}\Sigma^2)\mathbf{V}^T$$

Thus eigenvalues of S are diagonal entries of $(\frac{1}{n-1}\Sigma^2)$. As SVD is computationally efficient, all built-in functions use SVD to compute the loading matrix and then use the loading matrix to find principal components.

In the interest of keeping the post at a reasonable length, we will stop our exposition of theory here. Whatever we have discussed is only a fraction of everything. Entire books have been written on PCA. Interested readers who want to pursue this further can refer the references of this post as a starting point. Readers are encouraged to bring any errors or omissions to my notice.

References

1. I.T. Jolliffe, Principal component analysis, 2nd ed, Springer, New York,2002.
2. Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), 433-459.

Last modified: May 5, 2021