# Principal Component Analysis

Helen Cho

May 2020

## Introduction

Principal component analysis (PCA) is an extremely useful multivariate statistical technique that identifies a new set of uncorrelated variables through orthogonal diagonalization. This new set of variables can represent the data set with fewer dimensions while preserving the most important information and highlights the variables that "explain" most of the variation in the data [1]. PCA is particularly prevalent in machine learning, chemistry, and other scientific disciplines that work with large, multivariate data sets. In practice, the new set of variables is found through the singular value decomposition. This project will explore the linear algebra behind PCA with an example and briefly discuss applications of PCA in machine learning.

## Some Statistical Background and Notation

Because PCA is a statistical technique, this section will define some important statistical terms in terms of linear algebra, which come from Lay's linear algebra textbook [2] unless noted otherwise.

Collected data can be represented in a $p \times N$ matrix $A$, known as a **matrix of observations**, where $p$ is the number of variables and $N$ is the number of observations. For example, suppose five (hypothetical) Harvard undergraduates were randomly selected, and their GPAs and the average number of hours they sleep per night were recorded in the following table

| **Student** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| GPA | 3.0 | 3.5 | 3.3 | 3.8 | 3.6 |
| Average Hours of Sleep per Night | 6 | 7 | 9 | 8 | 8 |

Then, the matrix of observations $A$ is a $2 \times 5$ matrix, where $p = 2$ and $N = 5$. Each observation can be represented as an **observation vector**, $\mathbf{X}_j$, for $j = 1, \dots, N$. Then, the matrix of observations $A = \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix}$.

Before performing PCA on a data set, the data set must be "preprocessed." Depending on the type of PCA, the data will be preprocessed differently. Here, data from the matrix of observations will be used to construct a covariance matrix for covariance PCA.

First, we calculate the **sample mean**, $\mathbf{M}$, of the data set for observation vectors $\mathbf{X}_1, \cdots, \mathbf{X}_N$, which is given by

$$\mathbf{M} = \frac{1}{N}(\mathbf{X}_1 + \cdots + \mathbf{X}_N)$$

Then, for $k = 1, \ldots, N$, we define

$$\widehat{\mathbf{X}}_k = \mathbf{X}_k - \mathbf{M}$$

We can now define a matrix in **mean-deviation form**, given by

$$B = \begin{bmatrix} \widehat{\mathbf{X}}_1 & \cdots & \widehat{\mathbf{X}}_N \end{bmatrix}$$

This process effectively "centers" the data, that is, the sample mean of $B$ is 0.

Finally, we define the **(sample) covariance matrix**, $S$, which is a positive semidefinite $p \times p$ matrix given by

$$S = \frac{1}{N-1} BB^T$$

Let $\mathbf{X}$ be a vector in $\mathbb{R}^p$ with components $x_1, \ldots, x_p$, where each component represents the corresponding value in any given observation $\mathbf{X}_j$. Then, each entry $s_{ij}$ in $S$ with $i \neq j$, represents the covariance between $x_i$ and $x_j$. The **covariance** between two different variables describes the extent to which the deviation of one variable from its mean matches that of another [3]. If $s_{ij} > 0$, then $w_i$ and $w_j$ are either both above or below their respective means. If $s_{ij} < 0$, then $w_i$ and $w_j$ are inversely related. If $s_{ij} = 0$, then $w_i$ and $w_j$ are said to be **uncorrelated**. (If this sounds similar to correlation, this intuition is accurate because correlation follows directly from covariance.) Furthermore, if the magnitude of the covariance is large, then we can predict the value of one variable knowing the value of the other variable fairly accurately. Then, the information encoded by both variables is redundant, and one variable could be removed without losing too much information [5].

The covariance between a variable and itself is known as **variance**, which measures the spread of the values of a variable. In $S$, the variance of each variable falls along the diagonal of $S$. The **total variance** is the sum of the variances for each variable, or the **trace** of $S$ ($\operatorname{tr}(S)$), the sum of the entries along the diagonal of a square matrix.

In our example, [1]

$$\mathbf{M} = \begin{bmatrix} 3.44 \\ 7.6 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.44 & 0.06 & -0.14 & 0.36 & 0.16 \\ -1.6 & -0.6 & 1.4 & 0.4 & 0.4 \end{bmatrix}, \text{ and}$$

$$S = \begin{bmatrix} 0.093 & 0.17 \\ 0.17 & 1.3 \end{bmatrix}$$

## Principal Component Analysis

Note that we will be following much of the process set out in Lay Section 7.5 [2].

Recall that the goals of PCA are to reduce the dimensionality (or redundancy) of the data and to find a set of uncorrelated variables that account for the most variance in the data. To accomplish

[1] To see the source code for these calculations in python, visit https://github.com/helencho19/Math22bFinalProject

this, we want to find an orthogonal $p \times p$ matrix $P = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_p \end{bmatrix}$ such that $P$ acts as a change of variables matrix (a similar concept to the change of coordinates matrix from Ch 4 in Lay) from the original set of variables $\mathbf{X}$ to the new set of variables $\mathbf{Y}$ where $\mathbf{X} = P\mathbf{Y}$. Note that $\mathbf{Y} \in \mathbb{R}^p$, where $\mathbf{Y}$ represents uncorrelated variables $y_1, \dots, y_p$ arranged in order of decreasing variance.

With this change of variables, for each observation $\mathbf{X}_k$ where $k = 1, \cdots, N$, $\mathbf{X}_k = P\mathbf{Y}_k$, where $\mathbf{Y}_k$ is the coordinate vector of $\mathbf{X}_k$ with respect to the columns of $P$. Put another way, $\mathbf{Y}_k$ is a representation of $\mathbf{X}_k$ in terms of a new basis defined by the columns of $P$. Since $P$ is an orthogonal matrix, $P^{-1} = P^T$. Then, $\mathbf{Y}_k = P^T\mathbf{X}_k$. As we will see later, the basis described by the columns of $P$ determine the new set of variables $\mathbf{Y}$ we are interested in finding.

That being said, important questions remain: what exactly is $P$? How do we find it? To answer these questions, we first must zoom out and consider what sorts of properties we would like the set of variables $\mathbf{Y}$ to have.

In considering the properties of the new set of variables $\mathbf{Y}$, recall that we would like $y_1, \dots, y_p$ to be uncorrelated. This reduces the redundancy of the data because the covariance between $y_i$ and $y_j$ for $i \neq j$ must be 0 in order for $y_i$ and $y_j$ to be uncorrelated. We would also like the variables to account for the maximum variance of the data, arranged in decreasing order, thus concerning us with the variance of each variable. By incorporating these two properties, we accomplish the goals of PCA.

Recall that in the section on statistical background and notation, we introduced a data structure that pertains to both the covariance and variance, namely, the sample covariance matrix $S$. Thus, we want to "design" a covariance matrix $S_Y$ with the properties aforementioned [5]. (For clarity's sake, we rename the sample covariance matrix $S$ of the original set of variables as $S_X$.) Since we want the covariances between all of the new variables to be 0 and the variances to be some value, it seems reasonable that $S_Y$ should be a diagonal matrix. We also know that $S_Y = \frac{1}{N-1}YY^T$, where $Y = \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_N \end{bmatrix}$ and $\mathbf{Y}_k$ for $k = 1, \dots, N$ is the original observation $\mathbf{X}_k$ expressed in terms of the new basis. Given this information, we can show that $S_Y = P^T S_X P$:

$$\begin{aligned}
S_Y &= \frac{1}{N-1}YY^T \\
&= \frac{1}{N-1}\begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_N \end{bmatrix}\begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_N \end{bmatrix}^T \\
&= \frac{1}{N-1}\begin{bmatrix} P^T\mathbf{X}_1 & \cdots & P^T\mathbf{X}_N \end{bmatrix}\begin{bmatrix} P^T\mathbf{X}_1 & \cdots & P^T\mathbf{X}_N \end{bmatrix}^T \text{ since } \mathbf{Y}_k = P^T\mathbf{X}_k \\
&= \frac{1}{N-1}P^T\begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix}\left(P^T\begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix}\right)^T \\
&= \frac{1}{N-1}P^T\begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix}\begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_N \end{bmatrix}P \\
&= P^T\left(\frac{1}{N-1}\begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix}\begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_N \end{bmatrix}\right)P \\
&= P^T S_X P
\end{aligned}$$

Now we have a better answer for what $P$ is: we want a $P$ such that $P^T S_X P$ is diagonal. Note that $S_X = P S_Y P^T$. This decomposition should be very familiar. Recall that the Diagonalization Theorem tells us that if an $n \times n$ matrix $A$ is diagonalizable, then $A = PDP^{-1}$ with $D$ diagonal if and only if the columns of $P$ are the $n$ eigenvectors of $A$. In this case, the diagonal entries of $D$ are the corresponding eigenvalues of $A$. Using this theorem as inspiration, we can pick the $p$ normalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_p$ of $S_X$ to form the columns of $P$, and let the diagonal entries of $S_Y$ be the corresponding eigenvalues arranged in decreasing order. (Note: for more on why $S_X$ can be diagonalized, see Section 7.2 of Lay.) We have finally found $P$.

The unit eigenvectors of $S_X$ are called **principal components**. The eigenvector corresponding to the largest eigenvalue is known as the **first principal component**, the eigenvector corresponding to the second largest eigenvalue is the **second principal component**, and so on. The first principal component "explains" the most variation in the data since its variance (the corresponding eigenvalue) is the largest [2].

We use $\mathbf{Y} = P^T\mathbf{X}$ to determine the new set of variables $y_1, \ldots, y_p$. Since each row of $P^T$ corresponds to a principal component, each variable $y_k$ is a linear combination of the original set of variables $x_1, \ldots, x_p$ with the corresponding eigenvector $\mathbf{u}_k$ as weights.

To finish up our example, we diagonalize $S_X$ to get

$$S_X = \begin{bmatrix} 0.137 & -0.99 \\ 0.99 & 0.137 \end{bmatrix} \begin{bmatrix} 1.323 & 0 \\ 0 & 0.069 \end{bmatrix} \begin{bmatrix} 0.137 & 0.99 \\ -0.99 & 0.137 \end{bmatrix}$$

where the first principal component is $\begin{bmatrix} 0.137 \\ 0.99 \end{bmatrix}$ and the second principal component is $\begin{bmatrix} -0.99 \\ 0.137 \end{bmatrix}$. Visually, the two principal components form new axes for the centered data:

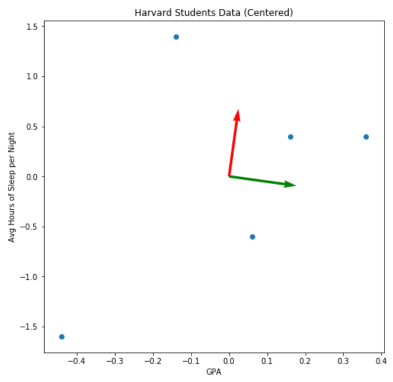

Figure 1: Principal Components for Centered Example Data

The red vector represents the first principal component, and the green vector represents the second principal component.

We can calculate how much of the variance each principal component accounts for by dividing the variance of the variable (the eigenvalue) by the total variance (the trace of $S_Y$):

$$\begin{aligned} \mathrm{tr}(S_Y) &= 1.323 + 0.069 = 1.392 \\ \text{first component: } \frac{1.323}{1.392} &= 0.95 \\ \text{second component: } \frac{0.069}{1.392} &= 0.05 \end{aligned}$$

Then, the new variable formed by the first principal component "explains" 95% of the variance in the data, and the variable formed by the second principal component "explains" 5% of the variance. Because the first principal component accounts for that much variance, we can project the observations onto the first principal component while retaining most of the information. This is how PCA can reduce the dimensionality of the data.

## Applications of PCA in Machine Learning

Now that we have a better understanding of how PCA works mathematically, we turn to some applications of PCA in machine learning (as explained by Andrew Ng in Machine Learning on Coursera [4]) to conclude the project.

We have seen how PCA could potentially be used for dimensionality reduction, and it is this aspect of PCA that machine learning capitalizes on. Oftentimes, we use machine learning techniques to better understand complex data sets that could have hundreds or thousands of dimensions. One important way we understand data is by visualizing it. However, it's a bit difficult to visualize anything in more than 3 dimensions. Thus, we can use PCA to find the first few principal components, project the data onto those components, and represent the data with the principal components as the new axes. PCA is also helpful in data compression. It's quite likely that some of the dimensions are highly correlated with each other and are therefore redundant. As we explored earlier, PCA removes this redundant information, and the resulting data is smaller and takes up less memory. Finally, PCA can speed up supervised learning algorithms by reducing the dimensions of the training set.

## Conclusion

This project aimed to give a brief introduction to PCA, but has barely scratched the surface of the subject or its applications. For those interested in the linear algebra behind PCA, we recommend looking into the singular value decomposition and how PCA can be viewed as a constrained optimization problem. For those interested in exploring more applications of PCA, fellow Math 22 classmate Emma Freeman has written an interesting paper on facial recognition software. Other notable topics (of many!) include image processing, chemometrics, and quantitative finance.

## References

[1] Hervé Abdi and Lynne J. Williams. “Principal component analysis”. In: *WIREs Computational Statistics* 2 (4 2010), pp. 433–459. DOI: 10.1002/wics.101.

[2] David C. Lay. *Linear Algebra and its Applications.* Addison-Wesley, 2011.

[3] Will Monroe. *Lecture Notes #15: Covariance and Correlation.* URL: https://web.stanford.edu/class/archive/cs/cs109/cs109.1178/lectureHandouts/150-covariance.pdf. (accessed: 05.12.2020).

[4] Andrew Ng. *Week 8: Dimensionality Reduction.* URL: https://www.coursera.org/learn/machine-learning/home/week/8. (accessed: 05.12.20).

[5] Jonathan Schlens. *A Tutorial on Principal Component Analysis.* URL: https://arxiv.org/pdf/1404.1100.pdf. (accessed: 05.12.2020).