Principal Component Analysis

ZTS
plote5024@gmail.com

Abstract

Principal component analysis (PCA) is a mainstay of modern data analysis.but (sometimes) poorly understood. This manuscript crystallizes this knowledge by deriving from simple intuitions, the mathematics behind PCA. readers of all levels will be able to gain a better understanding of PCA as well as the when, the how and the why of applying this technique.

I. INTRODUCTION

Principal component analysis (PCA) is a standard tool in modern data analysis - in fields as diverse as neuroscience to computer graphics - because it is a simple, nonparametric method for extracting relevant information from messy data sets. I record my intuition for learning PCA. I will refer to the author's article (link is in the appendix) to start with a simple example and provide an intuitive explanation of the PCA goal. Combined with my own knowledge, using mathematical rigor, the framework of linear algebra, I gradually understand the clear solution. At the same time, I clearly understand why and how PCA is closely related to the mathematical technique of singular value decomposition (SVD). The formulas in the notes combine my own derivation in the learning process with the understanding of the author's article and notes, and record what I think is a more rigorous and correct formula. Finally, I introduce a sentence from the author: Although the proofs are not so important for this tutorial, they are provided for adventurous readers who want a more comprehensive understanding of the mathematics.

II. Basic idea

The goal of principal component analysis is to identify the most meaningful basis to re-express a data set. The hope is that this new basis will filter out the noise and reveal hidden structure.

now state more precisely what PCA asks: **Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?**

A close reader might have noticed the conspicuous addition of the word linear. Indeed, PCA makes one stringent but powerful assumption: linearity. Linearity vastly simplifies the problem by restricting the set of potential bases. With this assumption PCA is now limited to re-expressing the data as a linear combination of its basis vectors. Let X be the original data set, where each column is a single sample (or moment in time) of our data set (i.e. X). Let Y be another m × n matrix related by a linear transformation P. X is the original recorded data set and Y is a new representation of that data set.

$$\boldsymbol{PX} = \boldsymbol{Y}$$

Also let us define the following quantities.

- $\boldsymbol{p_i}$ are the rows of $\boldsymbol{P}$
- $\boldsymbol{x_i}$ are the columns of $\boldsymbol{X}$ (or individual $\vec{x}$).
- $\boldsymbol{y_i}$ are the columns of $\boldsymbol{Y}$.

I think the basic understanding is also here.

→ **$\boldsymbol{P}$ is a matrix that transforms $\boldsymbol{X}$ into $\boldsymbol{Y}$.**

→ **Geometrically, $\boldsymbol{P}$ is a rotation and a stretch which again transforms $\boldsymbol{X}$ into $\boldsymbol{Y}$**

I assume a transformation matrix S here. You can see that its function is to lengthen the x-axis of the original coordinate system by a times and the y-axis by b times.Here S plays the role of R.

$$S = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

Similarly, I now want to perform a rotation transformation,The multiplication of the r matrix means that the original basis vector i changes from $\binom{1}{0}$ to $\binom{\cos(\Theta)}{\sin(\Theta)}$, and j changes from $\binom{0}{1}$ to $\binom{-\sin(\Theta)}{\cos(\Theta)}$, which naturally corresponds to a rotation transformation in the coordinate system.

$$R = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{bmatrix}$$

The matrix naturally realizes the characteristics of translation, expansion, rotation, etc., so for PCA, finding a good result often requires multiple matrices to be superimposed on each other.

III. VARIANCE AND THE GOAL

What is Covariance

The covariance measures the degree of the linear relationship between two variables. A large positive value indicates positively correlated data. Likewise, a large negative value denotes negatively correlated data. The absolute magnitude of the covariance measures the degree of redundancy. Some additional facts about the covariance.

Covariance Matrix

Population Covariance

$$\operatorname{Cov}(x, y) = \sum_{i=1}^{n} \frac{(x_i - \overline{x})(y_i - \overline{y})}{n}$$

The denominator here is n, which means that we have all the sample data, that is, we are studying the overall data situation

Sample Covariance

$$\text{Cov}(x, y) = \sum_{i=1}^{n} \frac{(x_i - \overline{x})(y_i - \overline{y})}{n - 1}$$

The denominator here is $n-1$, which is called the **Bessel correction**. This is because when we use only a sample (rather than the entire population) to estimate the covariance, we need to correct the estimate. The specific reason is as follows:

- ✓ **Unbiased estimate:** If we use the sample mean $\overline{x}$ and $\overline{y}$ instead of the population mean, then directly using the formula with a denominator of n will lead to a biased estimate of the covariance, that is, **the mean is lower than the actual value.** In order to obtain an unbiased estimate, we use $n-1$ to correct this bias.
- ✓ **Degrees of freedom:** The sample means $\overline{x}$ and $\overline{y}$ are calculated based on n samples, and **they themselves also occupy one degree of freedom.** Therefore, when calculating the sample covariance, the actual degree of freedom is $n-1$ instead of n.

Covariance cleverly expresses the relationship between variables with simple numbers, making it easier to see.It's actually the sign of the covariance that matters:

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

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(x, y) & \text{cov}(y, y) \end{bmatrix}$$

also because:

$$\text{cov}(x, y) = \sum_{i=1}^{n} \frac{x_i y_i}{n - 1}$$

$$\text{cov}(x, x) = \sum_{i=1}^{n} \frac{x_1^2}{n - 1}$$

Substitute:

$$C = \begin{bmatrix} \sum_{i=1}^{n} \frac{x_1^2}{n-1} & \sum_{i=1}^{n} \frac{x_i y_i}{n-1} \\ \sum_{i=1}^{n} \frac{x_i y_i}{n-1} & \sum_{i=1}^{n} \frac{y_1^2}{n-1} \end{bmatrix}$$

Expand to get:

$$C_X = \frac{1}{n-1} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

$$= \frac{1}{n-1} DD^T$$

DD^T is an $m \times m$ matrix, and each element (i, j) of this matrix represents the inner product between feature i and feature j. Specifically, the (i, j) element of DD^T represents the sum of the dot products between the ith feature and the jth feature, which is actually the covariance of the two feature vectors (without centering). $n-1$is used to find the average value.

In principal component analysis (PCA), this covariance matrix is used to find the principal components of the data, thereby reducing the dimensionality of the data and re-

moving redundant information.
We begin by rewriting C_Y in terms of the unknown variable.

$$\begin{aligned} C_Y &= \frac{1}{n-1} YY^T \\ &= \frac{1}{n-1} (PX)(PX)^T \\ &= \frac{1}{n-1} PXX^T P^T \\ &= P\left(\frac{1}{n-1} XX^T\right) P^T \end{aligned}$$

$$C_Y = PC_X P^T$$

Note that we have identified the covariance matrix of X in the last line.

Now comes the trick. We select the matrix P to be a matrix where each row π is an eigenvector of $\frac{1}{n} XX^T$.

By this selection,$\boldsymbol{P \equiv E^T}$. With this relation and Theorem 1 of Appendix $A(P^{-1} = P^T)$ we can finish evaluating C_Y

$$\begin{aligned} C_Y &= PC_X P^T \\ &= P(E^T DE)P^T \\ &= P(P^T DP)P^T \\ &= (PP^T)D(PP^T) \\ &= (PP^{-1})D(PP^{-1}) \end{aligned}$$

It is evident that the choice of P diagonalizes C_Y. This was the goal for PCA. We can summarize the results of PCA in the matrices P and C_Y.

- ♬ The principal components of X are the eigenvectors of $C_X = \frac{1}{n} XX^T$
- ♬ The i^{th} diagonal value of C_Y is the variance of X along π.

In practice computing PCA of a data set X entails

- ⋙ subtracting off the mean of each measurement type
- ⋙ computing the eigenvectors of C_X.