

Master of Information and Data Science
DATASCI 281: Computer Vision
Spring 2022

Professor Hany Farid
University of California, Berkeley

Principal Component Analysis

Principal component analysis (PCA) is the classic approach to reducing the complexity of analyzing high-dimensional data by projection into a lower-dimensional linear subspace. PCA projects data onto axes of maximal data variance. In so doing, the dimensionality of data is reduced while minimizing the loss of information or distortion. For example, shown in the left panel of Figure 1 are data in a 2-D space. Shown in the right panel is this data projected onto a 1-D linear subspace (dashed black line) as determined by PCA. Notice that this projection axis coincides with the axis along which the data varies the most. As a result, when the data is projected onto this axis it reduces the dimensionality while preserving as much of the original structure as possible.

In the language of linear algebra, PCA is a change of basis where the new basis reduces redundancy in the data. We quantify the amount of redundancy as the amount of co-variation between the components of each data point. In the left panel of Figure 1, for example, the x - and y -components of the data clearly covary and so knowledge of one component provides knowledge of the other – these components are somewhat redundant. In the transformed space in the right panel of Figure 1, the y -component of the data has been collapsed to 0 and so there is no covariation and we can eliminate this dimension.

We say that the goal of PCA is to diagonalize the *covariance matrix*. Without loss of generality, assume that the m 2-D data points in the above example are centered at the origin (i.e., are zero-mean). The covariance

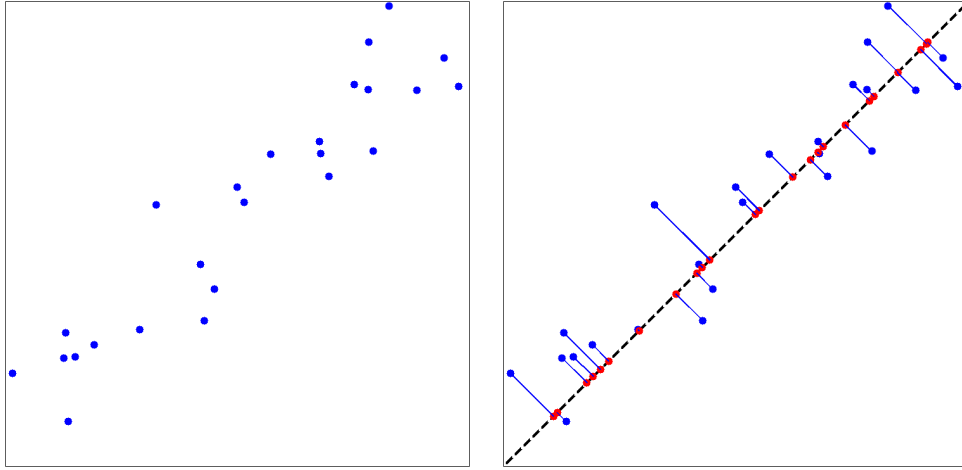


Figure 1: Dimensionality reduction from 2-D (blue) to 1-D (red).

matrix C is:

$$\begin{aligned}
 C &= MM^T \\
 &= \begin{pmatrix} x_1 & \dots & x_m \\ y_1 & \dots & y_m \end{pmatrix} \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_m & y_m \end{pmatrix} \\
 &= \begin{pmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 \end{pmatrix}. \tag{1}
 \end{aligned}$$

We seek to transform the data points (x_i, y_i) such that the off-diagonal elements of the matrix C are zero.

Each column of the $n \times m$ data matrix M contains a n -dimensional data point. A change of basis from the original canonical basis to a new orthonormal basis is computed as:

$$\tilde{M} = EM, \tag{2}$$

where each row of the matrix E contains a new basis vector. We seek the matrix E that diagonalizes the covariance matrix of our transformed data:

$$\tilde{C} = \tilde{M}\tilde{M}^T. \tag{3}$$

We will show that if the rows of the matrix E contain the eigenvectors of the covariance matrix C , then the covariance matrix \tilde{C} of the transformed data will be diagonal.

To begin, let's rewrite the covariance matrix \tilde{C} in terms of the original data matrix:

$$\begin{aligned}
\tilde{C} &= \tilde{M}\tilde{M}^T \\
&= (EM)(EM)^T \\
&= (EM)(M^T E^T) \\
&= E(MM^T)E^T \\
&= ECE^T.
\end{aligned} \tag{4}$$

Now, consider the singular value decomposition (SVD) of the symmetric matrix C :

$$C = USV = USU^T, \tag{5}$$

where, U and V are orthonormal matrices, S is a diagonal matrix, and because C is symmetric, $V = U^T$. The columns of the matrix U contain the eigenvectors of C . Recall that the rows of the matrix E contain the eigenvectors of C . As such we see that $E^T = U$. Substituting this SVD decomposition into the above expression for \tilde{C} yields:

$$\begin{aligned}
\tilde{C} &= ECE^T \\
&= E(USU^T)E^T \\
&= E(E^T S E)E^T \\
&= (EE^T)S(EE^T).
\end{aligned} \tag{6}$$

Because the matrix E is orthonormal, its transpose is its inverse ($E^T = E^{-1}$), so:

$$\begin{aligned}
\tilde{C} &= (EE^T)S(EE^T) \\
&= (EE^{-1})S(EE^{-1}) \\
&= S,
\end{aligned} \tag{7}$$

where recall that S is a diagonal matrix. We see, therefore, that projecting the original data M onto the eigenvectors E of the covariance matrix C results in a new diagonal covariance matrix \tilde{C} . As desired, the components of the transformed data do not covary.

Note that by formulating the problem of dimensionality reduction in terms of maximizing projected variance, it is being implicitly assumed that the original data is Gaussian distributed. Significant deviations of data from this assumption can yield highly undesirable results in which significant distortions are introduced into the projected data.

Implementation: Implementing PCA is relatively straight-forward. Denote column vectors $\vec{x}_i \in \mathcal{R}^n$, $i = 1, \dots, m$ as the input data. The overall mean is:

$$\vec{\mu} = \frac{1}{m} \sum_{i=1}^m \vec{x}_i \quad (8)$$

The zero-meaned data is packed into a $n \times m$ matrix:

$$M = (\vec{x}_1 - \vec{\mu} \quad \vec{x}_2 - \vec{\mu} \quad \dots \quad \vec{x}_m - \vec{\mu}) \quad (9)$$

The $n \times n$ covariance matrix is computed as:

$$C = MM^T. \quad (10)$$

The principle components are the eigenvectors \vec{e}_j of the covariance matrix (i.e., $C\vec{e}_j = \lambda_j\vec{e}_j$), where the eigenvalue, λ_j is proportional to the variance of the original data along the j^{th} eigenvector. The dimensionality of each \vec{x}_i is reduced from n to p by projecting (via an inner product) each \vec{x}_i onto the top p eigenvalue-eigenvectors. The resulting p -dimensional vector is the reduced-dimension representation. This, of course, is simply a change of basis, where the basis is now computed directly from the underlying data.

If the dimensionality of n is larger than the number of data points m , then the $n \times n$ covariance matrix MM^T may become prohibitively large. In this case, the eigenvectors of the smaller $m \times m$ covariance $M^T M$ can be computed, from which the desired eigenvectors of MM^T can be efficiently determined. Let \vec{e}_j be an eigenvector of the smaller $M^T M$ covariance matrix:

$$M^T M \vec{e}_j = \lambda \vec{e}_j. \quad (11)$$

Left multiplying each side of the above equation by M yields:

$$\begin{aligned} M(M^T M)\vec{e}_j &= \lambda M\vec{e}_j \\ (MM^T)M\vec{e}_j &= \lambda M\vec{e}_j \\ CM\vec{e}_j &= \lambda M\vec{e}_j, \end{aligned} \quad (12)$$

from which we see that $M\vec{e}_j$ is the eigenvector of the desired covariance matrix.