

# Reading Report:

## *A Tutorial on Principal Component Analysis*

Boyuan Du (2024151470021)  
School of Software, **Sichuan University**  
Email: 2024151470021@stu.scu.edu.cn

September 24, 2025

**Course / Assignment:** Reading report on PCA.

**Primary Source:** J. Shlens, “A Tutorial on Principal Component Analysis,” 2014.

### Abstract

PCA is a powerful, fundamental technique for extracting useful information from complex data by reducing dimensionality to the data’s essential features. The essence of PCA lies in its ability to identify the directions of maximum variance in the data, represented by the principal components. By representing the data in these components’ coordinates, PCA effectively reduces redundancy and highlights the most informative aspects of the dataset. The tutorial emphasizes two main approaches to PCA: eigen-decomposition (ED) of the covariance matrix and singular value decomposition (SVD) of the data matrix. Both methods yield the same principal components. However, they share a common limitation: they assume principal components are orthogonal, which restricts them from capturing non-orthogonal structure; Independent Component Analysis (ICA) can address this by seeking statistically independent components.

#### **About the experiment on MNIST.**

We follow the tutorial pipeline: we load MNIST and arrange the data matrix  $X \in \mathbb{R}^{784 \times n}$  with rows as pixel features and columns as samples. We then subtract the mean of each feature to center the data. Next, we compute PCA via SVD by forming  $Y = X^\top / \sqrt{n-1}$  and taking  $\text{SVD}(Y) = U\Sigma V^\top$ ; the columns of  $V$  are the principal directions. Using the first 2,000 training images and 2,000 test images, we project the data onto the first two principal components ( $k = 2$ ) to obtain a 2D embedding  $Z = V_{1:2}^\top X$ . In our notation,  $V_{1:2}$  denotes the submatrix consisting of the first two columns of  $V$  (the first two principal components), so  $Z = V_{1:2}^\top X$  is the projection of  $X$  onto this 2D PCA subspace. Finally, we plot the 2D points, using different colors or markers to indicate the ten digit classes.

**Keywords:** PCA, covariance, eigen-decomposition, SVD, variance.

# Contents

<b>1</b>	<b>Foundation &amp; Settings</b>	<b>3</b>
<b>2</b>	<b>Covariance, Redundancy, and SNR</b>	<b>3</b>
2.1	Covariance . . . . .	3
2.2	Covariance, Redundancy, and Noise . . . . .	4
2.3	SNR & Variance . . . . .	4
<b>3</b>	<b>Assumptions &amp; Limits of PCA</b>	<b>4</b>
<b>4</b>	<b>PCA as a Change of Basis</b>	<b>5</b>
<b>5</b>	<b>Two Powerful Routes to PCA: ED &amp; SVD</b>	<b>6</b>
<b>6</b>	<b>The Viability of PCA in Dimensionality Reduction</b>	<b>7</b>
<b>7</b>	<b>Experiments on MNIST</b>	<b>7</b>
<b>8</b>	<b>Conclusion</b>	<b>8</b>
<b>A</b>	<b>Mathematical Notations</b>	<b>9</b>

# 1 Foundation & Settings

Principal Component Analysis (PCA) is a technique to extract useful and hidden information and knowledge from complex and large datasets. It is entirely based on linear algebra to achieve the goal of dimensionality reduction, in which vectors and matrices are crucial and fundamental elements. Like the example of a fluctuating spring, PCA can extract the main movement along a single axis  $x$  from three different path records at different angles, which are both noisy and redundant. A matrix in PCA can serve multiple roles, such as

$$X = [x^{(1)} \ \dots \ x^{(n)}] \in \mathbb{R}^{m \times n}, \quad (1)$$

in which columns are samples; rows are features and they are centered to zero mean. and **linear transformations** such as stretching, rotation, and orthogonal changes of basis:

$$C_Y = PC_X P^\top P P^\top = I_m. \quad (2)$$

$P$  maps  $C_X$  to a new basis in which the off-diagonal elements are zero. In the process of PCA, we measure the dataset along orthonormal directions and find the direction of the largest variance, which is called the **principal component (PC)**.

$$p_i = \operatorname{argmax}_{p_i} p_i^\top C_X p_i$$

Thus, in linear algebra, we use a set of orthonormal vectors as a basis to formalize how we measure the data. The essence of PCA is to find a linear transformation  $P$  that re-expresses the data  $X$  in a new basis,  $Y = PX$ , in which the covariance  $C_Y$  becomes a diagonal matrix, in order to reduce redundancy and concentrate variance while isotropic noise is de-emphasized.

## 2 Covariance, Redundancy, and SNR

### 2.1 Covariance

#### Intuitive Definition of Covariance

Covariance represents the joint variability. In PCA, it reflects the degree to which different dimensions of a dataset shared information between features, which we call *redundancy*. When two variables share the same scale, the larger the covariance is, the more two dimensions are correlated and strongly correlated, which means greater redundancy.

$$C_X = \frac{1}{n} X X^\top, \quad (C_X)_{ij} = \frac{1}{n} \sum_{\ell=1}^n X_{i\ell} X_{j\ell}. \quad (3)$$

In the equation above,  $i$  is the  $i$ -th row of  $X$  and  $j$  is the  $j$ -th row of  $X$ . The diagonal elements of  $C_X$  represent the variance of each feature, while the off-diagonal elements represent the covariance between different features. The size of the covariance indicates the degree of redundancy between two features. The size of the elements in the diagonal of  $C_X$  indicates the variance along each feature, which reflects the useful information of the data.

When two variables are measured on different scales, we use the *correlation coefficient* to measure the degree of correlation between two variables.

## 2.2 Covariance, Redundancy, and Noise

The data contains both the useful information(signal) and the irrelevant information(noise). And in PCA we expect the noise we accept to approach zero. We adopt an additive-noise model.

$$X = S + N$$

Then the covariance simplifies to

$$C_X = \text{Cov}(X) = \text{Cov}(S) + \text{Cov}(N) = C_S + C_N.$$

So we first divide the covariance matrix into two parts: diagonal part and off-diagonal part. The former consists of the variances of the signal direction and the noise direction.

$$C_X = C_S + C_N \quad (4)$$

where  $C_S$  is the covariance of the signal and  $C_N$  is the covariance of the noise. The off-diagonal part consists of the covariances between different features. Now it is clear that PCA aims to reduce both the irrelevant information (noise) and the repetitive information (redundancy) by finding a new basis to re-express the data in, where the covariance matrix becomes diagonal and the variance along each dimension is ordered from large to small.

$$\text{diag}(\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m)$$

## 2.3 SNR & Variance

Before we take steps to decrease noise, we measure the quality of the data by comparing the variance between the signal and the noise. Thus, we define the *Signal-to-Noise Ratio (SNR)* as:

$$\text{SNR} = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2} \quad (5)$$

where  $\sigma_{\text{signal}}^2$  is the variance along the signal direction and  $\sigma_{\text{noise}}^2$  is the variance along the noise direction. A higher SNR indicates a cleaner and more reliable measurement, while a lower SNR suggests that the data are more contaminated by noise.

In PCA, we make an effort to increase the SNR by seeking directions where  $\sigma_{\text{noise}}^2$  approaches zero.

## 3 Assumptions & Limits of PCA

### I. Linearity

PCA assumes a **linear change of basis**. With a dataset  $X \in \mathbb{R}^{m \times n}$  and covariance  $C_X = \frac{1}{n}XX^\top$ , it seeks a linear transformation  $P$  to re-express  $X$  in a new basis  $Y = PX$  where the covariance  $C_Y = \frac{1}{n}YY^\top$  becomes diagonal.

$$Y = PX, \quad P \in O(m), \quad P^\top P = I_m. \quad (6)$$

However, this assumption restricts PCA to extracting only linear features from data; Independent Component Analysis (ICA) can further tackle this problem efficiently.

## II. Large Variance = Important Structure

PCA assumes that directions with the largest variance correspond to the most important underlying structure, which contains the most valuable information. We assume that the useful information is embedded in direction  $G$ , so this assumption is:

$$\forall g_{ij} \in G, \sigma^2 = \mathbf{g}_{ij} C_X \mathbf{g}_{ij}^\top \rightarrow \infty. \quad (7)$$

However, this assumption may not always be correct, and in real cases, it often leads to the wrong direction.

## III. The principal components are orthogonal

PCA assumes that the principal components are orthogonal to each other. Now we demonstrate this assumption in the process of PCA: PCA seeks a linear transformation  $P$  to re-express  $X$  in a new basis  $Y = PX$ , where

$$C_Y = PC_X P^\top \quad (8)$$

becomes diagonal. We assume that

$$P \in O(m), \quad p_i^\top p_j = 0 \quad (\forall i \neq j), \quad P^\top P = I_m \quad (\forall i = j, p_i^\top p_j = 1). \quad (9)$$

This assumption is crucial because it ensures that the principal components are uncorrelated, which simplifies the analysis and interpretation of the data. However, it also limits PCA's ability to capture non-orthogonal features in the data.

## 4 PCA as a Change of Basis

**Change-of-Basis Formulation.**

$$\mathbf{y} = \mathbf{P}\mathbf{x}, \quad \mathbf{C}_Y = \mathbf{P}\mathbf{C}_X\mathbf{P}^\top \quad (10)$$

Principal component analysis is the process by which we change the basis of the data matrix; it is the way we change our directions of measurement and evaluation of the data, reducing noise and redundancy to approach an optimal condition. Thus, the principal component information is embedded in the matrix  $\mathbf{P}$  that makes  $\mathbf{C}_Y$  diagonal. Each row of  $\mathbf{P}$  is a principal component, and the variance along that component is given by the corresponding diagonal element of  $\mathbf{C}_Y$ .

### Re-expression

The way we re-express the data in the new basis is by mapping the data onto each principal component. The mapping of a data point  $\mathbf{x}$  onto a principal component  $\mathbf{p}_i$  is given by the dot product  $\langle \mathbf{p}_i, \mathbf{x} \rangle$ . This gives us the coordinate of the data point in the new basis along that principal component. We can visualize this as a rotation and stretching of the basis vectors.

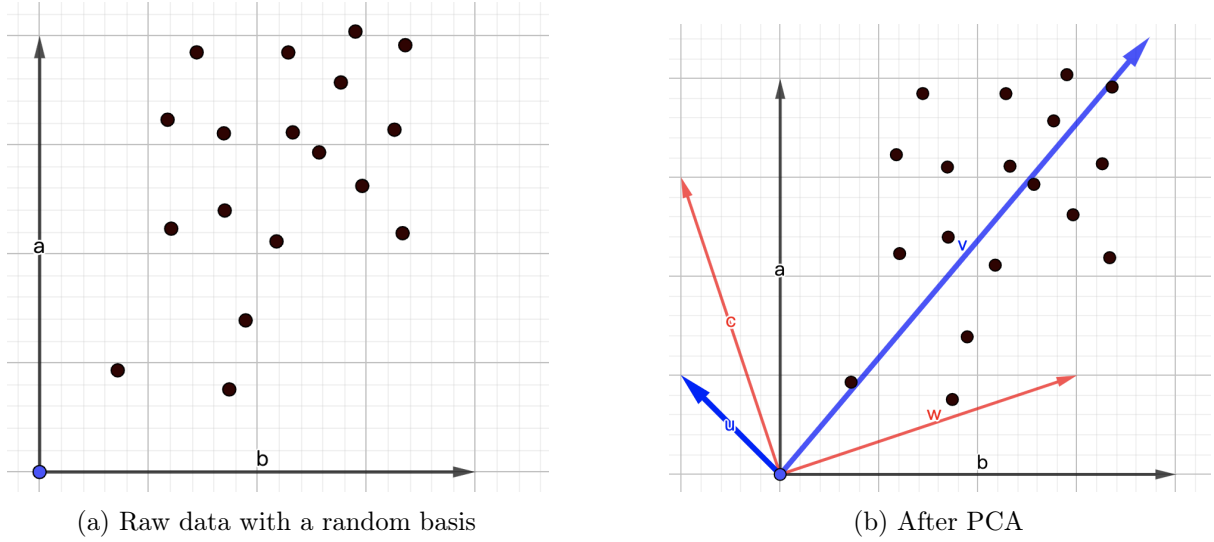


Figure 1: Comparison of data before and after PCA. Figure (a) shows the raw data in a random basis, where the axes are not aligned with the data's main variance direction, leading to a noisy and redundant representation. Figure (b) illustrates the same data after PCA, where the axes are aligned with the principal components, resulting in a clearer and more structured representation.

### Relation to Covariance

PCA is a linear transformation that re-expresses the dataset in a new orthonormal basis so that the covariance matrix becomes diagonal. In this process, the off-diagonal elements, which represent redundancy between variables, are eliminated, and the diagonal elements are the variances along the principal components. In conclusion, PCA is a tool for changing the basis of the data, and the covariance matrix is a measure of how well the data are represented in that basis.

## 5 Two Powerful Routes to PCA: ED & SVD

### ED Route (*Eigen Decomposition*)

As mentioned before, PCA seeks a linear transformation  $\mathbf{P}$  to re-express  $\mathbf{X}$  in a new basis  $\mathbf{Y} = \mathbf{P}\mathbf{X}$ , where the covariance  $\mathbf{C}_Y = \frac{1}{n}\mathbf{Y}\mathbf{Y}^\top$  becomes diagonal. In linear algebra, we know that a symmetric matrix can be diagonalized by its eigenvectors. Thus, we can find the eigenvectors of the covariance matrix  $\mathbf{C}_X = \frac{1}{n}\mathbf{X}\mathbf{X}^\top$  to form the change-of-basis matrix  $\mathbf{P}$ . The eigenvectors of  $\mathbf{C}_X$  are orthogonal and can be used as the new basis for the data. Intuitively, the eigenvectors represent the directions of maximum variance in the data, and the corresponding eigenvalues represent the amount of variance along those directions. By expressing the data in the coordinates of the eigenvectors, we can effectively reduce redundancy and highlight the most informative aspects of the dataset.

$$\mathbf{C}_X = \mathbf{E}\mathbf{D}\mathbf{E}^\top, \quad \text{PCs} \equiv \text{columns of } \mathbf{E}, \text{ variance} \equiv \text{diag}(\mathbf{D}) \quad (11)$$

### SVD Route (*Singular Value Decomposition*)

Another viable route to PCA is to use the singular value decomposition (SVD) of the data matrix  $\mathbf{X}$ . The left singular vectors  $u_i$  and right singular vectors  $v_i$  of  $\mathbf{X}$  are the eigenvectors of  $\mathbf{X}\mathbf{X}^\top$  and  $\mathbf{X}^\top\mathbf{X}$ , respectively. The singular values  $\sigma_i$  are the square roots of the eigenvalues of  $\mathbf{X}^\top\mathbf{X}$ .

SVD decomposes  $\mathbf{X}$  into three matrices:  $\mathbf{U}$ ,  $\mathbf{\Sigma}$ , and  $\mathbf{V}^\top$ . The columns of  $\mathbf{V}$  are the right singular vectors of  $\mathbf{X}$ , which correspond to the principal components. The diagonal elements of  $\mathbf{\Sigma}$  are the singular values, which are related to the variance along each principal component.

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad \text{PCs} \equiv \text{columns of } \mathbf{V} \quad (12)$$

Up to now, it is clear and easy to see the pros and cons of the two methods: **ED** is more intuitive and directly related to the covariance matrix, but the realization of **ED** requires a couple of extra conditions on the matrix, namely that it must be square and symmetric. **SVD** is more general and can handle any data matrix, but it is less intuitive, and the decomposition into three matrices can be more computationally intensive.

## 6 The Viability of PCA in Dimensionality Reduction

PCA, as its name suggests, is a method to find the principal components of the data. But the way we define “*principal*” is not entirely clear. In the context of dimensionality reduction, the degree to which a reduced representation can predict the original data is how we define success; this, in turn, reveals how “principal” the components we use to simplify the data are. Quantitatively, we use the **Mean Squared Error (MSE)** to measure the difference between the original data and the data predicted from the simplified representation.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \|x^{(i)} - \hat{x}^{(i)}\|^2 \quad (13)$$

To minimize the MSE, we need to maximize the variance along the principal components we choose to keep. The mathematical result is that we should choose the top  $k$  eigenvectors of the covariance matrix  $\mathbf{C}_X$  or the top  $k$  right singular vectors of the data matrix  $\mathbf{X}$ , as in PCA.

$$\text{MSE} = \sum_{i=k+1}^m \lambda_i \quad (14)$$

Thus, PCA is a viable method for dimensionality reduction because it effectively captures the most important features of the data while minimizing the loss of information.

## 7 Experiments on MNIST

**Setup** We use the first 2k training and 2k test images from MNIST. Each MNIST picture has  $28 \times 28$  gray pixels (numbers from 0 to 255). We first divide by 255 so the values are between 0 and 1. Then we simplify the picture: read the pixels row by row and put them one after another to make a long list of 784 numbers. This long list is one sample. We place many such lists side by side to build a big matrix  $X$  where rows are pixel positions and columns are images. Before running PCA, for each row, which represents each pixel position, we subtract its average value across all images so the data are centered. Let  $X \in \mathbb{R}^{m \times n}$  (rows = pixels, columns = samples). We center  $X$  by columns and compute PCA via SVD following [1]: let  $Y = \frac{1}{\sqrt{n}}X^\top$ , perform  $\text{SVD}(Y) = U\mathbf{\Sigma}V^\top$ ; the columns of  $V$  are principal components (PCs).

**Procedure** Using the PCs learned from the training set, we project both training and test data onto the first two components ( $k = 2$ ) to obtain a 2D embedding  $Z = V_{1:2}^\top X$ . In our notation,  $V_{1:2}$  denotes the submatrix consisting of the first two columns of  $V$  (the first two principal

components), so  $Z = V_{1:2}^\top X$  is the projection of  $X$  onto this 2D PCA subspace. We then plot the 2D points and use different colors or markers to indicate the ten digit classes (0–9).

**Results** Fig. 2 shows cluster-like patterns with partial overlap among classes. Digits with similar strokes (e.g., 4 and 9) tend to lie closer, while more distinct digits (e.g., 0 and 1) are farther apart.

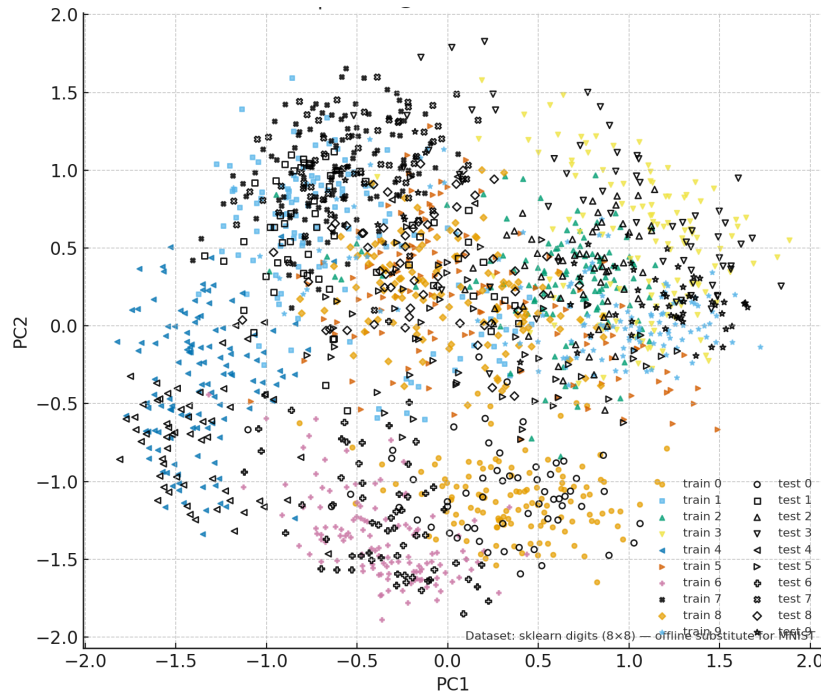


Figure 2: MNIST projected to 2D by PCA.

**Analysis**  $V_{1:2}$  just means “the first two columns of  $V$ .” These two columns are the first two main directions. When we compute  $Z = V_{1:2}^\top X$ , we turn each image into two numbers. So  $Z$  is simply the 2D version of our data. The degree of overlap is high to some extent given 2D is a frankly low dimension for 784D data.

**Conclusion of the experiment** PCA makes a simple rotation to new axes and puts them in order from “changes the most” to “changes less.” Showing only the first two axes gives an easy 2D picture that lets us see rough groups. But because PCA is a linear method and the two axes must be at right angles, it cannot show more complex or curved patterns.

## 8 Conclusion

PCA is a powerful and fundamental technique for extracting useful information from complex, high-dimensional data by transforming the data into a new basis in which the covariance matrix becomes diagonal. The essence of PCA lies in its ability to identify the directions of maximum variance in the data, which are represented by the principal components. By expressing the data in these components’ coordinates, PCA effectively reduces redundancy and highlights the most informative aspects of the dataset. ED and SVD are viable and powerful methods for achieving PCA as well as dimensionality reduction. However, PCA has limitations, such as its reliance on the assumption that principal components are orthogonal. In cases where this assumption does not hold, alternative techniques such as Independent Component Analysis (ICA) may be more appropriate.



## A Mathematical Notations

Table 1: Alphabetical summary of mathematical notations used in the PCA tutorial

Notation	Definition	Corresponds to
$A, B$	Two general matrices used to define or explain other definitions below	tool matrices
$a_i, b_i$	$i$ -th samples of $A$ and $B$	scalar observations in the illustrative example
$C_X = \frac{1}{n}XX^\top$ $C_Y = \frac{1}{n}YY^\top$	Covariance matrix of $X$ Covariance of $Y = PX$ in the new basis	reveals redundancy in dataset $X$ covariance matrix after change of basis
$D$	Diagonal matrix of eigenvalues in eigen-decomposition	variances along principal components
$E$	Matrix whose columns are eigenvectors of $C_X$	principal directions
$I$	Identity matrix	orthonormal basis of $\mathbb{R}^m$
$k$	Target dimension for reduction (also a scalar in the SVD example $Xa = kb$ )	
$m$	Number of features (measurement types)	dimensionality of the dataset
$n$	Number of samples (trials)	size of the dataset
$P = [p_1^\top \cdots p_m^\top]$	Rotation and stretching that transform $X$ into $Y$	change-of-basis matrix $p_i$ , $i$ -th principal component (row of $P$ ), principal axis
$r$	Rank of $X$ (or $X^\top X$ )	intrinsic dimensionality
$\text{SNR} = \sigma_{\text{signal}}^2 / \sigma_{\text{noise}}^2$	Signal-to-noise ratio	measurement quality
$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$	Diagonal matrix of singular values	covariance in the new basis
$\sigma_i$	$i$ -th singular value of $X$ ; $\lambda_i = \sigma_i^2/n$ (if $C_X = \frac{1}{n}XX^\top$ )	scale of mode $i$
$\sigma^2$	Variance of a scalar variable/sequence	spread/energy
$U$	Left singular vectors of $X$	orthonormal basis of the column space
$V$	Right singular vectors of $X$	orthonormal basis of the row space
$\hat{u}_i$	$i$ -th left singular vector; $\hat{u}_i = \frac{1}{\sigma_i}X\hat{v}_i$	output direction of mode $i$
$\hat{v}_i$	$i$ -th eigenvector of $X^\top X$	input direction of mode $i$
$X \in \mathbb{R}^{m \times n}$	Data matrix	stacked measurements dataset
$x^{(j)}$	$j$ -th sample vector (a column of $X$ )	per-sample measurement
$Y = PX$	Data expressed in PCA coordinates	PC scores (coordinates along PCs)
$Z = U^\top X$	Coordinates in the left-singular basis	transformed data
$\lambda_i$	$i$ -th eigenvalue of $C_X$	variance along the $i$ -th PC
$\delta_{ij}$	Kronecker delta ( $= 1$ if $i = j$ , else $0$ )	orthogonality indicator
$\ \cdot\ $	Euclidean norm	vector length
$(\cdot)^\top$	Transpose	matrix transpose
$\cdot$	Dot product	inner product

## References

- [1] J. Shlens, *A Tutorial on Principal Component Analysis*, 2014.