A Complete Tutorial on Eigenvalues, Eigenvectors, and Principal Component Analysis

From Fundamentals to Applications

Prepared by Shashikant Chaudhary
https://www.linkedin.com/in/chaudharysk/

enhanced by Generative AI

A Comprehensive Guide with Examples and Visualizations

December 13, 2025

Abstract

This tutorial provides a comprehensive journey from the fundamental concepts of eigenvalues and eigenvectors to their powerful application in Principal Component Analysis (PCA). We begin with intuitive explanations, progress through rigorous mathematical definitions, work through detailed examples with step-by-step solutions, and conclude with practical applications in data science. Throughout the tutorial, we include visualizations to build geometric intuition and Python code snippets for implementation.

Contents

1 Introduction: Why Study Eigenvalues?

Before diving into definitions, let's understand *why* eigenvalues and eigenvectors matter. Consider a matrix as a **transformation** that acts on vectors. When we multiply a matrix by a vector, the vector typically changes both its direction and magnitude.

However, there exist special vectors that, when transformed by a matrix, only get *scaled*—they don't rotate or change direction. These special vectors are called **eigenvectors**, and the scaling factors are called **eigenvalues**.

The Big Picture

Eigenvalues and eigenvectors reveal the **intrinsic properties** of linear transformations. They tell us:

- Which directions are preserved under the transformation
- How much stretching or compression occurs along those directions
- Whether the transformation is stable, oscillatory, or explosive

1.1 Applications Across Fields

Eigenvalues appear everywhere in science and engineering:

- **Physics**: Quantum mechanics (energy levels), vibration analysis
- **Engineering**: Structural stability, control systems
- **Data Science**: PCA, spectral clustering, PageRank algorithm
- **Computer Graphics**: 3D transformations, facial recognition
- **Economics**: Input-output models, Markov chains

2 Matrices as Linear Transformations

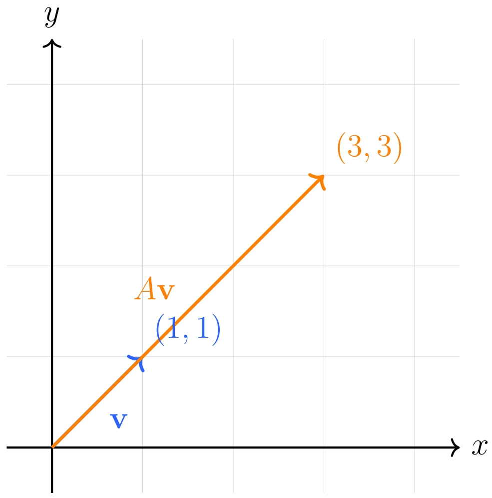
Before studying eigenvalues, we must understand what matrices *do* to vectors.

2.1 Geometric Interpretation

A 2×2 matrix A transforms every point in the plane to a new location. Consider:

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

When we apply A to a vector $\mathbf{v}$, we compute $A\mathbf{v}$.



Same direction, scaled by 3

Let's verify: if $\mathbf{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, then:

$$A\mathbf{v} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \cdot 1 + 1 \cdot 1 \\ 1 \cdot 1 + 2 \cdot 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The vector $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is special—it only gets scaled by 3!

3 Formal Definitions

Definition 3.1 (Eigenvalue and Eigenvector)**.** Let A be an $n \times n$ matrix. A scalar λ is called an **eigenvalue** of A if there exists a nonzero vector $\mathbf{v}$ such that:

$$\boxed{A\mathbf{v} = \lambda\mathbf{v}}$$

The nonzero vector $\mathbf{v}$ is called an **eigenvector** corresponding to λ.

Critical Observation

The equation $A\mathbf{v} = \lambda\mathbf{v}$ says that applying A to $\mathbf{v}$ has the same effect as multiplying $\mathbf{v}$ by the scalar λ. The vector $\mathbf{v}$ **does not change direction**—it only gets stretched (if $|\lambda| > 1$), compressed (if $|\lambda| < 1$), or flipped (if $\lambda < 0$).

3.1 Etymology

The word "eigen" comes from German, meaning "own" or "characteristic." Eigenvalues are the *characteristic values* that belong intrinsically to the matrix.

4 Finding Eigenvalues: The Characteristic Equation

4.1 Deriving the Method

Starting from $A\mathbf{v} = \lambda\mathbf{v}$, we want to find λ. Rearranging:

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} && (1) \\ A\mathbf{v} - \lambda\mathbf{v} &= \mathbf{0} && (2) \\ (A - \lambda I)\mathbf{v} &= \mathbf{0} && (3) \end{aligned}$$

where I is the identity matrix. This is a system of linear equations. For a *nonzero* solution $\mathbf{v}$ to exist, the matrix $(A - \lambda I)$ must be **singular** (non-invertible). This happens exactly when:

$$\boxed{\det(A - \lambda I) = 0}$$

Definition 4.1 (Characteristic Polynomial)**.** The **characteristic polynomial** of matrix A is:

$$p(\lambda) = \det(A - \lambda I)$$

The eigenvalues of A are the roots of $p(\lambda) = 0$, called the **characteristic equation**.

5 Step-by-Step Examples

5.1 Example 1: A 2×2 Matrix

Complete Worked Example

Find all eigenvalues and eigenvectors of:

$$A = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$$

Step 1: Set up the characteristic equation

$$A - \lambda I = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{pmatrix}$$

Step 2: Compute the determinant

$$\begin{aligned} \det(A - \lambda I) &= (4-\lambda)(3-\lambda) - (2)(1) && (4) \\ &= 12 - 4\lambda - 3\lambda + \lambda^2 - 2 && (5) \\ &= \lambda^2 - 7\lambda + 10 && (6) \end{aligned}$$

Step 3: Solve the characteristic equation

$$\lambda^2 - 7\lambda + 10 = 0$$

Using the quadratic formula or factoring:

$$(\lambda - 5)(\lambda - 2) = 0$$

Therefore: $\boxed{\lambda_1 = 5}$ and $\boxed{\lambda_2 = 2}$

Step 4: Find eigenvectors for $\lambda_1 = 5$
Solve $(A - 5I)\mathbf{v} = \mathbf{0}$:

$$\begin{pmatrix} 4-5 & 2 \\ 1 & 3-5 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} -1 & 2 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

From the first equation: $-v_1 + 2v_2 = 0 \Rightarrow v_1 = 2v_2$
Let $v_2 = 1$, then $v_1 = 2$. So:

$$\boxed{\mathbf{v}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}} \text{ (or any scalar multiple)}$$

Step 5: Find eigenvectors for $\lambda_2 = 2$
Solve $(A - 2I)\mathbf{v} = \mathbf{0}$:

$$\begin{pmatrix} 4-2 & 2 \\ 1 & 3-2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

From the first equation: $2v_1 + 2v_2 = 0 \Rightarrow v_1 = -v_2$
Let $v_2 = 1$, then $v_1 = -1$. So:

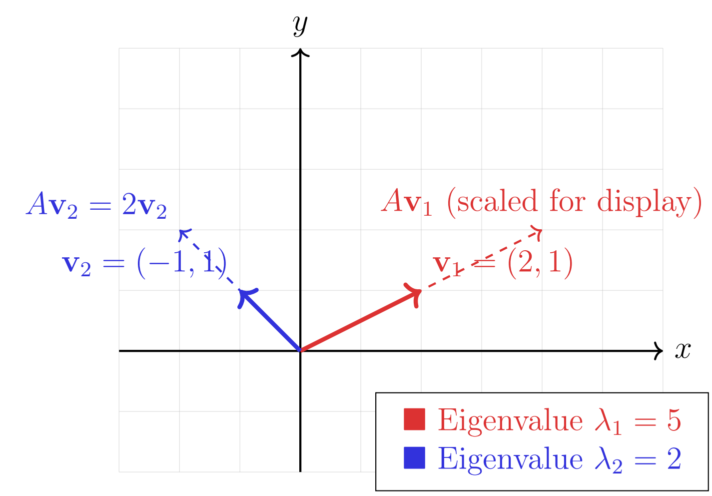
$$\boxed{\mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}} \text{ (or any scalar multiple)}$$

Verification:

$$A\mathbf{v}_1 = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 5 \end{pmatrix} = 5 \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 5\mathbf{v}_1 \quad \checkmark$$

$$A\mathbf{v}_2 = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix} = 2 \begin{pmatrix} -1 \\ 1 \end{pmatrix} = 2\mathbf{v}_2 \quad \checkmark$$

5.2 Geometric Visualization



5.3 Example 2: A 3×3 Matrix

Higher Dimensional Example

Find the eigenvalues of:

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & 1 & 4 \end{pmatrix}$$

Step 1: Form $A - \lambda I$

$$A - \lambda I = \begin{pmatrix} 1-\lambda & 2 & 0 \\ 0 & 3-\lambda & 0 \\ 2 & 1 & 4-\lambda \end{pmatrix}$$

Step 2: Compute the determinant
Since the matrix has zeros, we expand along the first row:

$$\det(A - \lambda I) = (1-\lambda)\det\begin{pmatrix} 3-\lambda & 0 \\ 1 & 4-\lambda \end{pmatrix} - 2\det\begin{pmatrix} 0 & 0 \\ 2 & 4-\lambda \end{pmatrix} + 0 \quad (7)$$

$$= (1-\lambda)(3-\lambda)(4-\lambda) - 2(0) \quad (8)$$

$$= (1-\lambda)(3-\lambda)(4-\lambda) \quad (9)$$

Step 3: Solve
The eigenvalues are: $\boxed{\lambda_1 = 1, \quad \lambda_2 = 3, \quad \lambda_3 = 4}$

Remark 5.1. Although this matrix is not triangular, the presence of zeros in the second row significantly simplifies the calculation. Expanding the determinant along the second row (which contains $0, 3-\lambda, 0$) is the most efficient method here. For triangular matrices, the eigenvalues are simply the diagonal entries.

5.4 Example 3: Complex Eigenvalues

Rotation Matrix

Find the eigenvalues of the 90° rotation matrix:

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Characteristic equation:

$$\det(R - \lambda I) = \det\begin{pmatrix} -\lambda & -1 \\ 1 & -\lambda \end{pmatrix} = \lambda^2 + 1 = 0$$

Solution:

$$\lambda^2 = -1 \Rightarrow \boxed{\lambda = \pm i}$$

Geometric Interpretation

Complex eigenvalues indicate **rotation**. A pure rotation (no scaling) has eigenvalues on the unit circle in the complex plane. The eigenvalue $i = e^{i\pi/2}$ corresponds to a 90° rotation.

6 Important Properties of Eigenvalues

Theorem 6.1 (Trace and Determinant)**.** For an $n \times n$ matrix A with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$:

1. $\mathrm{tr}(A) = \lambda_1 + \lambda_2 + \cdots + \lambda_n \quad$ (sum of eigenvalues = trace)
2. $\det(A) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n \quad$ (product of eigenvalues = determinant)

Verification with Example 1:

- Trace: $4 + 3 = 7 = 5 + 2 \quad \checkmark$
- Determinant: $4 \cdot 3 - 2 \cdot 1 = 10 = 5 \cdot 2 \quad \checkmark$

Theorem 6.2 (Eigenvalues of Special Matrices)**.**

1. **Symmetric matrices** ($A = A^T$) have only *real* eigenvalues
2. **Orthogonal matrices** ($A^T A = I$) have eigenvalues with $|\lambda| = 1$
3. **Positive definite matrices** have all *positive* eigenvalues
4. **Triangular matrices** have eigenvalues equal to the diagonal entries

7 Eigendecomposition (Diagonalization)

Definition 7.1 (Diagonalizable Matrix)**.** A matrix A is **diagonalizable** if it can be written as:

$$\boxed{A = PDP^{-1}}$$

where D is a diagonal matrix of eigenvalues and P is a matrix whose columns are the corresponding eigenvectors.

7.1 Constructing the Decomposition

Using our Example 1 with $\lambda_1 = 5$, $\mathbf{v}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $\lambda_2 = 2$, $\mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$:

$$P = \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix}$$

$$P^{-1} = \frac{1}{2+1} \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$$

Why Diagonalization Matters

1. **Matrix powers**: $A^n = PD^nP^{-1}$ (diagonal matrices are trivial to exponentiate)
2. **Matrix exponential**: $e^A = Pe^DP^{-1}$ (crucial for differential equations)
3. **Understanding transformations**: Diagonalization reveals the "natural axes" of the transformation

8 Transition to Principal Component Analysis

Now we connect eigenvalues to one of the most important techniques in data science: **Principal Component Analysis (PCA)**.

8.1 The Problem PCA Solves

Imagine you have data with many features (dimensions). PCA answers:

1. Which directions in the data have the *most variation*?
2. Can we reduce dimensions while preserving important information?
3. What are the underlying patterns in the data?

The Core Insight

PCA finds the eigenvectors of the **covariance matrix**. These eigenvectors point in the directions of maximum variance in the data. The corresponding eigenvalues tell us how much variance exists along each direction.

9 Mathematical Foundation of PCA

9.1 Covariance Matrix

Definition 9.1 (Covariance — Intuitive Explanation)**. Covariance** measures how two variables change together.

- **Positive covariance**: When X goes up, Y tends to go up too
- **Negative covariance**: When X goes up, Y tends to go down
- **Zero covariance**: X and Y have no linear relationship

How to Calculate Covariance (Step by Step):
Given n data points for variables X and Y:

1. **Find the mean** of each variable: $\bar{X}$ and $\bar{Y}$
2. **Center each value**: subtract the mean from each data point
3. **Multiply the centered values** for each pair: $(x_i - \bar{X}) \times (y_i - \bar{Y})$
4. **Add up all the products**
5. **Divide by** $(n - 1)$ to get the covariance

Covariance Calculation Example

Suppose we have 4 data points:

$$X = \{2, 4, 6, 8\}, \quad Y = \{1, 3, 4, 6\}$$

Step 1: Find means: $\bar{X} = 5$, $\bar{Y} = 3.5$
Step 2 & 3: Center and multiply each pair:

x_i	y_i	$(x_i - 5)$	$(y_i - 3.5)$	Product
2	1	-3	-2.5	$(-3) \times (-2.5) = 7.5$
4	3	-1	-0.5	$(-1) \times (-0.5) = 0.5$
6	4	$+1$	$+0.5$	$(+1) \times (+0.5) = 0.5$
8	6	$+3$	$+2.5$	$(+3) \times (+2.5) = 7.5$

Step 4: Sum of products $= 7.5 + 0.5 + 0.5 + 7.5 = 16$
Step 5: Covariance $= \frac{16}{4-1} = \frac{16}{3} \approx 5.33$
The positive covariance confirms: as X increases, Y increases too!

Definition 9.2 (Covariance Matrix)**.** For a dataset with p features, the **covariance matrix** Σ is a $p \times p$ table where each entry shows the covariance between two features:

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_p) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_p, X_1) & \mathrm{Cov}(X_p, X_2) & \cdots & \mathrm{Var}(X_p) \end{pmatrix}$$

Note: Variance is just the covariance of a variable with itself: $\mathrm{Var}(X) = \mathrm{Cov}(X, X)$

9.2 Key Properties of the Covariance Matrix

1. **Symmetric**: $\Sigma = \Sigma^T$ (so $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$)
2. **Positive semi-definite**: All eigenvalues are ≥ 0
3. **Diagonal entries**: $\Sigma_{ii} = \mathrm{Var}(X_i)$ (variance of feature i)

10 The PCA Algorithm

Algorithm 1 Principal Component Analysis

Require: Data matrix X with n samples and p features
Ensure: Principal components and transformed data

1: **Center the data**: Subtract the mean of each feature

$$X_{\text{centered}} = X - \bar{X}$$

2: **Compute covariance matrix**:

$$\Sigma = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}}$$

3: **Find eigenvalues and eigenvectors** of Σ
4: **Sort** eigenvectors by decreasing eigenvalue
5: **Select top k eigenvectors** to form projection matrix W
6: **Project data**: $X_{\text{new}} = X_{\text{centered}} \cdot W$

11 Complete PCA Example

Step-by-Step PCA Calculation

Consider the following 2D dataset with 5 points:

Point	x_1	x_2
1	2.5	2.4
2	0.5	0.7
3	2.2	2.9
4	1.9	2.2
5	3.1	3.0

Step 1: Center the Data
Calculate means:

$$\bar{x}_1 = \frac{2.5 + 0.5 + 2.2 + 1.9 + 3.1}{5} = 2.04$$

$$\bar{x}_2 = \frac{2.4 + 0.7 + 2.9 + 2.2 + 3.0}{5} = 2.24$$

Centered data:

$$X_{\text{centered}} = \begin{pmatrix} 0.46 & 0.16 \\ -1.54 & -1.54 \\ 0.16 & 0.66 \\ -0.14 & -0.04 \\ 1.06 & 0.76 \end{pmatrix}$$

Step 2: Compute Covariance Matrix
The formula is: $\Sigma = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}}$ where $n = 5$ (number of points).

First, let's write out both matrices clearly:

$$X_{\text{centered}} = \begin{pmatrix} 0.46 & 0.16 \\ -1.54 & -1.54 \\ 0.16 & 0.66 \\ -0.14 & -0.04 \\ 1.06 & 0.76 \end{pmatrix} \quad \text{(5 rows, 2 columns)}$$

$$X_{\text{centered}}^T = \begin{pmatrix} 0.46 & -1.54 & 0.16 & -0.14 & 1.06 \\ 0.16 & -1.54 & 0.66 & -0.04 & 0.76 \end{pmatrix} \quad \text{(2 rows, 5 columns)}$$

Now multiply $X_{\text{centered}}^T \times X_{\text{centered}}$ (result will be 2×2):

$$X_{\text{centered}}^T X_{\text{centered}} = \begin{pmatrix} 0.46 & -1.54 & 0.16 & -0.14 & 1.06 \\ 0.16 & -1.54 & 0.66 & -0.04 & 0.76 \end{pmatrix} \begin{pmatrix} 0.46 & 0.16 \\ -1.54 & -1.54 \\ 0.16 & 0.66 \\ -0.14 & -0.04 \\ 1.06 & 0.76 \end{pmatrix}$$

Entry (1,1): First row of X^T · First column of X:

$$(0.46)(0.46) + (-1.54)(-1.54) + (0.16)(0.16) + (-0.14)(-0.14) + (1.06)(1.06)$$

$$= 0.212 + 2.372 + 0.026 + 0.020 + 1.124 = 3.752$$

Entry (1,2) and (2,1): First row of X^T · Second column of X:

$$(0.46)(0.16) + (-1.54)(-1.54) + (0.16)(0.66) + (-0.14)(-0.04) + (1.06)(0.76)$$

$$= 0.074 + 2.372 + 0.106 + 0.006 + 0.806 = 3.362$$

Entry (2,2): Second row of X^T · Second column of X:

$$(0.16)(0.16) + (-1.54)(-1.54) + (0.66)(0.66) + (-0.04)(-0.04) + (0.76)(0.76)$$

$$= 0.026 + 2.372 + 0.436 + 0.002 + 0.578 = 3.412$$

So:

$$X_{\text{centered}}^T X_{\text{centered}} = \begin{pmatrix} 3.752 & 3.362 \\ 3.362 & 3.412 \end{pmatrix}$$

Finally, divide by $(n-1) = 4$:

$$\boxed{\Sigma = \frac{1}{4} \begin{pmatrix} 3.752 & 3.362 \\ 3.362 & 3.412 \end{pmatrix} = \begin{pmatrix} 0.938 & 0.8405 \\ 0.8405 & 0.853 \end{pmatrix}}$$

What Each Entry Means

- $\Sigma_{11} = 0.938$ is the **variance** of x_1 (how spread out x_1 values are)
- $\Sigma_{22} = 0.853$ is the **variance** of x_2 (how spread out x_2 values are)
- $\Sigma_{12} = \Sigma_{21} = 0.8405$ is the **covariance** between x_1 and x_2 (positive means

they increase together)

Step 3: Find Eigenvalues
Characteristic equation $\det(\Sigma - \lambda I) = 0$:

$$(0.938 - \lambda)(0.853 - \lambda) - 0.8405^2 = 0$$

$$\lambda^2 - 1.791\lambda + (0.800 - 0.706) = 0 \implies \lambda^2 - 1.791\lambda + 0.094 = 0$$

Using the quadratic formula:

$$\lambda = \frac{1.791 \pm \sqrt{1.791^2 - 4(0.094)}}{2} = \frac{1.791 \pm \sqrt{2.832}}{2} = \frac{1.791 \pm 1.683}{2}$$

$$\boxed{\lambda_1 \approx 1.737} \quad \boxed{\lambda_2 \approx 0.054}$$

Step 4: Find Eigenvectors
For $\lambda_1 = 1.737$:

$$(\Sigma - 1.737I)\mathbf{v} = \begin{pmatrix} -0.799 & 0.8405 \\ 0.8405 & -0.884 \end{pmatrix} \mathbf{v} = \mathbf{0}$$

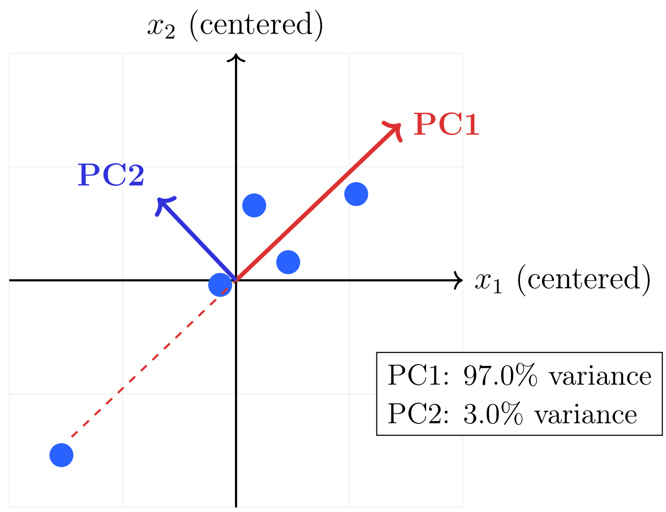
From the first equation: $-0.799v_1 + 0.8405v_2 = 0 \implies v_1 \approx 1.05v_2$

$$\mathbf{v}_1 \approx \begin{pmatrix} 0.725 \\ 0.689 \end{pmatrix} \quad \text{(normalized)}$$

For $\lambda_2 = 0.054$ (orthogonal to $\mathbf{v}_1$):

$$\mathbf{v}_2 \approx \begin{pmatrix} -0.689 \\ 0.725 \end{pmatrix} \quad \text{(normalized)}$$

Step 5: Interpret Results



Variance Explained:

$$\text{PC1 explains: } \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1.737}{1.791} \approx 97.0\%$$

$$\text{PC2 explains: } \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{0.054}{1.791} \approx 3.0\%$$

Key Finding

The first principal component (PC1) captures 97.0% of the variance in the data! This means we could reduce from 2D to 1D and lose only 3.0% of the information. The direction of PC1 shows that x_1 and x_2 are highly correlated—as one increases, so does the other.

Step 6: Project 2D Data to 1D (The Actual Transformation!)

Now we transform our 5 points from 2D to 1D. To do this, we multiply each centered point by the first eigenvector (PC1).

How Projection Works

To project a 2D point onto a 1D line (PC1), we compute the **dot product**:

$$\text{New 1D value} = (x_1^{\text{centered}}) \times (v_1) + (x_2^{\text{centered}}) \times (v_2)$$

where $\mathbf{v}_1 = \begin{pmatrix} 0.725 \\ 0.689 \end{pmatrix}$ is our first eigenvector.

The matrix multiplication:

$$X_{\text{new}} = X_{\text{centered}} \times \mathbf{v}_1 = \begin{pmatrix} 0.46 & 0.16 \\ -1.54 & -1.54 \\ 0.16 & 0.66 \\ -0.14 & -0.04 \\ 1.06 & 0.76 \end{pmatrix} \times \begin{pmatrix} 0.725 \\ 0.689 \end{pmatrix}$$

Calculate each point's new 1D coordinate:

Point	2D Centered (x_1, x_2)	Calculation	1D Value
1	$(0.46, 0.16)$	$(0.46 \times 0.725) + (0.16 \times 0.689)$	$0.334 + 0.110 = \mathbf{0.444}$
2	$(-1.54, -1.54)$	$(-1.54 \times 0.725) + (-1.54 \times 0.689)$	$-1.117 + (-1.061) = \mathbf{-2.178}$
3	$(0.16, 0.66)$	$(0.16 \times 0.725) + (0.66 \times 0.689)$	$0.116 + 0.455 = \mathbf{0.571}$
4	$(-0.14, -0.04)$	$(-0.14 \times 0.725) + (-0.04 \times 0.689)$	$-0.102 + (-0.028) = \mathbf{-0.129}$
5	$(1.06, 0.76)$	$(1.06 \times 0.725) + (0.76 \times 0.689)$	$0.769 + 0.524 = \mathbf{1.292}$

Result:

$$X_{\text{new}} = \begin{pmatrix} 0.444 \\ -2.178 \\ 0.571 \\ -0.129 \\ 1.292 \end{pmatrix} \quad \text{(5 points, now in 1D!)}$$

BEFORE		AFTER
$(2.5, 2.4)$		0.444
$(0.5, 0.7)$	Center data	-2.178
$(2.2, 2.9)$	$\longrightarrow$	0.571
$(1.9, 2.2)$	$\times$ eigenvector	-0.129
$(3.1, 3.0)$		1.292
2 numbers per point		1 number per point
2D (2 features)		**1D** (1 feature)

What Just Happened?

Each 2D point got "squashed" onto the PC1 line. The dot product measures *how far along the PC1 direction* each point lies:

- **Point 2** has the most negative value (-2.178) — it's furthest in the negative PC1 direction
- **Point 5** has the most positive value (1.292) — it's furthest in the positive PC1 direction
- **Point 4** is near zero (-0.129) — it's close to the center

We reduced from 2 numbers per point to 1 number per point, while keeping 97% of the information!

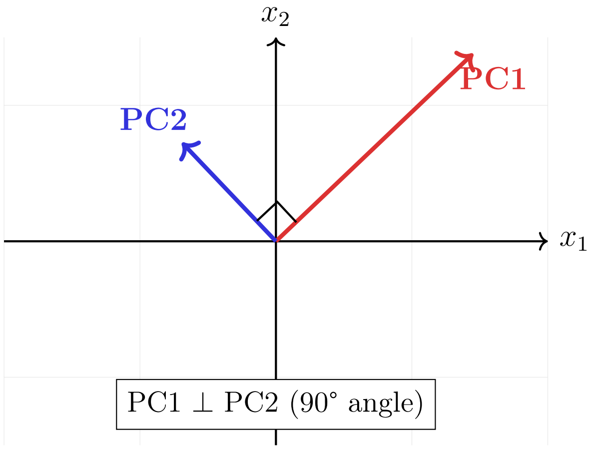
11.1 What is PC2? (The Second Principal Component)

PC2 Explained

PC2 is the **second principal component** — the direction of maximum variance that is **perpendicular (orthogonal)** to PC1.
Think of it this way:

- **PC1** captures the "main trend" in your data (97% of variance)
- **PC2** captures "what's left over" after PC1 (3% of variance)

Why is PC2 perpendicular to PC1?



By making PC2 perpendicular to PC1, we ensure that:

1. PC2 captures *different* information than PC1 (no redundancy)
2. The two components are **uncorrelated**

Calculating PC2 values (same method as PC1):

Our second eigenvector is $\mathbf{v}_2 = \begin{pmatrix} -0.689 \\ 0.725 \end{pmatrix}$

$$\text{PC2 value} = (x_1^{\text{centered}}) \times (-0.689) + (x_2^{\text{centered}}) \times (0.725)$$

Point	2D Centered	PC1 value	PC2 value
1	$(0.46, 0.16)$	0.444	$(0.46)(-0.689) + (0.16)(0.725) = -0.201$
2	$(-1.54, -1.54)$	-2.178	$(-1.54)(-0.689) + (-1.54)(0.725) = -0.055$
3	$(0.16, 0.66)$	0.571	$(0.16)(-0.689) + (0.66)(0.725) = 0.368$
4	$(-0.14, -0.04)$	-0.129	$(-0.14)(-0.689) + (-0.04)(0.725) = 0.067$
5	$(1.06, 0.76)$	1.292	$(1.06)(-0.689) + (0.76)(0.725) = -0.180$

If we keep BOTH PC1 and PC2:

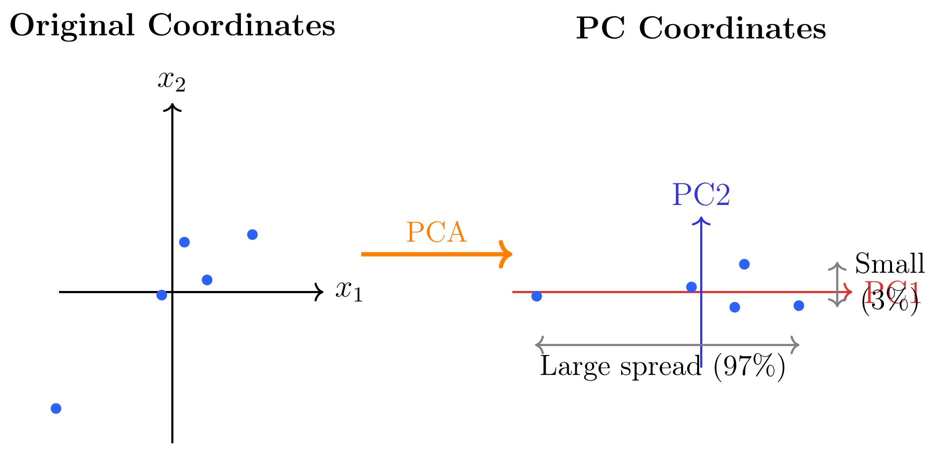
$$X_{\text{transformed}} = \begin{pmatrix} 0.444 & -0.201 \\ -2.178 & -0.055 \\ 0.571 & 0.368 \\ -0.129 & 0.067 \\ 1.292 & -0.180 \end{pmatrix} \quad \leftarrow \text{Still 2D, but in a } \textit{new coordinate system}!$$

When to Use PC2?

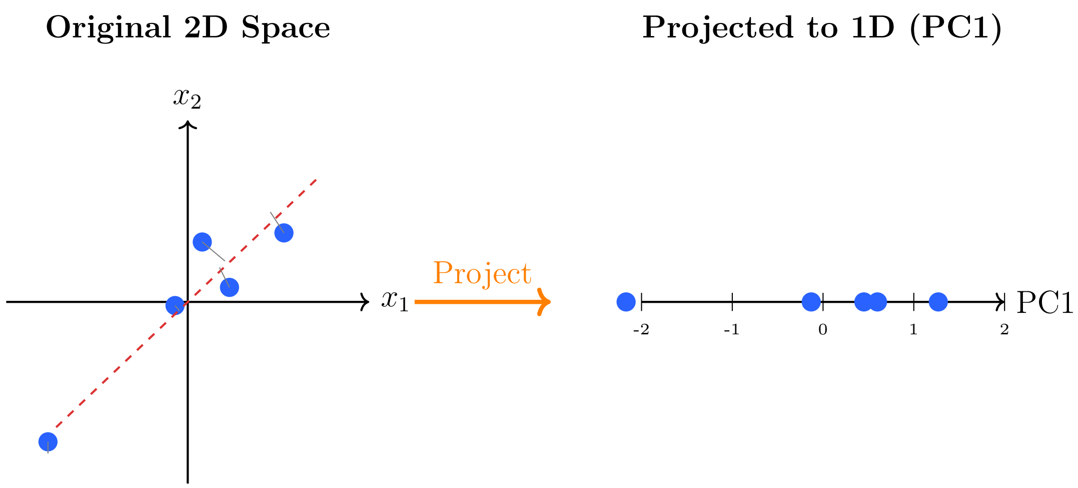
- **Keep only PC1**: When you want maximum compression (2D → 1D). You lose 3% of information.
- **Keep PC1 + PC2**: When you want to rotate your data to a new coordinate system but keep all information. The new axes (PC1, PC2) are aligned with

the "natural directions" of your data.

In higher dimensions (e.g., 100 features), you might keep PC1 through PC10 to reduce from 100D to 10D while retaining 95% of the variance.



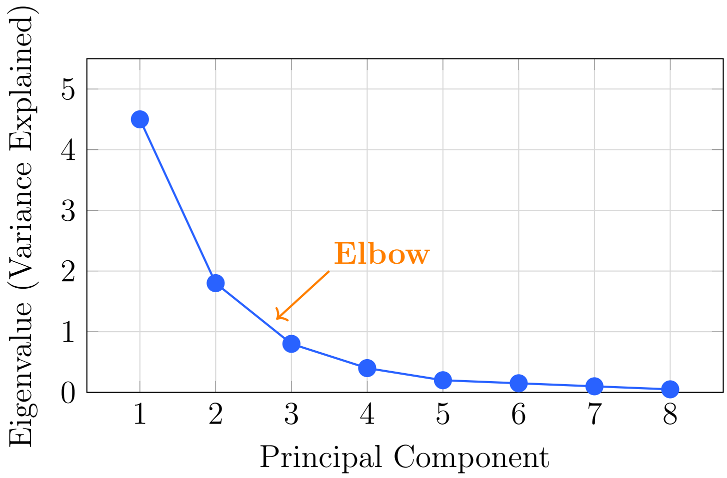
12 Visualizing PCA Dimensionality Reduction



13 Choosing the Number of Components

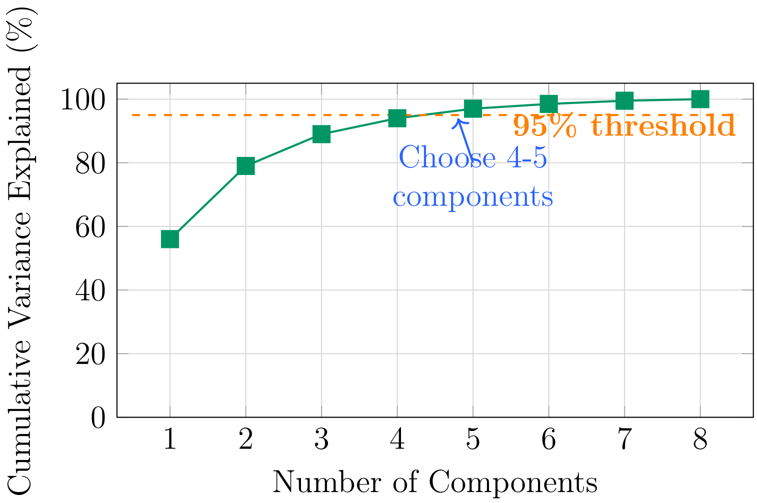
13.1 Scree Plot

A **scree plot** shows eigenvalues in decreasing order. The "elbow" suggests where to cut off.



13.2 Cumulative Variance

Choose k components to explain a target percentage (e.g., 95%) of variance:



14 Python Implementation

14.1 PCA from Scratch

Listing 1: PCA Implementation from Scratch

```python
import numpy as np

def pca_from_scratch(X, n_components=None):
    """
    Perform PCA on dataset X.

    Parameters:
    -----------
    X : array-like, shape (n_samples, n_features)
        Input data
```

```
    n_components : int, optional
        Number of components to keep

    Returns:
    --------
    X_transformed : Transformed data
    components : Principal component directions
    explained_variance : Variance explained by each component
    """
    # Step 1: Center the data
    X_centered = X - np.mean(X, axis=0)

    # Step 2: Compute covariance matrix
    n_samples = X.shape[0]
    cov_matrix = (X_centered.T @ X_centered) / (n_samples - 1)

    # Step 3: Eigendecomposition
    eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)

    # Step 4: Sort by eigenvalue (descending)
    sorted_idx = np.argsort(eigenvalues)[::-1]
    eigenvalues = eigenvalues[sorted_idx]
    eigenvectors = eigenvectors[:, sorted_idx]

    # Step 5: Select top k components
    if n_components is not None:
        eigenvectors = eigenvectors[:, :n_components]
        eigenvalues = eigenvalues[:n_components]

    # Step 6: Project data
    X_transformed = X_centered @ eigenvectors

    # Calculate explained variance ratio
    total_var = np.sum(eigenvalues)
    explained_variance_ratio = eigenvalues / total_var

    return X_transformed, eigenvectors, explained_variance_ratio


# Example usage
np.random.seed(42)
X = np.array([
    [2.5, 2.4],
    [0.5, 0.7],
    [2.2, 2.9],
    [1.9, 2.2],
    [3.1, 3.0]
])

X_pca, components, var_ratio = pca_from_scratch(X, n_components=1)
print("Transformed data:\n", X_pca)
```

```python
print("\nPrincipal components:\n", components)
print("\nVariance explained:", var_ratio)
```

14.2 Using Scikit-Learn

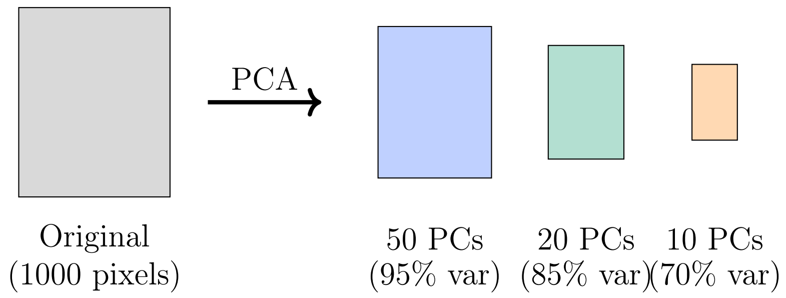
Listing 2: PCA with Scikit-Learn

```python
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load and prepare data
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target

# Standardize features (important for PCA!)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Results
print("Explained variance ratio:", pca.explained_variance_ratio_)
print("Cumulative variance:",
    np.cumsum(pca.explained_variance_ratio_))

# Visualize
plt.figure(figsize=(10, 6))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1],
                      c=y, cmap='viridis', alpha=0.7)
plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]:.1%}
    variance)')
plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]:.1%}
    variance)')
plt.title('Iris Dataset - PCA Visualization')
plt.colorbar(scatter, label='Species')
plt.show()
```

15 Applications of PCA

15.1 Image Compression (Eigenfaces)

PCA can compress images by keeping only the most important components:



15.2 Noise Reduction

By keeping only principal components with high eigenvalues, PCA filters out noise:

$$X_{\text{denoised}} = X_{\text{projected}} \cdot W^T + \bar{X}$$

15.3 Feature Engineering

PCA creates uncorrelated features, which can help many machine learning algorithms:

- Removes multicollinearity for regression
- Speeds up training by reducing dimensions
- Prevents overfitting in high-dimensional data

16 Limitations and Considerations

When NOT to Use PCA

1. **Non-linear relationships**: PCA only captures linear correlations. Use kernel PCA or t-SNE for non-linear data.
2. **Interpretability needed**: Principal components are linear combinations of all features, making interpretation difficult.
3. **Different scales**: Always standardize features first! Otherwise, high-variance features dominate.
4. **Categorical data**: PCA is designed for continuous numerical data.

17 Summary and Key Takeaways

Chapter Summary

Eigenvalues and Eigenvectors:

- $A\mathbf{v} = \lambda\mathbf{v}$ defines the eigenvalue problem
- Found by solving $\det(A - \lambda I) = 0$

- Reveal intrinsic properties of linear transformations
- Enable matrix decomposition: $A = PDP^{-1}$

Principal Component Analysis:

- Finds directions of maximum variance in data
- PCs are eigenvectors of the covariance matrix
- Eigenvalues indicate variance along each PC
- Used for dimensionality reduction, visualization, denoising

Workflow:

1. Center (and optionally standardize) data
2. Compute covariance matrix
3. Find eigenvalues/eigenvectors
4. Sort by eigenvalue magnitude
5. Keep top k components
6. Project data onto new basis

18 Practice Problems

1. Find the eigenvalues and eigenvectors of:

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

Verify that eigenvectors are orthogonal (since A is symmetric).

2. For the matrix $B = \begin{pmatrix} 2 & -1 \\ 1 & 4 \end{pmatrix}$:

   (a) Find eigenvalues
   (b) Find eigenvectors
   (c) Compute B^5 using diagonalization

3. Given the following dataset, perform PCA by hand:

$$X = \begin{pmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \\ 4 & 4 \end{pmatrix}$$

What percentage of variance is explained by PC1?

4. Prove that if λ is an eigenvalue of A, then λ^2 is an eigenvalue of A^2.
5. Explain why the covariance matrix is always positive semi-definite.

A Quick Reference: Formulas

Concept	**Formula**
Eigenvalue equation	$A\mathbf{v} = \lambda\mathbf{v}$
Characteristic polynomial	$p(\lambda) = \det(A - \lambda I)$
Trace	$\text{tr}(A) = \sum_i \lambda_i$
Determinant	$\det(A) = \prod_i \lambda_i$
Diagonalization	$A = PDP^{-1}$
Matrix power	$A^n = PD^nP^{-1}$
Covariance matrix	$\Sigma = \frac{1}{n-1}X^TX$
Variance explained	$\frac{\lambda_k}{\sum_i \lambda_i}$
PCA projection	$X_{\text{new}} = X_{\text{centered}} \cdot W$