PMAI LAB

Mathematics • Python • Artificial Intelligence

LINEAR ALGEBRA FOR MACHINE LEARNING

Lecture 07

Eigenvalues, Eigenvectors, Diagonalization, and Stable Transformation Directions

Lecture focus Identifying directions preserved by a transformation, decomposing matrices into simpler modes, and analysing repeated transformations, flow, stability, and data variation.

Dr. Hira Benish

Associate Professor of Mathematics

Student Learning Handout | Includes NumPy practice and review exercises

1 Lecture Overview

Most vectors change both direction and length when multiplied by a matrix. Eigenvectors are special directions whose orientation is preserved: a transformation only scales them. Eigenvalues record the corresponding scale factors.

These concepts reveal the natural modes of a matrix. They support dynamical-system analysis, PCA, spectral clustering, graph ranking, covariance analysis, and the efficient computation of repeated transformations.

Learning Outcomes

After completing this lecture, students should be able to:

- define eigenvalues, eigenvectors, and eigenspaces;
- calculate eigenvalues using the characteristic equation;
- calculate an eigenvector from a known eigenvalue;
- interpret eigenvalues as directional scaling factors;
- relate eigenvalues to trace, determinant, rank, and stability;
- determine whether a matrix is diagonalizable;
- apply the spectral theorem to a real symmetric matrix;
- analyse repeated transformations through matrix powers; and
- compute and verify eigendecompositions using NumPy.

1.1 Interpretation at a Glance

Concept	Meaning
Eigenvector	Direction preserved by the transformation
Eigenvalue	Scale applied along that direction
Eigenspace	All eigenvectors associated with one eigenvalue, plus zero
Diagonalization	Change to coordinates where the transformation acts independently
Dominant eigenvalue	Mode controlling long-term repeated behaviour

2 Eigenvalues and Eigenvectors

For a square matrix $A \in \mathbb{R}^{n \times n}$, a non-zero vector $\mathbf{v}$ is an **eigenvector** if

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

for some scalar λ. The scalar λ is the corresponding **eigenvalue**.

2.1 Geometric Meaning

- $\lambda > 1$: the eigenvector direction is stretched.
- $0 < \lambda < 1$: it is contracted.
- $\lambda < 0$: it is scaled and reversed.
- $\lambda = 0$: it is mapped to zero, so A is singular.
- $|\lambda| = 1$: its length is preserved along that direction.

Any non-zero multiple of an eigenvector is another eigenvector for the same eigenvalue.

2.2 Characteristic Equation

Rearrange the eigenvalue equation:

$$(A - \lambda I)\mathbf{v} = \mathbf{0}.$$

A non-zero solution exists only when $A - \lambda I$ is singular, so

$$\det(A - \lambda I) = 0.$$

This is the **characteristic equation**.

2.3 Eigenspace

For an eigenvalue λ, the eigenspace is

$$E_\lambda = \text{Null}(A - \lambda I).$$

Its non-zero vectors are exactly the eigenvectors associated with λ.

Machine-Learning Connection

Eigenvectors expose directions that a transformation treats independently. In covariance analysis, these directions identify patterns of variation; in graph methods, they identify structural modes of a network.

3 Worked Eigenvalue Example

Let

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

3.1 Step 1: Find the Eigenvalues

$$\det(A - \lambda I) = \det \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} = (2-\lambda)^2 - 1.$$

Therefore,

$$\lambda^2 - 4\lambda + 3 = 0 \quad \Rightarrow \quad (\lambda - 1)(\lambda - 3) = 0.$$

The eigenvalues are $\lambda_1 = 3$ and $\lambda_2 = 1$.

3.2 Step 2: Find the Eigenvectors

For $\lambda_1 = 3$,

$$(A - 3I)\mathbf{v} = \mathbf{0} \quad \Rightarrow \quad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \mathbf{0}.$$

Thus $v_1 = v_2$, so one eigenvector is

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

For $\lambda_2 = 1$, $v_1 = -v_2$, so one eigenvector is

$$\mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The normalized eigenvectors are

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Interpretation

The matrix scales the direction $[1, 1]^T$ by 3 and the perpendicular direction $[1, -1]^T$ by 1. Any input can be decomposed into these two eigenvector directions and transformed component by component.

4 Properties and Special Cases

For an $n \times n$ matrix, eigenvalues are counted with algebraic multiplicity.

4.1 Trace and Determinant

$$\text{tr}(A) = \sum_{i=1}^{n} \lambda_i, \qquad \det(A) = \prod_{i=1}^{n} \lambda_i.$$

Thus A is singular exactly when at least one eigenvalue is zero.

4.2 Triangular Matrices

The eigenvalues of a triangular matrix are its diagonal entries.

4.3 Complex Eigenvalues

A real matrix need not have real eigenvalues. For example, a 90° rotation has eigenvalues i and $-i$. It has no real direction that remains unchanged under the rotation.

4.4 Symmetric Matrices

If $A = A^T$ is real and symmetric, then:

- all eigenvalues are real;
- eigenvectors associated with distinct eigenvalues are orthogonal; and
- an orthonormal eigenvector basis exists.

4.5 Multiplicity

- **Algebraic multiplicity**: repetition of an eigenvalue in the characteristic polynomial.
- **Geometric multiplicity**: dimension of its eigenspace.

The geometric multiplicity is at least 1 and cannot exceed the algebraic multiplicity.

> Eigenvectors are not unique: their scale and sign may change. Numerical software may return $\mathbf{v}$ or $-\mathbf{v}$, and both are correct. Compare directions or verify $A\mathbf{v} \approx \lambda\mathbf{v}$.

Machine-Learning Connection

Covariance matrices are symmetric positive semidefinite. Their eigenvalues are non-negative and quantify variance along orthogonal eigenvector directions.

5 Diagonalization and the Spectral Theorem

A matrix A is **diagonalizable** when it has n linearly independent eigenvectors. Place them as columns of

$$P = \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{bmatrix},$$

and place the corresponding eigenvalues in

$$D = \text{diag}(\lambda_1, \ldots, \lambda_n).$$

Then

$$A = PDP^{-1}.$$

5.1 Meaning of the Decomposition

1. P^{-1} changes the vector into eigenvector coordinates.
2. D scales each coordinate independently.
3. P changes the result back to the original coordinates.

5.2 Efficient Matrix Powers

$$A^k = PD^kP^{-1}, \qquad D^k = \text{diag}(\lambda_1^k, \ldots, \lambda_n^k).$$

5.3 Spectral Theorem

Every real symmetric matrix has an orthogonal diagonalization

$$A = Q\Lambda Q^T,$$

where $Q^TQ = I$ and Λ contains real eigenvalues.

For the worked example,

$$Q = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}.$$

> Distinct eigenvalues guarantee independent eigenvectors and therefore diagonalizability. Repeated eigenvalues do not automatically prevent diagonalization; enough independent eigenvectors may still exist.

6 Repeated Transformations and Stability

Consider a discrete dynamical system

$$\mathbf{x}_{k+1} = A\mathbf{x}_k.$$

After k steps,

$$\mathbf{x}_k = A^k \mathbf{x}_0.$$

If A is diagonalizable, each eigenvector component is multiplied by λ_i^k.

6.1 Long-Term Behaviour

- $|\lambda_i| < 1$: that component decays.
- $|\lambda_i| > 1$: it grows.
- $\lambda_i < 0$: signs alternate while magnitude changes.
- $|\lambda_i| = 1$: it persists without magnitude decay.

The **spectral radius** is

$$\rho(A) = \max_i |\lambda_i|.$$

It often controls the long-term growth rate of repeated transformations.

6.2 Transportation-Flow Example

Suppose the state vector records people in two locations and

$$A = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}.$$

Each column describes how people in one location are distributed at the next step. The eigenvalues are 1 and 0.7.

The eigenvalue 1 represents a persistent equilibrium direction proportional to

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

The component associated with 0.7 decays, so the long-run distribution approaches two-thirds in the first location and one-third in the second.

Machine-Learning Connection

This same dominant-eigenvector principle appears in Markov chains, ranking algorithms, population models, iterative optimization analysis, and graph centrality.

7 Python and NumPy Practice

7.1 Eigenpairs of a Symmetric Matrix

```python
import numpy as np

A = np.array([[2.0, 1.0],
              [1.0, 2.0]])

eigenvalues, Q = np.linalg.eigh(A)

print("Eigenvalues:", eigenvalues)
print("Eigenvectors:\n", Q)
print("Q^T Q:\n", Q.T @ Q)
```

`np.linalg.eigh` is designed for real symmetric or complex Hermitian matrices. It returns real eigenvalues in ascending order and orthonormal eigenvectors.

7.2 Verifying the Eigendecomposition

```python
Lambda = np.diag(eigenvalues)

print("A Q = Q Lambda:",
      np.allclose(A @ Q, Q @ Lambda))
print("Reconstruction:",
      np.allclose(A, Q @ Lambda @ Q.T))
```

7.3 Computing a Matrix Power

```python
k = 10
A_power_spectral = Q @ np.diag(eigenvalues ** k) @ Q.T
A_power_direct = np.linalg.matrix_power(A, k)

print(np.allclose(A_power_spectral, A_power_direct))
```

> Use `np.linalg.eig` for a general square matrix and `np.linalg.eigh` for a symmetric matrix. In NumPy, eigenvectors are returned as columns.

8 Applications: Flow and Data Variation

8.1 Simulating Transportation Flow

```python
A_flow = np.array([[0.9, 0.2],
                   [0.1, 0.8]])

initial = np.array([100.0, 0.0])
after_20 = np.linalg.matrix_power(A_flow, 20) @ initial

values, vectors = np.linalg.eig(A_flow)
dominant_index = np.argmax(np.abs(values))
steady = np.real(vectors[:, dominant_index])
steady = steady / np.sum(steady)

print("State after 20 steps:", after_20)
print("Equilibrium proportions:", steady)
```

8.2 PCA Preview with a Covariance Matrix

For centered data X_c, the sample covariance matrix is

$$C = \frac{1}{m-1} X_c^T X_c.$$

```python
X = np.array([
    [2.0, 0.0], [3.0, 1.0], [4.0, 1.0],
    [5.0, 3.0], [6.0, 4.0]
])

X_centered = X - X.mean(axis=0)
C = X_centered.T @ X_centered / (len(X) - 1)

values, directions = np.linalg.eigh(C)
order = np.argsort(values)[::-1]
values = values[order]
directions = directions[:, order]

explained_ratio = values / values.sum()
scores = X_centered @ directions[:, 0]

print("Variance explained:", explained_ratio)
print("First-component scores:", scores)
```

Machine-Learning Connection

The leading covariance eigenvector is the direction of greatest sample variance. PCA keeps leading directions to create a lower-dimensional representation; the next lecture develops this connection through SVD.

9 Review and Independent Practice

Practice Tasks

1. Find the eigenvalues and eigenvectors of $\begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$.
2. Verify that trace equals the sum and determinant equals the product of the eigenvalues in Question 1.
3. Determine whether $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is diagonalizable.
4. Orthogonally diagonalize $\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$.
5. Use diagonalization to calculate the fifth power of a diagonalizable 2×2 matrix.
6. For a system $\mathbf{x}_{k+1} = A\mathbf{x}_k$, explain the roles of eigenvalues with magnitudes below, equal to, and above 1.
7. Compute the leading covariance eigenvector for a two-feature dataset and interpret its direction.

9.1 Key Takeaways

- Eigenvectors are directions preserved by a matrix transformation.
- Eigenvalues measure scaling along their eigenvector directions.
- Diagonalization expresses a transformation as independent directional scalings.
- Real symmetric matrices have real eigenvalues and an orthonormal eigenbasis.
- Matrix powers are controlled by powers of the eigenvalues.
- Dominant eigenvalues determine long-term modes in many iterative systems.
- Covariance eigenvectors identify directions of data variation.

9.2 Key Terms

Eigenvalue	Eigenvector	Eigenspace
Characteristic equation	Diagonalization	Spectral theorem
Algebraic multiplicity	Geometric multiplicity	Spectral radius
Dominant eigenvalue	Matrix power	Covariance matrix

Next lecture: Singular value decomposition, PCA, low-rank approximation, and data compression.