PMAI LAB

Mathematics • Python • Artificial Intelligence

LINEAR ALGEBRA FOR MACHINE LEARNING

Lecture 05

Inner Products, Orthogonality, Orthonormal Bases, and Vector Projections

Lecture focus Measuring vector geometry, constructing orthonormal directions, separating components, and projecting data onto useful subspaces.

Dr. Hira Benish

Associate Professor of Mathematics

Student Learning Handout | Includes NumPy practice and review exercises

1 Lecture Overview

Inner products add geometry to vector spaces. They allow us to measure length, angle, similarity, and perpendicularity. Orthogonal directions separate information cleanly, while projection identifies the component of a vector that lies inside a chosen subspace.

These concepts support similarity search, QR factorization, least-squares regression, signal decomposition, and dimensionality reduction. This lecture develops the mathematical ideas and applies them using Python and NumPy.

Learning Outcomes

After completing this lecture, students should be able to:

- define an inner product and state its main properties;
- obtain norms, distances, and angles from an inner product;
- determine whether vectors or subspaces are orthogonal;
- explain orthogonal complements and the Pythagorean relation;
- distinguish between orthogonal and orthonormal sets;
- construct an orthonormal basis using Gram-Schmidt;
- calculate vector and subspace projections;
- interpret projection matrices and residuals; and
- implement these operations using NumPy.

1.1 Conceptual Roadmap

Concept	Purpose
Inner product	Measures interaction and defines geometry
Norm	Measures vector length
Orthogonality	Identifies perpendicular, non-overlapping directions
Orthonormal basis	Gives independent unit directions and simple coordinates
Projection	Finds the closest component inside a subspace
Residual	Measures the component left outside the subspace

2 Inner Product Spaces

An **inner product** on a real vector space V assigns a scalar $\langle \mathbf{x}, \mathbf{y} \rangle$ to each pair of vectors. For all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$ and scalars a, b, it satisfies:

1. **Symmetry:** $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$;
2. **Linearity:** $\langle a\mathbf{x} + b\mathbf{y}, \mathbf{z} \rangle = a\langle \mathbf{x}, \mathbf{z} \rangle + b\langle \mathbf{y}, \mathbf{z} \rangle$; and
3. **Positive definiteness:** $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$, with equality only when $\mathbf{x} = \mathbf{0}$.

2.1 Standard Dot Product

In $\mathbb{R}^n$, the standard inner product is

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^{n} x_i y_i.$$

2.2 Weighted Inner Product

If M is symmetric positive definite, then

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{x}^T M \mathbf{y}$$

defines another inner product. It changes the geometry by assigning different importance to different directions.

For $M = \text{diag}(4, 1)$, movement in the first coordinate contributes four times as much to the squared norm as movement in the second coordinate.

2.3 Geometry Generated by an Inner Product

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}, \qquad d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

For non-zero vectors,

$$\cos\theta = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}.$$

Machine-Learning Connection

Similarity measures define the geometry used by a model. The standard dot product treats all coordinates equally, while a learned or weighted metric can emphasize directions that matter more for a particular task.

3 Orthogonality

Vectors $\mathbf{x}$ and $\mathbf{y}$ are **orthogonal** when

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0.$$

With the standard dot product, this means they meet at a right angle.

For example,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \qquad \mathbf{x}^T\mathbf{y} = 2 - 2 = 0.$$

3.1 Orthogonal Sets

A set of non-zero vectors is orthogonal when every distinct pair is orthogonal. Every orthogonal set of non-zero vectors is automatically linearly independent.

3.2 Pythagorean Relation

If $\mathbf{x} \perp \mathbf{y}$, then

$$\|\mathbf{x} + \mathbf{y}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2.$$

Orthogonal components contribute separately to the total squared length.

3.3 Orthogonal Complement

For a subspace $W \subseteq \mathbb{R}^n$,

$$W^\perp = \{\mathbf{x} : \mathbf{x}^T\mathbf{w} = 0 \text{ for every } \mathbf{w} \in W\}.$$

Every vector $\mathbf{x} \in \mathbb{R}^n$ can be decomposed uniquely as

$$\mathbf{x} = \mathbf{x}_W + \mathbf{x}_{W^\perp},$$

where $\mathbf{x}_W \in W$ and $\mathbf{x}_{W^\perp} \in W^\perp$.

Fundamental Subspace Relation

For $A \in \mathbb{R}^{m \times n}$,

$$\text{Null}(A) = \text{Row}(A)^\perp.$$

Directions in the null space are orthogonal to every row of A, which is why the transformation cannot detect them.

Machine-Learning Connection

Orthogonal features do not duplicate the same linear direction. This can simplify analysis, although statistical independence is stronger than orthogonality and should not be confused with it.

4 Orthonormal Bases

A set $\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ is **orthonormal** when

$$\mathbf{q}_i^T \mathbf{q}_j = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Thus the vectors are mutually orthogonal and each has length 1.

If the vectors are columns of Q, then

$$Q^T Q = I.$$

4.1 Coordinates Become Simple

If $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ is an orthonormal basis and

$$\mathbf{x} = c_1 \mathbf{q}_1 + \cdots + c_n \mathbf{q}_n,$$

then each coefficient is obtained directly:

$$c_i = \mathbf{q}_i^T \mathbf{x}.$$

Therefore,

$$\mathbf{x} = \sum_{i=1}^{n} (\mathbf{q}_i^T \mathbf{x}) \mathbf{q}_i.$$

4.2 Orthogonal Matrices

A square matrix Q with orthonormal columns satisfies

$$Q^T Q = Q Q^T = I, \qquad Q^{-1} = Q^T.$$

It preserves dot products, lengths, angles, and distances:

$$\|Q\mathbf{x}\| = \|\mathbf{x}\|.$$

Rotation and reflection matrices are orthogonal matrices.

Why Orthonormal Coordinates Are Stable

Orthonormal vectors neither repeat directions nor magnify small coordinate errors through an ill-conditioned basis. This makes orthogonal factorizations valuable in numerical linear algebra.

Machine-Learning Connection

Orthonormal directions appear in QR factorization, PCA, SVD, signal bases, and constrained neural-network methods. They provide clean, non-redundant coordinate directions.

5 Gram-Schmidt Orthogonalization

The Gram-Schmidt process converts an independent set $\mathbf{v}_1, \dots, \mathbf{v}_k$ into an orthonormal set $\mathbf{q}_1, \dots, \mathbf{q}_k$ spanning the same subspace.

5.1 Procedure

$$\mathbf{u}_1 = \mathbf{v}_1, \qquad \mathbf{q}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}.$$

For $j \geq 2$,

$$\mathbf{u}_j = \mathbf{v}_j - \sum_{i=1}^{j-1} (\mathbf{q}_i^T \mathbf{v}_j) \mathbf{q}_i, \qquad \mathbf{q}_j = \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|}.$$

Each projection onto an earlier direction is removed before normalization.

5.2 Worked Example

Let

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \qquad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

First,

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Next,

$$\mathbf{u}_2 = \mathbf{v}_2 - (\mathbf{q}_1^T \mathbf{v}_2) \mathbf{q}_1 = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{bmatrix}.$$

After normalization,

$$\mathbf{q}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}.$$

We can verify $\mathbf{q}_1^T \mathbf{q}_2 = 0$ and both norms equal 1.

> Classical Gram-Schmidt explains the geometry clearly. Numerical libraries often use modified Gram-Schmidt or Householder reflections because they are more stable for nearly dependent vectors.

6 Vector and Subspace Projections

The projection of $\mathbf{x}$ onto a non-zero vector $\mathbf{u}$ is

$$\text{proj}_{\mathbf{u}}(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}\mathbf{u}.$$

If $\mathbf{u}$ is a unit vector, this simplifies to $(\mathbf{x}^T\mathbf{u})\mathbf{u}$.

6.1 Worked Example

Let

$$\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Then

$$\text{proj}_{\mathbf{u}}(\mathbf{x}) = \frac{4}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

The residual is

$$\mathbf{r} = \mathbf{x} - \text{proj}_{\mathbf{u}}(\mathbf{x}) = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

and $\mathbf{r}^T\mathbf{u} = 0$.

6.2 Projection onto a Subspace

If the columns of Q form an orthonormal basis for W, then

$$\text{proj}_W(\mathbf{x}) = QQ^T\mathbf{x}.$$

The projection matrix is $P = QQ^T$ and satisfies

$$P^T = P, \qquad P^2 = P.$$

If a basis matrix A is not orthonormal but has independent columns, then

$$P = A(A^TA)^{-1}A^T.$$

Machine-Learning Connection

The projection is the point in W closest to $\mathbf{x}$ under Euclidean distance. This closest-point principle is the geometric foundation of least-squares regression.

7 Python and NumPy Practice

7.1 Inner Products and Orthogonality

```python
import numpy as np

x = np.array([1.0, 2.0])
y = np.array([2.0, -1.0])

inner_product = x @ y
orthogonal = np.isclose(inner_product, 0.0)

print("Inner product:", inner_product)
print("Orthogonal:", orthogonal)
```

7.2 Projection onto a Vector

```python
def project_onto_vector(x, u):
    denominator = u @ u
    if np.isclose(denominator, 0):
        raise ValueError("Projection direction is zero.")
    return ((x @ u) / denominator) * u

x = np.array([3.0, 1.0])
u = np.array([1.0, 1.0])

projection = project_onto_vector(x, u)
residual = x - projection

print("Projection:", projection)
print("Residual:", residual)
print("Residual orthogonal:",
      np.isclose(residual @ u, 0.0))
```

7.3 Checking an Orthonormal Matrix

```python
Q = np.array([
    [1 / np.sqrt(2),  1 / np.sqrt(2)],
    [1 / np.sqrt(2), -1 / np.sqrt(2)]
])

print(np.allclose(Q.T @ Q, np.eye(2)))
```

> Floating-point results are approximate. Use `np.isclose` for scalars and `np.allclose` for vectors or matrices rather than exact equality.

8 ML Case Study: Orthonormal Feature Directions

8.1 Gram-Schmidt in NumPy

```
def gram_schmidt(A, tol=1e-10):
    Q = []

    for j in range(A.shape[1]):
        u = A[:, j].astype(float).copy()
        for q in Q:
            u = u - (q @ u) * q

        length = np.linalg.norm(u)
        if length <= tol:
            raise ValueError("Columns are linearly dependent.")
        Q.append(u / length)

    return np.column_stack(Q)

A = np.array([
    [1.0, 1.0],
    [1.0, 0.0],
    [0.0, 1.0]
])

Q = gram_schmidt(A)
print("Q^T Q:\n", Q.T @ Q)
```

8.2 Projecting Row-Based Data

If $Q \in \mathbb{R}^{n \times k}$ contains orthonormal feature directions and observations are rows of X, then

$$X_{\text{projected}} = XQQ^T.$$

```
X = np.array([
    [2.0, 1.0, 0.0],
    [1.0, 2.0, 1.0],
    [0.0, 1.0, 3.0]
])

X_projected = X @ Q @ Q.T
residual = X - X_projected

print("Projected data:\n", X_projected)
print("Residual orthogonal to Q:",
      np.allclose(residual @ Q, 0.0))
```

Machine-Learning Connection

Projection separates each observation into a represented component and an orthogonal residual. Keeping only selected directions can compress data; the residual measures the information not represented by that subspace.

9 Review and Independent Practice

Practice Tasks

1. Calculate the dot product, angle, and distance between $[1, 2, 2]^T$ and $[2, 0, 1]^T$.
2. Determine whether $[1, 1, 0]^T$, $[1, -1, 0]^T$, and $[0, 0, 2]^T$ form an orthogonal set. Are they orthonormal?
3. Verify the Pythagorean relation for $[1, 2]^T$ and $[2, -1]^T$.
4. Apply Gram-Schmidt to $[1, 0, 1]^T$ and $[1, 1, 0]^T$.
5. Project $[4, 2]^T$ onto $[1, 1]^T$ and verify that the residual is orthogonal to the direction.
6. Construct $P = QQ^T$ for an orthonormal matrix Q and verify $P^2 = P$ and $P^T = P$ using NumPy.
7. Explain why projection loses information when the target subspace has lower dimension than the original space.

9.1 Key Takeaways

- An inner product defines length, angle, distance, and orthogonality.
- Orthogonal non-zero vectors are linearly independent.
- Orthonormal bases give unit directions and simple coordinate formulas.
- Gram-Schmidt removes existing directional components and then normalizes.
- A projection is the closest vector in a chosen subspace.
- The projection residual is orthogonal to the target subspace.
- Projection matrices are symmetric and idempotent.

9.2 Key Terms

Inner product	Orthogonality	Orthogonal complement
Orthonormal basis	Orthogonal matrix	Gram-Schmidt
Vector projection	Projection matrix	Residual
Symmetric matrix	Idempotent matrix	Closest point

Next lecture: Least-squares problems, normal equations, linear regression, and model fitting.