PMAI LAB

Mathematics • Python • Artificial Intelligence

LINEAR ALGEBRA FOR MACHINE LEARNING

Lecture 03

Vector Spaces, Subspaces, Linear Independence, Basis, and Dimension

Lecture focus Understanding valid vector spaces, detecting redundancy, constructing efficient bases, and interpreting rank and dimension in machine-learning feature spaces.

Dr. Hira Benish

Associate Professor of Mathematics

Student Learning Handout | Includes NumPy practice and review exercises

1 Lecture Overview

Machine-learning data often contains repeated, dependent, or unnecessary information. Vector-space concepts provide a precise language for identifying this structure. Linear independence tells us whether features contribute genuinely new directions, while a basis gives a compact set of directions sufficient to represent the entire space.

This lecture introduces vector spaces, subspaces, linear independence, basis, coordinates, dimension, rank, and nullity. These ideas are connected with feature redundancy, multicollinearity, learned representations, and dimension reduction.

Learning Outcomes

After completing this lecture, students should be able to:

- explain the defining properties of a vector space;
- determine whether a subset is a subspace;
- distinguish between linearly independent and dependent sets;
- test independence using a matrix and its rank;
- define a basis and calculate coordinates relative to a basis;
- explain dimension, rank, nullity, and the rank-nullity relation;
- identify redundant features in a data matrix; and
- implement these concepts using Python and NumPy.

1.1 Conceptual Roadmap

Concept	Central question
Vector space	Which objects and operations obey the linear-algebra rules?
Subspace	Which smaller set remains a vector space?
Independence	Does every vector add a genuinely new direction?
Basis	What is the smallest complete coordinate system?
Dimension	How many independent directions does the space contain?
Rank	How many independent directions are carried by a matrix?
Nullity	How many input directions are mapped to zero?

2 Vector Spaces

A **vector space** V over $\mathbb{R}$ is a set equipped with vector addition and scalar multiplication. These operations must behave consistently with ordinary algebra.

For all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and $a, b \in \mathbb{R}$:

1. $\mathbf{u} + \mathbf{v} \in V$ and $a\mathbf{u} \in V$ (closure);
2. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ and $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$;
3. a zero vector $\mathbf{0} \in V$ exists and $\mathbf{v} + \mathbf{0} = \mathbf{v}$;
4. every $\mathbf{v}$ has an additive inverse $-\mathbf{v}$;
5. $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$;
6. $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$;
7. $a(b\mathbf{v}) = (ab)\mathbf{v}$ and $1\mathbf{v} = \mathbf{v}$.

2.1 Common Examples

$\mathbb{R}^n$	All real vectors having n entries
P_n	All real polynomials of degree at most n
$M_{m\times n}$	All real $m \times n$ matrices
Function spaces	Functions satisfying specified linear conditions
Solution of $A\mathbf{x} = \mathbf{0}$	The null space of a matrix

2.2 Non-Examples

The positive quadrant $\{(x, y) : x \geq 0, y \geq 0\}$ is not a vector space because multiplying a non-zero vector by -1 leaves the set. Similarly, $\{(x, y) : x + y = 1\}$ is not a vector space because it does not contain the zero vector.

Machine-Learning Connection

Feature vectors usually live in $\mathbb{R}^n$. Embeddings, parameter vectors, gradients, and activation vectors are therefore elements of vector spaces, which allows ML algorithms to add, scale, compare, and transform them consistently.

3 Subspaces

A subset W of a vector space V is a **subspace** if W is itself a vector space under the same operations.

3.1 Subspace Test

A non-empty subset $W \subseteq V$ is a subspace if:

1. $\mathbf{0} \in W$;
2. $\mathbf{u}, \mathbf{v} \in W \Rightarrow \mathbf{u} + \mathbf{v} \in W$; and
3. $c \in \mathbb{R}$ and $\mathbf{u} \in W \Rightarrow c\mathbf{u} \in W$.

Important Subspaces Associated with a Matrix

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

- the **column space** is the span of the columns of A and lies in $\mathbb{R}^m$;
- the **row space** is the span of the rows of A and lies in $\mathbb{R}^n$; and
- the **null space** is $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}$.

3.2 Worked Example

Consider

$$W = \{(x, y, z) \in \mathbb{R}^3 : x + 2y - z = 0\}.$$

The zero vector satisfies the equation. The defining equation is homogeneous and is preserved by addition and scalar multiplication, so W is a subspace.

Let $y = s$ and $z = t$. Then $x = -2s + t$, giving

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = s \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Therefore,

$$W = \text{span}\left\{ \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\}.$$

Solution sets of homogeneous systems $A\mathbf{x} = \mathbf{0}$ are always subspaces. Solution sets of non-homogeneous systems $A\mathbf{x} = \mathbf{b}$ with $\mathbf{b} \neq \mathbf{0}$ are generally not subspaces because they do not contain the zero vector.

4 Linear Independence

Vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ are **linearly independent** if

$$c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k = \mathbf{0}$$

has only the trivial solution $c_1 = \cdots = c_k = 0$.

They are **linearly dependent** if a non-trivial solution exists. In that case, at least one vector can be expressed as a linear combination of the others.

4.1 Simple Examples

1. $[1, 0]^T$ and $[0, 1]^T$ are independent because neither repeats the direction of the other.
2. $[1, 2]^T$ and $[2, 4]^T$ are dependent because the second is twice the first.
3. Any set containing the zero vector is dependent.
4. More than n vectors in $\mathbb{R}^n$ must be dependent.

4.2 Matrix Rank Test

Place the vectors as columns of

$$A = \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_k \\ | & & | \end{bmatrix}.$$

The vectors are independent exactly when

$$\text{rank}(A) = k.$$

4.3 Polynomial Example

The polynomials $1 + x$, $x + x^2$, and $1 + 2x + x^2$ are dependent because

$$(1 + x) + (x + x^2) - (1 + 2x + x^2) = 0.$$

Machine-Learning Connection

Dependent feature columns carry redundant information. Perfect dependence causes rank deficiency and can make model parameters non-unique. Near dependence, called multicollinearity, can make estimates unstable even when the matrix is technically full rank.

5 Basis and Coordinates

A set $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ is a **basis** for a vector space V when:

1. the vectors in B are linearly independent; and
2. B spans V.

A basis is therefore a complete representation with no redundant directions.

5.1 Standard Basis

The standard basis of $\mathbb{R}^3$ is

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Every $[x, y, z]^T$ has the unique representation

$$x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3.$$

5.2 Coordinates Relative to a Basis

Let

$$B = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}, \qquad \mathbf{x} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

We solve

$$c_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

This gives $c_1 = 3$ and $c_2 = 1$, so the coordinate vector of $\mathbf{x}$ relative to B is

$$[\mathbf{x}]_B = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Why Independence Is Essential

If the basis vectors were dependent, a vector could have multiple coordinate descriptions. Independence guarantees uniqueness; spanning guarantees that every vector in the space can be represented.

Machine-Learning Connection

Changing a basis changes coordinates without changing the underlying object. PCA later constructs a new orthogonal basis aligned with directions of greatest variation in the data.

6 Dimension, Rank, and Nullity

The **dimension** of a finite-dimensional vector space is the number of vectors in any basis of that space. Different bases may contain different vectors, but they always contain the same number of vectors.

$$\text{dim}(\mathbb{R}^n) = n, \qquad \text{dim}(P_n) = n + 1.$$

6.1 Rank

For a matrix A,

$$\text{rank}(A) = \text{dim}(\text{Col}(A)) = \text{dim}(\text{Row}(A)).$$

Rank is the number of independent columns, equivalently the number of independent rows. A matrix has **full column rank** when every column adds an independent direction.

6.2 Nullity and Rank-Nullity

The **nullity** of A is the dimension of its null space. If A has n columns, then

$$\text{rank}(A) + \text{nullity}(A) = n.$$

If A has four columns and rank 2, then its nullity is 2. This means two independent input directions are mapped to zero.

6.3 Data Dimension versus Intrinsic Dimension

A dataset may have 100 recorded features but lie close to a much lower-dimensional structure. The number of columns is its **ambient dimension**; the number of meaningful independent directions is related to its **intrinsic dimension**.

Rank Is Not Always Exact in Real Data

Perfect dependence produces exact rank deficiency. Real datasets more often contain approximate dependence, so singular values and numerical tolerances are used to decide which directions are practically significant.

Machine-Learning Connection

Dimensionality-reduction methods seek a smaller set of informative directions. This reduces storage and computation, supports visualization, and may reduce noise or multicollinearity.

7 Python and NumPy Practice

7.1 Testing Linear Independence

```python
import numpy as np

def are_independent(vectors, tol=None):
    A = np.column_stack(vectors)
    rank = np.linalg.matrix_rank(A, tol=tol)
    return rank == len(vectors)

v1 = np.array([1.0, 2.0])
v2 = np.array([2.0, 4.0])
v3 = np.array([0.0, 1.0])

print(are_independent([v1, v2]))  # False
print(are_independent([v1, v3]))  # True
```

7.2 Coordinates Relative to a Basis

```python
B = np.array([
    [1.0,  1.0],
    [1.0, -1.0]
])

x = np.array([4.0, 2.0])
coordinates = np.linalg.solve(B, x)

print("Coordinates:", coordinates)
print("Reconstruction:", B @ coordinates)
```

Expected coordinates are `[3. 1.]`, and the reconstruction equals the original vector.

7.3 Rank and Nullity

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

rank = np.linalg.matrix_rank(A)
nullity = A.shape[1] - rank

print("Rank:", rank)
print("Nullity:", nullity)
```

> `np.linalg.matrix_rank` uses singular values and a numerical tolerance. It is suitable for floating-point data, but a nearly dependent matrix may be classified differently under different tolerances.

8 ML Case Study: Redundant Features

Suppose a dataset records normalized attendance, quiz performance, and a third feature created by adding the first two. The third column contains no new information.

```python
attendance = np.array([0.80, 0.60, 0.90, 0.70])
quiz = np.array([0.70, 0.85, 0.65, 0.75])
combined = attendance + quiz

X = np.column_stack((attendance, quiz, combined))

print("Data matrix:\n", X)
print("Number of features:", X.shape[1])
print("Rank:", np.linalg.matrix_rank(X))
```

The matrix has three columns but rank 2 because

$$\mathbf{x}_3 = \mathbf{x}_1 + \mathbf{x}_2.$$

8.1 Selecting Independent Columns

```python
def independent_column_indices(A, tol=None):
    selected = []
    current = np.empty((A.shape[0], 0))
    current_rank = 0

    for j in range(A.shape[1]):
        candidate = np.column_stack((current, A[:, j]))
        candidate_rank = np.linalg.matrix_rank(candidate, tol=tol)
        if candidate_rank > current_rank:
            selected.append(j)
            current = candidate
            current_rank = candidate_rank

    return selected

print("Independent column indices:",
      independent_column_indices(X))
```

Machine-Learning Connection

Removing an exactly redundant feature does not remove information. However, feature selection in real projects should consider prediction value, interpretability, noise, fairness, and domain meaning, not rank alone.

Dummy-variable trap: if one-hot encoding keeps every category together with an intercept, the encoded columns may become linearly dependent. Many statistical workflows drop one reference category to avoid this issue.

9 Review and Independent Practice

Practice Tasks

1. Determine whether $W = \{(x, y, z) : x - y + z = 0\}$ is a subspace of $\mathbb{R}^3$. Find a spanning set for it.
2. Determine whether $\{[1, 2]^T, [2, 1]^T\}$ is linearly independent using both an equation and matrix rank.
3. Explain why $\{[1, 0]^T, [0, 1]^T, [1, 1]^T\}$ spans $\mathbb{R}^2$ but is not a basis.
4. Find the coordinates of $[5, 1]^T$ relative to $B = \{[1, 1]^T, [1, -1]^T\}$.
5. Decide whether $1 + x$, $x + x^2$, and $1 + 2x + x^2$ are independent in P_2.
6. Create a 5×4 NumPy matrix whose fourth column is the sum of the first two. Calculate its rank and nullity.
7. Explain the difference between the recorded feature dimension and intrinsic dimension of a dataset.

9.1 Key Takeaways

- A vector space is closed under vector addition and scalar multiplication and satisfies the linear-algebra rules.
- A subspace contains the zero vector and remains closed under both operations.
- Independent vectors contribute distinct directions; dependent vectors contain redundancy.
- A basis is an independent spanning set and provides unique coordinates.
- Dimension counts basis directions, rank counts independent matrix directions, and nullity counts lost input directions.
- Feature matrices with dependent columns are rank deficient and may produce non-unique model parameters.

9.2 Key Terms

Vector space	Subspace	Closure
Linear independence	Linear dependence	Basis
Coordinates	Dimension	Rank
Nullity	Column space	Intrinsic dimension

Next lecture: Matrices as linear transformations, composition, inverse transformations, and geometric effects.