Linear Transformations, Matrices, and Composition

Research by: Jose Portillo

$$\Big\langle \langle K, +_g, \cdot \rangle,\ \langle V, +_v \rangle,\ \circ \Big\rangle$$

The system above is called a *vector space* over the field K [2]. Elements of K are called *scalars*, and elements of V are called *vectors*. The operation $\circ : K \times V \to V$ is scalar multiplication, and $+_v$ is vector addition. In what follows, symbols $\alpha_1, \ldots, \alpha_k$ always denote scalars in K, and symbols $x_1, \ldots, x_k$ always denote vectors in V.

Informally, a vector space is a set of vectors that can be added together and scaled by elements of a field.

Example (Coordinate n-space, [3]).
Let Γ be a field. Consider the set

$$\Gamma^n = \Gamma \times \cdots \times \Gamma$$

of n-tuples $(\xi^1, \ldots, \xi^n)$, with $\xi^i \in \Gamma$. Define addition by

$$(\xi^1, \ldots, \xi^n) + (\eta^1, \ldots, \eta^n) = (\xi^1 + \eta^1, \ldots, \xi^n + \eta^n)$$

and scalar multiplication by

$$\lambda(\xi^1, \ldots, \xi^n) = (\lambda\xi^1, \ldots, \lambda\xi^n), \quad \lambda \in \Gamma.$$

With these operations, Γ^n is a vector space over Γ, called the *n-space over* Γ. In particular, Γ itself is a vector space over Γ, where scalar multiplication coincides with field multiplication.

A finite set of vectors $\{b_1, \ldots, b_n\} \subset V$ is called a *basis* of the vector space V if every vector $x \in V$ can be written uniquely as a linear combination

$$x = \alpha_1 \circ b_1 + \alpha_2 \circ b_2 + \cdots + \alpha_n \circ b_n, \quad \alpha_1, \ldots, \alpha_n \in K.$$

A basis therefore provides a minimal and non-redundant set of building blocks for the entire space. Once a basis is fixed, every vector is completely determined by its coefficients relative to that basis.

Given vectors $x_1, \ldots, x_k \in V$ and scalars $\alpha_1, \ldots, \alpha_k \in K$, an expression of the form

$$\alpha_1 \circ x_1 + \alpha_2 \circ x_2 + \cdots + \alpha_k \circ x_k$$

is called a *linear combination*.

Let $\Big\langle \langle K, +_g, \cdot \rangle,\ \langle E, +_E \rangle,\ \circ_E \Big\rangle$ and $\Big\langle \langle K, +_g, \cdot \rangle,\ \langle G, +_G \rangle,\ \circ_G \Big\rangle$ be vector spaces over the same field K. A function $\psi : E \to G$ is called a *linear transformation* if it preserves linear combinations [1]:

$$\psi(\alpha_1 \circ_E x_1 + \cdots + \alpha_k \circ_E x_k) = \alpha_1 \circ_G \psi(x_1) + \cdots + \alpha_k \circ_G \psi(x_k).$$

Linear transformations preserve the algebraic structure of vector spaces independently of any choice of coordinates.

Let $B_E = \{e_1, \ldots, e_n\}$ and $B_G = \{g_1, \ldots, g_m\}$ be bases of E and G. Each image $\psi(e_i)$ can be written uniquely as $\psi(e_i) = a_{1i}g_1 + \cdots + a_{mi}g_m$. The scalars a_{ji} form an $m \times n$ matrix

$$A = (a_{ji}),$$

called the *matrix of ψ relative to the chosen bases.*

Matrices thus do not define linear transformations; rather, they are concrete representations of linear transformations once bases (coordinate systems) have been chosen.

In many applications, a vector does not merely represent an abstract element of a vector space, but the *state* of a system relative to a chosen reference system. Algebraically, a reference system is nothing more than a choice of basis.

Let E be a finite-dimensional vector space and let

$$B = \{e_1, \ldots, e_n\}$$

be a basis of E. Any vector $x \in E$ can be written uniquely as

$$x = \alpha_1 e_1 + \cdots + \alpha_n e_n.$$

The scalars $(\alpha_1, \ldots, \alpha_n)$ are the *coordinates* of the state x relative to the reference system B. Writing these coordinates as a column vector,

$$[x]_B = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix},$$

is a notational convenience: it encodes the action of linear combinations relative to the chosen basis.

Suppose now that a second reference system

$$B' = \{e'_1, \ldots, e'_n\}$$

is chosen for the same space E. The vector x is the same abstract object, but its coordinates relative to B' form a different column vector $[x]_{B'}$.

The passage from $[x]_B$ to $[x]_{B'}$ is not arbitrary. It is determined by a linear transformation

$$T : K^n \longrightarrow K^n,$$

whose matrix depends only on the relation between the two bases. Thus,

$$[x]_{B'} = A\,[x]_B,$$

where A is the matrix of the change of reference system.

This is the operational origin of matrix–vector multiplication: a matrix acts on a column vector because it represents a linear transformation acting on the coordinate representation of a state.

The requirement that the number of columns of A equal the number of components of $[x]_B$ is not a convention; it expresses the fact that the codomain of one linear map must match the domain of the next.

In this sense, column vectors represent states, matrices represent transformations between reference systems, and matrix multiplication represents successive changes of coordinates. The familiar computational rules are therefore consequences of the abstract structure introduced earlier, not independent algebraic tricks.

This abstract formulation is independent of geometry, yet it underlies most quantitative models in science and engineering. In modern machine learning, for example, embeddings and weight matrices are linear transformations on high-dimensional vector spaces, composed with non-linear activations. Even where non-linearity dominates, linear transformations provide the structural backbone—the universality of a man-made linear structure, even though the universe itself is not linear.

The set of all linear transformations from E into the base field K is denoted by

$$E^* = L(E, K)$$

and is called the *dual space* of E. An element $x^* \in E^*$ is called a *linear functional*. Thus,

$$x^* : E \to K$$

satisfies

$$x^*(\alpha_1 x_1 + \cdots + \alpha_k x_k) = \alpha_1 x^*(x_1) + \cdots + \alpha_k x^*(x_k).$$

Linear functionals assign scalar values to vectors while preserving linear structure.

If E is finite-dimensional with basis $B_E = \{e_1, \ldots, e_n\}$, then every $x^* \in E^*$ is uniquely determined by

$$x^*(e_1), \ldots, x^*(e_n).$$

These scalars form a row vector

$$[\, x^*(e_1) \ \ldots \ x^*(e_n) \,].$$

In contrast, vectors in E are represented as column vectors. These two objects live in different spaces: E and E^*.

Column vectors represent elements of a vector space E, whereas row vectors represent linear functionals, elements of the dual space E^*. These are algebraically different objects, even though they may have the same number of components.

In many situations, however, a vector is used to define a linear functional. This requires additional structure. If E is equipped with an inner product $\langle \cdot, \cdot \rangle$, then each vector $x \in E$ determines a linear functional

$$x^*(y) = \langle x, y \rangle.$$

Thus the inner product provides a mapping

$$E \longrightarrow E^*, \quad x \longmapsto x^*.$$

When vectors are written in coordinates as column vectors, the corresponding linear functional must be written as a row vector. The operation that performs this conversion in coordinates is the *transpose*.

Example.
Let

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in K^2.$$

With the standard inner product on K^2 (e.g. $K = \mathbb{R}$),

$$\langle x, y \rangle = x_1 y_1 + x_2 y_2,$$

the associated linear functional is represented by the row vector

$$x^T = [\, x_1 \ \ x_2 \,].$$

Its action on a vector $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ is given by

$$x^T y = \langle x, y \rangle.$$

The transpose operation therefore exists to convert vectors into linear functionals when an inner product is present. It is not intrinsic to vector spaces themselves, but arises from additional geometric structure imposed on them.

Matrix multiplication is not an arbitrary algebraic rule; it is the coordinate expression of *composition of linear maps*.

Note: Linear transformations are functions that are linear in a single argument. More generally, one can consider functions that are linear in several arguments, called multilinear maps. *A multilinear map*

$$T : E_1 \times \cdots \times E_k \longrightarrow G$$

is linear in each argument separately. In particular, functions of the form

$$T : E \times \cdots \times E \longrightarrow K$$

generalize linear functionals and bilinear forms. A tensor *of type* (k, ℓ) *on* E *is a multilinear map*

$$T : \underbrace{E^* \times \cdots \times E^*}_{k} \times \underbrace{E \times \cdots \times E}_{\ell} \longrightarrow K.$$

Special cases include scalars $(0,0)$, *vectors* $(1,0)$, *linear functionals* $(0,1)$, *linear transformations* $(1,1)$, *and bilinear forms* $(0,2)$. *Once a basis of* E *is chosen, a tensor can be represented by a multidimensional array of scalars; these arrays depend on the chosen basis, but the tensor itself does not. In numerical computation and machine learning, such arrays are commonly called* tensors, *although they represent the coordinate expressions of the abstract objects. NumPy and TensorFlow tensors are multidimensional arrays that represent tensors after a choice of basis; the tensor itself is a coordinate-free object.*

A column vector represents an element of K^n, a row vector represents an element of $(K^n)^*$, and an $m \times n$ matrix represents a linear map

$$K^n \xrightarrow{\psi} K^m.$$

A product such as

$$[\, x^* \,]\, A$$

is defined because it corresponds to the composition

$$K^n \xrightarrow{\psi} K^m \xrightarrow{x^*} K$$

The dimensions match precisely because the codomain of the first map equals the domain of the second.

By contrast, a product of two row vectors would attempt to compose

$$K^n \to K \quad \text{with} \quad K^n \to K,$$

which is meaningless: the output of the first map does not lie in the domain of the second. The product is therefore undefined not by convention, but by necessity.

Matrix dimensions encode compatibility of domains and codomains. One may multiply matrices if and only if the corresponding linear maps can be composed.

Concrete numerical example.

Let

$$x^*(x, y) = 3x + 2y$$

be a linear functional on K^2. Relative to the canonical basis, it is represented by the row vector

$$[\,3\;\,2\,].$$

Let

$$A = \begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix}$$

be the matrix of a linear transformation $\psi : K^2 \to K^2$.

The product

$$[\,3\;\,2\,]A = [\,3\;\,2\,] \begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix} = [\,19\;\,17\,]$$

is defined because it represents the composition

$$K^2 \xrightarrow{\ \psi\ } K^2 \xrightarrow{\ x^*\ } K.$$

The result is again a row vector, corresponding to the linear functional

$$(x, y) \longmapsto 19x + 17y.$$

By contrast, the product

$$[\,3\;\,2\,][\,a\;\,b\,]$$

is undefined because it would attempt to compose two maps of the form

$$K^2 \to K,$$

which is impossible: the output of the first map is a scalar, not a vector in K^2.

Note. An AI language model was used as a support tool for discussion, clarification of concepts, and stylistic refinement. All mathematical choices and final wording are the responsibility of the author.

References

[1] Iribarren Ignacio L., *Álgebra Lineal*, Editorial Equinoccio, 2018.

[2] Cohen, Leon and Ehrlich, Gertrude, *The Structure of the Real Number System*, D. Van Nostrand Company, 1963.

[3] Greub, Werner, *Linear Algebra*, Springer-Verlag, 1975.