Chapter 3

Fundamentals of Multilinear Subspace Learning

The previous chapter covered background materials on linear subspace learning. From this chapter on, we shall proceed to multiple dimensions with tensor-level computational thinking. Multilinear algebra is the foundation of multilinear subspace learning (MSL). Thus, we first review the basic notations and operations in multilinear algebra, as well as popular tensor decompositions. In the presentation, we include some discussions of the second-order case (for matrix data) as well, which can be understood in the context of linear algebra. Next, we introduce the important concept of *multilinear projections* for direct mapping of tensors to a lower-dimensional representation, as shown in Figure 3.1. They include elementary multilinear projection (EMP), tensor-to-vector projection (TVP), and tensor-to-tensor projection (TTP), which project an input tensor to a scalar, a vector, and a tensor, respectively. Their relationships are analyzed in detail subsequently. Finally, we extend commonly used vector-based scatter measures to tensors and scalars for optimality criterion construction in MSL.

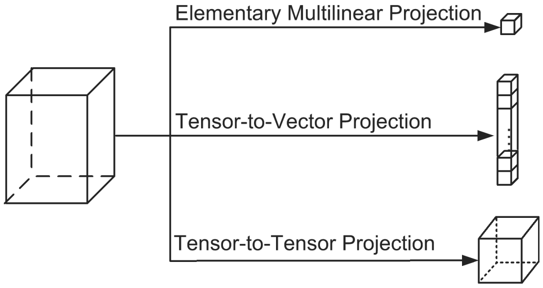

Elementary Multilinear Projection
Tensor-to-Vector Projection
Tensor-to-Tensor Projection

FIGURE 3.1: Multilinear subspace learning finds a lower-dimensional representation by direct mapping of tensors through a multilinear projection.

3.1 Multilinear Algebra Preliminaries

Multilinear algebra, the basis of tensor-based computing, has been studied in mathematics for several decades [Greub, 1967]. A *tensor* is a multidimensional (multiway) array. As pointed out by Kolda and Bader [2009], this notion of tensors is different from the same term referring to *tensor fields* in physics and engineering. In the following, we review the notations and some basic multilinear operations needed in introducing MSL.

3.1.1 Notations and Definitions

In this book, we have tried to remain consistent with the notations and terminologies in applied mathematics, particularly the seminal paper by De Lathauwer et al. [2000b] and the recent SIAM review paper on tensor decomposition by Kolda and Bader [2009].

Vectors are denoted by lowercase boldface letters, for example, **a**; matrices by uppercase boldface, for example, **A**; and tensors by calligraphic letters, for example, $\mathcal{A}$. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index whenever appropriate, for example, $n = 1, 2, ..., N$. Throughout this book, we restrict the discussions to real-valued vectors, matrices, and tensors.

Definition 3.1. *The number of dimensions (ways) of a tensor is its* ***order****, denoted by N. Each dimension (way) is called a* ***mode****.*

As shown in Figure 3.2, a scalar is a zero-order tensor ($N = 0$), a vector is a first-order tensor ($N = 1$), and a matrix is a second-order tensor ($N = 2$). Tensors of order three or higher are called *higher-order tensors*. An Nth-order tensor is an element in a *tensor space* of degree N, the *tensor product* (*outer product*) of N vector spaces (Section A.1.4) [Lang, 1984].

Mode addressing: There are two popular ways to refer to a *mode*: n-mode or mode-n. In [De Lathauwer et al., 2000b], only n-mode is used. In [De Lathauwer et al., 2000a], n-mode and mode-n are used interchangeably. In [Kolda and Bader, 2009], n-mode and mode-n are used in different contexts. In this book, we prefer to use mode-n to indicate the nth mode for clarity.

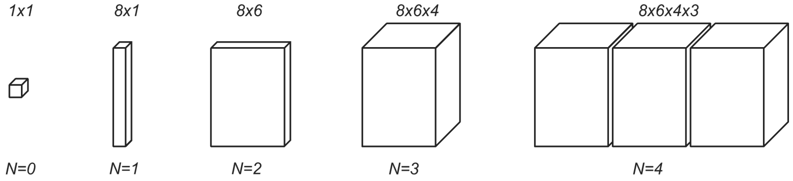


FIGURE 3.2: Illustration of tensors of order $N = 0, 1, 2, 3, 4$.

For example, “1-mode” (referring to the mode index) may be confused with one mode (referring to the number of modes).

An Nth-order tensor has N indices $\{i_n\}$, $n = 1, ..., N$, with each index $i_n (= 1, ..., I_n)$ addressing mode-n of $\mathcal{A}$. Thus, we denote an Nth-order tensor explicitly as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$.

When we have a set of N vectors or matrices, one for each mode, we denote the nth (i.e., mode-n) vector or matrix using a superscript in parenthesis, for example, as $\mathbf{u}^{(n)}$ or $\mathbf{U}^{(n)}$ and the whole set as $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, ..., \mathbf{u}^{(N)}\}$ or $\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, ..., \mathbf{U}^{(N)}\}$, or more compactly as $\{\mathbf{u}^{(n)}\}$ or $\{\mathbf{U}^{(n)}\}$. Figures 3.5, 3.9, and 3.10 provide some illustrations.

Element addressing: For clarity, we adopt the MATLAB style to address elements (entries) of a tensor (including vector and matrix) with indices in parentheses. for example, a single element (entry) is denoted as $\mathbf{a}(2)$, $\mathbf{A}(3, 4)$, or $\mathcal{A}(5, 6, 7)$ in this book, which are denoted as a_2, a_{34}, or a_{567} in conventional notations[1]. To address part of a tensor (a subarray), “:” denotes the full range of the corresponding index and $i : j$ denotes indices ranging from i to j, for example, $\mathbf{a}(2 : 5)$, $\mathbf{A}(3, :)$ (the third row), or $\mathcal{A}(1 : 3, :, 4 : 5)$.

Definition 3.2. *The **mode-n vectors**[2] of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n while keeping all the other indices fixed.*

For example, $\mathcal{A}(:, 2, 3)$ is a mode-1 vector. For second-order tensors (matrices), mode-1 and mode-2 vectors are the column and row vectors, respectively. Figures 3.3(b), 3.3(c), and 3.3(d) give visual illustrations of the mode-1, mode-2, and mode-3 vectors of the third-order tensor $\mathcal{A}$ in Figure 3.3(a), respectively.

Definition 3.3. *The i_nth **mode-n slice** of $\mathcal{A}$ is defined as an $(N-1)$th-order tensor obtained by fixing the mode-n index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, ..., :, i_n, :, ..., :)$.*

For example, $\mathcal{A}(:, 2, :)$ is a mode-2 slice of $\mathcal{A}$. For second-order tensors (matrices), a mode-1 slice is a mode-2 (row) vector, and a mode-2 slice is a mode-1 (column) vector. Figures 3.4(b), 3.4(c), and 3.4(d) give visual illustrations of the mode-1, mode-2, and mode-3 slices of the third-order tensor $\mathcal{A}$ in Figure 3.4(a), respectively. This definition of a slice is consistent with that in [Bader and Kolda, 2006]; however, it is different from the definition of a slice in [Kolda and Bader, 2009], where a slice is defined as a two-dimensional section of a tensor.

Definition 3.4. *A **rank-one tensor** $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ equals the outer product[3] of N vectors:*

$$\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}, \tag{3.1}$$

[1] In Appendix A, we follow the conventional notations of an element.

[2] Mode-n vectors are renamed as mode-n *fibers* in [Kolda and Bader, 2009].

[3] Here, we use a notation '$\circ$' different from the conventional notation '$\otimes$' to better differentiate the outer product of vectors from the Kronecker product of matrices.

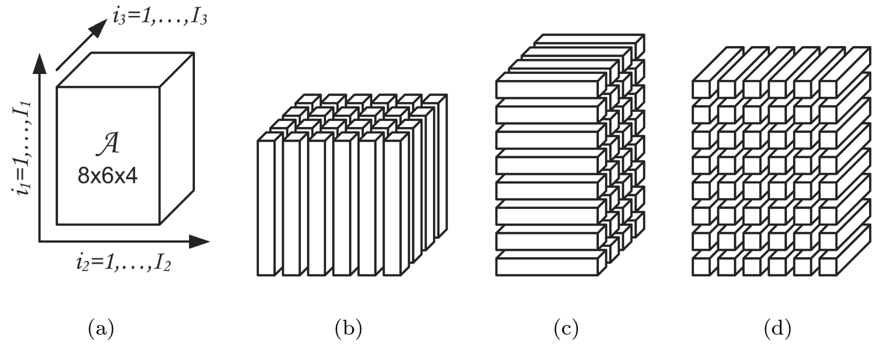


FIGURE 3.3: Illustration of the mode-n vectors: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$, (b) the mode-1 vectors, (c) the mode-2 vectors, and (d) the mode-3 vectors.

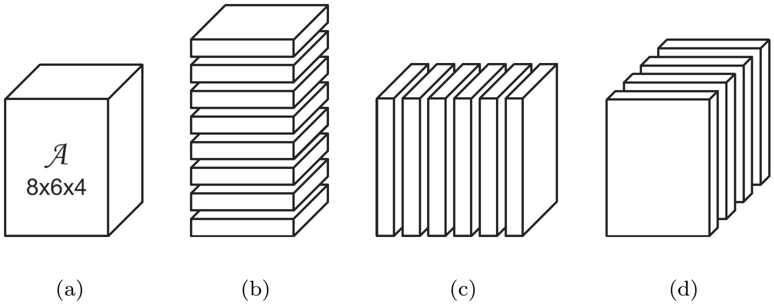


FIGURE 3.4: Illustration of the mode-n slices: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$, (b) the mode-1 slices, (c) the mode-2 slices, and (d) the mode-3 slices.

which means that

$$\mathcal{A}(i_1, i_2, ..., i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot ... \cdot \mathbf{u}^{(N)}(i_N) \tag{3.2}$$

for all values of indices.

A rank-one tensor for $N = 2$ (i.e., a second-order rank-one tensor) is a rank-one matrix, with an example shown in Figure 3.5. Some examples of rank-one tensors for $N = 2, 3$ learned from real data are shown in Figures 1.10(c) and 1.11(c) in Chapter 1.

Definition 3.5. *A **cubical tensor** $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times...\times I_N}$ has the same size for every mode, that is, $I_n = I$ for $n = 1, ..., N$ [Kolda and Bader, 2009].*

Thus, a square matrix is a second-order cubical tensor by this definition.

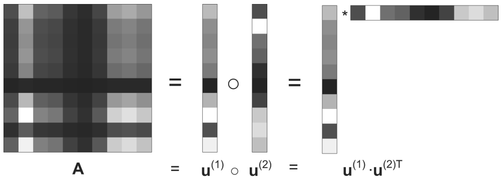


FIGURE 3.5: An example of second-order rank-one tensor (that is, rank-one matrix): $\mathbf{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} = \mathbf{u}^{(1)}\mathbf{u}^{(2)^T}$.

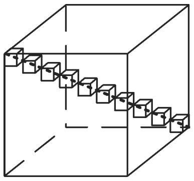


FIGURE 3.6: The diagonal of a third-order cubical tensor.

Definition 3.6. *A* ***diagonal tensor*** $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ *has non-zero entries, i.e.,* $\mathcal{A}(i_1, i_2, ..., i_N) \neq 0$*, only for* $i_1 = i_2 = ... = i_N$ *[Kolda and Bader, 2009].*

A diagonal tensor of order 2 ($N = 2$) is simply a diagonal matrix. A diagonal tensor of order 3 ($N = 3$) has non-zero entries only along its diagonal as shown in Figure 3.6. A vector $\mathbf{d} \in \mathbb{R}^I$ consisting of the diagonal of a cubical tensor can be defined as

$$\mathbf{d} = diag(\mathcal{A}), \text{ where } \mathbf{d}(i) = \mathcal{A}(i, i, ..., i). \tag{3.3}$$

3.1.2 Basic Operations

Definition 3.7 (Unfolding: tensor to matrix transformation[4]). *A tensor can be unfolded into a matrix by rearranging its mode-n vectors. The* ***mode-n unfolding*** *of* $\mathcal{A}$ *is denoted by* $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times ... \times I_{n-1} \times I_{n+1} \times ... \times I_N)}$*, where the column vectors of* $\mathbf{A}_{(n)}$ *are the mode-n vectors of* $\mathcal{A}$*.*

The (column) order of the mode-n vectors in $\mathbf{A}_{(n)}$ is usually not important as long as it is consistent throughout the computation. For a second-order tensor (matrix) $\mathbf{A}$, its mode-1 unfolding is itself $\mathbf{A}$ and its mode-2 unfolding is its transpose $\mathbf{A}^T$. Figure 3.7 shows the mode-1 unfolding of the third-order tensor $\mathcal{A}$ on the left.

Definition 3.8 (Vectorization: tensor to vector transformation). *Similar to the vectorization of a matrix, the* ***vectorization*** *of a tensor is a linear*

[4] Unfolding is also known as flattening or matricization [Kolda and Bader, 2009].

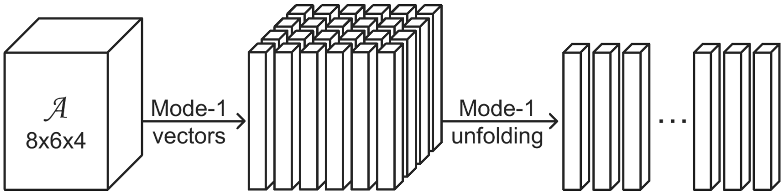


FIGURE 3.7: Visual illustration of the mode-1 unfolding.

transformation that converts the tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *into a column vector* $\mathbf{a} \in \mathbb{R}^{\prod_{n=1}^{N} I_n}$*, denoted as* $\mathbf{a} = vec(\mathcal{A})$*.*

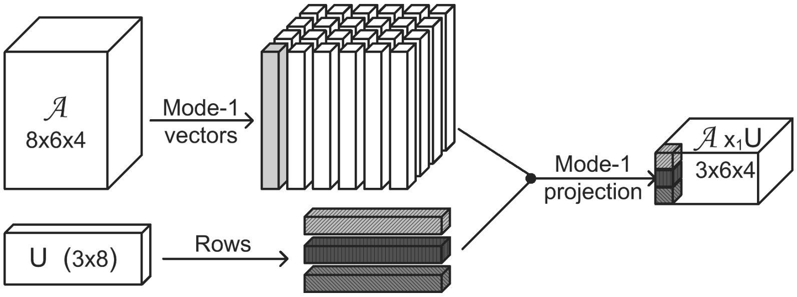


FIGURE 3.8: Visual illustration of the mode-n (mode-1) multiplication.

Definition 3.9 (**Mode-n product: tensor matrix multiplication**). *The* ***mode-n product*** *of a tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *by a matrix* $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ *is a tensor* $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N}$*, denoted as*

$$\mathcal{B} = \mathcal{A} \times_n \mathbf{U}, \tag{3.4}$$

where each entry of $\mathcal{B}$ *is defined as the sum of products of corresponding entries in* $\mathcal{A}$ *and* $\mathbf{U}$*:*

$$\mathcal{B}(i_1, ..., i_{n-1}, j_n, i_{n+1}, ..., i_N) = \sum_{i_n} \mathcal{A}(i_1, ..., i_N) \cdot \mathbf{U}(j_n, i_n). \tag{3.5}$$

This is equivalent to premultiplying each mode-n vector of $\mathcal{A}$ by $\mathbf{U}$. Thus, the mode-n product above can be written using the mode-n unfolding as

$$\mathbf{B}_{(n)} = \mathbf{U}\mathbf{A}_{(n)}. \tag{3.6}$$

For second order tensors (matrices) $\mathbf{A}$ and $\mathbf{U}$ of proper sizes,

$$\mathbf{A} \times_1 \mathbf{U} = \mathbf{U}\mathbf{A}, \quad \mathbf{A} \times_2 \mathbf{U} = \mathbf{A}\mathbf{U}^T. \tag{3.7}$$

Figure 3.8 demonstrates how the mode-1 multiplication $\mathcal{A} \times_1 \mathbf{U}$ is obtained.

The product $\mathcal{A} \times_1 \mathbf{U}$ is computed as the inner products between the mode-1 vectors of $\mathcal{A}$ and the rows of $\mathbf{U}$. In the mode-1 multiplication in Figure 3.8, each mode-1 vector of $\mathcal{A}$ ($\in \mathbb{R}^8$) is projected by $\mathbf{U} \in \mathbb{R}^{3\times 8}$ to obtain a vector ($\in \mathbb{R}^3$), as the differently shaded vector indicates in the right of the figure.

Tensor matrix multiplication has the following two properties [De Lathauwer et al., 2000b].

Property 3.1. *Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times \ldots \times I_N}$, and two matrices $\mathbf{U} \in \mathbb{R}^{J_n\times I_n}$ and $\mathbf{V} \in \mathbb{R}^{J_m\times I_m}$, where $m \neq n$, we have*

$$(\mathcal{A} \times_m \mathbf{U}) \times_n \mathbf{V} = (\mathcal{A} \times_n \mathbf{V}) \times_m \mathbf{U}. \tag{3.8}$$

Property 3.2. *Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times \ldots \times I_N}$, and two matrices $\mathbf{U} \in \mathbb{R}^{J_n\times I_n}$ and $\mathbf{V} \in \mathbb{R}^{K_n\times I_n}$, we have*

$$(\mathcal{A} \times_n \mathbf{U}) \times_n \mathbf{V} = \mathcal{A} \times_n (\mathbf{V} \cdot \mathbf{U}). \tag{3.9}$$

Definition 3.10 (**Mode-n product: tensor vector multiplication**). *The **mode-n product** of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times \ldots \times I_N}$ by a vector $\mathbf{u} \in \mathbb{R}^{I_n\times 1}$ is a tensor $\mathcal{C} \in \mathbb{R}^{I_1\times \ldots \times I_{n-1}\times 1\times I_{n+1}\times \ldots \times I_N}$, denoted as*

$$\mathcal{C} = \mathcal{A} \times_n \mathbf{u}^T, \tag{3.10}$$

where each entry of $\mathcal{C}$ is defined as

$$\mathcal{C}(i_1, ..., i_{n-1}, 1, i_{n+1}, ..., i_N) = \sum_{i_n} \mathcal{A}(i_1, ..., i_N) \cdot \mathbf{u}(i_n). \tag{3.11}$$

Multiplication of a tensor by a vector can be viewed as a special case of tensor matrix multiplication with $J_n = 1$ (so $\mathbf{U} \in \mathbb{R}^{J_n\times I_n} = \mathbf{u}^T$). This product $\mathcal{A} \times_1 \mathbf{u}^T$ can be computed as the inner products between the mode-1 vectors of $\mathcal{A}$ and $\mathbf{u}$. Note that the nth dimension of $\mathcal{C}$ is 1, so effectively the order of C is reduced to $N-1$. The *squeeze*() function in MATLAB can remove all modes with dimension equal to one.

Definition 3.11. *The **scalar product** (inner product) of two same-sized tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1\times I_2\times \ldots \times I_N}$ is defined as*

$$< \mathcal{A}, \mathcal{B} >= \sum_{i_1} \sum_{i_2} \ldots \sum_{i_N} \mathcal{A}(i_1, i_2, ..., i_N) \cdot \mathcal{B}(i_1, i_2, ..., i_N). \tag{3.12}$$

This can be seen as a generalization of the inner product in linear algebra (Section A.1.4).

Definition 3.12. *The **Frobenius norm** of $\mathcal{A}$ is defined as*

$$\parallel \mathcal{A} \parallel_F = \sqrt{< \mathcal{A}, \mathcal{A} >}. \tag{3.13}$$

This is a straightforward extension of the matrix Frobenius norm (Section A.1.5).

3.1.3 Tensor/Matrix Distance Measure

The Frobenius norm can be used to measure the distance between tensors $\mathcal{A}$ and $\mathcal{B}$ as

$$dist(\mathcal{A}, \mathcal{B}) = \parallel \mathcal{A} - \mathcal{B} \parallel_F . \tag{3.14}$$

Although this is a tensor-based measure, it is equivalent to a distance measure of corresponding vector representations denoted as $vec(\mathcal{A})$ and $vec(\mathcal{B})$, as to be shown in the following. We first derive a property regarding the scalar product between two tensors:

Proposition 3.1. *$< \mathcal{A}, \mathcal{B} > = < vec(\mathcal{A}), vec(\mathcal{B}) > = [vec(\mathcal{B})]^T vec(\mathcal{A})$.*

Proof. Let $\mathbf{a} = vec(\mathcal{A})$ and $\mathbf{b} = vec(\mathcal{B})$ for convenience. From Equation (3.12), $< \mathcal{A}, \mathcal{B} >$ is the summing the products between all corresponding entries in $\mathcal{A}$ and $\mathcal{B}$. We can have the same results by the sum of products between all corresponding entries in $\mathbf{a}$ and $\mathbf{b}$, their vectorizations. Thus, we have

$$\begin{aligned} < \mathcal{A}, \mathcal{B} > &= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} \mathcal{A}(i_1, i_2, ..., i_N) \cdot \mathcal{B}(i_1, i_2, ..., i_N) \\ &= \sum_{i=1}^{\prod_{n=1}^{N} I_n} \mathbf{a}(i) \cdot \mathbf{b}(i) \\ &= < \mathbf{a}, \mathbf{b} > \\ &= [\mathbf{b}]^T \mathbf{a}. \end{aligned}$$

□

Then, it is straightforward to show the equivalence.

Proposition 3.2. *$dist(\mathcal{A}, \mathcal{B}) = \parallel vec(\mathcal{A}) - vec(\mathcal{B}) \parallel_2$.*

Proof. From Proposition 3.1,

$$\begin{aligned} dist(\mathcal{A}, \mathcal{B}) &= \parallel \mathcal{A} - \mathcal{B} \parallel_F \\ &= \sqrt{< (\mathcal{A} - \mathcal{B}), (\mathcal{A} - \mathcal{B}) >} \\ &= \sqrt{< vec(\mathcal{A}) - vec(\mathcal{B}), vec(\mathcal{A}) - vec(\mathcal{B}) >} \\ &= \parallel vec(\mathcal{A}) - vec(\mathcal{B}) \parallel_2 . \end{aligned}$$

□

Proposition 3.2 indicates that the Frobenius norm of the difference between two tensors equals the Euclidean distance between their vectorized representations. The tensor Frobenius norm is a point-based measurement [Lu et al., 2004] without taking the tensor structure into account.

For second-order tensors, that is, matrices, their distance can be measured by the matrix Frobenius norm, which equals the square root of the trace of the difference matrix:

$$dist(\mathbf{A}, \mathbf{B}) = \| \mathbf{A} - \mathbf{B} \|_F = \sqrt{\mathrm{tr}\left((\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{B})\right)}. \tag{3.15}$$

An alternative for matrix distance is the so-called *volume measure* used in [Meng and Zhang, 2007]. The volume measure between matrices **A** and **B** is defined as

$$dist(\mathbf{A}, \mathbf{B}) = vol(\mathbf{A} - \mathbf{B}) = \sqrt{|(\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{B})|}, \tag{3.16}$$

where $|\cdot|$ denotes the determinant.

The matrix Frobenius norm is further generalized as the *assembled matrix distance* (AMD) in [Zuo et al., 2006] as

$$d_{AMD}(\mathbf{A}, \mathbf{B}) = \left(\sum_{i_1=1}^{I_1} \left(\sum_{i_2=1}^{I_2} (\mathbf{A}(i_1, i_2) - \mathbf{B}(i_1, i_2))^2 \right)^{p/2} \right)^{1/p}, \tag{3.17}$$

where the power p weights the differences between elements. AMD is a variation of the p-norm for vectors and it treats a matrix as a vector effectively. AMD with $p = 2$ is equivalent to the matrix Frobenius norm and the Euclidean norm for vectors. The AMD measure can be generalized to general higher-order tensors, or it can also be modified to take data properties (such as shape and connectivity) into account as in [Porro-Muñoz et al., 2011].

Distance measures are frequently used by classifiers to measure similarity or dissimilarity. Furthermore, it is possible to design classifiers by taking into account the matrix/tensor representation or structure. For example, Wang et al. [2008] proposed a classifier specially designed for matrix representations of patterns and showed that such a classifier has advantages in features extracted from matrix representations.

3.2 Tensor Decompositions

Multilinear subspace learning is based on tensor decompositions. This section reviews two most important works in this area.

3.2.1 CANDECOMP/PARAFAC

Hitchcock [1927a,b] first proposed the idea of expressing a tensor as the sum of rank-one tensors in polyadic form. It became popular in the psychometrics community with the independent introduction of canonical decomposition (CANDECOMP) by Carroll and Chang [1970] and parallel factors (PARAFAC) by Harshman [1970].

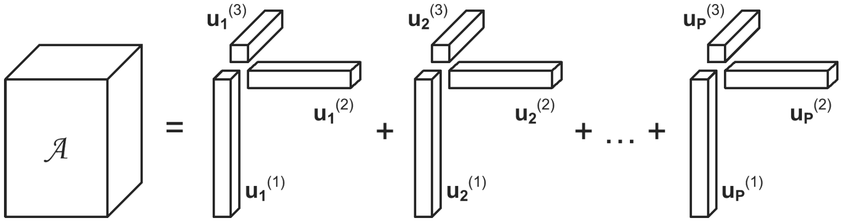


FIGURE 3.9: The CANDECOMP/PARAFAC decomposition of a third-order tensor.

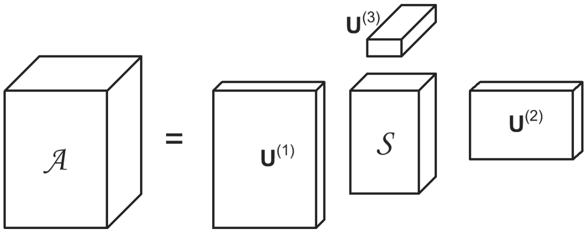


FIGURE 3.10: The Tucker decomposition of a third-order tensor.

With the CANDECOMP/PARAFAC decomposition (CP decomposition), a tensor $\mathcal{A}$ can be factorized into a linear combination of P rank-one tensors:

$$\mathcal{A} = \sum_{p=1}^{P} \lambda_p \mathbf{u}_p^{(1)} \circ \mathbf{u}_p^{(2)} \circ ... \circ \mathbf{u}_p^{(N)}, \tag{3.18}$$

where $P \leq \prod_{n=1}^{N} I_n$. Figure 3.9 illustrates this decomposition.

3.2.2 Tucker Decomposition and HOSVD

The Tucker decomposition [Tucker, 1966] was introduced in the 1960s. It decomposes an Nth-order tensor $\mathcal{A}$ into a core tensor $\mathcal{S}$ multiplied by N matrices, one in each mode:

$$\mathcal{A} \quad = \quad \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times ... \times_N \mathbf{U}^{(N)}, \tag{3.19}$$

where $P_n \leq I_n$ for $n = 1, ..., N$, and $\mathbf{U}^{(n)} = \left[\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} ... \mathbf{u}_{P_n}^{(n)}\right]$ is an $I_n \times P_n$ matrix often assumed to have orthonormal column vectors. Figure 3.10 illustrates this decomposition.

The Tucker decomposition was investigated mainly in psychometrics after its initial introduction. It was reintroduced by De Lathauwer et al. [2000b] as the *higher-order singular value decomposition* (HOSVD) to the communities of numerical algebra and signal processing, followed by many other disciplines. When $P_n = I_n$ for $n = 1, ..., N$ and $\{\mathbf{U}^{(n)}, n = 1, ..., N\}$ are all orthogonal

$I_n \times I_n$ matrices, then from Equation (3.19), the core tensor can be written as

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}. \tag{3.20}$$

Because $\mathbf{U}^{(n)}$ has orthonormal columns, we have [De Lathauwer et al., 2000a]

$$\| \mathcal{A} \|_F^2 = \| \mathcal{S} \|_F^2 . \tag{3.21}$$

A matrix representation of this decomposition can be obtained by unfolding $\mathcal{A}$ and $\mathcal{S}$ as

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{S}_{(n)} \cdot \left(\mathbf{U}^{(n+1)} \otimes ... \otimes \mathbf{U}^{(N)} \otimes \mathbf{U}^{(1)} \otimes ... \otimes \mathbf{U}^{(n-1)} \right)^T , \tag{3.22}$$

where $\otimes$ denotes the Kronecker product (Section A.1.4). The decomposition can also be written as

$$\mathcal{A} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} \mathcal{S}(i_1, i_2, ..., i_N) \mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ ... \circ \mathbf{u}_{i_N}^{(N)}, \tag{3.23}$$

that is, any tensor $\mathcal{A}$ can be written as a linear combination of $\prod_{n=1}^{N} I_n$ rank-one tensors. Comparison of Equation (3.23) against Equation (3.18) reveals the *equivalence* between Tucker decomposition and CP decomposition.

3.3 Multilinear Projections

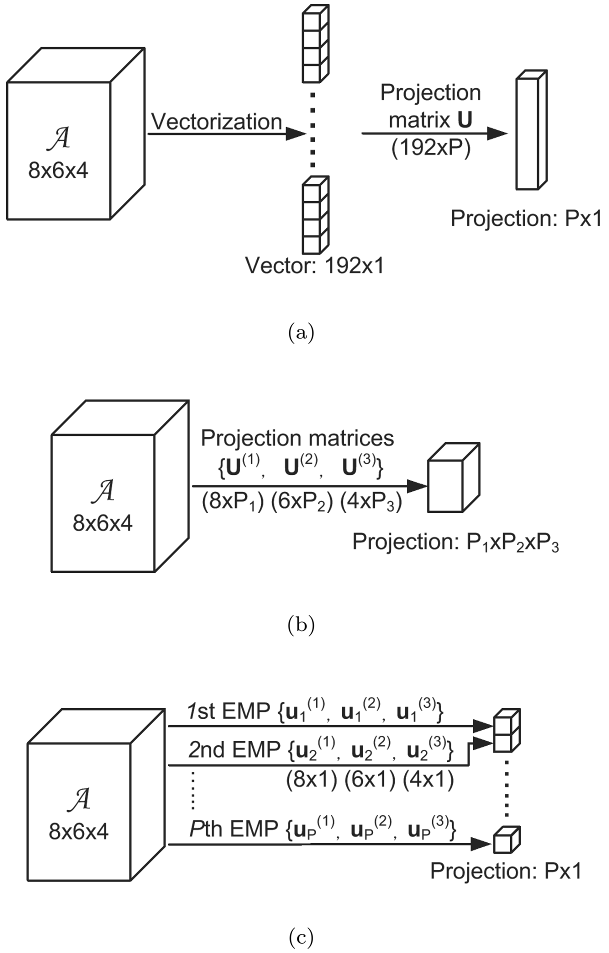
A *tensor subspace* is defined through a multilinear projection that maps the input data from a high-dimensional space to a low-dimensional space [He et al., 2005a]. Therefore, multilinear projection is an important concept to grasp before proceeding to multilinear subspace learning.

This section presents three basic multilinear projections named in terms of the input and output representations of the projection: the traditional vector-to-vector projection, the tensor-to-tensor projection, and the tensor-to-vector projection. Furthermore, we investigate the relationships among these projections.

3.3.1 Vector-to-Vector Projection

Linear projection is a standard transformation used widely in various applications [Duda et al., 2001; Moon and Stirling, 2000]. A linear projection takes a vector $\mathbf{x} \in \mathbb{R}^I$ as input and projects it to a vector $\mathbf{y} \in \mathbb{R}^P$ using a projection matrix $\mathbf{U} \in \mathbb{R}^{I \times P}$:

$$\mathbf{y} = \mathbf{U}^T \mathbf{x} = \mathbf{x} \times_1 \mathbf{U}^T. \tag{3.24}$$



(a)

(b)

(c)

FIGURE 3.11: Illustration of (a) vector-to-vector projection, (b) tensor-to-tensor projection, and (c) tensor-to-vector projection, where EMP stands for elementary multilinear projection.

If we name this projection according to its input and output representations, linear projection is a vector-to-vector projection because it maps a vector to another vector. When the input data is a matrix or a higher-order tensor, it needs to be vectorized (reshaped into a vector) before projection. Figure 3.11(a) illustrates the vector-to-vector projection of a tensor object $\mathcal{A}$.

Denote each column of $\mathbf{U}$ as $\mathbf{u}_p$, so $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ ...\ \mathbf{u}_P]$. Then each column $\mathbf{u}_p$ projects $\mathbf{x}$ to a scalar $\mathbf{y}(p)$ or y_p:

$$y_p = \mathbf{y}(p) = \mathbf{u}_p^T \mathbf{x}. \tag{3.25}$$

3.3.2 Tensor-to-Tensor Projection

In addition to traditional vector-to-vector projection, a tensor can also be projected to another tensor (of the same order usually), called the tensor-to-tensor projection (TTP). It is formulated based on the Tucker decomposition.

Consider the second-order case first. A second-order tensor (matrix) $\mathbf{X}$ resides in the *tensor space* denoted as $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2}$, which is defined as the *tensor product* (*outer product*) of two vector spaces $\mathbb{R}^{I_1}$ and $\mathbb{R}^{I_2}$. For the projection of a matrix $\mathbf{X}$ in a tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2}$ to another tensor $\mathbf{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2}$, where $P_n \leq I_n$ for $n = 1, 2$, two projection matrices $\mathbf{U}^{(1)} \in \mathbb{R}^{I_1 \times P_1}$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{I_2 \times P_2}$ (usually with orthonormal columns) are used so that [De Lathauwer et al., 2000a]

$$\mathbf{Y} = \mathbf{X} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} = \mathbf{U}^{(1)^T} \mathbf{X} \mathbf{U}^{(2)}. \tag{3.26}$$

For the general higher-order case, an Nth-order tensor $\mathcal{X}$ resides in the tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ [De Lathauwer et al., 2000b], which is the *tensor product* (*outer product*) of N vector spaces $\mathbb{R}^{I_1}$, $\mathbb{R}^{I_2}$, ..., $\mathbb{R}^{I_N}$. For the projection of a tensor $\mathcal{X}$ in a tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} ... \bigotimes \mathbb{R}^{P_N}$, where $P_n \leq I_n$ for all n, N projection matrices $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ (usually with orthonormal columns) are used so that [De Lathauwer et al., 2000a]

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}. \tag{3.27}$$

These N projection matrices used for TTP can be concisely written as $\{\mathbf{U}^{(n)}\}$. Figure 3.11(b) demonstrates the TTP of a tensor object $\mathcal{A}$ to a smaller tensor of size $P_1 \times P_2 \times P_3$. This multilinear projection can be carried out through N mode-n multiplications, as illustrated in Figure 3.8.

3.3.3 Tensor-to-Vector Projection

The third multilinear projection is from a tensor space to a vector space, and it is called the tensor-to-vector projection (TVP)[5]. It is formulated based on the CANDECOMP/PARAFAC model.

As a vector can be viewed as multiple scalars, the projection from a tensor to a vector can be viewed as multiple projections, each of which projects a tensor to a scalar, as illustrated in Figure 3.11(c). In the figure, the TVP of a tensor $\mathcal{A} \in \mathbb{R}^{8 \times 6 \times 4}$ to a $P \times 1$ vector consists of P projections, each projecting $\mathcal{A}$ to a scalar. Thus, the projection from a tensor to a scalar is the building block for TVP and it is considered first.

A second-order tensor (matrix) $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ can be projected to a scalar y through two unit projection vectors $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ as

$$y = \mathbf{X} \times_1 \mathbf{u}^{(1)^T} \times_2 \mathbf{u}^{(2)^T} = \mathbf{u}^{(1)^T} \mathbf{X} \mathbf{u}^{(2)}, \ \| \mathbf{u}^{(1)} \| = \| \mathbf{u}^{(2)} \| = 1, \tag{3.28}$$

[5] The tensor-to-vector projection is referred to as the rank-one projections in some works [Wang and Gong, 2006; Tao et al., 2006; Hua et al., 2007].

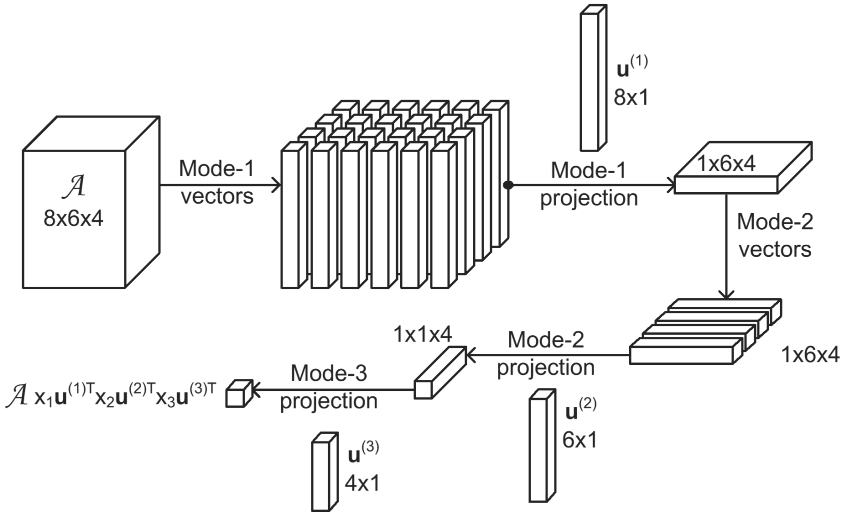


FIGURE 3.12: Illustration of an elementary multilinear projection.

where $\| \cdot \|$ is the Euclidean norm for vectors (see Section A.1.4). It can be written as the inner product between $\mathbf{X}$ and the outer products of $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$:

$$y =< \mathbf{X}, \mathbf{u}^{(1)}\mathbf{u}^{(2)^T} > . \tag{3.29}$$

A general tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ can be projected to a point y through N unit projection vectors $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, ..., \mathbf{u}^{(N)}\}$, which can also be written as $\{\mathbf{u}^{(n)}, n = 1, ..., N\}$ or simply as $\{\mathbf{u}^{(n)}\}$:

$$y = \mathcal{X} \times_1 \mathbf{u}^{(1)^T} \times_2 \mathbf{u}^{(2)^T} ... \times_N \mathbf{u}^{(N)^T}, \;\; \| \mathbf{u}^{(n)} \| = 1 \text{ for } n = 1, ..., N. \tag{3.30}$$

It can be written as the scalar product (Equation (3.12)) of $\mathcal{X}$ and the outer product of $\{\mathbf{u}^{(n)}\}$:

$$y =< \mathcal{X}, \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)} > . \tag{3.31}$$

Denote $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, then $y =< \mathcal{X}, \mathcal{U} >$. This multilinear projection through $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, ..., \mathbf{u}^{(N)}\}$ is named as an *elementary multilinear projection* (EMP) [Lu et al., 2009c], which is a projection of a tensor to a scalar[6]. It is a rank-one projection and it consists of one projection vector in each mode. Figure 3.12 illustrates an EMP of a tensor $\mathcal{A} \in \mathbb{R}^{8 \times 6 \times 4}$.

Thus, the TVP of a tensor $\mathcal{X}$ to a vector $\mathbf{y} \in \mathbb{R}^P$ in a P-dimensional vector space consists of P EMPs,

$$\{\mathbf{u}_p^{(1)}, \mathbf{u}_p^{(2)}, ..., \mathbf{u}_p^{(N)}\}, p = 1, ..., P, \tag{3.32}$$

[6] We call it EMP rather than tensor-to-scalar projection for two reasons. One is that it is used as a building block in TVP. The other is that we want to emphasize that it is an elementary operation.

which can be written concisely as $\{\mathbf{u}_p^{(n)}, n = 1, ..., N\}_{p=1}^{P}$ or simply as $\{\mathbf{u}_p^{(n)}\}_N^P$. The TVP from $\mathcal{X}$ to $\mathbf{y}$ is then written as

$$\mathbf{y} = \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)}, n = 1, ..., N\}_{p=1}^{P} = \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)}\}_N^P, \tag{3.33}$$

where the pth entry of $\mathbf{y}$ is obtained from the pth EMP as

$$y_p = \mathbf{y}(p) = \mathcal{X} \times_1 \mathbf{u}_p^{(1)^T} \times_2 \mathbf{u}_p^{(2)^T} ... \times_N \mathbf{u}_p^{(N)^T} = \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)}\}. \tag{3.34}$$

Figure 3.11(c) shows the TVP of a tensor $\mathcal{A}$ to a vector of size $P \times 1$.

3.4 Relationships among Multilinear Projections

With the introduction of the three basic multilinear projections, it is worthwhile to investigate their relationships.

Degenerated conditions: It is easy to verify that the vector-to-vector projection is the special case of the tensor-to-tensor projection and the tensor-to-vector projection for $N = 1$. The elementary multilinear projection is the degenerated version of the tensor-to-tensor projection with $P_n = 1$ for all n.

EMP view of TTP: Each projected element in the tensor-to-tensor projection can be viewed as the projection by an elementary multilinear projection formed by taking one column from each modewise projection matrix. Thus, a projected tensor in the tensor-to-tensor projection is obtained through $\prod_{n=1}^{N} P_n$ elementary multilinear projections with shared projection vectors (from the projection matrices) in effect, while in the tensor-to-vector projection, the P elementary multilinear projections do not have shared projection vectors.

Equivalence between EMP and VVP: Recall that the projection using an elementary multilinear projection $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, ..., \mathbf{u}^{(N)}\}$ can be written as

$$y =< \mathcal{X}, \mathcal{U} >=< vec(\mathcal{X}), vec(\mathcal{U}) >= [vec(\mathcal{U})]^T \, vec(\mathcal{X}), \tag{3.35}$$

by Proposition 3.1. Thus, an elementary multilinear projection is equivalent to a linear projection of $vec(\mathcal{X})$, the vectorized representation of $\mathcal{X}$, by a vector $vec(\mathcal{U})$ as in Equation (3.25). Because $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, Equation (3.35) indicates that the elementary multilinear projection is equivalent to a linear projection for $P = 1$ with a constraint on the projection vector such that it is the vectorized representation of a rank-one tensor.

Equivalence between TTP and TVP: Given a TVP $\{\mathbf{u}_p^{(n)}\}_N^P$, we can form N matrices $\{\mathbf{V}^{(n)}\}$, where

$$\mathbf{V}^{(n)} = [\mathbf{u}_1^{(n)}, ..., \mathbf{u}_p^{(n)}, ..., \mathbf{u}_P^{(n)}] \in \mathbb{R}^{I_n \times P}. \tag{3.36}$$

These matrices can be viewed as a TTP with $P_n = P$ for $n = 1, ..., N$ (equal subspace dimensions in all modes). Thus, the TVP of a tensor by $\{\mathbf{u}_p^{(n)}\}_N^P$ is equivalent to the diagonal of a corresponding TTP of the same tensor by $\{\mathbf{V}^{(n)}\}$ as defined in Equation (3.36):

$$\begin{aligned} \mathbf{y} &= \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)}\}_N^P && (3.37) \\ &= diag\left(\mathcal{X} \times_1 \mathbf{V}^{(1)^T} \times_2 \mathbf{V}^{(2)^T} ... \times_N \mathbf{V}^{(N)^T}\right). && (3.38) \end{aligned}$$

In the second-order case, this equivalence is

$$\mathbf{y} = diag\left(\mathbf{V}^{(1)^T}\mathbf{X}\mathbf{V}^{(2)}\right). \quad (3.39)$$

Number of parameters to estimate: The number of parameters to be estimated in a particular projection indicates model complexity, an important concern in practice. Compared with a projection vector of size $I \times 1$ in a VVP specified by I parameters ($I = \prod_{n=1}^{N} I_n$ for an Nth-order tensor), an EMP in a TVP is specified by $\sum_{n=1}^{N} I_n$ parameters. Hence, to project a tensor of size $\prod_{n=1}^{N} I_n$ to a vector of size $P \times 1$, TVP needs to estimate only $P \cdot \sum_{n=1}^{N} I_n$ parameters, while VVP needs to estimate $P \cdot \prod_{n=1}^{N} I_n$ parameters. The implication is that TVP has fewer parameters to estimate while being more constrained on the solutions, and VVP has less constraint on the solutions sought while having more parameters to estimate. In other words, TVP has a simpler model than VVP. For TTP with the same amount of dimensionality reduction $\prod_{n=1}^{N} P_n = P$, $\sum_{n=1}^{N} P_n \times I_n$ parameters need to be estimated. Thus, due to shared projection vectors, TTP may need to estimate even fewer parameters and its model can be even simpler.

Table 3.1 contrasts the number of parameters to be estimated by the three projections for the same amount of dimensionality reduction for several cases. Figure 3.13 further illustrates the first three cases, where the numbers of parameters are normalized with respect to that by VVP for better visualization. From the table and figure, we can see that for higher-order tensors, the conventional VVP model becomes extremely complex and parameter estimation becomes extremely difficult. This often leads to the small sample size (SSS) problem in practice when there are limited number of training samples available.

3.5 Scatter Measures for Tensors and Scalars

3.5.1 Tensor-Based Scatters

In analogy to the definitions of scatters in Equations (2.1), (2.27), and (2.25) for vectors used in linear subspace learning, we define tensor-based scatters to be used in multilinear subspace learning (MSL) through TTP.

TABLE 3.1: Number of parameters to be estimated by three multilinear projections.

Input	**Output**	**VVP**	**TVP**	**TTP**
$\prod_{n=1}^{N} I_n$	P	$P \cdot \prod_{n=1}^{N} I_n$	$P \cdot \sum_{n=1}^{N} I_n$	$\sum_{n=1}^{N} P_n \times I_n$
10×10	4	400	80	40 ($P_n = 2$)
100×100	4	40,000	800	400 ($P_n = 2$)
$100 \times 100 \times 100$	8	8,000,000	2,400	600 ($P_n = 2$)
$\prod_{n=1}^{4} 100$	16	1,600,000,000	6,400	800 ($P_n = 2$)

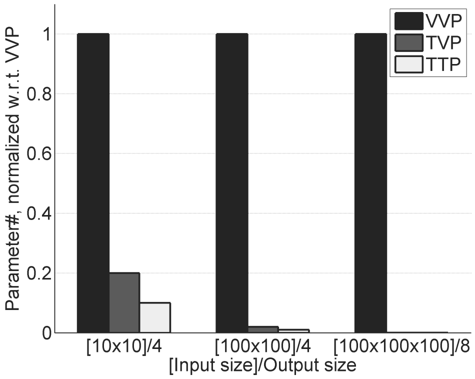


FIGURE 3.13: Comparison of the number of parameters to be estimated by VVP, TVP, and TTP, normalized with respect to the number by VVP for visualization.

Definition 3.13. *Let $\{\mathcal{A}_m, m = 1, ..., M\}$ be a set of M tensor samples in $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$. The total scatter of these tensors is defined as*

$$\Psi_{T_{\mathcal{A}}} = \sum_{m=1}^{M} \parallel \mathcal{A}_m - \bar{\mathcal{A}} \parallel_F^2, \tag{3.40}$$

where $\bar{\mathcal{A}}$ is the mean tensor *calculated as*

$$\bar{\mathcal{A}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{A}_m. \tag{3.41}$$

The mode-n total scatter matrix of these samples is then defined as

$$\mathbf{S}_{T_{\mathcal{A}}}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)} \right) \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)} \right)^T, \tag{3.42}$$

where $\mathbf{A}_{m(n)}$ *and* $\bar{\mathbf{A}}_{(n)}$ *are the mode-n unfolding of* $\mathcal{A}_m$ *and* $\bar{\mathcal{A}}$*, respectively.*

Definition 3.14. *Let* $\{\mathcal{A}_m, m = 1, ..., M\}$ *be a set of* M *tensor samples in* $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$*. The between-class scatter of these tensors is defined as*

$$\Psi_{B_{\mathcal{A}}} = \sum_{c=1}^{C} M_c \parallel \bar{\mathcal{A}}_c - \bar{\mathcal{A}} \parallel_F^2, \tag{3.43}$$

and the within-class scatter of these tensors is defined as

$$\Psi_{W_{\mathcal{A}}} = \sum_{m=1}^{M} \parallel \mathcal{A}_m - \bar{\mathcal{A}}_{c_m} \parallel_F^2, \tag{3.44}$$

where C *is the number of classes,* M_c *is the number of samples for class* c*,* c_m *is the class label for the* m*th sample* $\mathcal{A}_m$*,* $\bar{\mathcal{A}}$ *is the mean tensor, and the* class mean tensor *is*

$$\bar{\mathcal{A}}_c = \frac{1}{M_c} \sum_{m, c_m = c} \mathcal{A}_m. \tag{3.45}$$

Next, the mode-n between-class and within-class scatter matrices are defined accordingly.

Definition 3.15. *The mode-n between-class scatter matrix of these samples is defined as*

$$\mathbf{S}_{B_{\mathcal{A}}}^{(n)} = \sum_{c=1}^{C} M_c \cdot \left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)\left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)^T, \tag{3.46}$$

and the mode-n within-class scatter matrix of these samples is defined as

$$\mathbf{S}_{W_{\mathcal{A}}}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)\left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)^T, \tag{3.47}$$

where $\bar{\mathbf{A}}_{c_m(n)}$ *is the mode-n unfolding of* $\bar{\mathcal{A}}_{c_m}$*.*

From the definitions above, the following properties are derived:

Property 3.3. *Because* $\mathrm{tr}(\mathbf{A}\mathbf{A}^T) = \parallel \mathbf{A} \parallel_F^2$ *and* $\parallel \mathcal{A} \parallel_F^2 = \parallel \mathbf{A}_{(n)} \parallel_F^2$,

$$\Psi_{B_{\mathcal{A}}} = \mathrm{tr}\left(\mathbf{S}_{B_{\mathcal{A}}}^{(n)}\right) = \sum_{c=1}^{C} M_c \parallel \bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)} \parallel_F^2 \tag{3.48}$$

and

$$\Psi_{W_{\mathcal{A}}} = \mathrm{tr}\left(\mathbf{S}_{W_{\mathcal{A}}}^{(n)}\right) = \sum_{m=1}^{M} \parallel \mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)} \parallel_F^2 \tag{3.49}$$

for all n.

Scatters for matrices: As a special case, when $N = 2$, we have a set of M matrix samples $\{\mathbf{A}_m, m = 1, ..., M\}$ in $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2}$. The total scatter of these matrices is defined as

$$\Psi_{T_{\mathbf{A}}} = \sum_{m=1}^{M} \parallel \mathbf{A}_m - \bar{\mathbf{A}} \parallel_F^2, \tag{3.50}$$

where $\bar{\mathbf{A}}$ is the mean matrix calculated as

$$\bar{\mathbf{A}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{A}_m. \tag{3.51}$$

The between-class scatter of these matrix samples is defined as

$$\Psi_{B_{\mathbf{A}}} = \sum_{c=1}^{C} M_c \parallel \bar{\mathbf{A}}_c - \bar{\mathbf{A}} \parallel_F^2, \tag{3.52}$$

and the within-class scatter of these matrix samples is defined as

$$\Psi_{W_{\mathbf{A}}} = \sum_{m=1}^{M} \parallel \mathbf{A}_m - \bar{\mathbf{A}}_{c_m} \parallel_F^2, \tag{3.53}$$

where $\bar{\mathbf{A}}$ is the mean matrix, and the class mean matrix is

$$\bar{\mathbf{A}}_c = \frac{1}{M_c} \sum_{m, c_m = c} \mathbf{A}_m. \tag{3.54}$$

3.5.2 Scalar-Based Scatters

While the tensor-based scatters defined above are useful for developing MSL algorithms based on TTP, they are not applicable to those based on the TVP/EMP. Therefore, scalar-based scatters need to be defined for MSL through TVP/EMP. They can be viewed as the degenerated versions of the vector-based or tensor-based scatters.

Definition 3.16. *Let $\{a_m, m = 1, ..., M\}$ be a set of M scalar samples. The total scatter of these scalars is defined as*

$$S_{T_a} = \sum_{m=1}^{M} (a_m - \bar{a})^2, \tag{3.55}$$

where $\bar{a}$ is the mean scalar calculated as

$$\bar{a} = \frac{1}{M} \sum_{m=1}^{M} a_m. \tag{3.56}$$

Thus, the total scatter for scalars is simply a scaled version of the sample *variance.*

Definition 3.17. *Let $\{a_m, m = 1, ..., M\}$ be a set of M scalar samples. The between-class scatter of these scalars is defined as*

$$S_{B_a} = \sum_{c=1}^{C} M_c(\bar{a}_c - \bar{a})^2, \tag{3.57}$$

and the within-class scatter of these scalars is defined as

$$S_{W_a} = \sum_{m=1}^{M} (a_m - \bar{a}_{c_m})^2, \tag{3.58}$$

where

$$\bar{a}_c = \frac{1}{M_c} \sum_{m, c_m = c} a_m. \tag{3.59}$$

3.6 Summary

- An Nth-order tensor is an N-dimensional array with N modes.
- Most tensor operations can be viewed as operations on the mode-n vectors. This is key to understanding the connections between tensor operations and matrix/vector operations.
- Linear projection is a vector-to-vector projection. We can project a tensor directly to a tensor or vector through a tensor-to-tensor projection or a tensor-to-vector projection, respectively. Most of the connections among them can be revealed through elementary multilinear projection, which maps a tensor to a scalar.
- For the same amount of dimensionality reduction, TTP and TVP need to estimate many fewer parameters than VVP (linear projection) for higher-order tensors. Thus, TTP and TVP tend to have simpler models and lead to better generalization performance.
- Scatter measures (and potentially other measures/criteria) employed in VVP-based learning can be extended to tensors for TTP-based learning and to scalars for TVP-based learning.

3.7 Further Reading

De Lathauwer et al. [2000b] give a good introduction to multilinear algebra preliminaries for readers with a basic linear algebra background, so it is recommended to those unfamiliar with multilinear algebra. Those interested in HOSVD can find an in-depth treatment in this seminal paper. Its companion paper [De Lathauwer et al., 2000a] focuses on tensor approximation and it is also worth reading.

Kolda and Bader [2009] give a very comprehensive review of tensor decompositions. This paper also covers the preliminaries of multilinear algebra very well, with much additional material. It discusses many variations and various issues related to tensor decompositions. Cichocki et al. [2009] provides another good reference, where Section 1.4 covers the basics of multilinear algebra and Section 1.5 covers tensor decompositions.

We first named the tensor-to-vector projection and the elementary multilinear projection in [Lu et al., 2007b], commonly referred to as rank-one decomposition/projection. We then named the Tucker/HOSVD-style projection as the tensor-to-tensor projection in [Lu et al., 2009c] to suit subspace learning context better. Our survey paper [Lu et al., 2011] further examined the relationships among various projections, formulated the MSL framework, and gave a unifying view of the various scatter measures for MSL algorithms.

For multilinear algebra in a broader context, there are books that are several decades old [Greub, 1967; Lang, 1984], and there is also a book by mathematician Hackbusch [2012] with a modern treatment, which can be a good resource to consult for future development of MSL theories.