Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

An introduction to tensors for path signatures

Jack Beda[1], Gonçalo dos Reis[1], Nikolas Tapia[2]

submitted: February 3, 2025

[1] University of Edinburgh
James Clerk Maxwell Building
Peter Guthrie Tait Rd
Edinburgh EH9 3FD
United Kingdom
E-Mail: jack@beda.ca
g.dosreis@ed.ac.uk

[2] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: nikolasesteban.tapiamunoz@wias-berlin.de

No. 3173
Berlin 2025

2020 *Mathematics Subject Classification.* 60L10, 60L70, 60L99.

Key words and phrases. Signatures, tensors, linear algebra, tensor factorization, tensor algebra.

NT acknowledges support from DFG CRC/TRR 388 “Rough Analysis, Stochastic Dynamics and Related Fields”, Project B01. GdR acknowledges partial support by the Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/R511687/1], from the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the projects UIDB/00297/2020 and UIDP/00297/2020 (Center for Mathematics and Applications, NOVA Math), and by the UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding Guarantee [Project APP55638]. A GibHub repository related to the content of Sections 5 and 6 can be found in https://github.com/jfbeda/tensor_factoring.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

An introduction to tensors for path signatures

Jack Beda, Gonçalo dos Reis, Nikolas Tapia

Abstract

We present a fit-for-purpose introduction to tensors and their operations. It is envisaged to help the reader become acquainted with its underpinning concepts for the study of path signatures. The text includes exercises, solutions and many intuitive explanations. The material discusses direct sums and tensor products as two possible operations that make the Cartesian product of vectors spaces a vector space. The difference lies in linear Vs. multilinear structures – the latter being the suitable one to deal with path signatures. The presentation is offered to understand tensors in a deeper sense than just a multidimensional array. The text concludes with the prime example of an algebra in relation to path signatures: the *tensor algebra*. This manuscript is the extended version (with two extra sections) of a chapter to appear in Open Access in a forthcoming Springer volume "*Signatures Methods in Finance: An Introduction with Computational Applications*". The two additional sections here discuss the factoring of tensor product expressions to a minimal number of terms. This problem is not critical for path signatures theory, but is an elegant way of becoming familiar with the language of tensors and tensor products that are used throughout the forthcoming volume. A GitHub repository is attached.

1 Introduction

Throughout mathematics, computer science, and physics, the term *tensor* is used to describe a myriad of similar, but fundamentally different mathematical objects. For amusement, we invite the reader to visit the "100 Questions: A Mathematical Conventions Survey" and check[1] "Question 45: What is a tensor?" That answer's wide distribution seems to hint at a gap, as folks knowledge goes, to the mathematical meaning of a tensor (even among the informed?). It is thus necessary to be clear and unambiguous with our definitions and language so that at the end of this chapter the reader will be able to agree with us on the answer. For practical purposes, it often suffices to describe a tensor as a multidimensional array that extends the concept of a matrix [8]. This is surely true, but only after a certain structure on the underlying space is assumed. Tensors are much more, especially when the underlying spaces are infinite-dimensional [2, 3, 9].

For path signatures, seeing tensors as multidimensional arrays is a good starting point, but their power is only fully realized with an understanding of the operations that go with them: that is, with the algebra mixed in. The assumption behind the next sections is that the reader is largely unfamiliar with tensors (but having heard of multidimensional arrays).

Why do tensors come up in path signatures and how to read this chapter? We saw in [1] that path signatures rely heavily on iterated integrals and involve products of path components across different times. The path signature is a sequence of terms encoding information about a path at different levels of complexity: the first-level signature captures linear information via integrals of individual components; the

[1] Find the survey here: https://cims.nyu.edu/~tjl8195/survey/results.html#q45, at the time this manuscript was written.

DOI 10.20347/WIAS.PREPRINT.3173 Berlin 2025

second-level signature captures pairwise interactions, so-called bilinear relationships, like $\iint \mathrm{d}x^i \mathrm{d}x^j$, meaning they combine two inputs in a way that is linear in each (with x^i fixed, we have linearity in x^j, and vice-versa) whereas joint linearity does not hold; higher levels of the signature require more iterated integrals and in turn one requires multilinear relationships to describe these complicated interactions. Tensors and multilinear maps are natural and powerful mathematical tools to represent and analyze these interactions.

Section 2 introduces the basic definitions and properties of tensors by guiding the reader to first understand the difference between linear and bilinear operators, and how linear operators can be recovered from bilinear ones via the tensors product operation and the so-called *universal property*.

Section 3 then goes into a few other deeper properties and connects more explicitly tensors and algebras by introducing the *tensor algebra*. We will leave to subsequent chapters the development of further concepts like shuffle algebras, and power series to obtain exponentials and logarithms (alluded to in [1]).

2 A brief introduction to tensors

This section introduces two operations: *direct sums* and *tensor products*, two different ways of making new vector spaces out of old ones. Formally, each is a way of equipping the Cartesian product of vector spaces, $U \times V$, with a linear structure. The first leads to linear operators while the second leads to bilinear ones. While both are related, in many aspects they behave very differently[2]. In a nutshell, bilinear maps exhibit separate linearity in U and V while linear maps exhibit global linearity in $U \times V$. The distinction is particularly important in the algebra of tensors, where bilinear maps give rise to the tensor product structure through the so-called *universal property*. Linear maps correspond to mappings on the direct product space.

2.1 Direct sums

Let us recall that for two sets X and Y, their Cartesian product $X \times Y$ is defined as

$$X \times Y := \{(x, y) : x \in X, y \in Y\},$$

that is, the set of all ordered pairs where the first is an element of X and the second an element of Y.

Given vector spaces U and V, their Cartesian product does not immediately have a linear structure (i.e. is not immediately a vector space). In other words, after constructing the *set* $U \times V$ it is not clear how to add two ordered pairs, or multiply them by scalars. We must *define* a way to add and scale the elements of this set, and it turns out there are multiple, sensible and useful definitions. Direct sums are the simplest way to equip the Cartesian product of two (or more) vector spaces with a linear structure of its own.

Definition 2.1. Let U, V be vector spaces. On the Cartesian product $U \times V$ we define the following *linearity* operations: for $\boldsymbol{u}, \boldsymbol{u}_1, \boldsymbol{u}_2 \in U$, $\boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ and $\lambda \in \mathbb{R}$

$$(\boldsymbol{u}_1, \boldsymbol{v}_1) + (\boldsymbol{u}_2, \boldsymbol{v}_2) := (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}_1 + \boldsymbol{v}_2), \tag{1}$$

$$\lambda(\boldsymbol{u}, \boldsymbol{v}) := (\lambda \boldsymbol{u}, \lambda \boldsymbol{v}). \tag{2}$$

[2] For instance, there is no open mapping theorem for bilinear surjective maps, nor is there a Hahn-Banach theorem for bilinear continuous forms.

One can check that $U \times V$, equipped with the operations from eqs. (1) and (2), is a vector space, which it is customary to denote as $U \oplus V$, i.e the *direct sum* of U and V.

Exercise 2.2. State the axioms that define a *vector space* and show that $U \oplus V$ is indeed a vector space.

It can also be checked that if B_U and B_V are bases for U and V (respectively) with 0_U and 0_V denoting the zero elements of U and V (respectively) then the set

$$B = \{(\boldsymbol{u}, 0_V) : \boldsymbol{u} \in B_U\} \cup \{(0_U, \boldsymbol{v}) : \boldsymbol{v} \in B_V\}$$

is a basis for $U \oplus V$. It follows immediately that $\dim(U \oplus V) = \dim(U) + \dim(V)$.

Example 2.3. Let $U = \mathbb{R}^3$ and $V = \mathcal{M}_{2\times 2}(\mathbb{R})$ be the space of real 2-by-2 matrices. A generic element of $U \oplus V$ is an ordered pair $(\boldsymbol{u}, \boldsymbol{A})$ for a vector $\boldsymbol{u} \in \mathbb{R}^3$ and a matrix $\boldsymbol{A} \in \mathcal{M}_{2\times 2}(\mathbb{R})$. A concrete example would be letting

$$(\boldsymbol{u}, \boldsymbol{A}) = \left(\begin{bmatrix} 3 \\ -1 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}\right) \quad \text{and} \quad (\boldsymbol{v}, \boldsymbol{B}) = \left(\begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 5 & 3 \\ -1 & 2 \end{bmatrix}\right)$$

then we have

$$(\boldsymbol{u}, \boldsymbol{A}) + (\boldsymbol{v}, \boldsymbol{B}) = \left(\begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}\right).$$

A basis for $U \oplus V$ is (*with a slight abuse of notation for what '0' means*)

$$\left\{\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, 0\right), \left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, 0\right), \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, 0\right), \left(0, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right), \left(0, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\right), \left(0, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\right), \left(0, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right)\right\}.$$

We observe that indeed $\dim(U \oplus V) = 7 = \dim(\mathbb{R}^3) + \dim(\mathcal{M}_{2\times 2}(\mathbb{R}))$.

Remark 2.4 (On notation)**.** Elements of $U \oplus V$ can also be written *additively*, that is $\boldsymbol{u} + \boldsymbol{v}$ denotes the vector $(\boldsymbol{u}, \boldsymbol{v}) \in U \oplus V$. This notation is harmless because of Definition 2.1, as it behaves in the expected way. We can then restate the two linearity eqs. (1) and (2) in more natural notation:

$$\boldsymbol{u}_1 + \boldsymbol{v}_1 + \boldsymbol{u}_2 + \boldsymbol{v}_2 = \boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{v}_1 + \boldsymbol{v}_2, \qquad \lambda(\boldsymbol{u} + \boldsymbol{v}) = \lambda\boldsymbol{u} + \lambda\boldsymbol{v},$$

and we note that there is no confusion in the first equality with the two different meanings of + since due to the associativity and commutativity of addition, the four terms can be rearranged in an arbitrary way to give the same result[3]. Importantly, both the bracket notation, $(\boldsymbol{u}, \boldsymbol{v})$, and the additive notation, $\boldsymbol{u} + \boldsymbol{v}$, are used interchangeably in the path signature community.

Definition 2.1 generalizes easily to a finite number of summands: if $U_1, \ldots, U_n$ are vector spaces, the set $U_1 \times \cdots \times U_n$ carries a linear structure given by componentwise addition and multiplication by scalars. The resulting vector space is denoted by $U_1 \oplus \cdots \oplus U_n$. Extending this concept to infinite families requires some care.

[3] Formally, there is a canonical isomorphism between $U \oplus V$ and $V \oplus U$, so that the pairs $(\boldsymbol{u}, \boldsymbol{v})$ and $(\boldsymbol{v}, \boldsymbol{u})$ can be identified.

Definition 2.5. Consider an infinite index set I and let $(U_i : i \in I)$ be a family of vector spaces. The direct sum is defined to be the set of all sequences $(\boldsymbol{u}_i : i \in I)$ such that $\boldsymbol{u}_i \neq 0$ for finitely many indices $i \in I$. Addition and scalar multiplication are defined componentwise. The resulting vector space is denoted by

$$\bigoplus_{i \in I} U_i.$$

The finiteness constraint in this definition means that direct sum is a subset of the Cartesian product of the spaces, that is,

$$\bigoplus_{i \in I} U_i \subseteq \prod_{i \in I} U_i,$$

where $\prod_{i \in I} U_i$ is simply the set of all sequences indexed by I.

For our purposes it will be enough to consider countable families of vector spaces, that is, we will take $I = \mathbb{N}$. In this case, elements of $\bigoplus_{n \in \mathbb{N}} U_n$ may be denoted as

$$(\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \dots)$$

with the convention that there is only a finite number of non-zero elements in the sequence. The Cartesian product also carries the same linear structure and is indeed a vector space, whose elements consist of arbitrary $\mathbb{N}$-indexed sequences, which are still denoted as $(\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \dots)$ where all entries may be non-zero.

It should be noted that in the case that $I \subset \mathbb{N}$ is a finite set, say $I = \{1, \dots, N\}$, both spaces coincide but the inclusion becomes strict as soon as I is countable. In particular, when dealing with finite collections of spaces there is no ambiguity in the notation.

The main application of direct sums in the world of signatures is to decompose a vector space in terms of subspaces of objects sharing similar "shape" properties.

Definition 2.6. A vector space V is said to be *graded* if it can be decomposed as a direct sum:

$$V = \bigoplus_{n \in \mathbb{N}} V_n.$$

The subspace V_n is called the *homogeneous component of degree n*. For $v \in V_n$ we write $|v| = n$ for its degree.

We also note that this definition includes the case of finitely many summands, in which case there is $N \in \mathbb{N}$ such that $V_n = \{0\}$ for all $n > N$.

2.2 Tensor product and Tensors

We have now seen how the *direct product* is one way of equipping the Cartesian product of two vector spaces with a vector space structure. There is another way, the *tensor product*, in many ways similar, but with a structure such that it is compatible with multilinear functions in a way to be made precise later.

Definition 2.7. Let U, V be vector spaces. On the set $U \times V$ define the following *bilinearity* operations: for $\boldsymbol{u}, \boldsymbol{u}_1, \boldsymbol{u}_2 \in U$, $\boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ and $\lambda \in \mathbb{R}$

$$\begin{aligned}
(\boldsymbol{u}_1, \boldsymbol{v}) + (\boldsymbol{u}_2, \boldsymbol{v}) &\coloneqq (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}), \\
(\boldsymbol{u}, \boldsymbol{v}_1) + (\boldsymbol{u}, \boldsymbol{v}_2) &\coloneqq (\boldsymbol{u}, \boldsymbol{v}_1 + \boldsymbol{v}_2), \\
\lambda(\boldsymbol{u}, \boldsymbol{v}) \coloneqq (\lambda \boldsymbol{u}, \boldsymbol{v}) \quad &\text{and}^{4} \quad \lambda(\boldsymbol{u}, \boldsymbol{v}) \coloneqq (\boldsymbol{u}, \lambda \boldsymbol{v}).
\end{aligned}$$

As in the direct sum case (Definition 2.1), it can be verified that $U \times V$ *equipped with the bilinearity operations* is a vector space. We denote the *tensor product* of U and V as $U \otimes V$. For elements of the tensor product space, we write $\boldsymbol{u} \otimes \boldsymbol{v} := (\boldsymbol{u}, \boldsymbol{v}) \in U \otimes V$. By a slight abuse of language we also refer to $\boldsymbol{u} \otimes \boldsymbol{v}$ as the tensor product of the vectors $\boldsymbol{u} \in U$ and $\boldsymbol{v} \in V$. As we will see later in Section 3, this name is justified.

Exercise 2.8. Show that $U \otimes V$ is a vector space (recall Exercise 2.2).

Exercise 2.9. In this exercise we see why the $\otimes$ notation is an intuitive way of writing the tensor product: (a) Rewrite the bilinearity operations of Definition 2.7 using the notation $\boldsymbol{u} \otimes \boldsymbol{v}$ instead of $(\boldsymbol{u}, \boldsymbol{v})$; (b) Expand $(\boldsymbol{u}_1 + \boldsymbol{u}_2) \otimes (\boldsymbol{v}_1 + \boldsymbol{v}_2)$, where of course the addition $\boldsymbol{u}_1 + \boldsymbol{u}_2 \in U$ is simply the addition in U, and the same for V; (c) We have $\lambda \boldsymbol{u} \otimes \lambda \boldsymbol{v} \propto (\boldsymbol{u} \otimes \boldsymbol{v})$. What is the constant of proportionality?

Contrary to the direct sum case, for tensor products its not always possible to write $\boldsymbol{u}_1 \otimes \boldsymbol{v}_1 + \boldsymbol{u}_2 \otimes \boldsymbol{v}_2$ as a single tensor product of a vector in U with a vector in V [5].

Exercise 2.10. Show that $0_U \otimes \boldsymbol{v} = \boldsymbol{u} \otimes 0_V = 0_{U \otimes V}$ for all $\boldsymbol{u} \in U, \boldsymbol{v} \in V$.

In the same vein as the comment offered just before Definition 2.5, the Definition 2.7 admits a straightforward generalization to finitely many vector spaces and extending this concept to infinite families requires some care. In particular, we write

$$U^{\otimes n} := \underbrace{U \otimes U \otimes \cdots \otimes U}_{n \text{ times}}.$$

Proposition 2.11. *For U, V vector spaces with bases B_U, B_V, respectively, the set*

$$B = \{\boldsymbol{u} \otimes \boldsymbol{v} : \boldsymbol{u} \in B_U, \boldsymbol{v} \in B_V\}$$

is a basis for $U \otimes V$. It follows that $\dim(U \otimes V) = \dim(U) \cdot \dim(V)$.

We can tell immediately by the dimension of the spaces that the direct product and tensor product produce fundamentally different vector structures on the same set[6]. What distinguishes the tensor product vector space from all other possible linear structures is the following *universal property*.

Theorem 2.12. *Let U, V, W be vector spaces, and let $f : U \times V \to W$ be a bilinear map. That is, f satisfies for all $\boldsymbol{u}, \boldsymbol{u}_1, \boldsymbol{u}_2 \in U$, $\boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ and $\lambda \in \mathbb{R}$*

$$f(\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}) = f(\boldsymbol{u}_1, \boldsymbol{v}) + f(\boldsymbol{u}_2, \boldsymbol{v}), \qquad f(\boldsymbol{u}, \boldsymbol{v}_1 + \boldsymbol{v}_2) = f(\boldsymbol{u}, \boldsymbol{v}_1) + f(\boldsymbol{u}, \boldsymbol{v}_2),$$
$$\textit{and} \quad f(\lambda \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{u}, \lambda \boldsymbol{v}) = \lambda f(\boldsymbol{u}, \boldsymbol{v}).$$

Then, there exists a unique ***linear*** *function $\hat{f} : U \otimes V \to W$ such that $f(\boldsymbol{u}, \boldsymbol{v}) = \hat{f}(\boldsymbol{u} \otimes \boldsymbol{v})$.*

[4] This last element of the definition is actually *imposing* the vectors $(\lambda \boldsymbol{u}, \boldsymbol{v})$ and $(\boldsymbol{u}, \lambda \boldsymbol{v})$ to be equal in $U \otimes V$. This could be formalized by the use of quotient spaces, but doing such would involve a level of additional complexity not really necessary at the moment. (This also takes care of the apparent non-uniqueness of $0_{U \otimes V}$ hinted at in Exercise 2.10.)

[5] As an aside, it is this property of tensor product spaces that mathematically captures the phenomenon of entanglement of quantum particles. A quantum state like $|\varphi\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is entangled as it cannot be written as $\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix}$ for any scalars a, b, c, d. This entangled state says "I have two particles, and their spins are always opposite, but I cannot know which one is spin-up, and which one is spin-down".

[6] Recall $\dim(U \oplus V) = \dim(U) + \dim(V)$, whereas $\dim(U \otimes V) = \dim(U) \cdot \dim(V)$.

We say that bilinear functions *factor through* the tensor product. In fact, the tensor product is characterized by this property, in the sense that any other vector space Z equipped with a map $\underline{\otimes}: U \times V \to Z$ factorizing bilinear functions must be isomorphic to $U \otimes V$. In other words, the tensor product is the unique (up to isomorphism) vector space having this property. For the sake of simplicity we will omit the proof of this result, but refer the interested reader to the classical texts [2, 3]. At first sight bilinear maps are not quite as powerful as linear maps, nonetheless, the universal property fixes things as it allows to write the bilinear map as a linear map of the tensor product and thus recovers the neat results of linear maps that were not available (at the cost of using tensor products).

The next example highlights the difference between direct sums and tensor products.

Example 2.13 (Direct sums $\oplus$ Vs tensor products $\otimes$)**.** Let $U = V = \mathbb{R}$. The direct sum $U \oplus V$ satisfies $U \oplus V \cong \mathbb{R}^2$. Indeed, elements of $U \oplus V$ are ordered pairs of real numbers with component-wise addition and scalar multiplication. Moreover, a basis for $U \oplus V$ is $\{(1,0),(0,1)\}$ which is the canonical basis of $\mathbb{R}^2$, so $\dim(U \oplus V) = 2$.

On the contrary, we will show now that $U \otimes V$ satisfies $U \otimes V \cong \mathbb{R}$ which is *obviously not* $\mathbb{R}^2 \cong U \oplus V$. Consider the map $\varphi: U \times V \to \mathbb{R}$ given by $\varphi(x,y) = xy$; this map $\varphi(x,y)$ is clearly bilinear. By the universal property, there exists a unique map $\hat{\varphi}: U \otimes V \to \mathbb{R}$, given by $\hat{\varphi}(x \otimes y) = xy$. The map φ is injective since the equation $\hat{\varphi}(x \otimes y) = 0$ implies that either $x = 0$ or $y = 0$; in any case $x \otimes y = 0$ by Exercise 2.10. Finally, if $\lambda \in \mathbb{R}$ then $\hat{\varphi}(\lambda \otimes 1) = \lambda$ so that $\hat{\varphi}$ is surjective. In particular, $\mathbb{R} \otimes \mathbb{R}$ is spanned by the vector $1 \otimes 1$ so that $\dim(U \otimes V) = 1$.

Exercise 2.14. Show that if U is any vector space, then $U \otimes \mathbb{R}$ and $\mathbb{R} \otimes U$ are isomorphic to U. (*Hint: generalize Example 2.13.*)

The word *tensor* has many different meanings across different fields [2,3,9]. We will mostly be interested in the case where we are taking tensor products of a finite number of finite-dimensional vector spaces, represented as $\mathbb{R}^d$ for some integer $d \geq 1$. In this setting, tensors may be represented in a simpler, more concrete way, by working with canonical bases. It is at at this specific juncture (assuming a basis) that it is intuitive to define a tensor as a multidimensional array – see Remark 2.16.

Definition 2.15 (Tensor: order and shape)**.** Take $n \geq 1$, set $d_1, \ldots, d_n \geq 1$ and take the associated vector spaces $\mathbb{R}^{d_j}$ for $j = 1, \ldots, n$. An *order n tensor* of *shape* $(d_1, \ldots, d_n)$ is an element of the tensor product $\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_n}$.

It should be clear that elements of, say, $\mathbb{R}^2 \otimes \mathbb{R}^4$, $\mathbb{R}^4 \otimes \mathbb{R}^2$, and $\mathbb{R}^3 \otimes \mathbb{R}^3$, are all order 2 tensors, but their *shapes* are all very different – and, just like matrices, they cannot be added together as the shapes do not match.

Remark 2.16 (Basis, vectors and their components, and some notation)**.** Denote by $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$ the canonical basis of $\mathbb{R}^d$ for some $d \geq 1$. We have seen that the set

$$\left\{\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_n} : i_j \in \{1, \ldots, d_j\} \text{ for all } j = 1, \ldots, n\right\}$$

is a basis for $\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_n}$, which we call the canonical basis. In particular, an order n tensor $\boldsymbol{T}$ is determined by the n-dimensional array of its coefficients in this basis:

$$\boldsymbol{T} = \sum_{i_1=1}^{d_1} \cdots \sum_{i_n=1}^{d_n} \boldsymbol{T}^{i_1 \cdots i_n} \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_n}.$$

As long as we keep this in mind, the assignment $\boldsymbol{T} \mapsto (\boldsymbol{T}^{i_1 \cdots i_n})$ defines a one-to-one correspondence between elements of the tensor product $\boldsymbol{T} \in \mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_n}$ and multidimensional arrays $(\boldsymbol{T}^{i_1 \cdots i_n}) \in \mathbb{R}^{d_1 \times \cdots \times d_n}$. *We remark once again that this isomorphism depends on the fixing of a basis and is, in general, not canonical.*

Thus, **notation wise**, we shall refer to tensors using either a symbol (i.e. $\boldsymbol{T}$) or in component notation (i.e. $\boldsymbol{T}^{ijk}$ – using superscript notation for its components); for multiple tensors or vectors we use subscript notation, i.e., $\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots$ or $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$.

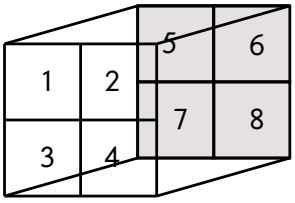
For example, $\boldsymbol{C}^{ij}$, $\boldsymbol{T}^{ijk}$ and $\boldsymbol{Q}^{ijkp}$ refer to the components of tensors $\boldsymbol{C}$, $\boldsymbol{T}$, and $\boldsymbol{Q}$ of order 2, 3, and 4 respectively. In particular, the order 1 tensors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ constitute the canonical basis of $\mathbb{R}^d$. The ith basis vector $\boldsymbol{e}_i$ has components given in the canonical basis by

$$\boldsymbol{e}_i^j = \begin{cases} 1 & j = i \\ 0 & \text{else} \end{cases} \qquad \text{for } j = 1, \ldots, d.$$

Example 2.17. We see that tensors of order 1 and 2 can be identified with column vectors and matrices, respectively. A scalar is by convention an order 0 tensor. For example, $\boldsymbol{u}$, $\boldsymbol{A}$, and $\boldsymbol{T}$ are tensors of order 1, 2, and 3 respectively:

$$\boldsymbol{u} := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3, \quad \boldsymbol{A} := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \in \mathbb{R}^2 \otimes \mathbb{R}^2 = (\mathbb{R}^2)^{\otimes 2} \cong \mathbb{R}^{2\times 2},$$

$$\boldsymbol{T} := \quad \in (\mathbb{R}^2)^{\otimes 3} \cong \mathbb{R}^{2\times 2\times 2}. \tag{3}$$



Remark 2.18. The tensor is an *intrinsic* object, in the sense that tensors do not depend on any choice of basis. From Remark 2.16, we see that a tensor can be uniquely described by a multidimensional array of numbers, but this is only true once we have fixed a basis. For many applications, it is possible, and practical, to think of tensors only in the terms of their components in a particular (e.g. the canonical) basis. Nonetheless, we encourage the reader to be mindful with the language and recognize when they are being loose with the concepts. This is exactly the same idea to how we can think of linear transformations from $\mathbb{R}^n$ to $\mathbb{R}^n$ in terms of an n-by-n square matrix, once we have fixed a basis. For example, the linear map $(x_1, x_2) \mapsto (x_1 + x_2, x_1 - x_2)$ can be represented by the matrix $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ in the standard basis, but in the eigenbasis, it is represented by the diagonal matrix $\begin{pmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{pmatrix}$. The characterizing property of tensors is that given a change of basis, we immediately know how the coordinate representation transforms. In this example we can go from the first to the second matrix representation by multiplication with an invertible matrix (diagonalization).

Writing explicit examples for the tensor product quickly becomes cumbersome, nevertheless we offer a few simple examples.

Example 2.19 (Tensor multiplication)**.** Let $\boldsymbol{u} \in \mathbb{R}^2$, $\boldsymbol{v} \in \mathbb{R}^3$ be vectors, that is, order 1 tensors of shapes (2) and (3), respectively. By definition, their tensor product is an order 2 tensor of shape $(2, 3)$: $\boldsymbol{u} \otimes \boldsymbol{v} \in \mathbb{R}^2 \otimes \mathbb{R}^3$. In the canonical basis they are represented by 2 and 3 coefficients, respectively. Namely

$$\boldsymbol{u} = \boldsymbol{u}^1 \boldsymbol{e}_1 + \boldsymbol{u}^2 \boldsymbol{e}_2, \quad \boldsymbol{v} = \boldsymbol{v}^1 \boldsymbol{e}_1 + \boldsymbol{v}^2 \boldsymbol{e}_2 + \boldsymbol{v}^3 \boldsymbol{e}_3,$$

or in the more traditional column vector notation,

$$\boldsymbol{u} = \boldsymbol{u}^1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \boldsymbol{u}^2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}^1 \\ \boldsymbol{v}^2 \\ \boldsymbol{v}^3 \end{bmatrix}.$$

By using the bilinearity of the tensor product (Definition 2.7) we may obtain the coordinates of $\boldsymbol{u} \otimes \boldsymbol{v}$ in the canonical basis. Indeed, recalling that the components of both vectors are scalars,

$$\begin{aligned} \boldsymbol{u} \otimes \boldsymbol{v} &= \left(\boldsymbol{u}^1 \boldsymbol{e}_1 + \boldsymbol{u}^2 \boldsymbol{e}_2\right) \otimes \left(\boldsymbol{v}^1 \boldsymbol{e}_1 + \boldsymbol{v}^2 \boldsymbol{e}_2 + \boldsymbol{v}^3 \boldsymbol{e}_3\right) \\ &= \boldsymbol{u}^1 \boldsymbol{v}^1 \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{u}^1 \boldsymbol{v}^2 \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 + \boldsymbol{u}^1 \boldsymbol{v}^3 \boldsymbol{e}_1 \otimes \boldsymbol{e}_3 \\ &\quad + \boldsymbol{u}^2 \boldsymbol{v}^1 \boldsymbol{e}_2 \otimes \boldsymbol{e}_1 + \boldsymbol{u}^2 \boldsymbol{v}^2 \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 + \boldsymbol{u}^2 \boldsymbol{v}^3 \boldsymbol{e}_2 \otimes \boldsymbol{e}_3. \end{aligned} \tag{4}$$

Thus, in the canonical basis the order 2 tensor $\boldsymbol{u} \otimes \boldsymbol{v}$ has components given by $(\boldsymbol{u} \otimes \boldsymbol{v})^{ij} = \boldsymbol{u}^i \boldsymbol{v}^j$. Identifying the canonical basis of order 2 tensors with matrices (i.e., $\boldsymbol{e}_i \otimes \boldsymbol{e}_j$ forming the standard 2-by-2 matrix basis), we may write

$$\boldsymbol{u} \otimes \boldsymbol{v} = \begin{bmatrix} \boldsymbol{u}^1 \boldsymbol{v}^1 & \boldsymbol{u}^1 \boldsymbol{v}^2 & \boldsymbol{u}^1 \boldsymbol{v}^3 \\ \boldsymbol{u}^2 \boldsymbol{v}^1 & \boldsymbol{u}^2 \boldsymbol{v}^2 & \boldsymbol{u}^2 \boldsymbol{v}^3 \end{bmatrix} \quad \text{and likewise} \quad \boldsymbol{v} \otimes \boldsymbol{u} = \begin{bmatrix} \boldsymbol{v}^1 \boldsymbol{u}^1 & \boldsymbol{v}^1 \boldsymbol{u}^2 \\ \boldsymbol{v}^2 \boldsymbol{u}^1 & \boldsymbol{v}^2 \boldsymbol{u}^2 \\ \boldsymbol{v}^3 \boldsymbol{u}^1 & \boldsymbol{v}^3 \boldsymbol{u}^2 \end{bmatrix} \in \mathbb{R}^3 \otimes \mathbb{R}^2.$$

Consider now the matrix

$$\boldsymbol{A} \coloneqq \begin{bmatrix} \boldsymbol{A}^{11} & \boldsymbol{A}^{12} \\ \boldsymbol{A}^{21} & \boldsymbol{A}^{22} \end{bmatrix} \in \mathbb{R}^2 \otimes \mathbb{R}^2.$$

Where, as before, the entries in the matrix notation corresponds to the coordinates in the canonical order 2 tensor basis: $\boldsymbol{A} = \boldsymbol{A}^{11} \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{A}^{12} \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 + \boldsymbol{A}^{21} \boldsymbol{e}_2 \otimes \boldsymbol{e}_1 + \boldsymbol{A}^{22} \boldsymbol{e}_2 \otimes \boldsymbol{e}_2$. Note that although $\boldsymbol{A}$ is an element of $\mathbb{R}^2 \otimes \mathbb{R}^2$, it does not necessarily mean that it can be written as $\boldsymbol{u} \otimes \boldsymbol{v}$ for some $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^2$. That is, the components $\boldsymbol{A}^{ij}$ are not necessarily of the form $\boldsymbol{A}^{ij} = \boldsymbol{u}^i \boldsymbol{v}^j$ for some vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of the appropriate dimensions. In the cases where it *is* possible to find such a decomposition, we say that $\boldsymbol{A}$ is a *rank 1 tensor*. On the other hand, $\boldsymbol{A}$ can always be written as a linear combination of sums of tensor products of some $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^2$, that is, sums of rank 1 tensors.

Continuing with the example, we may compute

$$\boldsymbol{u} \otimes \boldsymbol{A} = \begin{array}{|cc|cc|} \multicolumn{2}{c}{} & \multicolumn{2}{c}{} \\ \multicolumn{2}{c}{} & \multicolumn{2}{c}{} \\ \hline \multicolumn{2}{|c|}{\boldsymbol{u}^1 \boldsymbol{A}^{11}} & \multicolumn{2}{c|}{\boldsymbol{u}^1 \boldsymbol{A}^{12}} \\ \multicolumn{2}{|c|}{\boldsymbol{u}^1 \boldsymbol{A}^{21}} & \multicolumn{2}{c|}{\boldsymbol{u}^1 \boldsymbol{A}^{22}} \\ \hline \end{array} \begin{array}{|cc|} \multicolumn{2}{c}{} \\ \multicolumn{2}{c}{} \\ \hline \boldsymbol{u}^2 \boldsymbol{A}^{11} & \boldsymbol{u}^2 \boldsymbol{A}^{12} \\ \boldsymbol{u}^2 \boldsymbol{A}^{21} & \boldsymbol{u}^2 \boldsymbol{A}^{22} \\ \hline \end{array} \in \mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2 = (\mathbb{R}^2)^{\otimes 3},$$

where the expression on the right-hand side is a matrix-like notation for organizing the components of the order 3 tensor $\boldsymbol{u} \otimes \boldsymbol{A}$.

We will later see, in Section 3, that the computation performed in eq. (4) makes use of a larger structure. The tensor product can be thought of as a non-commutative analogue of polynomial multiplication, in the sense that it is an associative and bilinear operation. Note that it is, however, not commutative as the results of $\boldsymbol{u} \otimes \boldsymbol{v}$ and $\boldsymbol{v} \otimes \boldsymbol{u}$ are order 2 tensors of different shapes, at least in this example. In general, the results may differ even thought the shapes match.

Example 2.20 (Connecting to Signatures of [1])**.** Suppose $\boldsymbol{x}\colon [0,1] \longrightarrow \mathbb{R}^d$ is a smooth vector-valued path, which simply means, e.g., that for each $t \in [0,1]$ we may write $\boldsymbol{x}_t = (\boldsymbol{x}_t^1, \dots, \boldsymbol{x}_t^d) \in \mathbb{R}^d$ in the canonical basis. In particular, for each $t \in [0,1]$, $\boldsymbol{x}_t$ is a tensor of order 1.

We may use the tensor product to compactly write the collection of iterated integrals of $\boldsymbol{x}$. That is,

$$\int_0^t \int_0^s \mathrm{d}\boldsymbol{x}_u \otimes \mathrm{d}\boldsymbol{x}_s \in \mathbb{R}^d \otimes \mathbb{R}^d$$

is an order 2 tensor (i.e. a d-by-d matrix) with components (for $\boldsymbol{x}$ "sufficiently nice")

$$\left(\int_0^t \int_0^s \mathrm{d}\boldsymbol{x}_u \otimes \mathrm{d}\boldsymbol{x}_s\right)^{ij} = \int_0^t \int_0^s \dot{x}_u^i \dot{x}_s^j \,\mathrm{d}u\mathrm{d}s.$$

Before delving into tensors properties we can go back to "Question 45: What is a tensor?" of the "100 Questions: A Mathematical Conventions Survey". We hope to have convinced the reader that a tensor is nothing other than "an element of a tensor product of vector spaces".

3 A little bit more on tensors: The tensor algebra

We now venture into a few additional properties of tensors.

Definition 3.1. An associative algebra is a vector space A equipped with a bilinear map $m\colon A\times A \longrightarrow A$, called *product*, satisfying the associativity condition

$$m\big(m(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{z}\big) = m\big(\boldsymbol{x}, m(\boldsymbol{y},\boldsymbol{z})\big).$$

The universal property of the tensor product (see Theorem 2.12) yields that equivalently, it may be represented by a *linear* map $m\colon A \otimes A \longrightarrow A$, which is the one we will use from now on.

We say that an algebra A is *unital* if it has a distinguished element $1_A \in A$, called the *unit*, satisfying for all $\boldsymbol{x} \in A$

$$m(1_A \otimes \boldsymbol{x}) = \boldsymbol{x} = m(\boldsymbol{x} \otimes 1_A).$$

It is customary to write the product, denoted $\cdot_A$, of two elements of A using infix notation[7], that is, $\boldsymbol{x} \cdot_A \boldsymbol{y} := m(\boldsymbol{x} \otimes \boldsymbol{y})$. Oftentimes, when no confusion can arise, we write simply $\boldsymbol{x} \cdot \boldsymbol{y}$ or even omit the symbol completely and just write $\boldsymbol{x}\boldsymbol{y}$ instead. As we will mostly work with *unital associative algebras*, from here on we simply write *algebra*. In this notation, the condition for $(A, \cdot_A)$ to be an algebra can be written as

$$(\boldsymbol{x} \cdot_A \boldsymbol{y}) \cdot_A \boldsymbol{z} = \boldsymbol{x} \cdot_A (\boldsymbol{y} \cdot_A \boldsymbol{z})$$

for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in A$ and the unit satisfies $1 \cdot_A \boldsymbol{x} = \boldsymbol{x} \cdot_A 1 = \boldsymbol{x}$ for all $\boldsymbol{x} \in A$. We note that bilinearity of the product translates into the distributivity of $\cdot_A$ over $+$, e.g.,

$$(\boldsymbol{x} + \boldsymbol{y}) \cdot_A \boldsymbol{z} = \boldsymbol{x} \cdot_A \boldsymbol{z} + \boldsymbol{y} \cdot_A \boldsymbol{z}, \quad \lambda(\boldsymbol{x} \cdot_A \boldsymbol{y}) = (\lambda\boldsymbol{x}) \cdot_A \boldsymbol{y} = \boldsymbol{x} \cdot_A (\lambda\boldsymbol{y}),$$

and so on. We also remark that we *do not* require the product to be commutative, that is, we do not enforce that $\boldsymbol{x} \cdot_A \boldsymbol{y} = \boldsymbol{y} \cdot_A \boldsymbol{x}$ for every $\boldsymbol{x}, \boldsymbol{y} \in A$, although this may hold for some pairs of elements. In case this identity does hold for every $\boldsymbol{x}, \boldsymbol{y} \in A$ we say A is commutative (or Abelian).

[7] Infix notation is a way of writing mathematical (and logic) expressions where operators are placed between the operands they act upon. This is the most familiar notation to us humans and matches how we (humans) interpret math expressions.

Example 3.2. Let $\mathcal{M}_{n\times n}(\mathbb{R})$ be the space of n-by-n square matrices with real entries with its usual vector space structure (entry-wise addition and scalar multiplication). The matrix product $\boldsymbol{A}\cdot\boldsymbol{B} \coloneqq \boldsymbol{AB}$ equips A with the structure of a (non-commutative) associative algebra with unit $1_A = \boldsymbol{I}_n$, the n-by-n identity matrix.

Example 3.3. Denote by $\mathbb{R}[x]$ the space of polynomials in a single variable x, and for polynomials $\boldsymbol{p}(x)$ and $\boldsymbol{q}(x)$ define the multiplication rule by

$$(\boldsymbol{p}\cdot\boldsymbol{q})(x) \coloneqq \boldsymbol{p}(x)\boldsymbol{q}(x).$$

It is clear that this multiplication is bilinear in $\boldsymbol{p}$ and $\boldsymbol{q}$ and satisfies the associativity condition. The unit for this product is the constant polynomial $1_A(x) = 1$. For instance

$$(x^3+1)\cdot(x^2+x) = x^5 + x^4 + x^2 + x.$$

In fact, since the monomials $\{x^n : n \geq 1\}$ form a linear basis for A, the four terms on the right-hand side correspond simply to $x^3\cdot x^2$, $x^3\cdot x$ and so on, where we use the bilinearity of the product. This is an example of a commutative algebra.

Example 3.4. Many structures with which one is already very familiar are just algebras in disguise, Table 1 gives a few examples.

Vector Space	Bilinear Operator	Associative	Commutative	Unitary
$\mathbb{C}$	Complex product	Yes	Yes	Yes
$\mathbb{R}^3$	Vector cross product: $\vec{a}\times\vec{b}$	No	No	No
$\mathbb{R}[x]$	Multiplication	Yes	Yes	Yes
$\mathcal{M}_{n\times n}(\mathbb{R})$	Matrix multiplication	Yes	No	Yes

Table 1: Various examples of algebras. Some authors use the term algebra to refer to a vector space equipped with *any* bilinear operation, not necessarily associative. In that sense, the cross product is an algebra that is not associative.

The prime example of an algebra in relation to signatures is the *tensor algebra*.

Definition 3.5. Let V be a finite-dimensional vector space. The *tensor algebra over* V is the vector space

$$T(V) \coloneqq \bigoplus_{n\geq 0} V^{\otimes n} \quad \text{with} \quad V^{\otimes 0} \cong \mathbb{R}\mathbf{1}.$$

The product is simply the tensor product, and its unit is the vector $\mathbf{1}$ spanning $V^{\otimes 0}$.

The tensor algebra is therefore a *graded vector space* in the sense of Definition 2.6, where order n tensors are placed in degree n. We stress the fact (see Definition 2.5) that elements of $T(V)$ are *finite* sequences of tensors of arbitrary order. For this reason, vectors in $T(V)$ are usually called tensor (or non-commutative) polynomials. Later on we will construct the space of tensor series, which are infinite sequences.

Remark 3.6. When $V = \mathbb{R}^d$ the product can be written more explicitly in terms of the canonical basis $\{\boldsymbol{e}_1,\dots,\boldsymbol{e}_d\}$. Introducing the *word notation*, recall [1, Example 1.5] $\boldsymbol{e}_{i_1\cdots i_n} \coloneqq \boldsymbol{e}_{i_1}\otimes\cdots\otimes\boldsymbol{e}_{i_n} \in (\mathbb{R}^d)^{\otimes n}$ for $(i_1,\dots,i_n)\in\{1,\dots,d\}^n$, the product (denoted for now by $\cdot_{T(V)}$) is then defined as

$$\boldsymbol{e}_{i_1\cdots i_n}\cdot_{T(V)}\boldsymbol{e}_{j_1\cdots j_m} = \boldsymbol{e}_{i_1\cdots i_n j_1\cdots j_m}.$$

For this reason it is commonly known as the *concatenation product*. In this case, it corresponds to the product introduced in Definition 2.7. Common notations for $\boldsymbol{x}\cdot_{T(V)}\boldsymbol{y}$ include $\boldsymbol{x}\otimes\boldsymbol{y}$ and $\boldsymbol{xy}$.

Theorem 3.7. *The tensor algebra enjoys the following universal property: given any algebra A and any* linear *map $f : V \to A$, there exists a unique map $\hat{f} : T(V) \to A$, such that $f(\boldsymbol{u} \otimes \boldsymbol{v}) = f(\boldsymbol{u}) \cdot_A f(\boldsymbol{v})$ for all $\boldsymbol{u}, \boldsymbol{v} \in T(V)$.*

As is the case with the tensor product, this property actually characterizes the tensor algebra in the sense that any other algebra satisfying this property is necessarily isomorphic to $T(V)$ for some vector space V.

We note that even though V is finite-dimensional, $T(V)$ is always infinite dimensional since, owing to Proposition 2.11,

$$\dim V^{\otimes n} = (\dim V)^n.$$

For this reason, while the tensor algebra is a neat theoretical construction, it is not very useful for practical purposes. There are a couple of ways of obtaining finite-dimensional versions of $T(V)$ which preserve its structure. The most common in signature applications is *truncation*. The basic idea is that we want to preserve "low order" information while still retaining the algebra structure, where the meaning of "order" is in the sense of tensor level. Luckily, the straightforward idea of just discarding high-order information works, with the caveat that the product has to be slightly modified.

Definition 3.8. Given $N \geq 1$, the *level-N truncated tensor algebra* is the finite-dimensional graded vector space (recall Definition 2.6)

$$T^N(V) := \bigoplus_{n=0}^{N} V^{\otimes n} \quad \text{with product} \quad \boldsymbol{x} \cdot_N \boldsymbol{y} = \begin{cases} \boldsymbol{x} \otimes \boldsymbol{y} & \text{if } |\boldsymbol{x}| + |\boldsymbol{y}| \leq N \\ 0 & \text{else .} \end{cases}.$$

Following from Definition 2.6, we note that in particular every element of $T^N(V)$ can be written as a sequence of homogeneous elements, that is every $\boldsymbol{v} \in T^N(V)$ is of the form $\boldsymbol{v} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N)$ with $\boldsymbol{v}_n \in V^{\otimes n}$ (with some of them eventually zero). Hence, the product $\cdot_N$ is well-defined for all $\boldsymbol{x}, \boldsymbol{y} \in T^N(V)$ and not just for homogeneous tensors – elements in $T^N(V)$ are thus finite sequence of tensors of order up to N, with componentwise addition and multiplication by scalars.

It can be checked that $T^N(V)$ is an algebra[8] and

$$\dim T^N(V) = \frac{d^{N+1} - 1}{d - 1} \quad \text{where } \dim V = d.$$

Example 3.9. Let us take $N = 2$ and $V = \mathbb{R}^d$. The space $T^2(V) \cong \mathbb{R}\mathbf{1} \oplus \mathbb{R}^d \oplus (\mathbb{R}^d)^{\otimes 2}$ consists of elements of the form $(a, \boldsymbol{x}, \boldsymbol{A})$ with $a \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{A} \in \mathcal{M}_{d\times d}(\mathbb{R})$ [9].

The product reads

$$(a, \boldsymbol{x}, \boldsymbol{A}) \otimes (a', \boldsymbol{x}', \boldsymbol{A}') = (aa', a\boldsymbol{x}' + a'\boldsymbol{x}, a'\boldsymbol{A} + a\boldsymbol{A}' + \boldsymbol{x} \otimes \boldsymbol{x}').$$

We remark that this product is not commutative, meaning that in general the above expression will be different from that of $(a', \boldsymbol{x}', \boldsymbol{A}') \otimes (a, \boldsymbol{x}, \boldsymbol{A})$.

Exercise 3.10. Let $N = 2$ and $V = \mathbb{R}^d$. Show that an element $(a, \boldsymbol{x}, \boldsymbol{A}) \in T^2(V)$ is invertible if and only if $a \neq 0$, and compute its inverse.

[8] Technically speaking $T^N(V)$ is a quotient of $T(V)$ by a two-sided ideal.

[9] Note that we are tacitly identifying real valued 2 tensors (of shape $(2, 2)$) with real valued 2-by-2 matrices – looking back at Remark 2.16 and Examples 2.17 and 2.19, we have implicitly made the assumption of working with the canonical basis of $\mathbb{R}^2$.

Definition 3.11. The *extended tensor algebra* is the direct product

$$T((V)) \coloneqq \prod_{n=0}^{\infty} V^{\otimes n}.$$

We identify $T((V))$ with the space of infinite sequences $\boldsymbol{u} = (\boldsymbol{u}_0, \boldsymbol{u}_1, \dots)$ with $\boldsymbol{u}_0 \in \mathbb{R}$, $\boldsymbol{u}_1 \in V$, and so on. The product is induced by the product on $T(V)$ and is given, for $\boldsymbol{u} = (\boldsymbol{u}_0, \boldsymbol{u}_1, \dots)$ and $\boldsymbol{v} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \dots)$, by $\boldsymbol{u}\boldsymbol{v} = \boldsymbol{w} = (\boldsymbol{w}_0, \boldsymbol{w}_1, \dots)$ where

$$\boldsymbol{w}_n = \sum_{k=0}^{n} \boldsymbol{u}_k \otimes \boldsymbol{v}_{n-k} \in (V^*)^{\otimes n}.$$

This product mimics polynomial multiplication and is sometimes called the *Cauchy product* for this reason. Since this space contains arbitrarily long sequences of tensors, its elements are commonly called *tensor series*. We note that the tensor algebra $T(V)$ is a strict subspace of $T((V))$.

For each integer $N \geq 1$ there is a canonical projection $\pi_N \colon T((V)) \to T^N(V)$, preserving multiplication, given simply by discarding tensors of degree greater than N, that is,

$$\pi_N(\boldsymbol{u}_0, \boldsymbol{u}_1, \dots) = (\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_N).$$

In the realm of signatures, this projection is used to produce finite-dimensional versions of the signature (see Example 3.12 just below) that are suitable for its representation in a computer.

Example 3.12. Recall Example 2.20. Our prime example of an element in $T\left(\left(\mathbb{R}^d\right)\right)$ is the signature of a smooth $\mathbb{R}^d$-valued path $\boldsymbol{x} \colon [0,1] \to \mathbb{R}^d$. Its signature over the interval $[s,t] \subseteq [0,1]$, denoted by $S(\boldsymbol{x})_{s,t}$, is the tensor series of iterated integrals:

$$S(\boldsymbol{x})_{s,t} \coloneqq \left(1, \int_s^t \mathrm{d}\boldsymbol{x}_u, \int_s^t \int_s^{u_2} \mathrm{d}\boldsymbol{x}_{u_1} \otimes \mathrm{d}\boldsymbol{x}_{u_2}, \dots\right).$$

Projecting to the level-2 truncated tensor algebra we get

$$\pi_2 S(\boldsymbol{x})_{s,t} = \left(1, \int_s^t \mathrm{d}\boldsymbol{x}_u, \int_s^t \int_s^{u_2} \mathrm{d}\boldsymbol{x}_{u_1} \otimes \mathrm{d}\boldsymbol{x}_{u_2}\right).$$

4 Solutions to Exercises

Solution 4.1 (To Exercise 2.2)**.** We must check that the operations satisfy the axioms of a vector space, that is, that $+$ is associative and commutative, and that scalar multiplication distributes over $+$. Let $(\boldsymbol{u}_1, \boldsymbol{v}_1), (\boldsymbol{u}_2, \boldsymbol{v}_2) \in U \oplus V$. Then

$$(\boldsymbol{u}_1, \boldsymbol{v}_1) + (\boldsymbol{u}_2, \boldsymbol{v}_2) = (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}_1 + \boldsymbol{v}_2) = (\boldsymbol{u}_2 + \boldsymbol{u}_1, \boldsymbol{v}_2 + \boldsymbol{v}_1) = (\boldsymbol{u}_2, \boldsymbol{v}_2) + (\boldsymbol{u}_1, \boldsymbol{v}_1).$$

Moreover, if $(\boldsymbol{u}_3, \boldsymbol{v}_3) \in U \oplus V$ then

$$\begin{aligned}((\boldsymbol{u}_1, \boldsymbol{v}_1) + (\boldsymbol{u}_2, \boldsymbol{v}_2)) + (\boldsymbol{u}_3, \boldsymbol{v}_3) &= (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}_1 + \boldsymbol{v}_2) + (\boldsymbol{u}_3, \boldsymbol{v}_3) = (\boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{u}_3, \boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3) \\ &= (\boldsymbol{u}_1, \boldsymbol{v}_1) + (\boldsymbol{u}_2 + \boldsymbol{u}_3, \boldsymbol{v}_2 + \boldsymbol{v}_3) \\ &= (\boldsymbol{u}_1, \boldsymbol{v}_1) + ((\boldsymbol{u}_2, \boldsymbol{v}_2) + (\boldsymbol{u}_3, \boldsymbol{v}_3)).\end{aligned}$$

Likewise, for any $\lambda \in \mathbb{R}$ we have

$$\begin{aligned}\lambda((\boldsymbol{u}_1, \boldsymbol{v}_1) + (\boldsymbol{u}_2, \boldsymbol{v}_2)) &= \lambda(\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}_1 + \boldsymbol{v}_2) \\ &= (\lambda(\boldsymbol{u}_1 + \boldsymbol{u}_2), \lambda(\boldsymbol{v}_1 + \boldsymbol{v}_2)) = (\lambda\boldsymbol{u}_1 + \lambda\boldsymbol{u}_2, \lambda\boldsymbol{v}_1 + \lambda\boldsymbol{v}_2) \\ &= (\lambda\boldsymbol{u}_1, \lambda\boldsymbol{v}_1) + (\lambda\boldsymbol{u}_2, \lambda\boldsymbol{v}_2) = \lambda(\boldsymbol{u}_1, \boldsymbol{v}_1) + \lambda(\boldsymbol{u}_2, \boldsymbol{v}_2).\end{aligned}$$

We have used throughout that U and V are vector spaces.

Additive inverses are given simply by $-(\boldsymbol{u}, \boldsymbol{v}) = (-\boldsymbol{u}, -\boldsymbol{v})$ while the neutral element is $0_{U\oplus V} = (0_U, 0_V)$.

Solution 4.2 (To Exercise 2.8)**.** We have to check that the operations satisfy the axioms of a vector space. Since addition is defined symmetrically, we only check one side. Let $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ and $\boldsymbol{v} \in V$. Then

$$(\boldsymbol{u}_1, \boldsymbol{v}) + (\boldsymbol{u}_2, \boldsymbol{v}) = (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}) = (\boldsymbol{u}_2 + \boldsymbol{u}_1, \boldsymbol{v}) = (\boldsymbol{u}_2, \boldsymbol{v}) + (\boldsymbol{u}_1, \boldsymbol{v}).$$

Associativity follows in a similar way:

$$\begin{aligned}((\boldsymbol{u}_1, \boldsymbol{v}) + (\boldsymbol{u}_2, \boldsymbol{v})) + (\boldsymbol{u}_3, \boldsymbol{v}) &= (\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}) + (\boldsymbol{u}_3, \boldsymbol{v}) \\ &= \big((\boldsymbol{u}_1 + \boldsymbol{u}_2) + \boldsymbol{u}_3, \boldsymbol{v}\big) = \big(\boldsymbol{u}_1 + (\boldsymbol{u}_2 + \boldsymbol{u}_3), \boldsymbol{v}\big) \\ &= (\boldsymbol{u}_1, \boldsymbol{v}) + (\boldsymbol{u}_2 + \boldsymbol{u}_3, \boldsymbol{v}) = (\boldsymbol{u}_1, \boldsymbol{v}) + ((\boldsymbol{u}_2, \boldsymbol{v}) + (\boldsymbol{u}_3, \boldsymbol{v})).\end{aligned}$$

Now, for any $\lambda \in \mathbb{R}$ we see that (using throughout that U and V are vector spaces)

$$\begin{aligned}\lambda((\boldsymbol{u}_1, \boldsymbol{v}) + (\boldsymbol{u}_2, \boldsymbol{v})) &= \lambda(\boldsymbol{u}_1 + \boldsymbol{u}_2, \boldsymbol{v}) = (\lambda(\boldsymbol{u}_1 + \boldsymbol{u}_2), \boldsymbol{v}) \\ &= (\lambda\boldsymbol{u}_1 + \lambda\boldsymbol{u}_2, \boldsymbol{v}) = (\lambda\boldsymbol{u}_1, \boldsymbol{v}) + (\lambda\boldsymbol{u}_2, \boldsymbol{v}) = \lambda(\boldsymbol{u}_1, \boldsymbol{v}) + \lambda(\boldsymbol{u}_2, \boldsymbol{v}).\end{aligned}$$

Additive inverses are given by $-(\boldsymbol{u}, \boldsymbol{v}) = (-\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, -\boldsymbol{v})$.

Solution 4.3 (To Exercise 2.9)**.** (a) Let U, V be vector spaces. On the set $U \times V$ define the following bilinearity operations: for $\boldsymbol{u}, \boldsymbol{u}_1, \boldsymbol{u}_2 \in U$, $\boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ and $\lambda \in \mathbb{R}$

$$\begin{aligned}&(\boldsymbol{u}_1 \otimes \boldsymbol{v}) + (\boldsymbol{u}_2 \otimes \boldsymbol{v}) := (\boldsymbol{u}_1 + \boldsymbol{u}_2) \otimes \boldsymbol{v}, \\ &(\boldsymbol{u} \otimes \boldsymbol{v}_1) + (\boldsymbol{u} \otimes \boldsymbol{v}_2) := \boldsymbol{u} \otimes (\boldsymbol{v}_1 + \boldsymbol{v}_2) \text{ and } \lambda(\boldsymbol{u} \otimes \boldsymbol{v}) := (\lambda\boldsymbol{u}) \otimes \boldsymbol{v} =: \boldsymbol{u} \otimes (\lambda\boldsymbol{v}).\end{aligned}$$

(b) Set $\boldsymbol{u} := (\boldsymbol{u}_1 + \boldsymbol{u}_2)$. It is definitely not necessary to write $\boldsymbol{u} := (\boldsymbol{u}_1 + \boldsymbol{u}_2)$, but it may help to see how the axioms from Item (a) can be applied. We then have,

$$\begin{aligned}(\boldsymbol{u}_1 + \boldsymbol{u}_2) \otimes (\boldsymbol{v}_1 + \boldsymbol{v}_2) &= \boldsymbol{u} \otimes (\boldsymbol{v}_1 + \boldsymbol{v}_2) = \boldsymbol{u} \otimes \boldsymbol{v}_1 + \boldsymbol{u} \otimes \boldsymbol{v}_2 = (\boldsymbol{u}_1 + \boldsymbol{u}_2) \otimes \boldsymbol{v}_1 + (\boldsymbol{u}_1 + \boldsymbol{u}_2) \otimes \boldsymbol{v}_2 \\ &= \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 + \boldsymbol{u}_2 \otimes \boldsymbol{v}_1 + \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 + \boldsymbol{u}_2 \otimes \boldsymbol{v}_2.\end{aligned}$$

(c) We have $\lambda\boldsymbol{u} \otimes \lambda\boldsymbol{v} = \lambda^2(\boldsymbol{u} \otimes \boldsymbol{v})$. Compare this with the linear scaling eq. (2) in Definition 2.1.

Solution 4.4 (To Exercise 2.10)**.** It suffices to check that for any $\boldsymbol{u} \in U$ and any elementary tensor $\boldsymbol{u}' \otimes \boldsymbol{v}' \in U \otimes V$ it holds that $\boldsymbol{u} \otimes 0_V + \boldsymbol{u}' \otimes \boldsymbol{v}' = \boldsymbol{u}' \otimes \boldsymbol{v}'$.

Indeed, since we can write $0_V = \boldsymbol{v}' - \boldsymbol{v}'$ it follows that

$$\boldsymbol{u} \otimes 0_V + \boldsymbol{u}' \otimes \boldsymbol{v}' = \boldsymbol{u} \otimes (\boldsymbol{v}' - \boldsymbol{v}') + \boldsymbol{u}' \otimes \boldsymbol{v}' = \boldsymbol{u} \otimes \boldsymbol{v}' + \boldsymbol{u} \otimes (-\boldsymbol{v}') + \boldsymbol{u}' \otimes \boldsymbol{v}' = \boldsymbol{u}' \otimes \boldsymbol{v}'$$

where in the last equality we have used that $\boldsymbol{u} \otimes (-\boldsymbol{v})$ is the additive inverse of $\boldsymbol{u} \otimes \boldsymbol{v}$.

The check for $0_U \otimes \boldsymbol{v}$ can be done in a similar way.

Solution 4.5 (To Exercise 2.14)**.** We must find a bijective linear function $\Psi\colon \mathbb{R}\otimes U \to U$. It suffices to define $\Psi(\lambda\otimes \boldsymbol{u}) = \lambda \boldsymbol{u}$. Linearity follows from the fact that the right-hand side is bilinear in $(\lambda, \boldsymbol{u})$ and the properties of the tensor product. Injectivity is immediate since $\lambda\otimes\boldsymbol{u}\in\ker\Psi$ if and only if $\Psi(\lambda\otimes\boldsymbol{u}) = \lambda\boldsymbol{u} = 0_U$, which in turn implies that either $\lambda = 0$ or $\boldsymbol{u} = 0_U$ by the axioms of vector spaces, and in both cases this means that $\lambda\otimes\boldsymbol{u} = 0_{\mathbb{R}\otimes U}$. Therefore, it follows that $\ker\Psi = \{0\}$, i.e., Ψ is injective.

Bijectivity can be shown by noting that every $\boldsymbol{u}\in U$ can be obtained as $\boldsymbol{u} = \Psi(1\otimes\boldsymbol{u})$ (which in particular implies that the inverse map is $\Psi^{-1}(\boldsymbol{u}) = 1\otimes\boldsymbol{u}$), or by noting that since $\dim(\mathbb{R}\otimes U) = \dim(\mathbb{R})\cdot\dim(U) = \dim(U)$, by the rank-nullity theorem it follows that

$$\dim\big(\operatorname{im}(\Psi)\big) = \dim\big(\mathbb{R}\otimes U\big) - \dim\big(\ker(\Psi)\big) = \dim(U) \qquad \text{so that} \qquad \operatorname{im}(\Psi) = U.$$

Solution 4.6 (To Exercise 3.10)**.** The unit element in $T^2(V)$ is $\mathbf{1} = (1,0,0)$. From the product formula in Example 3.9 we see that the entries of the inverse element $(a',\boldsymbol{x}',\boldsymbol{A}') := (a,\boldsymbol{x},\boldsymbol{A})^{-1}$ must satisfy

$$aa' = 1, \quad a\boldsymbol{x}' + a'\boldsymbol{x} = 0 \quad \text{and} \quad \boldsymbol{A} + \boldsymbol{A}' + \boldsymbol{x}\otimes\boldsymbol{x}' = 0.$$

The first equation is solvable if and only if $a\neq 0$, in which case $a' = a^{-1}$. Inserting this in the second equation it follows that $\boldsymbol{x}' = -\frac{1}{a^2}\boldsymbol{x}$. Lastly, from the third equation we see that $\boldsymbol{A}' = -\frac{1}{a^2}\boldsymbol{A} + \frac{1}{a^3}\boldsymbol{x}\otimes\boldsymbol{x}$. Hence, in $T^2(V)$, we have that

$$(a,\boldsymbol{x},\boldsymbol{A})^{-1} = (a^{-1}, -a^{-2}\boldsymbol{x}, -a^{-2}\boldsymbol{A} + a^{-3}\boldsymbol{x}\otimes\boldsymbol{x}).$$

This can also be seen from the more general formula $\boldsymbol{A}^{-1} = \sum_{n\geq 0}(\mathbf{1}-\boldsymbol{A})^{\otimes n}$.

References

[1] I. Chevyrev and A. Kormilitzin, *A primer on the signature method in machine learning*, 2024, arXiv:1603.03788 [stat.ML], Revised version.

[2] J. Diestel, J. H. Fourie, and J. Swart, *The metric theory of tensor products*, American Mathematical Society, Providence, RI, 2008, Grothendieck's résumé revisited.

[3] W. Hackbusch, *Tensor spaces and numerical tensor calculus*, second ed., Springer Series in Computational Mathematics, vol. 56, Springer, Cham, 2019.

[4] J. Håstad, *Tensor rank is NP-complete*, Automata, languages and programming (Stresa, 1989), Lecture Notes in Comput. Sci., vol. 372, Springer, Berlin, 1989, pp. 451–460.

[5] R. A. Horn and C. R. Johnson, *Matrix analysis*, 2nd ed., Cambridge University Press, Cambridge, 2012.

[6] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev. **51** (2009), no. 3, 455–500.

[7] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and* N*-way arrays*, Multiway data analysis (Rome, 1988), North-Holland, Amsterdam, 1989, pp. 7–18.

[8] S. Rabanser, O. Shchur, and S. Günnemann, *Introduction to tensor decompositions and their applications in machine learning*, arXiv:1711.10781 (2017).

[9] M. Reed and B. Simon, *Methods of modern mathematical physics. I*, second ed., Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York, 1980, Functional analysis.

5 Extended Section: Tensor rank

The goal of Sections 5 and 6 is to introduce an applicable problem for the reader to grapple with: algorithmically factoring tensor product expressions to a minimal number of terms (i.e. $u \otimes v + u \otimes w \longrightarrow u \otimes (v + w)$). While the problem itself is not critical for path signatures, through studying the problem, we will get to apply the language of tensors and tensor products to be used continuously throughout the rest of the book. For a machine learning introduction to tensors focused around the issue of tensor decomposition we refer the reader to [8].

Sections 5 and 6 also include a few programming exercises in *Mathematica*. Many of these require nothing but a copy of *Mathematica*, but we also make use of a few useful functions which reside in a Github repository at

https://github.com/jfbeda/tensor_factoring .

From the repository, the reader can simply download and run `main.nb` to access to the functions.

5.1 Tensor rank

In linear algebra, the *rank* of a matrix, often defined as the number of linearly independent rows (or columns) of the matrix, is an incredibly useful property. We wish to generalize matrix rank to *tensor rank*, but it is not clear how the typical definition, in terms of linearly independent rows (or columns), can be easily generalized. Instead, we begin by recharacterizing matrix rank in a way that is easily generalized to tensors of arbitrary order.

Example 5.1. Consider the order 1 tensors $\boldsymbol{u}$ and $\boldsymbol{v}$, and the order 2 tensor $\boldsymbol{A}$ defined in the canonical basis (see Remark 2.16) by

$$\boldsymbol{u} := \begin{bmatrix} 1 \\ 2 \end{bmatrix} \in \mathbb{R}^2, \quad \boldsymbol{v} := \begin{bmatrix} 3 \\ 4 \end{bmatrix} \in \mathbb{R}^2, \quad \boldsymbol{A} := \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} \in \mathbb{R}^2 \otimes \mathbb{R}^2. \tag{5}$$

Notice that $\boldsymbol{A}$ may be written as a tensor product of $\boldsymbol{u}$ and $\boldsymbol{v}$. That is

$$\boldsymbol{A} = \boldsymbol{u} \otimes \boldsymbol{v}, \text{ or in terms of components, } \boldsymbol{A}^{ij} = \boldsymbol{u}^i \boldsymbol{v}^j \text{ for } i, j = 1, 2. \tag{6}$$

However, such a decomposition does not always exist. For example, consider $\boldsymbol{B}$ given by

$$\boldsymbol{B} := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \in \mathbb{R}^2 \otimes \mathbb{R}^2. \tag{7}$$

It is not possible to find order 1 tensors $\boldsymbol{w}, \boldsymbol{z} \in \mathbb{R}^2$ such that $\boldsymbol{B} = \boldsymbol{w} \otimes \boldsymbol{z}$. It *is* however, possible to decompose $\boldsymbol{B}$ into the *sum* of two tensor products:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{8}$$

In fact, **the minimum number of distinct tensor products that must be summed to give a specific matrix is another way of defining matrix rank**. This number is always equivalent to the number of independent rows (or columns). Notice that $\boldsymbol{A}$ has rank 1, and $\boldsymbol{B}$ has rank 2 with respect to both definitions. This definition of rank is easily generalized to tensors of arbitrary order.

Definition 5.2 (Tensor rank and rank decomposition)**.** Let $V_1, \ldots, V_m$ be vector spaces.

- An order m tensor $\boldsymbol{T} \in V_1 \otimes \cdots \otimes V_m$ is said to be of *rank 1* if it can be written as an outer product of m order 1 tensors:
$$\boldsymbol{T} = \boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_m \tag{9}$$
for some $\boldsymbol{v}_i \in V_i$.
- An order m tensor $\boldsymbol{T} \in V_1 \otimes \cdots V_m$ is said to be of *rank* r if it can be written as the sum of r rank 1 tensors, **and this is the smallest r for which such a decomposition is possible.** That is, r is the smallest integer such that
$$\boldsymbol{T} = \sum_{l=1}^{r} \boldsymbol{v}_{l,1} \otimes \cdots \otimes \boldsymbol{v}_{l,m} \tag{10}$$
for some $v_{l,i} \in V_i$. Such a decomposition of $\boldsymbol{T}$ into a minimum number of rank 1 tensors is called a *rank decomposition* or *rank factorization*.

Remark 5.3. For order 2 tensors, tensor rank is equivalent to the usual definition of matrix rank and thus provides a valid generalization. We omit a proof, but, as we saw in Example 5.1, $\operatorname{Rank}(\boldsymbol{A}) = 1$ and $\operatorname{Rank}(\boldsymbol{B}) = 2$ in both the tensor rank sense of Definition 5.2, and in the sense of matrix rank[10].

Remark 5.4. Definition 5.2 is inherently basis independent. That said, we will often fix a basis for each V_i, treating the tensors as multidimensional arrays (Remark 2.16.)

Remark 5.5 ('Order', 'level' and 'degree' are the same, but 'rank' is not.)**.** In some contexts, particularly in physics, it is common to use the term *rank* to refer to what we have called the *order* of a tensor. **Do not be confused!** In our case, the *order* is trivial to compute, and the *rank* is difficult. The terminology itself becomes more apparent after reading Section 3 and comes from viewing the tensor algebra over $V = \mathbb{R}^d$ as a graded vector space: $T(V) = \bigoplus_{n \geq 0} V^{\otimes n}$. Naturally, elements of "the homogeneous component of degree n", $V^{\otimes n}$, are called homogeneous tensors and n is the degree or level.

Example 5.6. Let us verify some of the above examples in Mathematica. Mathematica has a built-in `TensorProduct` function, that the reader may wish to to use. First, let us put the tensors $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ from Example 5.1 into the system.

Input

```
u = {1,2}
v = {3,4}
A = {{3,4},{6,8}}
B = {{1,0},{1,1}}
```

We can then indeed verify that $\boldsymbol{A} = \boldsymbol{u} \otimes \boldsymbol{v}$ with:

Input[11]

[10] The astute reader may complain that we have not *proven* that it is not possible to write $\boldsymbol{B}$ as a single tensor product, thus in the sense of tensor rank, eq. (8) only shows us $\operatorname{Rank}(\boldsymbol{B}) \leq 2$. Indeed, this is correct, and as we will see, while verifying an upper bound for tensor rank is easy (one must simply check the decomposition is valid), proving such a decomposition is minimal can be difficult.

[11] Recall that in many coding languages, including Mathematica, Python, and C, '$x = y$' sets the variable x to be y, whereas '$x == y$' evaluates to True if x and y are equal, and False otherwise.

```
TensorProduct[u, v] == A
```

Output

```
True
```

We have just shown that $\text{Rank}(\boldsymbol{A}) = 1$ by using the tensor product definition of rank. To compute the rank of $\boldsymbol{B}$, which we expect to be 2, we might be tempted to use Mathematica's in-built `TensorRank` function. However, just as we warned in Remark 5.5, this function in fact computes the *order* of the tensor it is fed, **not** the *rank*. To compute the tensor rank of Definition 5.2, we could of course just use the inbuilt Mathematica function `MatrixRank` because as we have seen, tensor rank and matrix rank are the same for tensors of order 2. Unsurprisingly, `MatrixRank[A]` and `MatrixRank[B]` return 1 and 2 respectively.

If we wish to compute a *rank decomposition* however, Mathematica does not have an in-built function. Instead, we turn to one of the functions in the provided git repository: e.g. `displayRankDecompositionOrderTwo`. This function computes the rank decomposition of an order 2 tensor using a procedure that will be illustrated in Section 5.3 (Algorithm 1).

Input

```
displayRankDecompositionOrderTwo[A]
```

Output

$$\begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{4}{3} \end{pmatrix}$$

Input

```
displayRankDecompositionOrderTwo[B]
```

Output

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We notice that, while the rank decomposition Mathematica gives in example 5.6 is the same as the one we gave in eq. (8), the decomposition it found for $\boldsymbol{A}$ in example 5.6 is not the one we gave in eq. (5). This illustrates the general point that rank decompositions are not unique.

5.2 Computing the rank decomposition of order 2 tensors

Section 5.1 provides a formal definition of tensor rank, however, in terms of practically computing a rank decomposition, it is useful to work with a slightly different formalism. As we will see, finding rank decompositions for tensors of order 2 is straightforward, but for tensors of order 2 or greater it is much more difficult, in fact, it is NP-hard. In this section we explore the question of algorithmically computing rank decompositions of order 2 tensors (viewed as matrices in a particular basis), and in Section 5.4,

we see how the problem becomes much more challenging when we consider tensors of arbitrary order.

Suppose $\boldsymbol{M} \in \mathbb{R}^n \otimes \mathbb{R}^m$ is an order 2 tensor of shape (n, m). By definition, we may write the components of $\boldsymbol{M}$ in any basis as

$$\boldsymbol{M}^{ij} = \sum_{l=1}^{r} \boldsymbol{u}^i_{l,1} \boldsymbol{u}^j_{l,2}. \tag{11}$$

This equation is simply eq. (10) but specified to the case of $m = 2$, and put into in coordinate form. For computational purposes, it is now useful to abandon the view of decomposing $\boldsymbol{M}$ into tensor products of order one tensors $\boldsymbol{u}_{l,k}$ by interpreting the components of the vectors $\boldsymbol{u}^i_{l,1}$ and $\boldsymbol{u}^j_{l,2}$ as components of matrices $\boldsymbol{D}^{li}_1$ and $\boldsymbol{D}^{lj}_2$ respectively. This lets us write the rank decomposition of $\boldsymbol{M}^{ij}$ as

$$\boldsymbol{M}^{ij} = \sum_{l=1}^{r} \boldsymbol{D}^{li}_1 \boldsymbol{D}^{lj}_2. \tag{12}$$

Now, the entries of the two matrices $\boldsymbol{D}^{li}_1$ and $\boldsymbol{D}^{lj}_2$ encode the rank decomposition of $\boldsymbol{M}$. We now notice that eq. (12) looks just like the matrix product in component form. That is, interpreting $\underset{n\times m}{\boldsymbol{M}}$ [12] as an n-by-m matrix of its coefficients in a particular basis, we have that $\text{Rank}(\boldsymbol{M}) = r$ is the smallest integer such that there exist matrices $\underset{r\times n}{\boldsymbol{D}_1}$ and $\underset{r\times m}{\boldsymbol{D}_2}$ that satisfy:

$$\underset{n\times m}{\boldsymbol{M}} = \left(\underset{r\times n}{\boldsymbol{D}_1}\right)^T \underset{r\times m}{\boldsymbol{D}_2}. \tag{13}$$

Indeed, it is perfectly valid to define matrix rank in terms of eq. (13) (e.g. see [5, p. 13]). We see furthermore that finding $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ for a given $\boldsymbol{M}$ amounts to finding a rank decomposition of $\boldsymbol{M}$ because the coordinates of $\boldsymbol{u}_{l,1}$ and $\boldsymbol{u}_{l,2}$ can be recovered via

$$\boldsymbol{u}^i_{l,1} = \boldsymbol{D}^{li}_1, \quad \text{and} \quad \boldsymbol{u}^j_{l,2} = \boldsymbol{D}^{lj}_2. \tag{14}$$

Remark 5.7. It is clear from eq. (13) that a rank decomposition of a matrix is not unique: given a rank decomposition $\boldsymbol{M} = \boldsymbol{D}_1^T \boldsymbol{D}_2$, then $\boldsymbol{M} = (\boldsymbol{P}^T \boldsymbol{D}_1)^T (\boldsymbol{P}^{-1} \boldsymbol{D}_2)$ is also a valid rank decomposition for any invertible r-by-r matrix $\boldsymbol{P}$. Stronger uniqueness conditions exist for tensors of higher rank [7].

5.3 Algorithms for computing the rank decomposition of a matrix

There are fast (i.e. polynomial time) algorithms for computing $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ from eq. (13) for an n-by-m matrix $\boldsymbol{M}$. One such algorithm is detailed below using the reduced row echelon form (RREF) of $\boldsymbol{M}$ [13]:

Another method, computationally slower, uses the singular value decomposition (SVD)[14]:

[12] We use the shorthand $\underset{n\times m}{\boldsymbol{M}}$ for *the n-by-m matrix* $\boldsymbol{M}$.

[13] Recall the reduced row echelon form of a matrix is the result of applying elementary row operations on a matrix to bring it into a reduced form. For example, the following matrices are all in reduced row echelon form:

$$\begin{bmatrix} 1 & 7 & 0 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

[14] The SVD of a matrix $\underset{n\times m}{\boldsymbol{M}}$ is a factorization to $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\underset{n\times n}{\boldsymbol{U}}$ and $\underset{m\times m}{\boldsymbol{V}}$ are orthogonal matrices and $\underset{n\times m}{\boldsymbol{\Sigma}}$ is a diagonal matrix with $\text{Rank}(\boldsymbol{M})$ non-zero entries.

Algorithm 1 Rank decomposition of a matrix using the reduced row echelon form

$\boldsymbol{M} \leftarrow$ some matrix
$\boldsymbol{R} \leftarrow \mathrm{RREF}(\boldsymbol{M})$ ▷ Reduced row echelon form
$\boldsymbol{D}_1^T \leftarrow$ all columns of $\boldsymbol{M}$ that are pivot columns of $\boldsymbol{R}$
$\boldsymbol{D}_2 \leftarrow$ all non-zero rows of $\boldsymbol{R}$
assert $\boldsymbol{M} == \boldsymbol{D}_1^T \boldsymbol{D}_2$
return $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$

Algorithm 2 Rank decomposition of a matrix using singular value decomposition

$\boldsymbol{M} \leftarrow$ some matrix
$\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T \leftarrow \mathrm{SVD}(\boldsymbol{M})$ ▷ Singular value decomposition
$r \leftarrow \mathrm{Rank}(\boldsymbol{M})$ ▷ Easily acquired from the SVD
$\boldsymbol{U}' \leftarrow$ first r columns of $\boldsymbol{U}$
$\boldsymbol{\Sigma}' \leftarrow$ top-left $r \times r$ diagonal block of $\boldsymbol{\Sigma}$
$\boldsymbol{V}' \leftarrow$ first r columns of $\boldsymbol{V}$
assert $C == \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{V}'^T$
$\boldsymbol{D}_1^T \leftarrow \boldsymbol{U}'$
$\boldsymbol{D}_2 \leftarrow \boldsymbol{\Sigma}'\boldsymbol{V}'^T$
assert $\boldsymbol{M} == \boldsymbol{D}_1^T \boldsymbol{D}_2$
return $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$

Implementations of both Algorithms 1 and 2 in Mathematica are available through the provided git repository via `rankDecompositionOrderTwoToMatricesRREF` and `rankDecompositionOrderTwoToMatricesS` respectively.

Exercise 5.8. Define $\boldsymbol{M}$ as

$$\boldsymbol{M} = \begin{bmatrix} 3 & 4 & 2 \\ 1 & 2 & 1 \\ 0 & -2 & -1 \end{bmatrix}.$$

(a) Compute the rank of $\boldsymbol{M}$ by considering the number of independent columns.

(b) Compute the rank of $\boldsymbol{M}$ by considering the number of independent rows.

(c) Does the decomposition of $\boldsymbol{M}$ into the form of eq. (16) below contradict your answers to Items (a) and (b)?

$$\begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \boldsymbol{M}. \tag{16}$$

5.4 Surprising properties of the tensor decomposition

Tensor rank is filled with surprising properties, not expected from a generalization of matrix rank.

▷ *Computing tensor decompositions of tensors of order greater than* 2 *is NP-hard* [4].

This is to say that, assuming $P \neq NP$, there is no polynomial time algorithm for computing the tensor rank of tensors of order greater than 2. For the case of order 2, the algorithms in Section 5.3 are both polynomial time.

▹ *The rank of a real tensor can differ over* $\mathbb{R}$ *and* $\mathbb{C}$ [7, p. 10].

We can show this through the following illustrating example from Kruskal [6, p. 464].

Example 5.9. Let $\boldsymbol{Z} \in \mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2$ be the order 3 tensor of shape $(2,2,2)$ defined in the canonical basis by

$$\boldsymbol{Z}^{ij1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{ij}, \quad \boldsymbol{Z}^{ij2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}^{ij}. \tag{17}$$

Define $\boldsymbol{u}_{l,i}$ by

$$\boldsymbol{u}_{1,1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{u}_{1,2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{u}_{1,3} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \tag{18}$$

$$\boldsymbol{u}_{2,1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{u}_{2,2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{u}_{2,3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{19}$$

$$\boldsymbol{u}_{3,1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \boldsymbol{u}_{3,2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{u}_{3,3} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{20}$$

Define $\boldsymbol{v}_{l,j}$ by

$$\boldsymbol{v}_{1,1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}, \quad \boldsymbol{v}_{1,2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \boldsymbol{v}_{1,3} = \begin{bmatrix} 1 \\ -i \end{bmatrix}, \tag{21}$$

$$\boldsymbol{v}_{2,1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \boldsymbol{v}_{2,2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}, \quad \boldsymbol{v}_{2,3} = \begin{bmatrix} 1 \\ i \end{bmatrix}. \tag{22}$$

We may decompose $\boldsymbol{Z}$ as

$$\boldsymbol{Z} = \boldsymbol{u}_{1,1} \otimes \boldsymbol{u}_{1,2} \otimes \boldsymbol{u}_{1,3} + \boldsymbol{u}_{2,1} \otimes \boldsymbol{u}_{2,2} \otimes \boldsymbol{u}_{2,3} + \boldsymbol{u}_{3,1} \otimes \boldsymbol{u}_{3,2} \otimes \boldsymbol{u}_{3,3} \tag{23}$$

Which is a decomposition in three terms, using only real coefficient. However, despite the fact that $\boldsymbol{Z}$ is real, we may also decompose $\boldsymbol{Z}$ as:

$$\boldsymbol{Z} = \boldsymbol{v}_{1,1} \otimes \boldsymbol{v}_{1,2} \otimes \boldsymbol{v}_{1,3} + \boldsymbol{v}_{2,1} \otimes \boldsymbol{v}_{2,2} \otimes \boldsymbol{v}_{2,3} \tag{24}$$

using only two terms. The price we pay for the two term decomposition is that the coefficients of the tensors in which we expand $\boldsymbol{Z}$ are complex. In this way we see clearly that the rank of a tensor depends on whether or not it is decomposed over $\mathbb{R}$ or over $\mathbb{C}$. The astute reader will indeed point out that we have not *proven* that no two-term decomposition of $\boldsymbol{Z}$ over $\mathbb{R}$ exists, but it can be shown that this is the case. This example serves only as an illustration.

Remark 5.10. The tensors $\boldsymbol{Z}^{ijk}$, $\boldsymbol{u}_{l,i}$ and $\boldsymbol{v}_{l,j}$ above are hard coded in the git repository, and can be accessed via `Z[[i,j,k]]`, `u[[k,i]]` and `v[[k,j]]` respectively. We can then verify eq. (23) and eq. (24) by running (recall footnote 11 for the '==' notation):

Input

```
Z==Sum[TensorProduct[u[[k,1]],u[[k,2]],u[[k,3]]],{k,3}]
Z==Sum[TensorProduct[v[[k,1]],v[[k,2]],v[[k,3]]],{k,2}]
```

Output

```
True
True
```

6 Extended Section: Factoring tensor product expressions

In this section, we introduce an elegant toy problem to increase our familiarity with tensors and tensor products. The problem is interesting in its own right, though the specifics are not necessary to signatures. The problem is to *find an algorithm to factor tensor product $\otimes$ expressions to a minimal number of terms*. As is often the case, this problem can be elegantly recast in the language of tensors: computing a minimal factorization amounts to finding a rank decomposition.

6.1 The main problem

Given a tensor product expression comprising many terms, what is the minimum number of terms in which it may be factored using the linearity rules of the tensor product?

Let us consider a few simple examples. Suppose $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \in U$, and $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3 \in V$ for two vector spaces U and V. Consider

$$\begin{aligned} \boldsymbol{X}_0 &\coloneqq \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \to (\boldsymbol{a}_1 + \boldsymbol{a}_2) \otimes (\boldsymbol{b}_1 + \boldsymbol{b}_2) \\ \boldsymbol{X}_1 &\coloneqq \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_3 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_3 \to \boldsymbol{a}_1 \otimes (\boldsymbol{b}_1 + \boldsymbol{b}_3) + \boldsymbol{a}_2 \otimes (\boldsymbol{b}_2 + \boldsymbol{b}_3) \end{aligned} \tag{25}$$

We see that $\boldsymbol{X}_0$, originally comprising 4 terms, can be factored into just 1 term. Whereas $\boldsymbol{X}_1$, also comprising 4 terms can only be factored into a minimum of 2 terms. It is not hard to convince yourself that there is no way to factor $\boldsymbol{X}_1$ into just a single term, but we would like to be able to *prove* this. That is, given a tensor product expression $\boldsymbol{X}$ we would like a systematic way of finding a factorization of $\boldsymbol{X}$ in a minimal number of terms. As it turns out, the answer to this question lies in computing the rank decomposition of $\boldsymbol{X}$. Let us start by formalizing what 'minimal factorization' means to then state the problem appropriately (see Problem 6.2).

Definition 6.1 (Minimal factorization)**.** Let U, V be vector spaces, and let $\boldsymbol{X} \in U \otimes V$ be a tensor product expression. A factorization of $\boldsymbol{X}$ is called *minimal* (or '*minimal factorization*') if there are no factorizations of $\boldsymbol{X}$ with fewer terms.

We note immediately that minimal factorizations are not unique. Observe that $\boldsymbol{X}_1$ from eq. (25) can also be minimally factored to two terms as

$$\boldsymbol{X}_1 = \frac{1}{2}(\boldsymbol{a}_1 + \boldsymbol{a}_2) \otimes (\boldsymbol{b}_1 + \boldsymbol{b}_2 + 2\boldsymbol{b}_3) + \frac{1}{2}(\boldsymbol{a}_2 - \boldsymbol{a}_1) \otimes (\boldsymbol{b}_2 - \boldsymbol{b}_1).$$

Problem 6.2 (Minimally factoring tensor product expressions)**.** Let U, V be vector spaces of dimension d_1 and d_2 with bases $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_j\}$ respectively. Given the unfactored tensor product expression:

$$\boldsymbol{X} \coloneqq \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}^{ij} (\boldsymbol{a}_i \otimes \boldsymbol{b}_j) = \boldsymbol{X}^{11} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \cdots + \boldsymbol{X}^{d_1 d_2} \boldsymbol{a}_{d_1} \otimes \boldsymbol{b}_{d_2} \in U \otimes V. \tag{26}$$

Determine the minimum $r \in \mathbb{N}$, such that there exist $\boldsymbol{u}_l \in U$, and $\boldsymbol{v}_l \in V$ such that we may write $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \sum_{l=1}^{r} \boldsymbol{u}_l \otimes \boldsymbol{v}_l. \tag{27}$$

6.2 The naive algorithm

Already you may be looking at Problem 6.2 and seeing similarities to the tensor decompositions of the previous section. For the moment however, we ignore this and try a simple algorithm for approaching the problem. Suppose the expression we need to factor is:

$$\boldsymbol{X}_0 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_3 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_3. \tag{28}$$

A naive algorithm might try grouping left terms ($\boldsymbol{a} \otimes \boldsymbol{c} + \boldsymbol{b} \otimes \boldsymbol{c} \to (\boldsymbol{a} + \boldsymbol{b}) \otimes \boldsymbol{c}$), then right terms ($\boldsymbol{a} \otimes \boldsymbol{b} + \boldsymbol{a} \otimes \boldsymbol{c} \to \boldsymbol{a} \otimes (\boldsymbol{b} + \boldsymbol{c})$), until no further terms match. Applying such an algorithm starting starting from the left would give:

$$\boldsymbol{X}_0 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 + (\boldsymbol{a}_1 + \boldsymbol{a}_2) \otimes \boldsymbol{b}_3 \qquad \text{(3 terms)} \tag{29}$$

Or from the right:

$$\boldsymbol{X}_0 = \boldsymbol{a}_1 \otimes (\boldsymbol{b}_1 + \boldsymbol{b}_3) + \boldsymbol{a}_2 \otimes (\boldsymbol{b}_2 + \boldsymbol{b}_3) \qquad \text{(2 terms)} \tag{30}$$

It turns out the factorization with 2 terms is minimal, but we already see the limitations of this algorithm: *it can get stuck*. The factorization in eq. (29) contains no terms that can be grouped, and yet is not minimal. This algorithm is *greedy*: it takes the locally optimal decision, but there is no guarantee that the final factorization it provides is minimal.

6.3 Using the power of tensor rank

We seek a better factorization algorithm. To proceed, we take the equation for the factored form of an arbitrary tensor product expression, eq. (26), and the equation for the unfactored form, eq. (27), and set them equal:

$$\boldsymbol{X} = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}^{ij} (\boldsymbol{a}_i \otimes \boldsymbol{b}_j) = \sum_{l=1}^{r} \boldsymbol{u}_l \otimes \boldsymbol{v}_l. \tag{31}$$

Comparing eq. (31) with Definition 5.2, we notice that finding the minimal factorization of $\boldsymbol{X}$, i.e. by finding $\boldsymbol{u}_l$ and $\boldsymbol{v}_l$, is exactly the problem of finding a rank decomposition of $\boldsymbol{X}$. We saw various algorithms in Section 5.3 for computing the rank decomposition of an order 2 tensor, and we can indeed use any of these to find a decomposition of $\boldsymbol{X}$.

To be more explicit, we may write $\boldsymbol{u}_l$ and $\boldsymbol{v}_l$ in terms of the bases $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_i\}$ respectively. That is,

$$\boldsymbol{u}_l = \sum_{i=1}^{d_1} u_l^i \boldsymbol{a}_i, \quad \text{and} \quad \boldsymbol{v}_l = \sum_{j=1}^{d_2} v_l^j \boldsymbol{b}_i. \tag{32}$$

Substituting this into eq. (31) gives:

$$\sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \boldsymbol{X}^{ij}(\boldsymbol{a}_i \otimes \boldsymbol{b}_j) = \sum_{l=1}^{r}\sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \boldsymbol{u}_l^i \boldsymbol{v}_l^j \boldsymbol{a}_i \otimes \boldsymbol{b}_j \tag{33}$$

$$0 = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \left(\boldsymbol{X}^{ij} - \sum_{l=1}^{r} \boldsymbol{u}_l^i \boldsymbol{v}_l^j\right) \boldsymbol{a}_i \otimes \boldsymbol{b}_j. \tag{34}$$

$$\implies \quad \boldsymbol{X}^{ij} = \sum_{l=1}^{r} \boldsymbol{u}_l^i \boldsymbol{v}_l^j. \tag{35}$$

Where in the last line we have used the fact that $\{\boldsymbol{a}_i \otimes \boldsymbol{b}_j\}$ are a basis for $U \otimes V$ (Proposition 2.11) and so the coefficients in eq. (34) must all vanish. We may now use any of the algorithms in Section 5.3 to find matrices $\underset{r\times n_1}{\boldsymbol{D}_1}$ and $\underset{r\times n_2}{\boldsymbol{D}_2}$ such that $\boldsymbol{X}^{ij} = (\boldsymbol{D}_1^T \boldsymbol{D}_2)^{ij}$, which allows us to recover the coefficients of $\boldsymbol{u}_l$ and $\boldsymbol{v}_l$ in the $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_i\}$ bases via

$$\boldsymbol{u}_l^i = \boldsymbol{D}_1^{li}, \quad \text{and} \quad \boldsymbol{v}_l^j = \boldsymbol{D}_2^{lj}. \tag{36}$$

Exercise 6.3. Let $U = \mathbb{R}[x]$, and $V = \mathbb{R}[y]$ be the vector spaces of polynomials in the single variable x, and y respectively. Consider the following expression:

$$\boldsymbol{E} \coloneqq -x \otimes y + 2x^2 \otimes y + 3x \otimes y^2 - 4x^2 \otimes y^2 + x^3 \otimes y^2 \in U \otimes V. \tag{37}$$

(a) Try to factor eq. (37) to a minimal number of terms by grouping.

(b) Confirm your factorization is indeed minimal by using Mathematica, or otherwise.

6.4 Generalization

We have solved the problem of factorizing tensor product expressions containing one tensor product symbol per term. That is, we have a computationally efficient[15] algorithm for solving Problem 6.2. We can easily generalize this to the case of expressions that contain more than one tensor product symbol in each term. For example

$$\boldsymbol{Z} \coloneqq \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 \otimes \boldsymbol{w}_1 + \boldsymbol{u}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{w}_2 - \boldsymbol{u}_2 \otimes \boldsymbol{v}_1 \otimes \boldsymbol{w}_2 + \boldsymbol{u}_2 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{w}_1. \tag{38}$$

Proceeding just as in Problem 6.2 we can state the question of factoring tensor product expressions involving more than one tensor product symbol.

Problem 6.4. Let $V_1, \cdots, V_m$ be vector spaces of dimension $d_1, \cdots, d_m$ with bases $\{\boldsymbol{a}_{1,i_1}\}, \cdots, \{\boldsymbol{a}_{m,i_m}\}$ respectively. Given the unfactored tensor product expression

$$\boldsymbol{Y} \coloneqq \sum_{i_1=1}^{d_1} \cdots \sum_{i_m=1}^{d_m} \boldsymbol{Y}^{i_1 i_2 \cdots i_m} \left(\boldsymbol{a}_{1,i_1} \otimes \cdots \otimes \boldsymbol{a}_{m,i_m}\right) \in V_1 \otimes \cdots \otimes V_m \tag{39}$$

[15] Since all of the algorithms in Section 5.3 are polynomial time, factorizations of tensor product expressions involving a single tensor product symbol are fast to compute.

what is the minimum integer r such that we may write $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = \sum_{l=1}^{r} \boldsymbol{v}_{l,1} \otimes \cdots \otimes \boldsymbol{v}_{l,m}. \tag{40}$$

Once again, we see that finding the minimum factorization of a tensor product expression $\boldsymbol{Y}$ amounts simply to finding the rank decomposition of $\boldsymbol{Y}$. The number of terms of terms in a minimal factorization is simply the tensor rank.

It is here we can make use of some of the important properties of the tensor rank that we saw in Section 5.4.

- Since minimally factoring a tensor product expression is equivalent to finding its rank decomposition, both problems are NP-hard for tensors of order greater than 2.
- The fact that the tensor rank of real tensors can differ over $\mathbb{R}$ and over $\mathbb{C}$ translates into the fact that tensor product expressions of order greater than 2 differ when the factorization is over $\mathbb{R}$ or over $\mathbb{C}$. As an example, take $\boldsymbol{Z}$ from equation eq. (38):

  $$\boldsymbol{Z} = \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 \otimes \boldsymbol{w}_1 + \boldsymbol{u}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{w}_2 - \boldsymbol{u}_2 \otimes \boldsymbol{v}_1 \otimes \boldsymbol{w}_2 + \boldsymbol{u}_2 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{w}_1. \tag{41}$$

  This purely real expression can be minimally *factored to three terms* over $\mathbb{R}$:

  $$\boldsymbol{Z} = \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 \otimes (\boldsymbol{w}_1 - \boldsymbol{w}_2) + \boldsymbol{u}_2 \otimes \boldsymbol{v}_2 \otimes (\boldsymbol{w}_1 + \boldsymbol{w}_2) + (\boldsymbol{u}_1 - \boldsymbol{u}_2) \otimes (\boldsymbol{v}_1 + \boldsymbol{v}_2) \otimes \boldsymbol{w}_2, \tag{42}$$

  but minimally *factored to two terms* over $\mathbb{C}$:

  $$\begin{aligned} \boldsymbol{Z} = \frac{1}{2}\Big(&(\boldsymbol{u}_1 - i\boldsymbol{u}_2) \otimes (\boldsymbol{v}_1 + i\boldsymbol{v}_2) \otimes (\boldsymbol{w}_1 - i\boldsymbol{w}_2) \\ &+ (\boldsymbol{u}_1 + i\boldsymbol{u}_2) \otimes (\boldsymbol{v}_1 - i\boldsymbol{v}_2) \otimes (\boldsymbol{w}_1 + i\boldsymbol{w}_2)\Big). \end{aligned} \tag{43}$$

7 Extended Section: Solutions to additional Exercises

Solution 7.1 (To Exercise 5.8)**.** (a) Label the columns $\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3$. Notice that $\boldsymbol{C}_2 = 2\boldsymbol{C}_3$ and hence $\boldsymbol{C}_2$ is linearly dependent on $\boldsymbol{C}_3$. Since $\boldsymbol{C}_1$ and $\boldsymbol{C}_3$ are linearly independent, the matrix $\boldsymbol{M}$ has a total of 2 linearly independent columns. Thus the rank of $\boldsymbol{M}$ is 2.

(b) Label the rows $\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3$. Note that $\boldsymbol{R}_1 = 3\boldsymbol{R}_2 + \boldsymbol{R}_3$, thus $\boldsymbol{R}_1$ is linearly dependent on $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$. Since $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ are linearly independent, the rank of $\boldsymbol{M}$ is 2.

(c) One might think that eq. (16) provides a rank decomposition of $\boldsymbol{M}$ in three terms, and thus conclude that $\text{Rank}(\boldsymbol{M}) = 3$, contradicting our results from Items (a) and (b). However, just because the decomposition in eq. (16) does correctly produce $\boldsymbol{M}$, it is not a decomposition with a minimal number of terms. In fact, from the previous parts of the question we *know* that $\boldsymbol{M}$ can be decomposed into the sum of two outer products. Finding these can be difficult (and the reader is not expected to be able to do so in their head), but here is one such decomposition:

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \boldsymbol{M}. \tag{44}$$

We can guarantee that there is no decomposition of $\boldsymbol{M}$ in fewer than 2 terms because we saw earlier that $\mathrm{Rank}(\boldsymbol{M}) = 2$. We can also use the provided Mathematica code to find a rank decomposition of $\boldsymbol{M}$.

Input

```
M = {{3,4,2},{1,2,1},{0,-2,-1}};
displayRankDecompositionOrderTwo[M]
```

Output

$$\begin{pmatrix} 3 & 4 & 2 \\ 1 & 2 & 1 \\ 0 & -2 & -1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 4 \\ 2 \\ -2 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ \frac{1}{2} \end{pmatrix}$$

This is different, valid rank decomposition of $\boldsymbol{M}$.

Solution 7.2 (To Exercise 6.3)**.** (a) We begin with:

$$\boldsymbol{E} \coloneqq -x \otimes y + 2x^2 \otimes y + 3x \otimes y^2 - 4x^2 \otimes y^2 + x^3 \otimes y^2 \in U \otimes V. \tag{45}$$

Grouping from the left gives:

$$\boldsymbol{E} = x \otimes (3y^2 - y) + x^2 \otimes (-4y^2 + 2y) + x^3 \otimes y^2. \tag{46}$$

Which cannot be further grouped. Trying instead from the right gives:

$$\boldsymbol{E} = (-x + 2x^2) \otimes y + (3x - 4x^2 + x^3) \otimes y^2. \tag{47}$$

Thus the rank of $\boldsymbol{E}$ is no larger than 2.

(b) Let us assign the obvious bases $\{x, x^2, x^3, \cdots\}$ and $\{y, y^2, y^3, \cdots\}$ to $U = \mathbb{R}[x]$ and $V = \mathbb{R}[y]$. Written in this basis we have

$$\boldsymbol{E} = \sum_{i=1}^{3} \sum_{j=1}^{2} \boldsymbol{E}^{ij} x^i \otimes y^j. \tag{48}$$

Where the coefficients $\boldsymbol{E}^{ij}$ can be written in a matrix as:

$$\boldsymbol{E}^{ij} = \begin{bmatrix} -1 & 3 \\ 2 & -4 \\ 0 & 1 \end{bmatrix}. \tag{49}$$

Letting Mathematica compute the rank of this matrix yields 2, which tells us that our factorization with 2 terms is minimal.