Tensors: Old and New Motivations from Linear Algebra

John Sheekey

UCD

ILAS, March 2022

Premise

- Tensors are fundamental objects in linear algebra, which appear in various branches of maths and science.
- Tensors are both easier and more difficult than you think.
- Students sometimes first encounter tensors through physics or differential geometry.
- There are some interesting accessible motivations for studying tensors from a linear algebraic standpoint.

Tensors: They’re not just for Physics!

Notation

Throughout this talk:

- F denotes a field;
- V denotes an F-vector space of (usually) finite dimension n.
- $\mathcal{B} = \{e_1, \ldots, e_n\}$ denotes a basis for V over F.

Matrices, Linear Maps, and Bilinear Forms

We usually see the following basic facts in introductory linear algebra courses:

- Every linear map t from V to itself can be represented uniquely as a matrix, that we will denote by T with entries T_{ij}:
$$t(e_j) = \sum_i T_{ij} e_i.$$
- Every bilinear form b from $V \times V$ to F can be represented uniquely as a matrix, that we will denote by B with entries B_{ij}:
$$b(e_i, e_j) = B_{ij}.$$

Tensors, Multilinear Maps, and Multilinear Forms

It is not a big leap to show the following:

- Every bilinear map t from $V \times V$ to V can be represented uniquely as a 3-dimensional array of field elements, that we will denote by T with entries T_{ijk}:

$$t(e_j, e_k) = \sum_i T_{ijk} e_i.$$

- Every trilinear form b from $V \times V \times V$ to F can be represented uniquely as a 3-dimensional array of field elements, that we will denote by B with entries B_{ijk}:

$$b(e_i, e_j, e_k) = B_{ijk}.$$

Tensors as Multidimensional Arrays

We can write a three-dimensional array as a sequence of matrices, for example

$$\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right]$$

We call these matrices the *slices* of the tensor.

The space of all trilinear forms has dimension n^3.

Pure Tensors, Rank, and Group Actions

Let $V^{\otimes 2} = V \otimes V$ denote the F-vector space spanned by elements $e_i \otimes e_j$, with the rule that

$$u \otimes (v + \lambda v') = u \otimes v + \lambda u \otimes v'$$
$$(u + \lambda u') \otimes v = u \otimes v + \lambda u' \otimes v$$

for all $u, v, u', v' \in V$, and all $\lambda \in F$.

Via the correspondence $u \otimes v \leftrightarrow uv^t$, where we view elements of V as column vectors with coordinates with respect to the basis $\mathcal{B}$, we see that two-fold tensors of the form $u \otimes v$ correspond to rank-one matrices.

The linear map defined by $u \otimes v$ is the map $x \mapsto u^*(x)y$.

The bilinear form defined by $u \otimes v$ is the map $(x, y) \mapsto u^*(x)v^*(y)$; a product of linear forms.

Pure Tensors, Rank, and Group Actions

More generally, the space $V^{\otimes k}$ has dimension n^k, with basis elements $e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_k}$.

Tensors of the form $u_1 \otimes u_2 \otimes \cdots \otimes u_k$ are called *pure*, *fundamental*, or *rank-one* tensors.

Every tensor is a sum of pure tensors; the fewest pure tensors required to generate a tensor is called its *rank*; in the matrix case this corresponds precisely to the usual rank of a matrix; this can be an interesting exercise.

For a tensor product of more than two spaces, it is extremely difficult to calculate the rank, or even to know what the maximum possible rank is.

Pure Tensors, Rank, and Group Actions

A rank-one matrix can be recognised by noting that its rows are all scalar multiples of each other.

$$\begin{bmatrix} c \\ d \end{bmatrix} \otimes \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ce & cf \\ de & df \end{bmatrix}$$

A rank-one tensor in $V \otimes V \otimes V$ can be recognised by noting that its slices are all scalar multiples of each other, and are all rank one matrices.

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} \otimes \begin{bmatrix} e \\ f \end{bmatrix} = \left[\begin{bmatrix} ace & acf \\ ade & adf \end{bmatrix}, \begin{bmatrix} bce & bcf \\ bde & bdf \end{bmatrix} \right]$$

Pure Tensors, Rank, and Group Actions

Consider $W = V^{\otimes k} = V \otimes \cdots \otimes V$. Invertible linear maps are elements of $\mathrm{GL}(W) = \mathrm{GL}(n^k, F)$.

The group $G = GL(V) \times \cdots \times GL(V)$ acts naturally on W:

$$\boxed{(g_1, \ldots, g_k) : v_1 \otimes \cdots \otimes v_k \mapsto g_1(v_1) \otimes \cdots \otimes g_k(v_k).}$$

We can write $g_1 \otimes \cdots \otimes g_k$ for this map. If we identify $F^n \otimes F^n$ with F^{n^2} in a sensible way, then the matrix corresponding to $g_1 \otimes g_2$ is the Kronecker product of the matrices corresponding to g_1 and g_2 respectively.

Such a map fixes the set of pure tensors, and preserves tensor rank.

In the case $k = 2$, this corresponds to multiplying a matrix on the left and right by an invertible matrix.

Algebras and Linear Maps

- An algebra can be viewed as a subalgebra of a matrix algebra via its *regular representation*.
- This can be used as an illustration of how each axiom influence the properties, and what happens when we relax the rules.
- We can naturally associate a tensor to an algebra; also vice-versa, if we omit multiplicative associativity!
- Natural notions like tensor rank can tell us something about the algebra; multiplicative complexity.
- However, these notions can be incredibly difficult to work with!

Algebras and Linear Maps

Let A denote an F-algebra of dimension n. Then “Multiplication on the left by y” defines a map on A:

$$x \mapsto y \circ x =: L_y(x)$$

Then for $x, z \in A$, $\lambda \in F$,

$$\begin{aligned} L_y(x + \lambda z) &= L_y(x) + \lambda L_y(z) \\ L_{y+\lambda z}(x) &= L_y(x) + \lambda L_z(x). \end{aligned}$$

- Each L_y is an F-linear map on A, and so can be identified with an element of $M_n(F)$.
- The set $C(A) = \{L_y : y \in A\}$ is an n-dimensional subspace of $M_n(F)$.

Algebras and Linear Maps

The set $C(A) = \{L_y : y \in A\}$ is an n-dimensional subspace of $M_n(F)$.

$$C(\mathbb{C}) = \left\{ \begin{bmatrix} a & -b \\ b & a \end{bmatrix} : a, b \in \mathbb{R} \right\}$$

$$C(\mathbb{H}) = \left\{ \begin{bmatrix} a & -b & -c & d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} : a, b, c, d \in \mathbb{R} \right\}$$

$$C(M_2(F)) = \left\{ \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \\ c & 0 & d & 0 \\ 0 & c & 0 & d \end{bmatrix} : a, b, c, d \in \mathbb{R} \right\}$$

Algebras and Linear Maps

The set $C(A) = \{L_y : y \in A\}$ is an n-dimensional subspace of $M_n(F)$.

- If A is associative, then $C(A)$ is a sub-algebra of $M_n(F)$.
- If A is a division algebra, then every nonzero element of $C(A)$ is invertible.

Non-associative division algebras (*semifields*) correspond precisely to n-dimensional subspace of $M_n(F)$ where every nonzero element is invertible.

Algebras and Tensors

By choosing a basis for $C(A)$, we can represent the algebra A as a tensor $T(A)$.

$$T(\mathbb{C}) = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right]$$

$$T(\mathbb{H}) = \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \right]$$

$$T(M_2(F)) = \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right]$$

Algebras and Tensor Rank

We can decompose the tensor $T(\mathbb{C})$ into the sum of three rank-one tensors.

$$\begin{aligned}
T(\mathbb{C}) &= \left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}\right] \\
&= \frac{1}{2}\left[\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\right] \\
&+ \frac{1}{2}\left[\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}, \begin{bmatrix}-1 & 1\\1 & -1\end{bmatrix}\right] \\
&+ 2\left[\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}, \begin{bmatrix}0 & -1\\0 & 0\end{bmatrix}\right]
\end{aligned}$$

Does this tell us anything useful?

Algebras and Tensor Rank

In order to calculate the product

$$(a+bi)(c+di) = (ac-bd)+(ad+bc)i,$$

naively it takes four real multiplications.

However, we can achieve this using fewer multiplications (although more additions) of the unknowns:

$$\begin{aligned} ac-bd &= ((a+b)(c+d)+(a-b)(c-d))/2-2bc \\ ad+bc &= ((a+b)(c+d)-(a-b)(c-d))/2 \end{aligned}$$

The tensor rank of an algebra measures the complexity of multiplication in the nonscalar model.

Algebras and Tensor Rank

The tensor rank of an algebra measures the complexity of multiplication in the nonscalar model.

Similarly, multiplying 2×2 matrices naively takes 8 multiplications.

However, Strassen's famous algorithm shows that it can be done in 7 multiplications (and 7 are necessary): the tensor rank of $T(M_2(F))$ is 7.

This is very important for efficient computation of larger matrices. The tensor rank is not known even for 3×3 matrices; all we know is that it is at least 19 and at most 23.

Algebras and Tensor Rank

Over finite fields, multiplication in field extensions plays an important role in many applications.

In this setting the problem has interesting connections to coding theory and algebraic geometry.

In a recent paper (Lavrauw-JS), it was shown that there exist non-associative division algebras (semifields) with lower tensor rank than the field of the same order.

What is a determinant?

Students are (in some universities) first introduced to determinants when attempting to calculate the inverse of a matrix.

They learn algorithms for calculating larger determinants, learn that a non-zero determinant implies invertibility, and that the determinant of a product is the product of the determinants.

[One visual way of explaining determinants is as the area/volume of the image of a unit square/cube after multiplying by the given matrix - GeoGebra]

However, a proof of the product rule is often deemed too complicated, perhaps proven only in the 2×2 case.

Alternating Multilinear Forms

Students will often encounter the notion of a symmetric, or skew-symmetric/antisymmetric/alternating bilinear form, and how they correspond to symmetric and skew-symmetric matrices respectively.

They may encounter exterior/wedge products in physics-motivated settings.

However a straightforward description of the determinant and its important properties can be achieved without too much complicated setup.

Alternating Multilinear Forms

- A multilinear form is *alternating* if it is zero whenever any two inputs are equal.
- The space of alternating form on V^n is one-dimensional; choose D as one such (nonzero) form.
- The form $D_T(v_1, \ldots, v_n) := D(T(v_1), \ldots, T(v_n))$ is alternating; thus there exists a unique $d(T) \in F$ such that $D_T = d(T)D$.
- Then $D_{ST} = d(ST)D$. It is also straightforward to see that $D_{ST} = d(S)d(T)D$. Thus d is multiplicative.
- It is reasonably straightforward to see that $d(T) = 0$ if and only if T is not invertible.

$$d(T) = \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) \prod_{i=1}^{n} T_{i\sigma(i)}$$

Tensors and Quantum Information

Classically, an object with two possible states (on/off, up/down etc) is represented by a 0 or 1, and a system of n such objects is represented by a string in $\{0, 1\}^n$.

In the quantum world, objects can be in a *superposition* of states. An *observation* returns one of the classical states with some *probability*.

We represent a quantum state by a vector $\alpha_0 e_0 + \alpha_1 e_1 \in F^2$. A measurement will return the vector e_i with probability $|\alpha_i|$.

A quantum system of n objects is represented by a vector in the tensor power $(F^2)^{\otimes n}$. Classical states correspond to standard basis vectors $e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_n} \leftrightarrow (i_1, i_2, \ldots, i_n)$.

Tensors and Quantum Information

Quantum computations are then modelled by the application of a certain restricted class of linear maps; called *feasible*.

Often these are maps that act as the identity in some factors of the tensor product, and a *unitary matrix* acting on others.

The standard example is that of a Hadamard matrix

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

or products of the form $H \otimes I_2, H \otimes H$ etc.

Deutsch-Josza Algorithm

Suppose we have a binary function f from $\{0, 1\}^n$ to $\{0, 1\}$, and we are told that it is either *constant*, or *balanced*.

Classically, we would need to evaluate f at $2^{n/2}$ values in order to determine which of these properties it has.

Instead we create a superposition of all states, run it through a single iteration of a feasible operation, and then measure the output.

$$\sum T_{i_1 i_2 \dots, i_n}(e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_n}) \mapsto e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_n}$$

with probability $|T_{i_1 i_2 \dots, i_n}|$

Deutsch-Josza Algorithm

The key is to find a feasible operation such that the possible outputs of the measurement for a constant function are disjoint from those of a balanced function.

That is, the positions with nonzero coefficients in the resulting arrays are disjoint. Given the operation, calculating the coefficients is elementary linear algebra.

So the quantum algorithm can distinguish between constant and balanced in just one operation!

Tensors and Post-Quantum Cryptography

Quantum algorithms can break cryptosystems based on problems assumed difficult for classical computers, e.g. factorisation, elliptic curves.

Linear algebra over finite fields has recently become relevant to Post-Quantum Cryptography, via *rank-metric codes*.

Tensors may well play a role in constructing quantum-proof cryptosystems.

Tensors: Part of the solution, as well as part of the problem!

Thank you for your attention!