A Newton–Grassmann method for computing the Best Multi-Linear Rank-(r_1, r_2, r_3) Approximation of a Tensor

Lars Eldén and Berkant Savas

16 april 2007

LITH-MAT-R–2007-6–SE

Linköpings universitet

Matematiska Institutionen
Linköpings Universitet
581 83 Linköping

Department of Mathematics
Linköpings University
581 83 Linköping

A Newton–Grassmann method for computing the Best Multi-Linear Rank-(r_1, r_2, r_3) Approximation of a Tensor

Lars Eldén and Berkant Savas
Department of Mathematics, Linköping University

April 16, 2007

We derive a Newton method for computing the best rank-(r_1, r_2, r_3) approximation of a given $J \times K \times L$ tensor $\mathcal{A}$. The problem is formulated as an approximation problem on a product of Grassmann manifolds. Incorporating the manifold structure into Newton's method ensures that all iterates generated by the algorithm are points on the Grassmann manifolds. We also introduce a consistent notation for matricizing a tensor, for contracted tensor products and some tensor-algebraic manipulations, which simplify the derivation of the Newton equations and enable straightforward algorithmic implementation. Experiments show a quadratic convergence rate for the Newton-Grassmann algorithm.

1 Introduction

The problem of approximating a tensor $\mathcal{A} \in \mathbb{R}^{J\times K\times L}$ by another tensor $\mathcal{B}$ of equal dimension but of lower rank,

$$\min_{\mathcal{B}} \|\mathcal{A} - \mathcal{B}\|,$$

occurs e.g. in signal processing [14, 4], and pattern classification [18]. Throughout the paper, we will use the Frobenius norm (we will state the precise meaning of this and other concepts in Section 2). There is no unique definition of the rank of a tensor (as opposed to the case of matrices). Here we will deal with the concept of *multi-linear rank* [3] and assume that $\text{rank}(\mathcal{B}) = (r_1, r_2, r_3)$, which means that the tensor $\mathcal{B}$ can be written as a product of a *core* tensor $\mathcal{S}$ and three matrices,

$$\mathcal{B} = (X, Y, Z) \cdot \mathcal{S}, \tag{1}$$

with matrices of full column rank, $X \in \mathbb{R}^{J\times r_1}$, $Y \in \mathbb{R}^{K\times r_2}$, and $Z \in \mathbb{R}^{L\times r_3}$. The tensor $\mathcal{S}$ has dimension $r_1 \times r_2 \times r_3$. It is no restriction to assume that X, Y, and Z have orthonormal columns. Thus we want to solve the problem

$$\min_{\mathcal{S},X,Y,Z} \|\mathcal{A} - (X, Y, Z) \cdot \mathcal{S}\|, \quad \text{subject to} \quad X^T X = I,\ Y^T Y = I,\ Z^T Z = I. \tag{2}$$

The approximation problem is illustrated in Figure 1.

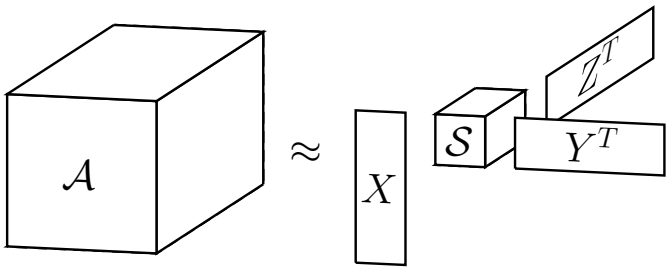

Figure 1: The approximation of a tensor $\mathcal{A}$ by another tensor $\mathcal{B} = (X, Y, Z) \cdot \mathcal{S}$ of lower multi-linear rank.

Unlike the matrix case, there is no known closed form solution of the approximation problem (2). It can be shown that the minimization problem is well-defined [14],[3, Corollary 4.5].

We will restrict ourselves to considering the approximation problem (2) for a $3-$tensor $\mathcal{A}$ in this paper. The main contribution is the derivation of a Newton method for the solution of (2). The constraints on the unknown matrices X, Y, and Z are taken into account by formulating the problem as an optimization problem on a product of three Grassmann manifolds. To be able to differentiate the objective function and derive the Newton equations without extensive index manipulation (as is sometimes used in tensor algebra) we develop an algebraic framework based on tensor contractions. Within this framework it is also straightforward to generalize the derivations to tensors of order 4 and higher, and we sketch this in Section 4.6.

In view of the lack of a standard terminology and notation in the field of tensor computations we define the concepts used in this paper in Section 2. There we also we also propose a "canonical" tensor matricization, contracted tensor products, and a few tensor-algebraic identities. The optimization problem problem on the product of three Grassmann manifolds is formulated in Section 3 and the Newton-Grassmann method is derived in Section 4.2. In Section 5 the numerical implementation of the method is briefly described, and some preliminary numerical experiments are reported.

2 Tensor Concepts and Identities

For simplicity of notation and presentation, we will mostly, in this and the following sections, present the basic concepts using examples in terms of $3-$tensors or $5-$tensors. Some more general definitions are given in [13, 1, 3]. We will use Roman letters written

with a calligraphic font to denote tensors, capital Roman letters to denote matrices (2−tensors), and lower case Roman letters to denote vectors. However, we will also use Roman letters in the the middle of the alphabet, $J, K, L, \ldots$, and $j, k, l, \ldots,$, to denote tensor dimensions and subscripts.

Let $\mathcal{A}$ denote a tensor in $\mathbb{R}^{J\times K\times L}$. The three "dimensions" of the tensor are referred to as *modes*. In the approximation problem (2) we will not consider the tensor as multi-linear operator[1], and therefore there is no need to make a distinction between *contravariant* and *covariant* tensor modes in the tensor notation. We will use both standard subscripts and "MATLAB-like" notation: a particular tensor element will be denoted in two equivalent ways:

$$\mathcal{A}(j,k,l) = a_{jkl}.$$

We will refer to subtensors in the following way. A subtensor obtained by fixing one of the indices is called a *slice*, e.g.,

$$\mathcal{A}(j,:,:).$$

A *fibre* is a subtensor, where all indices but one are fixed,

$$\mathcal{A}(j,:,l).$$

When in the following we use tensors, matrices and vectors in operations that we will define, it is assumed that the dimensions of the respective quantities are conforming in the sense that all the operations are well-defined.

2.1 Tensor–Matrix Multiplication

Even if our tensors are not primarily linear operators with contravariant and covariant modes, it is convenient to define two variants of tensor–matrix multiplication. We first define the mode−p *contravariant multiplication of a tensor by a matrix*. For concreteness we first let $p = 1$. The mode−1 product of a tensor $\mathcal{A} \in \mathbb{R}^{J\times K\times L}$ by a matrix $W \in \mathbb{R}^{M\times J}$ is defined by

$$\mathbb{R}^{M\times K\times L} \ni \mathcal{B} = (W)_1 \cdot \mathcal{A}, \qquad \mathcal{B}(m,k,l) = \sum_{j=1}^{J} a_{jkl} w_{mj}. \tag{3}$$

This means that all column vectors (mode−1 fibres) in the 3−tensor are multiplied by the matrix W. Similarly, mode−2 multiplication by a matrix X means that all row vectors (mode−2 fibres) are multiplied by the matrix X. Mode−3 multiplication is analogous.

It is easy to see that for integers $p \neq q$, mode−p and mode−q multiplication commute:

$$(W)_p \cdot ((X)_q \cdot \mathcal{A}) = (X)_q \cdot ((W)_p \cdot \mathcal{A}), \qquad p \neq q.$$

[1]However, when we derive the Newton equations for solving the minimization problem, then we will deal with a Hessian, which, of course, is a linear operator constructed in terms of tensors.

Therefore it makes sense to define

$$(W, X)_{p,q} \cdot \mathcal{A} = (W)_p \cdot ((X)_q \cdot \mathcal{A}).$$

Obviously the following identity holds,

$$(W_1)_p \cdot ((W_2)_p \cdot \mathcal{A}) = (W_1 W_2)_p \cdot \mathcal{A}, \tag{4}$$

where the matrix and tensor dimensions are assumed to be conforming, and the product $W_1 W_2$ is standard matrix multiplication.

In the case when tensor-matrix multiplication is performed in all modes in the same formula, we omit the subscripts and write

$$(X, Y, Z) \cdot \mathcal{A}, \tag{5}$$

where the mode of each multiplication is understood from the order in which the matrices are given. Thus, we have the identity

$$(Y)_2 \cdot \mathcal{A} = (I, Y, I) \cdot \mathcal{A}.$$

The notation (5) was suggested by Lim[2] [3].

One can also write the standard matrix multiplication of three matrices in the form

$$XFY^T = (X, Y) \cdot F, \tag{6}$$

where, at the same time, F is considered as a matrix and a $2-$tensor.

Covariant multiplication, cf. [11, Chapter 2], by a matrix $V \in \mathbb{R}^{J \times M}$ is defined

$$\mathbb{R}^{M \times K \times L} \ni \mathcal{C} = \mathcal{A} \cdot (W)_1, \qquad \mathcal{C}(m, k, l) = \sum_{j=1}^{J} a_{jkl} w_{jm}. \tag{7}$$

Obviously we have the following relation between contravariant and covariant multiplication:

$$(X^T)_p \cdot \mathcal{A} = \mathcal{A} \cdot (X)_p. \tag{8}$$

2.2 A "Canonical" Tensor Matricization

In the following sections we will occasionally rearrange the elements of a tensor so that they form a matrix[3]. We will refer to this as *matricizing* the tensor[4]. Sometimes the matricization is performed *along one specific mode* [13, 10, 17]. Given an $N-$tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ its matricization along the n–th mode is a matrix of dimensions $I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N$. Here we will introduce a more general tensor matricization[5] which is intuitively and directly related to the matrix–tensor multiplication. In

[2] An alternative notation was given in [13].

[3] In particular, when the Newton equations are to be solved numerically, they must be arranged as standard "matrix-vector" linear equations.

[4] Alternative terms are *unfolding* [13] or *flattening* [17].

[5] A similar, but not identical, generalized matricization is given in Bader and Kolda [1, 2]. The difference between the two definitions is explained later in this section.

this matricization we will map some modes of a tensor to the rows of the matrix and the rest to the columns.

Let $\mathsf{r} = \{r_1, \cdots, r_L\}$ be the modes of $\mathcal{A}$ mapped to the rows and $\mathsf{c} = \{c_1, \cdots, c_M\}$ be the modes of $\mathcal{A}$ mapped to the columns. The matricization is denoted

$$A^{(\mathsf{r};\mathsf{c})} \in \mathbb{R}^{J\times K}, \quad \text{where} \quad J = \prod_{i=1}^{L} I_{r_i}, \quad \text{and} \quad K = \prod_{i=1}^{M} I_{c_i}. \tag{9}$$

Of course many different one–to–one functions can map the tensor $\mathcal{A}$ onto a matrix with dimensions as specified in (9). The different maps differ in the ordering of the row– and column–indices of specific tensor elements.

We consider it useful, for analysis and consistency with tensor–matrix products, if the matricization operation has the following properties, which are best illustrated with a few examples. Let $\mathcal{A}$ be a 5–tensor and consider the product $\mathcal{B} = \mathcal{A} \cdot (X, Y, Z, U, V)$ where X, Y, Z, U, V are matrices of appropriate dimensions multiplied with $\mathcal{A}$ along its different modes. Here $\otimes$ denotes Kronecker product of matrices.

$$\begin{aligned}
B^{(2;1,3:5)} \equiv B^{(2)} &= Y^T A^{(2)}(X \otimes Z \otimes U \otimes V), && \mathsf{r} = \{2\},\ \mathsf{c} = \{1,3,4,5\},\\
B^{(3,2;1,4,5)} \equiv B^{(3,2)} &= (Z \otimes Y)^T A^{(3,2)}(X \otimes U \otimes V), && \mathsf{r} = \{3,2\},\ \mathsf{c} = \{1,4,5\},\\
B^{(2,4,1;5,3)} &= (Y \otimes U \otimes X)^T A^{(2,4,1;5,3)}(V \otimes Z), && \mathsf{r} = \{2,4,1\},\ \mathsf{c} = \{5,3\},\\
B^{(1,2,4;5,3)} \equiv B^{(;5,3)} &= (X \otimes Y \otimes U)^T A^{(;5,3)}(V \otimes Z), && \mathsf{r} = \{1,2,4\},\ \mathsf{c} = \{5,3\}.
\end{aligned}$$

Observe that the ordering of the matrices in the Kronecker products is specified by the matricization indices r and c. Specifying only the row (column) modes assumes the column (row) modes to be in increasing order. In the above examples we have used the covariant multiplication. For contravariant multiplication the transpose will be introduced on the other side. For instance, with $\mathcal{C} = (X, Y, Z, U, V) \cdot \mathcal{A}$ we have

$$\begin{aligned}
C^{(2)} &= Y A^{(2)}(X \otimes Z \otimes U \otimes V)^T, && \mathsf{r} = \{2\}, && \mathsf{c} = \{1,3,4,5\},\\
C^{(3,2)} &= (Z \otimes Y) A^{(3,2)}(X \otimes U \otimes V)^T, && \mathsf{r} = \{3,2\}, && \mathsf{c} = \{1,4,5\}.
\end{aligned}$$

For a given an N–tensor $\mathcal{A}$, the matricization to $A^{(\mathsf{r};\mathsf{c})}$ has the desired properties, if the element $\mathcal{A}(i_1, \ldots, i_N)$ is mapped to $A^{(\mathsf{r};\mathsf{c})}(j, k)$ where

$$j = 1 + \sum_{l=1}^{L} \left[\left(i_{r_{L-l+1}} - 1\right) \prod_{l'=1}^{l-1} I_{r_{L-l'+1}} \right], \tag{10}$$

$$k = 1 + \sum_{m=1}^{M} \left[\left(i_{c_{M-m+1}} - 1\right) \prod_{m'=1}^{m-1} I_{c_{M-m'+1}} \right]. \tag{11}$$

The matricization mapping presented in Bader and Kolda [1, 2] is different from ours in that it reverses the ordering of the matrices in both sides of matricized forms[6] of the matrix–tensor products.

[6]For example, in the Bader-Kolda mapping the matricization of $\mathcal{B}$ would be $B^{(2,4,1;5,3)} = (X \otimes U \otimes Y)^T A^{(2,4,1;5,3)}(Z \otimes V)$.

Applying the matricizing on the matrix products $B = (X, Y) \cdot A = XAY^T$ we obtain

$$B^{(1)} = XA^{(1)}Y^T, \qquad B^{(2)} = YA^{(2)}X^T.$$

Of course, this is trivial since for matrices $A^{(1)} \equiv A$ and $A^{(2)} \equiv A^T$.

Observe that this framework enables vectorization as well. Then one of r or c has to be the empty set, denoted $\emptyset$ and the other contains all modes. Consider first the matrix case, $B = (X, Y) \cdot A = XAY^T$. Vectorizing B with $\mathsf{r} = \{1, 2\}$ and $\mathsf{c} = \emptyset$ we obtain

$$B^{(1,2;\emptyset)} = (X \otimes Y)A^{(1,2;\emptyset)},$$

where $A^{(1,2;\emptyset)}$ and $B^{(1,2;\emptyset)}$ are the row-wise vectorizations of A and B, giving a column vector. Changing the row modes to $\mathsf{r} = \{2, 1\}$ we obtain the more familiar

$$B^{(2,1;\emptyset)} = \mathrm{vec}(B) = \mathrm{vec}(XAY^T) = (Y \otimes X)\,\mathrm{vec}(A) = (Y \otimes X)A^{(2,1;\emptyset)},$$

where, by convention, $\mathrm{vec}(\cdot)$ denotes the column-wise vectorization. Further, with a 3-tensor $\mathcal{B} = \mathcal{A} \cdot (X, Y, Z)$ we have

$$B^{(2,1,3;\emptyset)} = (Y \otimes X \otimes Z)^T A^{(2,1,3;\emptyset)}, \quad \text{and} \quad B^{(\emptyset;2,1,3)} = A^{(\emptyset;2,1,3)}(Y \otimes X \otimes Z),$$

where in the first case the vectorization gives a column vector and in the second the vectorization gives a row vector.

Finally, for later reference, we specify two special cases with tensor–matrix product along one mode only. Let $\mathcal{A}$ be a general N–tensor. Then

$$\mathcal{B} = \mathcal{A} \cdot (X)_p, \qquad \Leftrightarrow \qquad B^{(p)} = X^T A^{(p)}, \tag{12}$$

$$\mathcal{C} = (X)_p \cdot \mathcal{A}, \qquad \Leftrightarrow \qquad C^{(p)} = XA^{(p)}. \tag{13}$$

The notation in this paper emphasizes the connection between multi-linear tensor-matrix products and their matricized form. Other notations are found in [12, 13, 10, 8].

2.3 Inner Product, Tensor Product, and Contracted Product

Given two tensors $\mathcal{A}$ and $\mathcal{B}$ of the same dimensions, we define the *inner product*

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{j,k,l} a_{jkl} b_{jkl}. \tag{14}$$

The corresponding *tensor norm* is

$$\|\mathcal{A}\| = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}. \tag{15}$$

This *Frobenius norm* will be used throughout the paper. As in the matrix case the norm is invariant under orthogonal transformations, i.e.

$$\|\mathcal{A}\| = \|(U, V, W) \cdot \mathcal{A}\| = \|\mathcal{A} \cdot (U, V, W)\|,$$

for orthogonal matrices U, V, and W. This follows immediately from the fact that mode$-p$ multiplication by an orthogonal matrix does not change the Euclidean length of the mode$-p$ fibres.

The product of two tensors, $\mathcal{A} \in \mathbb{R}^{J\times K\times L}$ and $\mathcal{B} \in \mathbb{R}^{M\times N}$, say, is a tensor of higher dimensionality, here a 5$-$tensor,

$$\mathbb{R}^{J\times K\times L\times M\times N} \ni \mathcal{C} = \mathcal{A} \otimes \mathcal{B}, \qquad c_{jklmn} = a_{jkl} b_{mn}.$$

This is the *tensor product*, or outer product[7].

The inner product (14) can be considered as a special case of the *contracted product of two tensors*, which is a tensor (outer) product followed by a contraction along specified modes. Thus, if $\mathcal{A}$ and $\mathcal{B}$ are 3$-$tensors of equal dimensions, we define, using essentially the notation of [1],

$$\begin{aligned}
\mathcal{C} &= \langle \mathcal{A}, \mathcal{B} \rangle_1 , & c_{jklm} &= \sum_{\lambda} a_{\lambda jk} b_{\lambda lm} , & &(4\text{−tensor}) ,\\
D &= \langle \mathcal{A}, \mathcal{B} \rangle_{1:2} , & d_{jk} &= \sum_{\lambda,\mu} a_{\lambda\mu j} b_{\lambda\mu k} , & &(2\text{−tensor}),\\
e &= \langle \mathcal{A}, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \rangle_{1:3} , & e &= \sum_{\lambda,\mu,\nu} a_{\lambda\mu\nu} b_{\lambda\mu\nu} , & &(\text{scalar}).
\end{aligned}$$

We will refer to the first two as *partial contractions*.

Observe that we let the ordering of the modes in contracted tensor products be implicitly given in the summation. Thus given $\mathcal{A} \in \mathbb{R}^{J\times K\times L}$ and $\mathcal{B} \in \mathbb{R}^{J\times M\times N}$, then

$$\mathcal{C} = \langle \mathcal{A}, \mathcal{B} \rangle_1 \in \mathbb{R}^{K\times L\times M\times N}.$$

In general, the modes of the product are given by the ordering of the non-contracted modes of the first argument followed by the ordering of the non-contracted modes of the second argument.

We will also use negative subscripts when the the contraction is made in all but a few modes. For 3-tensors we have

$$\langle \mathcal{A}, \mathcal{B} \rangle_{2:3} \equiv \langle \mathcal{A}, \mathcal{B} \rangle_{-1} , \qquad \langle \mathcal{A}, \mathcal{B} \rangle_{2} \equiv \langle \mathcal{A}, \mathcal{B} \rangle_{-(1,3)} .$$

The contracted product can be defined also for tensors of different numbers of modes. For example, with a 4$-$tensor $\mathcal{F}$ and matrices (2$-$tensors) F and G,

$$\langle \mathcal{A}, F \rangle_{3:4;1:2} = G, \qquad \sum_{\mu,\nu} a_{jk\mu\nu} f_{\mu\nu} = g_{jk}, \tag{16}$$

defines a linear system of equations.

In the following sections we will need a number of lemmas. The first result relates contraction to matricization.

[7] Often the same notation, $\otimes$, is used for the Kronecker product of matrices, which is the tensor product of two matrices (2$-$tensors), followed by a particular matricization of the 4$-$tensor.

Lemma 2.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be N–tensors of matching dimensions in all but (possibly) the ith mode and $A^{(i)}$ and $B^{(i)}$ the corresponding ith mode matricizations. Then*

$$\langle \mathcal{A}, \mathcal{B} \rangle_{-i} = A^{(i)} B^{(i)T}, \tag{17}$$

and

$$\langle \mathcal{A}, \mathcal{B} \rangle = \mathrm{tr}(A^{(i)} B^{(i)T}) = \mathrm{tr}(\langle \mathcal{A}, \mathcal{B} \rangle_{-i}). \tag{18}$$

Proof. For simplicity we give the proof only for 3–tensors and partial contraction in all but the first mode. The general case is completely analogous. Let $\mathcal{A} \in \mathbb{R}^{J \times L \times M}$ and $\mathcal{B} \in \mathbb{R}^{K \times L \times M}$. Then

$$\langle \mathcal{A}, \mathcal{B} \rangle_{-1}(j,k) = \sum_{l,m} a_{jlm} b_{klm}. \tag{19}$$

With $C = A^{(1)} B^{(1)T}$ we get

$$C(j,k) = \sum_{\lambda} a^{(1)}_{j\lambda} b^{(1)}_{k\lambda}, \tag{20}$$

where $A^{(1)}(j,\lambda) = a^{(1)}_{j\lambda}$ and $B^{(1)}(k,\lambda) = b^{(1)}_{k\lambda}$. By equation (11) element $\mathcal{A}(j,l,m)$ is mapped to $A^{(1)}(j,\lambda)$ where $\lambda = m + (l-1)M$, and similarly for elements of $\mathcal{B}$. The equality of (17) follows by observing that the λ–summation for the right hand side actually consists of a summation over m and l.

The identity (18) follows from (19) by inspection. □

The partial contracted products of two matrices A and B are

$$\langle A, B \rangle_{-2} = \langle A, B \rangle_1 = A^T B, \qquad \langle A, B \rangle_{-1} = \langle A, B \rangle_2 = A B^T, \tag{21}$$

which shows that partial contraction is related to matrix transposition. In the next lemma we show that partial contractions play the role of taking the adjoint with respect to the inner product (14).

Lemma 2.2. *Let the N–tensors $\mathcal{B}$ and $\mathcal{C}$ and the matrix Q be of conforming dimensions. Then*

$$\langle \mathcal{B} \cdot (Q)_i, \mathcal{C} \rangle = \langle Q, \langle \mathcal{B}, \mathcal{C} \rangle_{-i} \rangle, \tag{22}$$

$$\langle \mathcal{B} \cdot (Q)_i, \mathcal{C} \rangle_{-i} = Q^T \langle \mathcal{B}, \mathcal{C} \rangle_{-i} = \langle Q, \langle \mathcal{B}, \mathcal{C} \rangle_{-i} \rangle_1, \tag{23}$$

$$\langle \mathcal{B}, \mathcal{C} \cdot (Q^T)_i \rangle_{-i} = \langle \mathcal{B}, \mathcal{C} \rangle_{-i} Q^T = \langle \langle \mathcal{B}, \mathcal{C} \rangle_{-i}, Q \rangle_2. \tag{24}$$

Proof. Equation (22) follows from

$$\begin{aligned} \langle Q, \langle \mathcal{B}, \mathcal{C} \rangle_{-i} \rangle &= \langle Q, B^{(i)} C^{(i)T} \rangle = \mathrm{tr}(Q^T B^{(i)} C^{(i)T}) \\ &= \mathrm{tr}((\mathcal{B} \cdot (Q)_i)^{(i)} C^{(i)T} \rangle = \langle \mathcal{B} \cdot (Q)_i, \mathcal{C} \rangle, \end{aligned}$$

where we have used (18) and (12). The second and third identities follow directly by matricizing the expressions along the ith mode and using (17). □

The following lemma can be motivated as follows: Obviously, from the definition of contracted product, the mapping

$$Q \longrightarrow \langle \mathcal{B} \cdot (Q)_j, \mathcal{C} \rangle_{-i}$$

is linear from matrices to matrices. In order to solve a linear system involving such a mapping we need to write it in the form (16).

Lemma 2.3. *Let the N–tensors $\mathcal{B}$ and $\mathcal{C}$ and the matrix Q be of conforming dimensions. If $j \neq i$ then*

$$\langle \mathcal{B} \cdot (Q)_j, \mathcal{C} \rangle_{-i} = \begin{cases} \langle \langle \mathcal{B}, \mathcal{C} \rangle_{-(i,j)}, Q \rangle_{1,3;1,2} & \text{if } j < i \\ \langle \langle \mathcal{B}, \mathcal{C} \rangle_{-(i,j)}, Q \rangle_{2,4;1,2} & \text{if } j > i \end{cases} \tag{25}$$

$$\langle \mathcal{B}, \mathcal{C} \cdot (Q)_j \rangle_{-i} = \begin{cases} \langle \langle \mathcal{B}, \mathcal{C} \rangle_{-(i,j)}, Q \rangle_{3,1;1,2} & \text{if } j < i \\ \langle \langle \mathcal{B}, \mathcal{C} \rangle_{-(i,j)}, Q \rangle_{4,2;1,2} & \text{if } j > i \end{cases} \tag{26}$$

The proof is given in the appendix.

2.4 Multi-Linear Rank and Higher Order SVD

The *multi-linear* rank of a $3-$tensor is a triplet (r_1, r_2, r_3) such that

$$r_i = \dim(R(A^{(i)}) = \operatorname{rank}(A^{(i)}), \qquad i = 1, 2, 3,$$

where $R(A) = \{y \,|\, y = Ax\}$ is the *range space* of the matrix A, and $\operatorname{rank}(A)$ is the matrix rank. Multi-linear rank is discussed in [3], as well as other rank concepts. In this paper we will only deal with multi-linear rank, and we will use the notation $\operatorname{rank}-(r_1, r_2, r_3)$, and $\operatorname{rank}(\mathcal{A}) = (r_1, r_2, r_3)$.

For matrices the rank is obtained via the *Singular Value Decomposition (SVD)*, see e.g. [7, Chapter 2]. One generalization of the SVD to tensors, the *Higher Order SVD*, was given in [13]. We here present the HOSVD for the case when $\mathcal{A}$ is a $3-$tensor. The general case is an obvious generalization.

Theorem 2.4 (HOSVD)**.** *Any $3-$tensor $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$ can be factorized*

$$\mathcal{A} = (U, V, W) \cdot \mathcal{S}, \tag{27}$$

where $U \in \mathbb{R}^{J \times J}$, $V \in \mathbb{R}^{K \times K}$, and $W \in \mathbb{R}^{L \times L}$, are orthogonal matrices, and $\mathcal{S} \in \mathbb{R}^{J \times K \times L}$ is all-orthogonal*: the matrices $\langle \mathcal{S}, \mathcal{S} \rangle_{-i}$, $i = 1, 2, 3$, are diagonal, and*

$$\|\mathcal{S}(1, :, :)\| \geq \|\mathcal{S}(2, :, :)\| \geq \cdots \geq 0, \tag{28}$$

$$\|\mathcal{S}(:, 1, :)\| \geq \|\mathcal{S}(:, 2, :)\| \geq \cdots \geq 0, \tag{29}$$

$$\|\mathcal{S}(:, :, 1)\| \geq \|\mathcal{S}(:, :, 2)\| \geq \cdots \geq 0, \tag{30}$$

are the $1-$mode, $2-$mode, and $3-$mode singular values, also denoted $\sigma_i^{(1)}$, $\sigma_i^{(2)}$, $\sigma_i^{(3)}$.

Partitioning the orthogonal matrices in terms of columns, $U = (u_1, \ldots, u_J)$, $V = (v_1, \ldots, v_K)$, $W = (w_1, \ldots, w_L)$, the HOSVD equation can be written

$$\mathcal{A} = \sum_{j,k,l} s_{jkl}\, u_j \otimes v_k \otimes w_l,$$

where and $\otimes$ denotes the tensor product: For vectors x, y and z we have

$$(x \otimes y \otimes z)_{\lambda\mu\nu} = x_\lambda y_\mu z_\nu.$$

Assume that the higher order singular values of $\mathcal{A}$ satisfy

$$\begin{aligned} \sigma^{(1)}_{r_1} > 0, \qquad & \sigma^{(1)}_{r_1+1} = 0, \\ \sigma^{(2)}_{r_2} > 0, \qquad & \sigma^{(2)}_{r_2+1} = 0, \\ \sigma^{(3)}_{r_3} > 0, \qquad & \sigma^{(3)}_{r_3+1} = 0, \end{aligned}$$

for some constants r_1, r_2, and r_3. It is easy to show that in this case the multi-linear rank of $\mathcal{A}$ is (r_1, r_2, r_3).

3 Best Rank$-(r_1, r_2, r_3)$ Approximation

Assume that we want to approximate, using the norm (15), the tensor $\mathcal{A}$ by another tensor $\mathcal{B}$ of rank (r_1, r_2, r_3). Thus we want to solve

$$\min_{\text{rank}(\mathcal{B})=(r_1,r_2,r_3)} \|\mathcal{A} - \mathcal{B}\|. \tag{31}$$

This problem is treated in [14]. In the matrix case, the solution of the corresponding problem is given by the truncated SVD (the Eckart-Young property; a simple proof is given in [6, Theorem 6.7]). In view of the fact that the HOSVD "orders the mass" of the tensor in a similar way as the SVD, see (28)-(30), one might think that a truncated HOSVD would give the solution of (31). However, this is not the case [14].

Some theoretical questions concerning the best rank$-(r_1, r_2, r_3)$ approximation problem are studied in [3]. In particular the following result is proved.

Proposition 3.1. *Every $k-$tensor $\mathcal{A}$ has a best approximation $\mathcal{B}$ with*

$$\text{rank}(\mathcal{B}) \leq (r_1, r_2, \ldots, r_k)$$

for any specified $(r_1, r_2, \ldots, r_k)$.

The rank constraint in (31) implies (see [3, 14] and Section 2.4) that $\mathcal{B}$ can be written

$$\mathcal{B} = (X, Y, Z) \cdot \overline{\mathcal{B}}, \qquad \overline{\mathcal{B}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$$

where $X \in \mathbb{R}^{J \times r_1}$, $Y \in \mathbb{R}^{K \times r_2}$, and $Z \in \mathbb{R}^{L \times r_3}$, with

$$X^T X = I, \qquad Y^T Y = I, \qquad Z^T Z = I. \tag{32}$$

The identity matrices in (32) have dimensions r_1, r_2, and r_3, respectively.

Define three orthogonal matrices,

$$\begin{aligned}
\widehat{X} &= \begin{pmatrix} X & X_\perp \end{pmatrix}, & X_\perp &\in \mathbb{R}^{J\times(J-r_1)},\\
\widehat{Y} &= \begin{pmatrix} Y & Y_\perp \end{pmatrix}, & Y_\perp &\in \mathbb{R}^{K\times(K-r_2)},\\
\widehat{Z} &= \begin{pmatrix} Z & Z_\perp \end{pmatrix}, & Z_\perp &\in \mathbb{R}^{L\times(L-r_3)}.
\end{aligned}$$

In transformed coordinates, i.e., with $\widehat{\mathcal{A}} = (\widehat{X}^T, \widehat{Y}^T, \widehat{Z}^T)\cdot\mathcal{A}$ and $\widehat{\mathcal{B}} = (\widehat{X}^T, \widehat{Y}^T, \widehat{Z}^T)\cdot\mathcal{B}$, the residual in the approximation problem becomes

$$\begin{aligned}
\|\widehat{\mathcal{A}} - \widehat{\mathcal{B}}\|^2 &= \sum_{j=1}^{r_1}\sum_{k=1}^{r_2}\sum_{l=1}^{r_3}(\hat{a}_{jkl} - \hat{b}_{jkl})^2 + \sum_{j=r_1+1}^{J}\sum_{k=r_2+1}^{K}\sum_{l=r_3+1}^{L}(\hat{a}_{jkl} - \hat{b}_{jkl})^2\\
&= \sum_{j=1}^{r_1}\sum_{k=1}^{r_2}\sum_{l=1}^{r_3}(\hat{a}_{jkl} - \hat{b}_{jkl})^2 + \sum_{j=r_1+1}^{J}\sum_{k=r_2+1}^{K}\sum_{l=r_3+1}^{L}\hat{a}_{jkl}^2\ ,
\end{aligned}$$

due to the rank constraint. The first term can be made equal to zero by choosing $\hat{a}_{jkl} - \hat{b}_{jkl}$, and the residual becomes

$$\|\widehat{\mathcal{A}} - \widehat{\mathcal{B}}\|^2 = \sum_{j=r_1+1}^{J}\sum_{k=r_2+1}^{K}\sum_{l=r_3+1}^{L}\hat{a}_{jkl}^2.$$

We now see that the problem of solving (31), i.e. making the residual as small as possible, is equivalent to determining X, Y, and Z so that

$$\|(X^T, Y^T, Z^T)\cdot\mathcal{A}\| = \|\mathcal{A}\cdot(X, Y, Z)\|$$

is maximized. We thus define the objective function to be maximized,

$$\Phi(X, Y, Z) = \frac{1}{2}\|\mathcal{A}\cdot(X, Y, Z)\|^2 = \frac{1}{2}\sum_{j,k,l}A_{jkl}^2, \quad A_{jkl} = \sum_{\lambda,\mu,\nu}a_{\lambda\mu\nu}x_{\lambda j}y_{\mu k}z_{\nu l}, \tag{33}$$

where $x_{\lambda j}$, $y_{\mu k}$, and $z_{\nu l}$ are elements of X, Y, and Z, respectively.

4 Solving the Maximization Problem by Newton's Method

It follows from the invariance of the norm under orthogonal transformations that

$$\Phi(X, Y, Z) = \Phi(XU, YV, ZW), \tag{34}$$

for orthogonal matrices $U \in \mathbb{R}^{r_1\times r_1}$, $V \in \mathbb{R}^{r_2\times r_2}$, and $W \in \mathbb{R}^{r_3\times r_3}$. This means that the problem of maximizing Φ under the orthogonality constraint (32) is is not yet well-defined: the problem is over-parameterized, and any straightforward constrained optimization method would have difficulties. It follows that we should maximize the function

Φ not just over matrices with orthonormal columns but over equivalence classes of such matrices, for instance,

$$[X] = \{XU \mid U \text{ orthogonal}\}. \tag{35}$$

This means that we should maximize over the *Grassmann manifold* [5], or more precisely, over a product of Grassmann manifolds.

4.1 Newton's Method on the Grassmann Manifold

The Grassmann manifold can be considered as a set of equivalence classes of matrices (35) with orthonormal columns that span the same subspace, see [5]. Here we give a very brief description of Newton's method for maximizing a function $G(X)$ defined on the Grassmann manifold, and then we state Newton's method on the product manifold.

Assume that $X \in \mathbb{R}^{J\times r_1}$ is a point on the Grassmann manifold $\mathrm{Gr}(J, r_1)$. A *tangent vector* $\Delta \in \mathbb{R}^{J\times r_1}$ at X satisfies

$$X^T\Delta = 0,$$

and the *tangent space* at X, consisting of all tangent vectors at X and denoted $\mathbb{T}_X$, is a linear space. The projection on the tangent space is

$$\Pi_X = I - XX^T. \tag{36}$$

The canonical metric (inner product) of the Grassmann manifold is

$$\langle \Delta_1, \Delta_2 \rangle = \mathrm{tr}(\Delta_1^T\Delta_2), \tag{37}$$

where Δ_1 and Δ_2 are tangent vectors.

Let Δ be a tangent vector at X, and let $X(t)$ be a parameterization of a geodesic curve in the direction Δ. With the thin SVD, $\Delta = U\Sigma V^T$, where $U \in \mathbb{R}^{J\times r_1}$, and $\Sigma \in \mathbb{R}^{r_1\times r_1}$, the geodesic is given by

$$X(t) = XV\cos(t\Sigma)V^T + U\sin(t\Sigma)V^T. \tag{38}$$

In Newton's method for maximizing a function $G(t)$, with $t = 0$ as a tentative maximizer, we approximate the function by the first three terms of the Maclaurin expansion,

$$G(t) \approx G(0) + t\left.\frac{dG}{dt}\right|_{t=0} + \frac{t^2}{2}\left.\frac{d^2G}{dt^2}\right|_{t=0},$$

and then we maximize the second degree polynomial in t. In Newton's method on the Grassmann manifold the objective of the quadratic approximation is to determine a tangent vector Δ that maximizes a second degree function

$$G(X(t)) \approx G(X) + \langle \Delta, \nabla G\rangle + \frac{1}{2}\langle \Delta, H(\Delta)\rangle, \tag{39}$$

where $\langle\cdot,\cdot\rangle$ is the inner product (37). ∇G is the gradient on the tangent space,

$$\nabla G = \Pi_X G_x, \qquad (G_x)_{jk} = \frac{\partial G}{\partial x_{jk}}, \tag{40}$$

and the Hessian $H(\Delta)$ is a linear operator on the tangent space, $\mathbb{T}_X \ni \Delta \longrightarrow H(\Delta) \in \mathbb{T}_X$. It is shown in [5] that the Newton equation for determining $\Delta \in \mathbb{T}_X$ is a Sylvester-like equation, which in our notation becomes

$$\Pi_X \langle \mathcal{G}_{xx}, \Delta \rangle_{1:2} - \Delta \langle X, G_x \rangle_1 = -\nabla G, \qquad (\mathcal{G}_{xx})_{jklm} = \frac{\partial^2 G}{\partial X_{jk}\, \partial X_{lm}}. \tag{41}$$

Here the contracted product of the 4–tensor $\mathcal{G}_{xx}$ and the matrix Δ defines a linear operator. $\langle \mathcal{G}_{xx}, \Delta \rangle_{1:2}$ is a matrix, which can be multiplied by Π_X to project it to the tangent space $\mathbb{T}_X$.

In order to solve the Newton equation (41) numerically, there are essentially three approaches:

Solve the problem in the ambient Euclidean space Using the coordinates given by X itself, we could disregard that the problem is defined on the Grassmann manifold and solve the Newton equation in the ambient Euclidean space $\mathbb{R}^{Jr_1}$. Since X is constrained, i.e. $X^T X = I$, the overparameterized coordinate representation will cause the Newton-Grassmann equation (41) to be singular. A pseudoinverse solution combined with a projection might be used to keep the iterates on the manifold.

Solve the problem on the tangent space The Newton-Grassmann equation (41) is non-singular in the neighbourhood of a local maximum when considered *on the tangent space* $\mathbb{T}_X$. Using a coordinate representation on the tangent space one can obtain a smaller problem with a full rank Hessian operator. We will do this in the case of a product manifold in Section 4.3.

Solve the problem by introducing Lagrange multipliers The third approach, which is more efficient for large problems with $J \gg r_1$, is to effectively introduce Lagrange multipliers for the constraint and simultaneously solve for those and Δ, see e.g. [15, Alg. 2].

4.2 Newton's Method on the Product Manifold

Our constrained optimization problem is

$$\max_{(X,Y,Z)\in \mathrm{Gr}^3} \Phi(X,Y,Z), \qquad \mathrm{Gr}^3 = \mathrm{Gr}(J, r_1) \times \mathrm{Gr}(K, r_2) \times \mathrm{Gr}(L, r_3), \tag{42}$$

where the objective function is defined in (33). The tangent space at (X, Y, Z) is $\mathbb{T}^3 = \mathbb{T}_X \times \mathbb{T}_Y \times \mathbb{T}_Z$, and the inner product is the sum of the inner products on the respective manifolds. We will now derive the Newton equation on the product manifold corresponding to (41). First we will differentiate Φ in the direction of a geodesic curve, and then we will identify the terms in the expansion corresponding to (39).

A geodesic curve in the direction $(\Delta_x, \Delta_y, \Delta_z)$ is given by $(X(t), Y(t), Z(t))$, where the components are defined according to (38). From the definition of a tangent vector

(see also (38)) we have

$$\frac{\partial x_{st}}{\partial t} = (\Delta_x)_{st},$$

and corresponding in the other two directions. We therefore get

$$\left(\frac{dX(t)}{dt}\,,\,\frac{dY(t)}{dt}\,,\,\frac{dZ(t)}{dt}\right) = (\Delta_x\,,\,\Delta_y\,,\,\Delta_z),$$

and since

$$\mathcal{A}\cdot(X,Y,Z)(j,k,l) = \sum_{\lambda,\mu,\nu} a_{\lambda\mu\nu}x_{\lambda j}y_{\mu k}z_{\nu l},$$

every $x_{\lambda j}$ etc. will be replaced by $(\Delta_x)_{\lambda j}$ etc. in the differentiation of $\mathcal{A}\cdot(X,Y,Z)$:

$$\frac{d(\mathcal{A}\cdot(X,Y,Z))}{dt} = \mathcal{A}\cdot(\Delta_x,Y,Z) + \mathcal{A}\cdot(X,\Delta_y,Z) + \mathcal{A}\cdot(X,Y,\Delta_z).$$

Grassmann Gradient The first derivative of Φ becomes

$$\begin{aligned}\frac{d\Phi}{dt} &= \frac{1}{2}\frac{d}{dt}\langle\mathcal{A}\cdot(X,Y,Z),\mathcal{A}\cdot(X,Y,Z)\rangle \\ &= \langle\mathcal{A}\cdot(\Delta_x,Y,Z),\mathcal{A}\cdot(X,Y,Z)\rangle &(43)\\ &+ \langle\mathcal{A}\cdot(X,\Delta_y,Z),\mathcal{A}\cdot(X,Y,Z)\rangle &(44)\\ &+ \langle\mathcal{A}\cdot(X,Y,\Delta_z),\mathcal{A}\cdot(X,Y,Z)\rangle. &(45)\end{aligned}$$

First we will identify the gradient, and to do this we need to rewrite (43)-(45) in the form of the derivative term in (39).

It is convenient to define the tensor $\mathcal{F} = \mathcal{A}\cdot(X,Y,Z)$, since it will be used in many expressions. From (22) we see that

$$\langle\mathcal{A}\cdot(\Delta_x,Y,Z),\mathcal{F}\rangle = \langle\Delta_x,\langle\mathcal{A}\cdot(I,Y,Z),\mathcal{F}\,\rangle_{-1}\rangle =: \langle\Delta_x,\Phi_x\rangle, \tag{46}$$

and corresponding for the other terms in (44) and (45). The X-part of the Grassmann gradient (see (40)) then becomes

$$\begin{aligned}\Pi_X\Phi_x &= \Pi_X\langle\mathcal{A}\cdot(I,Y,Z),\mathcal{F}\rangle_{-1} \\ &= \langle\mathcal{A}\cdot(I,Y,Z),\mathcal{A}\cdot(X,Y,Z)\rangle_{-1} - XX^T\langle\mathcal{A}\cdot(I,Y,Z),\mathcal{F}\rangle_{-1} \\ &= \langle\mathcal{A}\cdot(I,Y,Z),\mathcal{A}\cdot(I,Y,Z)\rangle_{-1}X - X\langle\mathcal{F},\mathcal{F}\rangle_{-1}, &(47)\end{aligned}$$

where we have used Lemma 2.1, (23) and (24). The factors in (47) have an interpretation in terms of subtensors: $\mathcal{F}$ is a tensor in $\mathbb{R}^{r_1\times r_2\times r_3}$ and the contracted product

$$\langle\mathcal{F},\mathcal{F}\rangle_{-1} = \langle\mathcal{F},\mathcal{F}\rangle_{2:3} = \langle\mathcal{A}\cdot(X,Y,Z),\mathcal{A}\cdot(X,Y,Z)\rangle_{2:3}$$

is a symmetric matrix in $\mathbb{R}^{r_1\times r_1}$, whose (j,k) element is the inner product between $\mathcal{F}(j,:,:)$ and $\mathcal{F}(k,:,:)$, that is first mode jth and kth slices of $\mathcal{F}$. Multiplying from the left by X results in an $J\times r_1$ matrix. Similarly, $\langle\mathcal{A}\cdot(I,Y,Z),\mathcal{A}\cdot(I,Y,Z)\rangle_{-1}$ is a

symmetric $J \times J$ matrix, where the elements are inner products between the slices of $\mathcal{A} \cdot (I, Y, Z)$ and $\mathcal{A} \cdot (I, Y, Z)$.

Using analogous reformulations for (44) and (45), the complete Grassmann gradient becomes

$$\nabla\Phi = \begin{pmatrix} \Pi_X \Phi_x \\ \Pi_Y \Phi_y \\ \Pi_Y \Phi_z \end{pmatrix} = \begin{pmatrix} \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (I, Y, Z) \rangle_{-1} X - X \langle \mathcal{F}, \mathcal{F} \rangle_{-1} \\ \langle \mathcal{A} \cdot (X, I, Z), \mathcal{A} \cdot (X, I, Z) \rangle_{-2} Y - Y \langle \mathcal{F}, \mathcal{F} \rangle_{-2} \\ \langle \mathcal{A} \cdot (X, Y, I), \mathcal{A} \cdot (X, Y, I) \rangle_{-3} Z - Z \langle \mathcal{F}, \mathcal{F} \rangle_{-3} \end{pmatrix}. \tag{48}$$

Grassmann Hessian Computing the second derivative of Φ, using the same technique as for the gradient we obtain

$$\begin{aligned} \frac{d^2\Phi}{dt^2} &= \langle \mathcal{A} \cdot (\Delta_x, Y, Z), \mathcal{A} \cdot (\Delta_x, Y, Z) \rangle + \langle \mathcal{A} \cdot (\Delta_x, \Delta_y, Z), \mathcal{A} \cdot (X, Y, Z) \rangle \\ &\quad + \langle \mathcal{A} \cdot (\Delta_x, Y, Z), \mathcal{A} \cdot (X, \Delta_y, Z) \rangle + \langle \mathcal{A} \cdot (\Delta_x, Y, \Delta_z), \mathcal{A} \cdot (X, Y, Z) \rangle \\ &\quad + \langle \mathcal{A} \cdot (\Delta_x, Y, Z), \mathcal{A} \cdot (X, Y, \Delta_z) \rangle + \cdots, \end{aligned} \tag{49}$$

where, for simplicity of the present discussion, we have omitted 10 analogous terms. The first term, which gives the "xx" derivative, can be dealt with using Lemma 2.2. We get

$$\begin{aligned} \langle \mathcal{A} \cdot (\Delta_x, Y, Z), \mathcal{A} \cdot (\Delta_x, Y, Z) \rangle &= \langle \Delta_x, \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (\Delta_x, Y, Z) \rangle_{-1} \rangle \\ &= \langle \Delta_x, \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (I, Y, Z) \rangle_{-1} \Delta_x \rangle. \end{aligned}$$

From (41) and (46) we now see that the "xx" part of the Grassmann Hessian is a Sylvester operator,

$$\begin{aligned} \mathcal{H}_{xx}(\Delta_x) &= \Pi_X \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (I, Y, Z) \rangle_{-1} \Delta_x - \Delta_x X^T \Phi_x \\ &= \Pi_X \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (I, Y, Z) \rangle_{-1} \Delta_x - \Delta_x \langle \mathcal{F}, \mathcal{F} \rangle_{-1}, \end{aligned} \tag{50}$$

where Φ_x is defined in (46), and we have used Lemma 2.2.

For the second term in (49) we get, using Lemmas 2.2 and 2.3,

$$\begin{aligned} \langle \mathcal{A} \cdot (\Delta_x, \Delta_y, Z), \mathcal{A} \cdot (X, Y, Z) \rangle &= \langle \Delta_x, \langle \mathcal{A} \cdot (I, \Delta_y, Z), \mathcal{A} \cdot (X, Y, Z) \rangle_{-1} \rangle \\ &= \langle \Delta_x, \langle \mathcal{F}^1_{xy}, \Delta_y \rangle_{2,4;1:2} \rangle. \end{aligned} \tag{51}$$

where $\mathcal{F}^1_{xy}$ is the $4-$tensor

$$\mathbb{R}^{J \times K \times r_1 \times r_2} \ni \mathcal{F}^1_{xy} = \langle \mathcal{A} \cdot (I, I, Z), \mathcal{A} \cdot (X, Y, Z) \rangle_{-(1,2)} = \langle \mathcal{A} \cdot (I, I, Z), \mathcal{A} \cdot (X, Y, Z) \rangle_3.$$

It is obvious that $\langle \mathcal{F}^1_{xy}, \cdot \rangle_{2,4;1:2}$ defines a linear operator that maps matrices on matrices.

The third term in (49) becomes, again using Lemmas 2.2 and 2.3,

$$\begin{aligned} \langle \mathcal{A} \cdot (\Delta_x, Y, Z), \mathcal{A} \cdot (X, \Delta_y, Z) \rangle &= \langle \Delta_x, \langle \mathcal{A} \cdot (I, Y, Z), \mathcal{A} \cdot (X, \Delta_y, Z) \rangle_{-1} \rangle \\ &= \langle \Delta_x, \langle \mathcal{F}^2_{xy}, \Delta_y \rangle_{4,2;1:2} \rangle. \end{aligned} \tag{52}$$

where $\mathcal{F}^2_{xy}$ is a 4−tensor,

$$\mathbb{R}^{J\times r_2\times r_1\times K} \ni \mathcal{F}^2_{xy} = \langle \mathcal{A}\cdot(I,Y,Z), \mathcal{A}\cdot(X,I,Z)\rangle_{-(1,2)} = \langle \mathcal{A}\cdot(I,Y,Z), \mathcal{A}\cdot(X,I,Z)\rangle_{3}.$$

We now have

$$\mathcal{F}_{xy}(\Delta_y) = \langle \mathcal{F}^1_{xy}, \Delta_y\rangle_{2,4;1:2} + \langle \mathcal{F}^2_{xy}, \Delta_y\rangle_{4,2;1:2}\,. \tag{53}$$

The fourth and fifth terms in (49) can be dealt with similarly and give the $\mathcal{F}_{xz}$ operator.

In order for the second derivative operators to be in the tangent space we must multiply $\mathcal{F}_{xy}$, and $\mathcal{F}_{xz}$ by Π_x. If we rewrite all the terms in the second derivative (49) in an analogous way, we get a Hessian operator, $\mathcal{H} : \mathbb{T}^3 \mapsto \mathbb{T}^3$, where

$$\mathcal{H}(\Delta) = \begin{pmatrix} \mathcal{H}_{xx}(\Delta_x) + \mathcal{H}_{xy}(\Delta_y) + \mathcal{H}_{xz}(\Delta_z) \\ \mathcal{H}_{yx}(\Delta_x) + \mathcal{H}_{yy}(\Delta_y) + \mathcal{H}_{yz}(\Delta_z) \\ \mathcal{H}_{zx}(\Delta_x) + \mathcal{H}_{zy}(\Delta_y) + \mathcal{H}_{zz}(\Delta_z) \end{pmatrix}, \tag{54}$$

and each "$\mathcal{H}_{**}$" is a linear operator to be specified below. The diagonal operators[8] are (recall that $\mathcal{F} = \mathcal{A}\cdot(X,Y,Z)$),

$$\begin{aligned}
\mathcal{H}_{xx}(\Delta_x) &= \Pi_x\langle \mathcal{B}_x, \mathcal{B}_x\rangle_{-1}\Delta_x - \Delta_x\langle \mathcal{F}, \mathcal{F}\rangle_{-1}, & \mathcal{B}_x &= \mathcal{A}\cdot(I,Y,Z),\\
\mathcal{H}_{yy}(\Delta_y) &= \Pi_y\langle \mathcal{B}_y, \mathcal{B}_y\rangle_{-2}\Delta_y - \Delta_y\langle \mathcal{F}, \mathcal{F}\rangle_{-2}, & \mathcal{B}_y &= \mathcal{A}\cdot(X,I,Z),\\
\mathcal{H}_{zz}(\Delta_z) &= \Pi_z\langle \mathcal{B}_z, \mathcal{B}_z\rangle_{-3}\Delta_z - \Delta_z\langle \mathcal{F}, \mathcal{F}\rangle_{-3}, & \mathcal{B}_z &= \mathcal{A}\cdot(X,Y,I).
\end{aligned} \tag{55}$$

Since the Hessian operator is selfadjoint[9] we give only the blocks of the "upper triangular part",

$$\begin{aligned}
\mathcal{H}_{xy}(\Delta_y) &= \Pi_x\left(\langle\langle \mathcal{C}_{xy}, \mathcal{F}\rangle_{-(1,2)}, \Delta_y\rangle_{2,4;1:2} + \langle\langle \mathcal{B}_x, \mathcal{B}_y\rangle_{-(1,2)}, \Delta_y\rangle_{4,2;1:2}\right),\\
\mathcal{H}_{xz}(\Delta_z) &= \Pi_x\left(\langle\langle \mathcal{C}_{xz}, \mathcal{F}\rangle_{-(1,3)}, \Delta_z\rangle_{2,4;1:2} + \langle\langle \mathcal{B}_x, \mathcal{B}_z\rangle_{-(1,3)}, \Delta_z\rangle_{4,2;1:2}\right),\\
\mathcal{H}_{yz}(\Delta_z) &= \Pi_y\left(\langle\langle \mathcal{C}_{yz}, \mathcal{F}\rangle_{-(2,3)}, \Delta_z\rangle_{2,4;1:2} + \langle\langle \mathcal{B}_y, \mathcal{B}_z\rangle_{-(2,3)}, \Delta_z\rangle_{4,2;1:2}\right),
\end{aligned}$$

where we have also introduced $\mathcal{C}_{xy} = \mathcal{A}\cdot(I,I,Z)$, $\mathcal{C}_{xz} = \mathcal{A}\cdot(I,Y,I)$ and $\mathcal{C}_{yz} = \mathcal{A}\cdot(X,I,I)$. Observe that diagonal operators are Sylvester operators, and the off-diagonal operators have the form of 4–tensors acting on matrices.

4.3 Coordinate Representation for Gradient the Hessian Operator on the Tangent Space

The Hessian (54) is still given in a coordinate-free form. In order to make it more concrete, and to obtain a linear system of equations with the correct dimension that is non-singular in a neighbourhood of a maximum, we introduce coordinate expressions

[8] Even if the Hessian is not a block matrix, we will refer to the operators $\mathcal{H}_{xx}$, $\mathcal{H}_{yy}$, etc. as diagonal operators and $\mathcal{H}_{xy}$, $\mathcal{H}_{xz}$, etc. as off-diagonal operators.

[9] The operator is still somewhat abstract in the sense that we have not specified any coordinate representation on the tangent space $\mathbb{T}^3$. However, considered as an operator on $\mathbb{T}^3$ it can be seen that the operator is selfadjoint.

for the unknowns. We first see that the projections onto the tangent spaces can be represented as

$$\Pi_X = X_\perp X_\perp^T, \qquad \Pi_Y = Y_\perp Y_\perp^T, \qquad \Pi_Z = Z_\perp Z_\perp^T,$$

where $X_\perp$, $Y_\perp$, $Z_\perp$, are defined as in Section 3. In order to get a coordinate representation for the unknown tangents, we write them as [5, Section 2.5]

$$\begin{aligned} \Delta_x &= X_\perp D_x, & D_x &\in \mathbb{R}^{(J-r_1)\times r_1}, \\ \Delta_y &= Y_\perp D_y, & D_y &\in \mathbb{R}^{(K-r_2)\times r_2}, \\ \Delta_z &= Z_\perp D_z, & D_x &\in \mathbb{R}^{(L-r_3)\times r_3}; \end{aligned} \tag{56}$$

(note that the coordinate matrices are *not assumed to be diagonal*, even if the notation might be interpreted in that direction). With these coordinate expressions we can repeat the derivation from after (49) and write the Hessian as a linear operator acting on D_x, D_y, and D_z. We get

$$\begin{aligned} \widehat{\mathcal{H}}(D) &:= \begin{pmatrix} X_\perp^T & 0 & 0 \\ 0 & Y_\perp^T & 0 \\ 0 & 0 & Z_\perp^T \end{pmatrix} \begin{pmatrix} \mathcal{H}_{xx}(\Delta_x) + \mathcal{H}_{xy}(\Delta_y) + \mathcal{H}_{xz}(\Delta_z) \\ \mathcal{H}_{yx}(\Delta_x) + \mathcal{H}_{yy}(\Delta_y) + \mathcal{H}_{yz}(\Delta_z) \\ \mathcal{H}_{zx}(\Delta_x) + \mathcal{H}_{zy}(\Delta_y) + \mathcal{H}_{zz}(\Delta_z) \end{pmatrix} \\ &= \begin{pmatrix} \widehat{\mathcal{H}}_{xx}(D_x) + \widehat{\mathcal{H}}_{xy}(D_y) + \widehat{\mathcal{H}}_{xz}(D_z) \\ \widehat{\mathcal{H}}_{yx}(D_x) + \widehat{\mathcal{H}}_{yy}(D_y) + \widehat{\mathcal{H}}_{yz}(D_z) \\ \widehat{\mathcal{H}}_{zx}(D_x) + \widehat{\mathcal{H}}_{zy}(D_y) + \widehat{\mathcal{H}}_{zz}(D_z) \end{pmatrix}, \end{aligned} \tag{57}$$

where each "$\widehat{\mathcal{H}}_{**}$" is a linear operator. The diagonal operators are

$$\begin{aligned} \widehat{\mathcal{H}}_{xx}(D_x) &= \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_x \rangle_{-1} D_x - D_x \langle \mathcal{F}, \mathcal{F} \rangle_{-1}, & \widehat{\mathcal{B}}_x &= \mathcal{A} \cdot (X_\perp, Y, Z), \\ \widehat{\mathcal{H}}_{yy}(D_y) &= \langle \widehat{\mathcal{B}}_y, \widehat{\mathcal{B}}_y \rangle_{-2} D_y - D_y \langle \mathcal{F}, \mathcal{F} \rangle_{-2}, & \widehat{\mathcal{B}}_y &= \mathcal{A} \cdot (X, Y_\perp, Z), \\ \widehat{\mathcal{H}}_{zz}(D_z) &= \langle \widehat{\mathcal{B}}_z, \widehat{\mathcal{B}}_z \rangle_{-3} D_z - D_z \langle \mathcal{F}, \mathcal{F} \rangle_{-3}, & \widehat{\mathcal{B}}_z &= \mathcal{A} \cdot (X, Y, Z_\perp). \end{aligned} \tag{58}$$

The Hessian operator $\widehat{\mathcal{H}}$ is selfadjoint with respect to the inner product,

$$\langle D, \widehat{\mathcal{H}}(D) \rangle_{\mathbb{T}^3} = \langle \widehat{\mathcal{H}}(D), D \rangle_{\mathbb{T}^3}.$$

where

$$\langle D, E \rangle_{\mathbb{T}^3} = \langle D_x, E_x \rangle + \langle D_y, E_y \rangle + \langle D_z, E_z \rangle,$$

and $D = (D_x, D_y, D_z)$ and $E = (E_x, E_y, E_z)$ are the coordinates for two tangents. Therefore we give only the blocks of the "upper triangular part",

$$\begin{aligned} \widehat{\mathcal{H}}_{xy}(D_y) &= \left(\langle \langle \widehat{\mathcal{C}}_{xy}, \mathcal{F} \rangle_{-(1,2)}, D_y \rangle_{2,4;1:2} + \langle \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_y \rangle_{-(1,2)}, D_y \rangle_{4,2;1:2} \right), \\ \widehat{\mathcal{H}}_{xz}(D_z) &= \left(\langle \langle \widehat{\mathcal{C}}_{xz}, \mathcal{F} \rangle_{-(1,3)}, D_z \rangle_{2,4;1:2} + \langle \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_z \rangle_{-(1,3)}, D_z \rangle_{4,2;1:2} \right), \\ \widehat{\mathcal{H}}_{yz}(D_z) &= \left(\langle \langle \widehat{\mathcal{C}}_{yz}, \mathcal{F} \rangle_{-(2,3)}, D_z \rangle_{2,4;1:2} + \langle \langle \widehat{\mathcal{B}}_y, \widehat{\mathcal{B}}_z \rangle_{-(2,3)}, D_z \rangle_{4,2;1:2} \right), \end{aligned} \tag{59}$$

where $\widehat{\mathcal{C}}_{xy} = \mathcal{A} \cdot (X_\perp, Y_\perp, Z)$, $\widehat{\mathcal{C}}_{xz} = \mathcal{A} \cdot (X_\perp, Y, Z_\perp)$ and $\widehat{\mathcal{C}}_{yz} = \mathcal{A} \cdot (X, Y_\perp, Z_\perp)$.

In the coordinate representation (56) the Grassmann gradient (48) is given by

$$\nabla\widehat{\Phi} = \begin{pmatrix} X_\perp^T & 0 & 0 \\ 0 & Y_\perp^T & 0 \\ 0 & 0 & Z_\perp^T \end{pmatrix} \begin{pmatrix} \Pi_X \Phi_x \\ \Pi_Y \Phi_y \\ \Pi_Y \Phi_z \end{pmatrix} = \begin{pmatrix} \langle \widehat{\mathcal{B}}_x, \mathcal{F} \rangle_{-1} \\ \langle \widehat{\mathcal{B}}_y, \mathcal{F} \rangle_{-2} \\ \langle \widehat{\mathcal{B}}_z, \mathcal{F} \rangle_{-3} \end{pmatrix}. \tag{60}$$

It is known [14] that in general the objective function (42) is not concave. In fact it is easy to construct non-concave examples using the coordinate representation of the Hessian.

Proposition 4.1. *The maximization problem* (42) *can have local maxima.*

Proof. Given any tensor $\mathcal{A}$ and a stationary point (X, Y, Z), for which the Hessian $\widehat{\mathcal{H}}$ is negative definite. Now create a new tensor $\widetilde{\mathcal{A}}$ by modifying $\mathcal{A}$ in such a way that a large element occurs in $\widetilde{\mathcal{A}} \cdot (X_\perp, Y_\perp, Z_\perp)$. Then, obviously, (X, Y, Z) is a stationary point for $\widetilde{\Phi}(X, Y, Z) = \frac{1}{2}\|\widetilde{\mathcal{A}} \cdot (X, Y, Z)\|^2$, but it cannot be a global maximum. □

4.4 Interpretation of Operators in the Hessian

The Hessian operator $\widehat{\mathcal{H}}$ consists of partial contractions involving the tensors

$$\begin{array}{llll} \mathcal{A} \cdot (X, Y, Z), & \mathcal{A} \cdot (X_\perp, Y, Z), & \mathcal{A} \cdot (X, Y_\perp, Z), & \mathcal{A} \cdot (X, Y, Z_\perp), \\ & \mathcal{A} \cdot (X, Y_\perp, Z_\perp), & \mathcal{A} \cdot (X_\perp, Y, Z_\perp), & \mathcal{A} \cdot (X_\perp, Y_\perp, Z). \end{array}$$

These are blocks of the tensor $\widehat{\mathcal{A}} = \mathcal{A} \cdot ((X\ X_\perp), (Y\ Y_\perp), (Z\ Z_\perp))$. The only block in $\widehat{\mathcal{A}}$ that does not occur in $\widehat{\mathcal{H}}$ is $\mathcal{A} \cdot (X_\perp, Y_\perp, Z_\perp)$. $\widehat{\mathcal{A}}$ is illustrated in Figure 2.

The partial contractions $\langle \cdot, \cdot \rangle_{-p}$ are matrices, whose elements are inner products between the slices in a subtensor. In Figure 2 we illustrate the inner products in $\widehat{\mathcal{H}}_{xx}$. In the off-diagonal operators the inner products are between fibres in subtensors. For instance, in $\widehat{\mathcal{H}}_{xy}$ the inner products in $\langle \widehat{\mathcal{C}}_{xy}, \mathcal{F} \rangle_{-(1,2)}$ are between fibers, illustrated with the symbol, from $\mathcal{A} \cdot (X_\perp, Y_\perp, Z)$ and $\mathcal{A} \cdot (X, Y, Z)$. Similarly the elements of $\langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_y \rangle_{-(1,2)}$ are inner products between the fibers from $\mathcal{A} \cdot (X_\perp, Y, Z)$ and $\mathcal{A} \cdot (X, Y_\perp, Z)$.

4.5 Matricizing the Hessian Operator

It is now straightforward to matricize the operators in the Hessian and vectorize D_x, D_y and D_z to obtain a standard matrix–vector linear system.

The "xx" block in (58) has the form

$$\widehat{\mathcal{H}}_{xx}(D_x) = \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_x \rangle_{-1} D_x - D_x \langle \mathcal{F}, \mathcal{F} \rangle_{-1}. \tag{61}$$

Observing that the contracted tensors are matrices and with straightforward vectorization of matrix products [9, Chapter 4.3] we get

$$\operatorname{vec}(\widehat{\mathcal{H}}_{xx}(D_x)) = \Big(I \otimes \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_x \rangle_{-1} + \langle \mathcal{F}, \mathcal{F} \rangle_{-1} \otimes I \Big)\, d_x \equiv \widehat{H}_{xx} d_x,$$

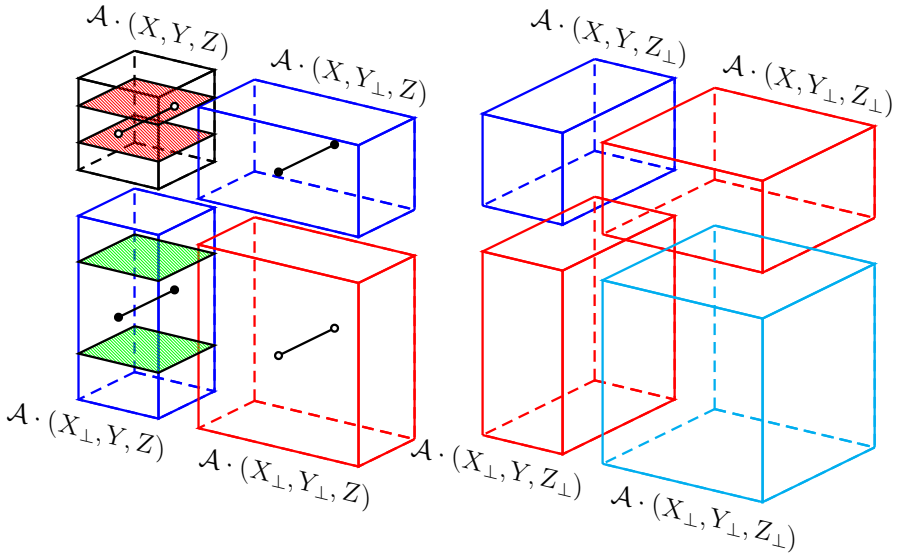

Figure 2: Illustration of the partial contractions in the Hessian. For better visibility we have slided the backward part of the the tensor $\widehat{\mathcal{A}}$ to the right.

where $d_x = \mathrm{vec}(D_x)$. The other diagonal blocks are treated analogously.

The off-diagonal blocks in (59) consist of two 4–tensors acting on matrices. The "xy" block is given by

$$\widehat{\mathcal{H}}_{xy}(D_y) = \left(\langle \widehat{\mathcal{H}}^1_{xy}, D_y\rangle_{2,4;1:2} + \langle \widehat{\mathcal{H}}^2_{xy}, D_y\rangle_{4,2;1:2}\right),$$

where $\widehat{\mathcal{H}}^1_{xy} = \langle \widehat{\mathcal{C}}_{xy}, \mathcal{F}\rangle_{-(1,2)}$ and $\widehat{\mathcal{H}}^2_{xy} = \langle \widehat{\mathcal{B}}_x, \widehat{\mathcal{B}}_y\rangle_{-(1,2)}$. In $\widehat{\mathcal{H}}^1_{xy}$, we map the first and third mode to the rows and second and forth mode to the columns of the matrix. In $\widehat{\mathcal{H}}^2_{xy}$ the ordering of the row modes is the same but the column modes are four and two. The vectorized form of the operation $\widehat{\mathcal{H}}_{xy}(D_y)$ is

$$\mathrm{vec}(\widehat{\mathcal{H}}_{xy}(D_y)) = \left(\widehat{H}^{1\,(1,3;2,4)}_{xy} + \widehat{H}^{2\,(1,3;4,2)}_{xy}\right) d_y \equiv \widehat{H}_{xy} d_y,$$

where $d_y = \mathrm{vec}(D_y)$.

After matricizing all blocks of $\widehat{\mathcal{H}}$ and vectorizing the gradients we obtain the matrix form for the Newton equation,

$$\widehat{H}\,d = \begin{pmatrix} \widehat{H}_{xx} & \widehat{H}_{xy} & \widehat{H}_{xz} \\ \widehat{H}_{yx} & \widehat{H}_{yy} & \widehat{H}_{yz} \\ \widehat{H}_{zx} & \widehat{H}_{zy} & \widehat{H}_{zz} \end{pmatrix} \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = - \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} = -g, \tag{62}$$

where $g_x = \mathrm{vec}\left(\langle \widehat{\mathcal{B}}_x, \mathcal{F}\rangle_{-1}\right)$, $g_y = \mathrm{vec}\left(\langle \widehat{\mathcal{B}}_y, \mathcal{F}\rangle_{-2}\right)$ and $g_z = \mathrm{vec}\left(\langle \widehat{\mathcal{B}}_z, \mathcal{F}\rangle_{-3}\right)$ are the vectorized gradients from (60).

4.6 Generalizing to Higher Order Tensors

Note that the representations for the Grassmann gradient and Hessian in Section 4.2 can easily be generalized to the case of 4−tensors and higher. Assume that the objective function $\Phi(X,Y,Z,W) = \frac{1}{2}\|\mathcal{A}\cdot(X,Y,Z,W)\|_F$ is to be maximized over a product of four Grassmann manifolds. Then the diagonal operators in the Hessian (55) have to be modified by introducing an extra matrix W, i.e. we put $\mathcal{B}_x = \mathcal{A}\cdot(I,Y,Z,W)$, etc., and then we add a fourth diagonal block,

$$\mathcal{H}_{ww}(\Delta_w) = \Pi_w\langle\mathcal{B}_w,\mathcal{B}_w\rangle_{-4}\Delta_w - \Delta_w\langle\mathcal{F},\mathcal{F}\rangle_{-4},$$

where $\mathcal{F} = \mathcal{A}\cdot(X,Y,Z,W)$, and $\mathcal{B}_w = \mathcal{A}\cdot(X,Y,Z,I)$. The off-diagonal operators are modified analogously. For instance

$$\mathcal{H}_{xw}(\Delta_w) = \Pi_x\left(\langle\langle\mathcal{C}_{xw},\mathcal{F}\rangle_{-(1,4)},\Delta_w\rangle_{2,4;1:2} + \langle\langle\mathcal{B}_x,\mathcal{B}_w\rangle_{-(1,4)},\Delta_w\rangle_{4,2;1:2}\right),$$

where $\mathcal{B}_x$ and $\mathcal{B}_w$ are as above, and $\mathcal{C}_{xw} = \mathcal{A}\cdot(I,Y,Z,I)$.

5 Implementation and Experimental Results

Given the analysis from the previous section together with the TensorToolbox[10] [1] the algorithmic implementation in MATLAB is straightforward. A pseudo–code is given in Algorithm 1.

Algorithm 1 Newton-Grassmann algorithm

Given tensor $\mathcal{A}$ and starting points $(X_0,Y_0,Z_0)\in \mathrm{Gr}^3$
repeat
  compute the Grassmann gradient $\nabla\widehat{\Phi}$ given in equation (60)
  compute the Grassmann Hessian $\widehat{\mathcal{H}}$ from equation (57)
  matricize $\widehat{\mathcal{H}}$ and vectorize $\nabla\widehat{\Phi}$ to form the Grassmann-Newton equations (62)
  solve $D = (D_x,D_y,D_z)$ from the Newton equation on the Grassmann manifolds
  take a geodesic step along the direction given by D to obtain new iterates (X,Y,Z)
until $\|\nabla\widehat{\Phi}\|/\Phi < \mathrm{TOL}$

In this section we report the results of a couple of preliminary numerical experiments, where we compare the Newton-Grassmann algorithm with higher order orthogonal iteration (HOOI) [14]. Each HOOI iteration consists of 3 steps, where in each step two of the unknown matrices are considered as fixed, and the third is updated.

Test 1 Our first experiment was tailored to simulate a "signal tensor" with low rank and added normally distributed noise. We used two $20\times20\times20$ tensors, $\mathcal{A}_1 = \mathcal{B}_1 + \rho\mathcal{E}_1$

[10] The TensorToolbox implements basic tensor operations as tensor–matrix multiplication and general matricization of tensors. Even though you have to make some minor notational modifications it is quite consistent with the presented framework of this paper.

and $\mathcal{A}_2 = \mathcal{B}_2 + \rho\mathcal{E}_2$, where we chose $\mathcal{B}_1$ and $\mathcal{B}_2$ as random tensors with ranks $(10, 10, 10)$ and $(15, 15, 15)$, respectively. Thus $\mathcal{B}_1$ was constructed from a $10 \times 10 \times 10$ tensor with normally distributed $(N(0, 1))$ elements; that tensor was then 'blown up' to dimension $20 \times 20 \times 20$ by multiplying it in each mode by a 20×10 matrix with orthonormal columns. The elements of the noise tensors $\mathcal{E}_1$ and $\mathcal{E}_2$ were chosen normally distributed $(N(0, 1))$ and the level of noise was controlled by ρ, which was taken equal to 0.1. In both cases we computed a rank$-(5, 5, 5)$ approximation. As initial approximation a random tensor was chosen and 10 HOOI iterations were performed before the Newton method was started. Figure 3 shows the convergence history of the Newton-Grassmann and HOOI methods.

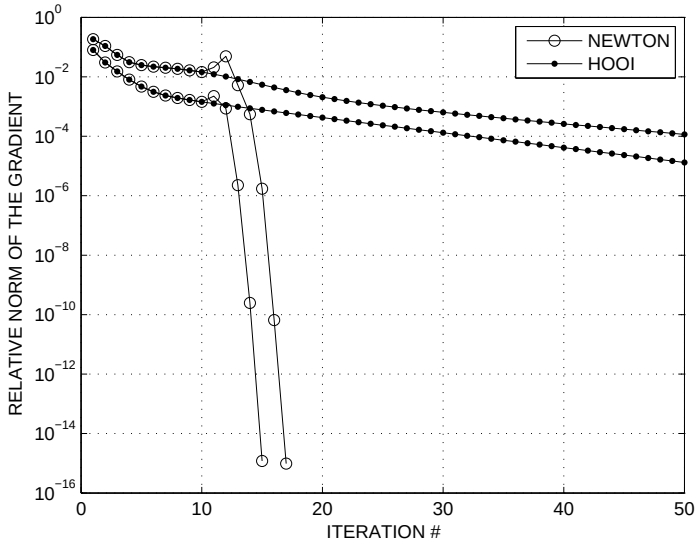

Figure 3: Convergence history for Test 1: number of iterations versus the relative gradient norm $\|\nabla\widehat{\Phi}\|/\Phi$. The lower pair of Newton-Grassmann and HOOI curves is for $\mathcal{A}_1$ and the upper pair is for $\mathcal{A}_2$.

We also performed tests where the ranks of the "signal tensor" and the approximating tensor coincided. Then, for small values of ρ, the convergence of HOOI was very rapid.

Test 2 We approximated a random $20 \times 20 \times 20$ tensor (the elements were in $N(0, 1)$) by a rank$-(5, 5, 5)$ tensor. Both algorithms were initialized by HOSVD and we performed 20 HOOI iterations before Newton–Grassmann was employed. Figure 4 shows the convergence history.

The quadratic convergence of the Newton–Grassmann algorithm is clearly visible in both plots.

In our experience the HOOI method may have acceptable convergence speed for low rank signal tensors with noise of small magnitude. In general, the closer the rank of the approximating tensor to the correct rank of the signal tensor the faster the convergence.

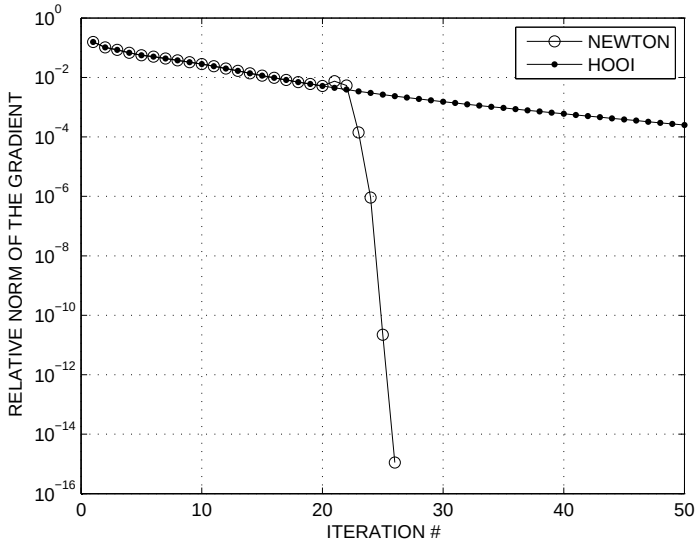

Figure 4: Convergence history for Test2.

On the other hand, approximating a full rank tensor with HOOI can have very slow convergence, see Figure 4, and in some cases HOOI requires a large number of iterations before the convergence is stabilized to a constant linear rate, cf. the upper curve in Figure 3.

Computational complexity Naturally, the price to be paid for the fast convergence of the Newton-Grassmann method is a higher computational cost per iteration. Assume, for simplicity, that we have an $n \times n \times n$ tensor which is approximated by a $r \times r \times r$ tensor. Each iteration in HOOI involves six tensor by matrix products and three maximization problems, e.g. $\mathcal{A} \cdot (I, Y, Z)$ and

$$\max_{X^T X = I} \|X^T A^{(1)} (Y \otimes Z)\|.$$

The solution is the dominant r–dimensional left singular subspace of the matrix $A^{(1)}(Y \otimes Z)$, which we assume is computed with SVD [7, Section 5.4.5]. Then, the approximate amount of flops (floating point additions and multiplications) per iteration is $6n^3 r$ for the tensor–matrix product and $18nr^4 + 33r^6$ for the dominant subspace (based on the table in [7, Section 5.4.5]; note that faster SVD algorithms are available and will be implemented in the next version of LAPACK), which gives

$$\text{flops(HOOI)} \approx 6n^3 r + 18nr^4 + 33r^6.$$

Each iteration in the Newton–Grassmann algorithm is dominated by the computation of the Hessian and the solution of the Newton's equations (62), which amounts to

$$\text{flops(Newton)} \approx 4n^4 + 9n^3 r^3.$$

Optimization Issues With the formulation of our problem (31) as an optimization problem on the product of Grassmann manifolds, and with the parametrization of Section 4.3 we have reduced it to an unconstrained optimization problem. When solving this one must deal with standard optimization issues such as obtaining good starting points, indefiniteness of the Hessian, line search, etc., see e.g. [16] for details.

As is always the case with Newton's method, the choice of a good starting point is important. One obvious alternative is to start with $X_0 = U(:, 1 : r_1)$, $Y_0 = V(:, 1 : r_2)$ and $Z_0 = W(:, 1 : r_3)$ where U, V and W are obtained from the HOSVD. But this choice is often not good enough. In our experiments with random tensors, the Hessian was almost always indefinite for points given by the HOSVD. When we performed initial HOOI iterations, then within a reasonable amount of steps we got to the proximity of the local minimum where we could employ the Newton algorithm. An alternative for HOOI could be to perform the initial steps with a conjugate gradient algorithm on the product Grassmann manifold.

6 Conclusion and future work

In this paper we have formulated the tensor approximation problem to be defined on product of Grassmann manifolds and derived the Newton's method for this problem. We have showed quadratic convergence of the algorithm in the proximity of a local minimizer.

The general tensor matricization introduced in Section 2.2, the contracted tensor products and the tensor algebraic identities from Section 2.3 have been very useful both for the analysis of the differentiated expressions of the objective function and for the algorithmic implementation. The generalization from 3–tensors to higher order tensors is straightforward with the presented tensor algebraic analysis.

Our present and future work include further analysis of the theoretical aspects of the best approximation problem. For computational and memory efficiency, the implementation details for the Newton-Grassmann algorithms need to be investigated. An alternative approach for this and similar problems, which we are presently pursuing, is to develop Quasi-Newton methods on (products of) Grassmann manifolds.

A Proof of Lemma 2.3

To prove the identities we will use the definition for contracted tensor product to verify that elements of the resulting matrices in both sides are the same. Let $j < i$ and assume we have the following dimensions;

$$\mathcal{B} \in \mathbb{R}^{K_1 \times \cdots \times K_j \times \cdots \times K_i \times \cdots \times K_N},$$
$$Q \in \mathbb{R}^{K_j \times L_j}.$$

The dimensions of the modes of the tensor $\mathcal{C}$ are assumed to be the same as those in $\mathcal{B}$, except modes j and i, which are taken to be L_j and L_i. We will show that

$$\langle \mathcal{B} \cdot (Q)_j, \mathcal{C} \rangle_{-i} = \left\langle \langle \mathcal{B}, \mathcal{C} \rangle_{-(i,j)}, Q \right\rangle_{1,3;1,2}. \tag{63}$$

Then, for the first argument on the left hand side we have

$$\mathcal{B} \cdot (Q)_j =: \mathcal{D} \in \mathbb{R}^{K_1 \times \cdots \times L_j \times \cdots \times K_i \times \cdots \times K_N}$$

where the elements are given by

$$d_{k_1 \cdots l_j \cdots k_i \cdots k_N} = \sum_{k_j} a_{k_1 \cdots k_j \cdots k_i \cdots k_N} q_{k_j l_j}.$$

The expression on the left hand side of (63) becomes

$$\langle \mathcal{D}, \mathcal{C} \rangle_{-i} =: \mathcal{E} \in \mathbb{R}^{K_i \times L_i},$$

where the entries are

$$\begin{aligned} e_{k_i l_i} &= \sum_{\substack{k_1,\ldots,k_{j-1},l_j \\ k_{j+1},\ldots,k_{i-1} \\ k_{i+1},\ldots,k_N}} d_{k_1 \cdots l_j \cdots k_i \cdots k_N} c_{k_1 \cdots l_j \cdots l_i \cdots k_N} \\ &= \sum_{\substack{k_1,\ldots,k_j,l_j \\ k_{j+1},\ldots,k_{i-1} \\ k_{i+1},\ldots,k_N}} a_{k_1 \cdots k_j \cdots k_i \cdots k_N} q_{k_j l_j} c_{k_1 \cdots l_j \cdots l_i \cdots k_N} \\ &= \sum_{k_j,l_j} q_{k_j l_j} \sum_{\substack{k_1,\ldots,k_{j-1} \\ k_{j+1},\ldots,k_{i-1} \\ k_{i+1},\ldots,k_N}} a_{k_1 \cdots k_j \cdots k_i \cdots k_N} c_{k_1 \cdots l_j \cdots l_i \cdots k_N}, \end{aligned}$$

which shows that (63) holds. The other cases are analogous.

References

[1] B. Bader and T. Kolda. Matlab tensor classes for fast algorithm prototyping. Technical Report SAND2004-5187, Sandia National Laboratories, Oct. 2004. To appear in ACM Trans. Math. Software.

[2] B. W. Bader and T. G. Kolda. Efficient matlab computations with sparse and factored tensors. Technical report, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006.

[3] V. de Silva and L. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl., to appear*, 2007.

[4] L. DeLathauwer, L. Hoegaerts, and J. Vandewalle. A Grassmann-Rayleigh quotient iteration for dimensionality reduction in ICA. In *Proc. 5th Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2004)*, pages 335–342, Granada, Spain, Sept. 2004.

[5] A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303–353, 1999.

[6] Lars Eldén. *Matrix Methods in Data Mining and Pattern Recognition.* Society for Industrial and Applied Mathematics, Philadeplhia, PA, Philadelphia, PA, USA, 2007.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations. 3rd ed.* Johns Hopkins Press, Baltimore, MD., 1996.

[8] R. A. Harshman. An index formalism that generalizes the capabilities of matrix notation and algebra to n-way arrays. *J. Chemometrics*, 15:689–714, 2001.

[9] R. J. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, Cambridge, 1991.

[10] H.A.L. Kiers. Towards a standardized notation and terminology in multiway analysis. *J. Chemometrics*, 14:106–125, 2000.

[11] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry.* Interscience Publisher, 1963.

[12] Tamara G. Kolda. Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, Sandia National Laboratories, April 2006.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensor. *SIAM J. Matrix Anal. Appl.*, 21:1324–1342, 2000.

[15] E. Lundström and L. Eldén. Adaptive eigenvalue computations using Newton's method on the Grassmann manifold. *SIAM J. Matrix Anal. Appl.*, 23:819–839, 2002.

[16] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[17] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. 7th European Conference on Computer Vision (ECCV'02)*, Lecture Notes in Computer Science, Vol. 2350, pages 447–460, Copenhagen, Denmark, 2002. Springer Verlag.

[18] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pages 93–99, Madison WI, 2003.