
II. Alternating Least Square (ALS) algorithm.

III. Tucker Decomposition and Higher Order Singular Value Decomposition (HOSVD).

I. Tools.

We will need various extra tools to go on and introduce them in this paragraph.

I.a) Matricizations of Tensors.

We can take all frontal slices and make out a matrix:

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\]
By definition \( T(1) = \text{matrix of fibers } T \circ (F^1) \text{ ordered as indicated above} \)

\[ = I_1 \times I_2 I_3 \text{ matrix.} \]

We can also take lateral slices and construct two other matrices:

\[ T(2) = \text{matrix of fibers } T \times x^2 \text{ put as} \]

columns: \[ \begin{array}{c}
\alpha y \\
\varphi \\
\varepsilon \end{array} \]

This is an \( I_2 \times I_1 I_3 \text{ matrix.} \)

\[ T(3) = \text{matrix of fibers } T \times x^3 \text{ put as} \]

columns: \[ \begin{array}{c}
\gamma \varepsilon \\
\alpha y \\
\alpha z \end{array} \]

This is an \( I_3 \times I_1 I_2 \text{ matrix.} \)

We say that \( T(1), T(2), T(3) \) are the three matrices of the tensor \( T \). See example.
Example.

\[
T_4 = \begin{bmatrix}
1 & 4 & 7 & 10 \\
2 & 5 & 8 & 11 \\
3 & 6 & 9 & 12
\end{bmatrix}
\]

\[
T_5 = \begin{bmatrix}
4 & 2 & 3 & 7 & 8 & 9
\end{bmatrix}
\]

\[
T_6 = \begin{bmatrix}
2 & 3 & 4 & 5 & 6
\end{bmatrix}
\]

We will need to express matrix entries of

\[
\sum_{r=1}^{R} \alpha_r \otimes \beta_r \otimes \gamma_r
\]

For this handy tool will be the Khatri-Rao product, and the Kronecker product.
15) **Kronecker Product of vectors**

\[ C \otimes_{K \mathbf{K}^0} b = \begin{bmatrix} C^1 & \cdots & C^I \end{bmatrix} \otimes_{K \mathbf{K}^0} \begin{bmatrix} b^1 \\ \vdots \\ b^I \end{bmatrix} = \begin{bmatrix} C^1 b \\ \vdots \\ C^I b \end{bmatrix} \]

\[ (C \otimes_{K \mathbf{K}^0} b)^T = C^T \otimes_{K \mathbf{K}^0} b^T = [C^1 b^T, \ldots, C^I b^T] \]

by definition

by def

\( I_3 \otimes \mathbf{I}_2 \) vector

**Exercise!**

Show that

\[ (e \otimes_{K \mathbf{K}^0} d)^T (C \otimes_{K \mathbf{K}^0} b) = (e^T c) (d^T b) \]

inner product

usual inner products

**Notation and Remark.**

1) \( \otimes \) and \( \otimes_{K \mathbf{K}^0} \) are basically equivalent except that the resulting object is presented as a matrix with \( C \otimes b \) and as a column vector with \( C \otimes_{K \mathbf{K}^0} b \). But these are isomorphic.

2) Sometimes, \( \otimes_{K \mathbf{K}^0} \) is called \( \otimes \) and \( \otimes \) is called \( \odot \).
I. c) Khatri-Rao product.

\[ C = \begin{bmatrix} c_1 & \cdots & c_R \end{bmatrix} I_3 \times R \]
\[ B = \begin{bmatrix} b_1 & \cdots & b_R \end{bmatrix} I_2 \times R \]

\[ C \otimes_{\text{KhR}} B = \begin{bmatrix} c_1 \otimes_{\text{KhR}} b_1 & c_2 \otimes_{\text{KhR}} b_2 & \cdots & c_R \otimes_{\text{KhR}} b_R \end{bmatrix} \]

\[ I_2 \otimes I_3 \times R \text{ matrix.} \]

Often one also use the notation \( \otimes_{\text{KhR}} = \circ \).

Properties \( (\text{exercise!}) \)

1) \( (E \otimes_{\text{KhR}} D)^T (C \otimes_{\text{KhR}} B) = (E^TC) \ast (D^TB) \)

where \( \ast \) is the Hadamard pointwise product between matrices, \( (A \ast B)_{ij} = A_{ij} B_{ij} \). For example

\[
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \ast \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{pmatrix}
\]

2) If \( C \) \& \( B \) are full column rank (i.e. their columns are linear independent) then \( C \otimes_{\text{KhR}} B \) is also full column rank i.e. \( c_1 \otimes_{\text{KhR}} b_1, \ldots, c_R \otimes_{\text{KhR}} b_R \) are linear independent.
(d) Matrixizations of $T = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r$:

We have

$$T_{(1)} = A \left( C \otimes_{K_1 R} B \right)^T,$$

$$T_{(2)} = B \left( A \otimes_{K_2 R} C \right)^T,$$

$$T_{(3)} = C \left( B \otimes_{K_3 R} A \right)^T.$$

Check of the first equation:

$$T_{(1)} = \begin{bmatrix}
\sum_{r=1}^{R} a_r^x b_r^y c_r^z \\
\vdots \\
\sum_{r=1}^{R} a_r^x b_r^y c_r^z
\end{bmatrix} = \begin{bmatrix}
A \\
\vdots \\
A
\end{bmatrix} \begin{bmatrix}
\begin{bmatrix}
b_1^T c_1^z \\
b_2^T c_1^z \\
\vdots \\
b_R^T c_1^z
\end{bmatrix} \\
\begin{bmatrix}
b_1^T c_2^z \\
b_2^T c_2^z \\
\vdots \\
b_R^T c_2^z
\end{bmatrix} \\
\begin{bmatrix}
b_1^T c_R^z \\
b_2^T c_R^z \\
\vdots \\
b_R^T c_R^z
\end{bmatrix}
\end{bmatrix} = A \left( C \otimes_{K_1 R} B \right)^T.$$
III. ALS algorithm for Tensor Decomposition.

We first introduce the definition of the Frobenius norm for tensors. This is nothing else than the Euclidean norm for the array \( T \) as \( \text{dim}(T) \) of numbers.

**Def:** Frobenius Norm

\[
\| T \|_F^2 = \sum_{\alpha=1}^I \sum_{\beta=1}^J \sum_{\gamma=1}^K |T_{\alpha\beta\gamma}|^2.
\]

**Property:** \( \| T \|_F \) is rotation invariant in the sense that \( T(R_1, R_2, R_3) \) has the same norm for rotation (orthogonal) matrices \( R_1, R_2, R_3 \). (see exercise).

Now suppose we are given an array of numbers \( T_{\alpha\beta\gamma} \) and suppose we knew there exist

\[
A = [a_1, \ldots, a_i, \ldots], \quad B = [b_1, \ldots, b_j, \ldots], \quad C = [c_1, \ldots, c_k]\n\]

s.t.

\[
T = \sum_{r=1}^R a_r \otimes b_r \otimes c_r. \quad \text{In order to find A, B, C we could try to minimize} \]

\[
\| T - \sum_{r=1}^R a_r \otimes b_r \otimes c_r \|_F^2
\]

over unknowns \( A, B, C \).
This is a highly non-convex minimization problem and in general we do not have algs with convergence guarantees. Convergence will in particular be highly dependent on initialization.

Note that obviously:

$$\|T\|_F = \sum_{r=1}^{R} |\mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r|$$

$$= \|T_{(1)} - A (C \otimes_{K\times R} B)^T\|_F^2$$

$$= \|T_{(2)} - B (A \otimes_{K\times R} C)^T\|_F^2$$

$$= \|T_{(3)} - C (B \otimes_{K\times R} A)^T\|_F^2$$

(When on the right-hand side we have Matrix Frobenius Norms).

Main idea of ALS: Go through the steps:

$$A \leftarrow \arg\min_A \|T_{(1)} - A (C \otimes_{K\times R} B)^T\|_F^2$$

$$B \leftarrow \arg\min_B \|T_{(2)} - B (A \otimes_{K\times R} C)^T\|_F^2$$

$$C \leftarrow \arg\min_C \|T_{(3)} - C (B \otimes_{K\times R} A)^T\|_F^2$$
Solving the least square problems (see later on) we can cast the ALS algorithm in the form:

- **Input**: $A_0$, $B_0$, $C_0$; **Output**: $A$, $B$, $C$

- Initialize with $B^{(0)}$, $C^{(0)}$ of full column rank and do:

$$
A^{m+1} = T^{(1)} (C^{(m)} \otimes_{KLR} B^{(m)}) (C^{mT} C^{mT} + B^{mT} B)^{-1}
$$

$$
B^{m+1} = T^{(2)} (A^{m+1} \otimes_{KLR} B^{m}) (A^{m+1T} A^{m+1T} + C^{mT} C^{mT})^{-1}
$$

$$
C^{m+1} = T^{(3)} (B^{m+1} \otimes A^{m+1}) (B^{m+1T} B^{m+1T} + A^{m+1T} A^{m+1T})^{-1}
$$

<Comment on ALS1; ALS2.>

**Derivation of These Equations.**

To derive these equations it suffices to see that

$$
\arg \min_A \| T^{(1)} - A (C \otimes_{KLR} B)^T \|_F
$$

$$
= T^{(1)} ((C \otimes_{KLR} B)^T)^+ \quad \text{Moore-Penrose pseudo inverse}
$$

and for $C$ & $B$ full column rank $\implies C \otimes_{KLR} B$ full column rank $\implies (C \otimes_{KLR} B)^T$ full row rank $\implies$

$$
(C \otimes_{KLR} B)^T)^+ = (C \otimes_{KLR} B) (C \otimes_{KLR} B)^T (C \otimes_{KLR} B)^{-1}
$$

$$
= (C \otimes_{KLR} B) (C^T C + B^T B)^{-1}
$$
Recap of least square solution

The problem is of the form

$$\begin{align*}
\text{argmin } & \| Y - X \phi \|_F^2 \\
\text{subject to } & X \in \mathbb{R}^{I \times R} \\
& I_1 \times I_2 \times I_3 \\
& I_1 \times I_2 \\
& I_1 \times I_3
\end{align*}$$

and \( \phi \) is full rank.

Set \( X = Y \phi^T (\phi \phi^T)^{-1} + \Xi \)

Note: Pseudo-inverse for \( \phi \) full rank.

$$\begin{align*}
\| Y - X \phi \|_F^2 &= \| Y - Y \phi^T (\phi \phi^T)^{-1} \phi \|_F^2 + 2 \Xi \|_F^2 \\
&= \| Y - Y \phi^T (\phi \phi^T)^{-1} \phi \|_F^2 + 2 \Xi \|_F^2 \\
&\geq \| Y - Y \phi^T (\phi \phi^T)^{-1} \phi \|_F^2 \\
\end{align*}$$

where we used in (x) that the term:

$$\begin{align*}
2 \text{Tr} \left[ (Y - Y \phi^T (\phi \phi^T)^{-1} \phi) (\phi \phi^T)^{-1} \phi \phi^T \Xi \right] \\
= 2 \text{Tr} \left[ (Y - \phi^T (\phi \phi^T)^{-1} \phi) (\phi \phi^T)^{-1} \phi \phi^T \Xi \right] \\
= 2 \text{Tr} \left[ (\phi - \phi^T (\phi \phi^T)^{-1} \phi \phi^T) \Xi \right] = 0
\end{align*}$$

Thus:

$$\| Y - X \phi \|_F^2 \geq \| Y - Y \phi^T (\phi \phi^T)^{-1} \phi \|_F^2$$

for all \( X \) and equality is attained with \( X = Y \phi^T (\phi \phi^T)^{-1} \).
III. Tucker Decomposition(s) and HOSVD.

III. A) Recap of Matrix SVD.

* A: MxN real matrix. We can always find
  
  \( \mathbf{U}, \mathbf{V} \) orthogonal (i.e. s.t. \( \mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I} \)
  \( \mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I} \)
  
  \( \mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T \)
  
  \( M \times N \quad M \times M \quad N \times N \)

  where \( \Sigma_{ij} \) has \( \Sigma_{ij} = \sigma_i \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(M,N)} \)
  and zero elsewhere are the singular values.

* Suppose \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 \) for \( R \leq \min(M,N) \)

  (so possibly some singular values are zero) we can
  write SVD as

  \[ \mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T \]

  \( \mathbf{U} \mathbf{U}^T = \mathbf{I} \quad \mathbf{V} \mathbf{V}^T = \mathbf{I} \)

  \( \mathbf{U} \mathbf{U}^T \mathbf{V} \mathbf{V}^T = \mathbf{I} \quad \mathbf{V} \mathbf{V}^T \mathbf{U} \mathbf{U}^T = \mathbf{I} \)

  are \( M \times R \) and \( N \times R \) arrays of orthonormal vectors.

* If the singular values are all distinct in this form the
  SVD decomposition is unique.
For matrices the Eckart–Young Theorem solves the following problem: Find the best rank approximation to a matrix.

**Eckart–Young Theorem**

Let \( K \leq K \). Then

\[
\arg \min_{A_K : \text{rank}(A_K) \leq K} \| A - A_K \|_F^2 = \sum_{i=1}^{K} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.
\]

where we truncated the SVD to the first \( K \) highest singular values \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_K \).

This Theorem is at the basis of dimensionality reduction techniques (e.g. PCA).

For tensors the problem of finding

\[
\arg \min_{T_K} \| T - T_K \|_F^2
\]

is not well defined. Indeed (see exercises) the space of tensors of rank \( K \) is not closed: we can find a sequence of rank \( K \) that converges to tensors of rank \( K+1 \).

(This jump of rank cannot happen for matrices.)
III.6) Concept of Multilinear Rank

To formulate analogs of SVD we need another concept of rank: the multilinear rank.
Consider a tensor $T$ and its matricizations $T_{(1)}$, $T_{(2)}$, $T_{(3)}$...

Let $R_1 = \text{dim space spanned by columns of } T_{(1)}$
$= \text{column rank of } T_{(1)}$.

$R_2 = \text{dim space spanned by columns of } T_{(2)}$
$= \text{column rank of } T_{(2)}$.

$R_3 = \text{idem}$.

(and so on for higher order tensors).

**Definition of Multilinear rank:**

$\text{rank} (T) = (R_1, R_2, R_3, ...)$

5. The multilinear rank is a collection of numbers representing the ranks of the matricizations.

(Recall: Tensor rank = $R =$ smallest number of terms in polyadic decomposition.

For matrices, the two concepts are equivalent $R_1 = R_2 = R$.)
III. C) Orthogonal Tucker Decomposition

**Theorem:** Let \( \text{rank}(T) = (R_1, R_2, R_3) \). It is always possible to decompose \( T \) as follows:

\[
T = \sum_{p=1}^{R_1} \sum_{q=1}^{R_2} \sum_{r=1}^{R_3} G_{pqr} \mathbf{u}_p \otimes \mathbf{v}_q \otimes \mathbf{w}_r
\]

where \( \left[ \mathbf{u}_1, \ldots, \mathbf{u}_{R_1} \right] \), \( \left[ \mathbf{v}_1, \ldots, \mathbf{v}_{R_2} \right] \), \( \left[ \mathbf{w}_1, \ldots, \mathbf{w}_{R_3} \right] \) are orthogonal arrays of vectors, and \( G_{pqr} \) is an \( R_1 \times R_2 \times R_3 \) array of numbers called the core tensor.

**Remarks:**

a) The core tensor is not diagonal. So this decomposition is different from the polyadic one. Also here \( u_i \)'s, \( v_i \)'s, \( w_i \)'s are \( 1 \) but in the polyadic they are not necessarily \( 1 \).

b) This decomposition is not unique. (Unlike sometimes, the SVD).

c) Truncations do not give a "best low-multilinear-rank approx" but a pretty good one.
Remark 6) Continued

If we use non-uniquely take three relative orthogonal metrics, $M^u_i \cdot R_i \times R_i$, $\tilde{M}^u_i \cdot R_i \times R_i$, $M^w \cdot R_i \times R_i$. Define

$$\tilde{\mu} p = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

$$\tilde{\nu} p = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

$$\tilde{\mu} p = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

and

$$\tilde{\nu} = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

and

$$\tilde{\nu} p = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

Check that

$$\sum_{p' \neq p} \tilde{\nu} p q r = \sum_{p' \neq p} \frac{R_1}{R_2} \left( M^{uT} \right)_{pp'} \frac{w'}{p'}$$

From the fact that $M^u_i \cdot M^{uT} = M^{uT} M^u = I$

iden for $M^u_i$, $M^{w_i}$. 
Remark c) continued: we will not prove here but just state two interesting facts:

- For best multilinear rank approx. In other words, given $T$ of multilinear rank $(R_1, R_2, R_3)$ of $K_1 \in R_1$, $K_2 \in R_2$, $K_3 \in R_3$, the following problem is well defined (the min is attained):

$$\arg \min_{\tilde{T}} \left\| \tilde{T} - \tilde{T} \right\|_F^2 = \tilde{T}$$

where $\tilde{T}$; rank $(K_1, K_2, K_3)$.

- Although the above problem is well defined we do not in general know of good algorithms. But the TUCKER-HOSVD (see next paragraph) gives a pretty good approximation in the following sense:

$$\left\| T - \tilde{T} \right\|_F^2 \leq \left( \text{min} \left( \text{max} \right) - 1 \right) \left\| T - \tilde{T} \right\|_F^2$$

where

$$\tilde{T} = \text{Trunc}_b \text{HOSVD}$$

to $\sum_{p \leq k_1} \sum_{q \leq k_2} \sum_{r \leq k_3}$.
III. d) Proof of Theorem (Tucker) and HOSVD

We provide a proof of the existence of a Tucker decomposition, which is constructive, and also gives us an algorithm. The particular decomposition obtained here is an "analog" of SVD and is called Higher Order Singular Value Decomposition (HOSVD).

Take the matrices $T_{(1)}$, $T_{(2)}$, $T_{(3)}$.

Perform usual matrix SVD on these matrices,

$$T_{(1)} = U_{I_1 \times R_1}^{(1)} \Sigma_{P_1 \times P_1}^{(1)} V_{R_1 \times I_2 \times I_3}^{(1)T}$$

$$T_{(2)} = U_{I_2 \times R_2}^{(2)} \Sigma_{P_2 \times P_2}^{(2)} V_{R_2 \times I_1 \times I_3}^{(2)T}$$

$$T_{(3)} = U_{I_3 \times R_3}^{(3)} \Sigma_{P_3 \times P_3}^{(3)} V_{R_3 \times I_1 \times I_2}^{(3)T}$$

which exist (recall the dimensions and ranks of $T_{(1)}$, $T_{(2)}$, $T_{(3)}$ to see this here).

Consider left singular vectors of $T_{(1)}$, $T_{(2)}$, $T_{(3)}$.

These are used in the matrices $U_{I_1 \times R_1}^{(1)}$, $U_{I_2 \times R_2}^{(2)}$, $U_{I_3 \times R_3}^{(3)}$, $V_{R_1 \times I_2 \times I_3}^{(1)T}$, $V_{R_2 \times I_1 \times I_3}^{(2)T}$, $V_{R_3 \times I_1 \times I_2}^{(3)T}$.
Finally compute the tensor

\[ G = T \left( \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right) \]

where by definition this means:

\[
G_{pqr} = \sum_{\alpha=1}^{I_1} \sum_{\beta=1}^{I_2} \sum_{\gamma=1}^{I_3} T_{\alpha \beta \gamma} U_{\alpha p}^{(1)} U_{\beta q}^{(2)} U_{\gamma r}^{(3)}
\]

The existence of the Tucker decomposition is then established by checking that with this \( G \) and the \( \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \) we must have (invertible relation using unitarity):

\[
T_{\alpha \beta \gamma} = \sum_{\rho=1}^{R_1} \sum_{\sigma=1}^{R_2} \sum_{\tau=1}^{R_3} G_{\rho \sigma \tau} U_{\alpha \rho}^{(1)} U_{\beta \sigma}^{(2)} U_{\gamma \tau}^{(3)}
\]

i.e. \( T = \sum_{pqr} G_{pqr} \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(3)} \).
This last proof gives an algorithm for obtaining a Tucker decomposition.

**HOSVD algorithm**

**Input** $T^{\times \times 3}$; **Output** $G_{\times \times 3}, U, V, W$.

1) From $T^{\times \times 3}$ consider matrices $T_{(1), T_{(2)}, T_{(3)}}$.

2) Compute left singular vectors of $T_{(1), T_{(2)}, T_{(3)}}$ from the matrix SVD. This yields the cores corresponding to non-zero singular values:

$$\begin{bmatrix} u_1 & \ldots & u_{r_1} \\ \vdots & \ddots & \vdots \\ u_{r_1} & \ldots & u_{r_3} \end{bmatrix}, \begin{bmatrix} v_1 & \ldots & v_{r_2} \\ \vdots & \ddots & \vdots \\ v_{r_2} & \ldots & v_{r_3} \end{bmatrix}, \begin{bmatrix} w_1 & \ldots & w_{r_3} \end{bmatrix}$$

3) Note that also gives a systematic way to compute the multilinear rank $(r_1, r_2, r_3)$.

3) Compute the "core tensor" from

$$G_{\times \times 3} = \sum_{\alpha \beta \gamma} T^{\times \times 3}_{\alpha \beta \gamma} u^\alpha v^\beta w^\gamma$$

This yields the HOSVD:

$$T = \sum_{\alpha \beta \gamma} G_{\times \times 3} u^\alpha \otimes v^\beta \otimes w^\gamma.$$

**Remark**: as explained before this is just one particular Tucker decomposition. These are not unique and can be obtained by rotations.