HOPLS-MILR Algorithm Documentation

Contents

1.1   Notation

- We use $\mathcal{G}$ to usually denote the core tensor in a Tucker decomposition. Tensors are also underlined bold capital letters, e.g. $\underline{\mathbf{X}}$.

1.2   Normal HOPLS

Recap: Original HOPLS Framework (Sequential Extraction of Component r)

We have:

$$\underline{\mathbf{X}} = \mathcal{G} \times_1 \mathbf{T} \times_2 \overline{\mathbf{P}}^{(1)} \times_3 \cdots \times_N \overline{\mathbf{P}}^{(N-1)} + \underline{\mathbf{E}}_R,$$

$$\underline{\mathbf{Y}} = \mathcal{D} \times_1 \mathbf{T} \times_2 \overline{\mathbf{Q}}^{(1)} \times_3 \cdots \times_M \overline{\mathbf{Q}}^{(M-1)} + \underline{\mathbf{F}}_R,$$

For each component r, operating on current residuals $\underline{\mathbf{E}}_r, \underline{\mathbf{F}}_r$:

1. **Find Loadings $\mathbf{P}_r^{(n)}, \mathbf{Q}_r^{(m)}$:**
   - The goal is to maximise $\|\mathcal{G}_r\|_F^2 \cdot \|\mathcal{D}_r\|_F^2$.
   - Using Proposition 3.3, this is equivalent to maximising $\|\langle \mathcal{G}_r, \mathcal{D}_r \rangle_{\{1;1\}}\|_F^2$.
   - Substituting the LS solutions for $\mathcal{G}_r, \mathcal{D}_r$ (from Proposition 3.1, assuming fixed $\mathbf{t}_r, \mathbf{P}_r^{(n)}, \mathbf{Q}_r^{(m)}$) into this, and using $\mathbf{t}_r^T \mathbf{t}_r = 1$, leads to the objective (Eq. 14 in HOPLS paper):

$$\max \left\| [\![\langle \underline{\mathbf{E}}_r, \underline{\mathbf{F}}_r \rangle_{\{1;1\}}; \mathbf{P}_r^{(1)T}, \ldots, \mathbf{P}_r^{(N-1)T}, \mathbf{Q}_r^{(1)T}, \ldots, \mathbf{Q}_r^{(M-1)T}]\!] \right\|_F^2$$

   - Let $\underline{\mathbf{C}}_r = \langle \underline{\mathbf{E}}_r, \underline{\mathbf{F}}_r \rangle_{\{1;1\}}$. The problem simplifies to:

$$\max_{\{\mathbf{P}_r^{(n)}\}, \{\mathbf{Q}_r^{(m)}\}} \left\| [\![\underline{\mathbf{C}}_r; \mathbf{P}_r^{(1)T}, \ldots, \mathbf{Q}_r^{(M-1)T}]\!] \right\|_F^2 \quad \text{s.t.} \quad \mathbf{P}_r^{(n)T} \mathbf{P}_r^{(n)} = \mathbf{I}_{L_{n+1}} \ \forall n, \ \mathbf{Q}_r^{(m)T} \mathbf{Q}_r^{(m)} = \mathbf{I}_{K_{m+1}} \ \forall m.$$

   - According to Proposition 3.2 (applied to $\underline{\mathbf{C}}_r$), this is equivalent to finding the best subspace approximation of $\underline{\mathbf{C}}_r$: $\min \left\| \underline{\mathbf{C}}_r - [\![\mathbf{G}_r^{(C_r)}; \mathbf{P}_r^{(1)}, \ldots, \mathbf{Q}_r^{(M-1)}]\!] \right\|_F^2$.

- This is solved by rank-$(L_2, \ldots, L_N, K_2, \ldots, K_M)$ HOOI on $\underline{\mathbf{C}}_r$ to get the orthonormal loadings $\mathbf{P}_r^{(n)}, \mathbf{Q}_r^{(m)}$ and the core $\underline{\mathbf{G}}_r^{(C_r)}$.
2. **Find Latent Vector $\mathbf{t}_r$ (from $\underline{\mathbf{E}}_r$):**
   - (Eq. 17 in HOPLS paper)

$$\mathbf{t}_r = \arg\min_{\mathbf{t}} \left\| \underline{\mathbf{E}}_r - [\![\mathcal{G}_r; \mathbf{t}, \mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}]\!] \right\|_F^2 \quad s.t. \quad \|\mathbf{t}_r\|_F = 1$$

   - Using Proposition 3.2, this is equivalent to:

$$\max_{\mathbf{t}} \left\| \underline{\mathbf{E}}_r \times_1 \mathbf{t}^T \times_2 \mathbf{P}_r^{(1)T} \ldots \right\|_F^2 \quad s.t. \quad \|\mathbf{t}_r\|_F = 1$$

   Solution: $\mathbf{t}_r$ is the first leading left singular vector of $(\underline{\mathbf{E}}_r \times_2 \mathbf{P}_r^{(1)T} \ldots \times_N \mathbf{P}_r^{(N-1)T})_{(1)}$.
3. **Find Core Tensors $\mathcal{G}_r, \mathcal{D}_r$ (Original LS - Proposition 3.1):**
   - $\mathcal{G}_{r,LS} = \underline{\mathbf{E}}_r \times_1 \mathbf{t}_r^T \times_2 \mathbf{P}_r^{(1)T} \ldots \times_N \mathbf{P}_r^{(N-1)T}$
   - $\mathcal{D}_{r,LS} = \underline{\mathbf{F}}_r \times_1 \mathbf{t}_r^T \times_2 \mathbf{Q}_r^{(1)T} \ldots \times_M \mathbf{Q}_r^{(M-1)T}$

1.3 Naive Ridge Idea:

To enhance stability and prevent overfitting in HOPLS, especially with high-dimensional or noisy data, we can introduce Ridge (L2) regularisation when estimating the core tensor $\mathcal{G}$ for the predictor tensor $\underline{\mathbf{X}}$ (and similarly for $\underline{\mathcal{D}}$ if $\underline{\mathbf{Y}}$ is a tensor).

The standard HOPLS estimates the core tensor $\mathcal{G}_{LS} \in \mathbb{R}^{1 \times L_2 \times \cdots \times L_N}$ for a single component by minimising:

$$\min_{\mathcal{G}} ||\underline{\mathbf{X}} - \mathcal{G} \times_1 \mathbf{t} \times_2 \mathbf{P}^{(1)} \cdots \times_N \mathbf{P}^{(N-1)}||_F^2$$

Where:

- Predictor tensor: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$
- Latent vector: $\mathbf{t} \in \mathbb{R}^{I_1}$, with $\mathbf{t}^\top \mathbf{t} = 1$ (orthonormal)
- Loading matrices: $\mathbf{P}^{(n)} \in \mathbb{R}^{I_{n+1} \times L_{n+1}}$ for $n = 1, \ldots, N-1$, each with orthonormal columns $(\mathbf{P}^{(n)\top} \mathbf{P}^{(n)} = \mathbf{I})$
- Target: Estimate the core tensor $\mathcal{G} \in \mathbb{R}^{1 \times L_2 \times \cdots \times L_N}$

The original HOPLS solution is (from Proposition 3.1 of the HOPLS paper):

$$\mathcal{G}_{LS} = \underline{\mathbf{X}} \times_1 \mathbf{t}^\top \times_2 \mathbf{P}^{(1)\top} \cdots \times_N \mathbf{P}^{(N-1)\top}$$

However, ridge-regularising formulates the problem with an additional penalty term:

Ridge-Regularised Formulation (Proposition 3.1-R): The objective function is modified to include an L2 penalty on the core tensor:

$$\min_{\mathcal{G}} \left(||\underline{\mathbf{X}} - \mathcal{G} \times_1 \mathbf{t} \times_2 \mathbf{P}^{(1)} \cdots \times_N \mathbf{P}^{(N-1)}||_F^2 + \lambda ||\mathcal{G}||_F^2 \right)$$

where $\lambda > 0$ is the regularisation parameter.

Solution: The solution for the Ridge-regularised core tensor $\mathcal{G}_{Ridge}$ is:

$$\mathcal{G}_{Ridge} = \frac{1}{1+\lambda} \mathcal{G}_{LS} = \frac{1}{1+\lambda} \left(\underline{\mathbf{X}} \times_1 \mathbf{t}^\top \times_2 \mathbf{P}^{(1)\top} \cdots \times_N \mathbf{P}^{(N-1)\top} \right)$$

Proof/Derivation: Let $L(\mathcal{G})$ be the Ridge-regularised objective function. We use mode-1 matricisation:

$$(\mathcal{G} \times_1 \mathbf{t} \times_2 \mathbf{P}^{(1)} \cdots \times_N \mathbf{P}^{(N-1)})_{(1)} = \mathbf{t}\mathcal{G}_{(1)}\mathbf{K}^\top$$

where $\mathcal{G}_{(1)}$ is the mode-1 matricisation of $\mathcal{G}$, and $\mathbf{K} = \mathbf{P}^{(N-1)} \otimes \cdots \otimes \mathbf{P}^{(1)}$. The objective becomes:

$$L(\mathcal{G}_{(1)}) = ||\underline{\mathbf{X}}_{(1)} - \mathbf{t}\mathcal{G}_{(1)}\mathbf{K}^\top||_F^2 + \lambda ||\mathcal{G}_{(1)}||_F^2$$

Taking the derivative with respect to $\mathcal{G}_{(1)}$ and setting to zero:

$$\frac{\partial L(\mathcal{G}_{(1)})}{\partial \mathcal{G}_{(1)}} = -2\mathbf{t}^\top (\underline{\mathbf{X}}_{(1)} - \mathbf{t}\mathcal{G}_{(1)}\mathbf{K}^\top)\mathbf{K} + 2\lambda \mathcal{G}_{(1)} = \mathbf{0}$$

Using $\mathbf{t}^\top \mathbf{t} = 1$ and $\mathbf{K}^\top \mathbf{K} = \mathbf{I}$ (due to orthonormality of $\mathbf{P}^{(n)}$):

$$-\mathbf{t}^\top \underline{\mathbf{X}}_{(1)} \mathbf{K} + (1)\mathcal{G}_{(1)}(\mathbf{I}) + \lambda \mathcal{G}_{(1)} = \mathbf{0}$$

$$\mathcal{G}_{(1)}(1 + \lambda)\mathbf{I} = \mathbf{t}^\top \underline{\mathbf{X}}_{(1)} \mathbf{K}$$

The term $\mathbf{t}^\top \underline{\mathbf{X}}_{(1)} \mathbf{K}$ is the mode-1 matricisation of the standard least-squares solution, $\mathcal{G}_{LS,(1)}$.

Thus, $\mathcal{G}_{Ridge,(1)} = \frac{1}{1+\lambda} \mathcal{G}_{LS,(1)}$. Converting back to tensor form gives the stated solution for $\mathcal{G}_{Ridge}$.

> The naive ridge solution is simply a scaled-down version of the original least-squares solution $\mathcal{G}_{LS}$. However only the magnitude is reduced, and the pattern of contributions from the core remain the same. Can we do better?

1.4 Mean Index-Weighted L2 Core Regularisation (MILR)

Steps 1 and 2 (finding loadings $\mathbf{P}_r, \mathbf{Q}_r$ and latent vector $\mathbf{t}_r$) that define the orthonormal basis for the current component r remain unchanged. The modification happens in **Step 3**.

1.4.1 Proposition 3.1-MILR - Mean Index-Weighted L2 Regularised Core:

Proposition 3.1 is a reworking of the original Proposition 3.1 from the HOPLS paper.

Given:

- A tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$
- Column orthonormal loading matrices $\mathbf{P}^{(n)} \in \mathbb{R}^{I_{n+1} \times L_{n+1}}, n = 1, \ldots, N-1$
  - (where $\mathbf{P}^{(n)}$ corresponds to mode $n+1$ of $\underline{\mathbf{X}}$ and has L_{n+1} columns)
- A latent vector $\mathbf{t} \in \mathbb{R}^{I_1}$ with $\|\mathbf{t}\|_F = 1$
- A global regularisation parameter $\lambda > 0$
- A weighting exponent $\alpha > 0$

1.4.2 Problem

The problem to find the core tensor $\mathcal{G} \in \mathbb{R}^{1 \times L_2 \times L_3 \times \cdots \times L_N}$ is:

$$\min_{\mathcal{G}} \left\| \underline{\mathbf{X}} - \underline{\mathbf{G}} \times_1 \mathbf{t} \times_2 \mathbf{P}^{(1)} \times_3 \mathbf{P}^{(2)} \cdots \times_N \mathbf{P}^{(N-1)} \right\|_F^2 + \lambda \sum_{l_2,\ldots,l_N} w_{l_2,\ldots,l_N} (\mathcal{G}_{1,l_2,\ldots,l_N})^2$$

where the weights $w_{l_2,\ldots,l_N}$ for the core tensor element $\mathcal{G}_{1,l_2,\ldots,l_N}$ (with $1 \le l_{j+1} \le L_{j+1}$ for $j = 1, \ldots, N-1$) are defined as:

$$w_{l_2,\ldots,l_N} = \begin{cases} \frac{1}{N-1} \sum_{j=1}^{N-1} \left(\frac{l_{j+1}}{L_{j+1}} \right)^\alpha & \text{if } N-1 > 0 \text{ (i.e., } N \ge 2) \\ 0 & \text{if } N-1 = 0 \text{ (i.e., } N = 1, \text{ where X is a vector, G is scalar)} \end{cases}$$

Here, l_{j+1} is the index along the mode of $\mathcal{G}$ that corresponds to the loading matrix $\mathbf{P}^{(j)}$ (which has L_{j+1} columns).

An element is $\mathcal{G}_{1,l_2,l_3,\ldots,l_N}$. This element interacts with column l_2 of $\mathbf{P}^{(1)}$ (which has L_2 columns), column l_3 of $\mathbf{P}^{(2)}$ (which has L_3 columns), $\ldots$, column l_N of $\mathbf{P}^{(N-1)}$ (which has L_N columns).

(Note: For $\mathcal{G} \in \mathbb{R}^{1 \times L_2 \times \cdots \times L_N}$, its indices are $(1, l_2, \ldots, l_N)$. $\mathbf{P}^{(1)}$ has L_2 columns, $\mathbf{P}^{(2)}$ has L_3 columns, $\ldots$, $\mathbf{P}^{(N-1)}$ has L_N columns. So the sum should be over the $N-1$ loading matrices $\mathbf{P}^{(1)}$ to $\mathbf{P}^{(N-1)}$. The k-th term in the sum corresponds to $\mathbf{P}^{(k)}$, whose columns are indexed by l_{k+1} and total columns L_{k+1}. We skip by one index as the first mode is for the latent vector $\mathbf{t}$.)

1.4.3 Solution

The solution for $\mathcal{G}$ (denoted $\mathcal{G}_{MILR}$) is given element-wise by:

$$(\mathcal{G}_{1,l_2,\ldots,l_N})_{MILR} = \frac{1}{1 + \lambda w_{l_2,\ldots,l_N}} (\mathcal{G}_{1,l_2,\ldots,l_N})_{LS}$$

This can be written compactly using element-wise (Hadamard) division and product:

$$\mathcal{G}_{MILR} = \mathcal{G}_{LS} \oslash (\mathbf{1} + \lambda \underline{\mathbf{W}}_{\mathcal{G}})$$

where $\oslash$ denotes element-wise division, and $\mathbf{1}$ is a tensor of ones with the same shape as $\mathcal{G}_{LS}$.

where the Least Squares solution (the best core tensor for given latent vector $\mathbf{t}$ and loadings $\mathbf{P}^{(n)}$) is:

$$(\mathcal{G})_{LS} = \underline{\mathbf{X}} \times_1 \mathbf{t}^T \times_2 \mathbf{P}^{(1)T} \times_3 \cdots \times_N \mathbf{P}^{(N-1)T}$$

1.4.4 *Proof of Proposition 3.1-MILR:*

Let $\mathbf{g} = \text{vec}(\mathcal{G}_{MILR}) \in \mathbb{R}^K$ and $\mathbf{x}_{proj} = \text{vec}((\mathcal{G})_{LS})$.

The orthonormality of $\mathbf{t}, \mathbf{P}^{(n)}$s implies $\mathbf{K}_{\mathcal{G}}^T \mathbf{K}_{\mathcal{G}} = \mathbf{I}$, which ensures the design matrix $\mathbf{K}_{\mathcal{G}}$ that maps $\mathbf{g}$ to $\text{vec}(\underline{\mathbf{X}}_{approx})$ satisfies $\mathbf{K}_{\mathcal{G}}^T \mathbf{K}_{\mathcal{G}} = \mathbf{I}$. Consequently, the unregularised objective $\|\underline{\mathbf{X}} - \underline{\mathbf{X}}_{approx}\|_F^2$ can be expressed in terms of $\mathbf{g}$ and its LS estimate $\mathbf{x}_{proj}$ as $\|\mathbf{x}_{proj} - \mathbf{g}\|_2^2 + \text{const}$, where $\text{const} = \|\underline{\mathbf{X}}\|_F^2 - \|\mathbf{x}_{proj}\|_F^2$. Thus, the minimisation problem becomes:

We can rewrite the problem as:

$$\min_{\mathbf{g}} \left[\|\mathbf{x}_{proj} - \mathbf{g}\|_2^2 + \lambda \mathbf{g}^T \mathbf{W}_{diag} \mathbf{g} \right]$$

where $\mathbf{W}_{diag} \in \mathbb{R}^{K \times K}$ is a diagonal matrix. The k-th diagonal entry of $\mathbf{W}_{diag}$, $(W_{diag})_{kk}$, is the weight w_k corresponding to the k-th element g_k of the vectorised core tensor (which itself corresponds to a specific $w_{l_2,\ldots,l_N}$). The objective function is separable with respect to the elements g_k and can be expressed as:

$$f(\mathbf{g}) = \sum_{k=1}^{K} \left(((x_{proj})_k - g_k)^2 + \lambda w_k g_k^2 \right)$$

Taking the derivative of the objective with respect to g_k and setting it to zero, we have:

$$\frac{\partial f(\mathbf{g})}{\partial g_k} = -2\left((x_{proj})_k - g_k\right) + 2\lambda w_k g_k = 0$$

$$(x_{proj})_k - g_k + \lambda w_k g_k = 0$$

$$(x_{proj})_k = g_k(1 + \lambda w_k)$$

$$\implies g_k = \frac{1}{1 + \lambda w_k}(x_{proj})_k$$

This proves the element-wise solution. ∎

Impact on Propositions 3.2, 3.3, 3.4:

- **Proposition 3.2:** Remains the same for the *unregularised* objective used to find the loadings via HOOI on $\underline{\mathbf{C}}_r$. If we tried choosing loadings $\mathbf{P}_r, \mathbf{Q}_r$ that maximise the norm of the *MILR-regularised* core tensor, the objective would be: $\max \left\| \mathcal{G}_{LS} \oslash (\mathbf{1} + \lambda_X \underline{\mathbf{W}}_{\mathcal{G}}) \right\|_F^2 \cdot \left\| \mathcal{D}_{LS} \oslash (\mathbf{1} + \lambda_Y \underline{\mathbf{W}}_{\mathcal{D}}) \right\|_F^2$ (where $\oslash$ denotes Hadamard (element-wise) division).

- **Proposition 3.3:** This algebraic property is unchanged.

- **Proposition 3.4 (Tensor-Matrix case for $\mathbf{t} = \mathbf{Yq}$):** Unchanged.

1.4.5   Why do we weight the core tensor elements by their indices?

- When performing HOOI/HOSVD on the cross covariance tensor $\underline{\mathbf{C}}_r$, one gets loading matrices $\mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}$.
- These matrices are ordered by importance due to the HOOI algorithm which works in an alternating least-squares manner and deflates away residuals as it decomposes the tensor.
- The index of a column in a loading matrix (e.g. l_2 for $\mathbf{P}_r^{(1)}$, l_3 for $\mathbf{P}_r^{(2)}$, etc) directly corresponds to its 'rank' or importance as determined by the HOOI decomposition on $\underline{\mathbf{C}}_r$.
- The core tensor $\mathcal{G}_r$ has elements like $\mathcal{G}_{1,l_2,\ldots,l_N}$ that quantifies the interaction between:
    - The latent vector $\mathbf{t}_r$ (mode 1)
    - The l_2-th column of $\mathbf{P}_r^{(1)}$ (mode 2)

– The l_3-th column of $\mathbf{P}_r^{(2)}$ (mode 3)

– ... and so on up to mode N.

- We define the weight for the core tensor element $\mathcal{G}_{1,l_2,...,l_N}$ as:

$$w_{l_2,...,l_N} = \frac{1}{N-1} \sum_{j=1}^{N-1} \left(\frac{l_{j+1}}{L_{j+1}} \right)^\alpha$$

- Here l_{j+1} is the index of the column in the loading matrix $\mathbf{P}_r^{(j)}$ that this specific core tensor element interacts with.
- L_{j+1} is the total number of columns (rank chosen) for that loading matrix $\mathbf{P}_r^{(j)}$.
- The term $\left(\frac{l_{j+1}}{L_{j+1}} \right)^\alpha$ is a normalised index which is closer to 0 for the first (most important) columns, and approaches 1 for the last (least important) columns.
- Exponent $\alpha > 0$ controls how much rapidly the penalty increases with the normalised index.
- The weights (shrinkage factor) scale up as the indexes increase

> **More intuition:** MILR is a heuristic where we assume that interactions involving less important loading vectors (higher indices) can be noisy and might contribute to overfitting. An analogy to consider two spectrums: On one end we have Principal Component Regression (PCR) where smaller components are discarded. The other end is Ridge Regression on PCA scores where all scores are used but penalised. MILR is a middle ground that doesn't discard, but it shrinks structurally, and selectively. Typical L2 regression would just shrink the entire core (which is easy to do but does not respect the hierarchies of the loadings in a decomposed tensor structure). While tensors can capture higher-order interactions, this flexibility is an overfitting risk. By selectively penalising interactions, it allows the stronger, more consistent higher-order signals (those primarily captured by the lower-indexed loadings and their interactions in the core tensor) to have a relatively larger influence on the final model and its predictions.

1.5 Algorithms For Mean Index-Weighted L2 Core Regularisation (HOPLS-MILR)

1.5.1 Algorithm: HOPLS-MILR (Tensor $\underline{\mathbf{X}}$ and Tensor $\underline{\mathbf{Y}}$)

REQUIRE: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}, \underline{\mathbf{Y}} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, $I_1 = J_1$. Number of latent vectors R. Ranks $\{L_k\}_{k=2}^{N}$ and $\{K_m\}_{m=2}^{M}$. Global regularisation strengths λ_X, λ_Y. Weighting exponent $\alpha > 0$. Convergence threshold ε.

ENSURE: $\{\mathbf{P}_r^{(n)}\}; \{\mathbf{Q}_r^{(m)}\}; \{\underline{\mathbf{G}}_r\}; \{\underline{\mathbf{D}}_r\}; \mathbf{T}$

1. **Initialisation:** $\underline{\mathbf{E}}_1 \leftarrow \underline{\mathbf{X}}, \quad \underline{\mathbf{F}}_1 \leftarrow \underline{\mathbf{Y}}, \quad \mathbf{T} \leftarrow []$.
2. **FOR** $r = 1$ **TO** R **DO**
   1. **IF** $\|\underline{\mathbf{E}}_r\|_F > \varepsilon$ AND $\|\underline{\mathbf{F}}_r\|_F > \varepsilon$ **THEN**
      1. $\underline{\mathbf{C}}_r \leftarrow \langle \underline{\mathbf{E}}_r, \underline{\mathbf{F}}_r \rangle_{\{1,1\}}$
      2. Rank-$(L_2, \ldots, L_N, K_2, \ldots, K_M)$ orthogonal Tucker decomposition of $\underline{\mathbf{C}}_r$ by HOOI:
         - $\underline{\mathbf{C}}_r \approx [\![\underline{\mathbf{G}}_r^{(C_r)}; \mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}, \mathbf{Q}_r^{(1)}, \ldots, \mathbf{Q}_r^{(M-1)}]\!]$
         - *(Yields orthonormal $\mathbf{P}_r^{(n)}$ [size $I_{n+1} \times L_{n+1}$] and $\mathbf{Q}_r^{(m)}$ [size $J_{m+1} \times K_{m+1}$])*
      3. $\underline{\mathbf{X}}_{proj,r} \leftarrow \underline{\mathbf{E}}_r \times_2 \mathbf{P}_r^{(1)T} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)T}$
      4. $\mathbf{t}_r \leftarrow$ first leading left singular vector of SVD($(\underline{\mathbf{X}}_{proj,r})_{(1)}$)
      5. $\mathbf{t}_r \leftarrow \mathbf{t}_r / \|\mathbf{t}_r\|_F$
      6. Append $\mathbf{t}_r$ to $\mathbf{T}$.
      7. **Calculate $\mathcal{G}_{r,LS}$:**
         - $\mathcal{G}_{r,LS} \leftarrow \underline{\mathbf{E}}_r \times_1 \mathbf{t}_r^T \times_2 \mathbf{P}_r^{(1)T} \times_3 \mathbf{P}_r^{(2)T} \ldots \times_N \mathbf{P}_r^{(N-1)T}$
         - *($\mathcal{G}_{r,LS}$ has size $1 \times L_2 \times L_3 \times \cdots \times L_N$)*
      8. **Construct Weight Tensor $\underline{\mathbf{W}}_{\mathcal{G},r}$ for $\mathcal{G}_{r,LS}$:**
         - Initialise $\underline{\mathbf{W}}_{\mathcal{G},r}$ with zeros, same size as $\mathcal{G}_{r,LS}$.
         - Number of loading matrices for $\mathcal{G}_r$ is $P_{count} = N - 1$.
         1. **IF** $P_{count} > 0$ **THEN**
            1. For each element $(\mathcal{G}_{r,LS})_{1,idx_2,...,idx_N}$ (where $1 \leq idx_j \leq L_j$): $P_{count} = 0$
               1. **FOR** $j = 1$ **TO** P_{count} **DO** *(Iterating through $\mathbf{P}_r^{(1)}$ to $\mathbf{P}_r^{(N-1)}$)*
                  1. $L_{current\_mode} = L_{j+1}$ *(Rank for mode $j + 2$ of $\underline{\mathbf{X}}$, corresponding to $\mathbf{P}_r^{(j)}$)*

2. $idx_{current\_mode} = idx_{j+1}$ *(Index for that mode in $\mathcal{G}_r$)*

3. sum_norm_indices $\leftarrow$ sum_norm_indices $+ (idx_{current\_mode}/L_{current\_mode})^\alpha$

     2. **END FOR**

     3. $(\underline{\mathbf{W}}_{\mathcal{G},r})_{1,idx_2,\ldots,idx_N} \leftarrow (1/\mathrm{P}_{\mathrm{count}}) * \mathrm{sum\_norm\_indices}$

   2. **ELSE** (if $N=1$, $\mathcal{G}_r$ is scalar): $(\underline{\mathbf{W}}_{\mathcal{G},r})_1 \leftarrow 0$ *(or 1, effectively making it standard ridge)*

   3. **END IF**

9. **Calculate MILR Core Tensor $\mathcal{G}_r$:** $\mathcal{G}_r \leftarrow \mathcal{G}_{r,LS}./(1 + \lambda_X * \underline{\mathbf{W}}_{\mathcal{G},r})$ *(element-wise operations)*

10. **Calculate $\mathcal{D}_{r,LS}$ and MILR $\mathcal{D}_r$ (analogously):**

   • $\mathcal{D}_{r,LS} \leftarrow \underline{\mathbf{F}}_r \times_1 \mathbf{t}_r^T \times_2 \mathbf{Q}_r^{(1)T} \ldots \times_M \mathbf{Q}_r^{(M-1)T}$

   • Construct $\underline{\mathbf{W}}_{\mathcal{D},r}$ based on indices of $\mathbf{Q}_r^{(m)}$ columns and ranks K_m.

   • Number of loading matrices for $\mathcal{D}_r$ is $\mathrm{Q}_{\mathrm{count}} = M - 1$.

   1. **IF** $\mathrm{Q}_{\mathrm{count}} > 0$ **THEN**

     1. For each element $(\mathcal{D}_{r,LS})_{1,kdx_2,\ldots,kdx_M}$: $w_{val} = (1/\mathrm{Q}_{\mathrm{count}}) * \sum_{j=1}^{\mathrm{Q}_{\mathrm{count}}} (kdx_{j+1}/K_{j+1})^\alpha$ $(\underline{\mathbf{W}}_{\mathcal{D},r})_{1,kdx_2,\ldots,kdx_M} \leftarrow w_{val}$

   2. **ELSE:** $(\underline{\mathbf{W}}_{\mathcal{D},r})_1 \leftarrow 0$

   3. **END IF**

   4. $\mathcal{D}_r \leftarrow \mathcal{D}_{r,LS}./(1 + \lambda_Y * \underline{\mathbf{W}}_{\mathcal{D},r})$

11. **Deflation:**

   • $\underline{\mathbf{E}}_{r+1} \leftarrow \underline{\mathbf{E}}_r - [\![\underline{\mathbf{G}}_r; \mathbf{t}_r, \mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}]\!]$

   • $\underline{\mathbf{F}}_{r+1} \leftarrow \underline{\mathbf{F}}_r - [\![\underline{\mathbf{D}}_r; \mathbf{t}_r, \mathbf{Q}_r^{(1)}, \ldots, \mathbf{Q}_r^{(M-1)}]\!]$

  2. **ELSE**

   1. Break

  3. **END IF**

3. **END FOR**

4. **Return** $\{\mathbf{P}_r^{(n)}\}; \{\mathbf{Q}_r^{(m)}\}; \{\underline{\mathbf{G}}_r\}; \{\underline{\mathbf{D}}_r\}; \mathbf{T}$

1.5.2   Algorithm: HOPLS2-MILR (Tensor $\underline{\mathbf{X}}$ and Matrix $\mathbf{Y}$)

In the Tensor-Matrix case, when the dependent variable $\mathbf{Y}$ is a matrix, the objective changes – we seek a rank-1 approximiation for $\mathbf{Y}$ in each component $d_r \mathbf{t}_r \mathbf{q}_r^T$.

$$\mathbf{Y} = \sum_{r=1}^{R} d_r \mathbf{t}_r \mathbf{q}_r^T + F_R$$

Recall Proposition 3.4 from the original HOPLS paper that for a given Y-loading vector $\mathbf{q}$ (with unit norm), the projection $\mathbf{t} = \mathbf{Y}\mathbf{q}$ provides the optimal (least-squares) rank-one approximation of the matrix $\mathbf{Y}$ that utilises $\mathbf{q}^T$ as its right singular vector (or loading). This justifies using $\mathbf{Y}\mathbf{q}$ as the representation of $\mathbf{Y}$'s information along the direction $\mathbf{q}$ when linking it to $\mathbf{t}$, which is the X-side latent structure in the HOPLS tensor-matrix algorithm.

We get a simplified cross-covariance tensor $\underline{\mathbf{C}}_r = \underline{\mathbf{E}}_r \times_1 \mathbf{F}_r^T$.

The optimisation problem becomes:

$$\max_{\mathbf{P}_r^{(n)}, \mathbf{q}_r} \left\| \underline{\mathbf{E}}_r \times_1 (\mathbf{F}_r \mathbf{q}_r)^T \times_2 \mathbf{P}_r^{(1)T} \cdots \times_N \mathbf{P}_r^{(N-1)T} \right\|_F^2$$

subject to $\|\mathbf{q}_r\|_F = 1$ and $\mathbf{P}_r^T \mathbf{P}_r = \mathbf{I}$ (orthonormality).

In the 2D case, subspace approximation is equal to low rank approximation. This is equivalent to performing a rank-$(1, L_2, \ldots, L_N)$ HOSVD on the cross-covariance tensor $\underline{\mathbf{C}}_r = \underline{\mathbf{E}}_r \times_1 \mathbf{F}_{1,mat}^T$, which yields the core tensor $\underline{\mathbf{G}}_r^{(C_r)}$ and the orthonormal X-loadings $\mathbf{P}_r^{(n)}$, the Y-loading vector $\mathbf{q}_r$ (which is the factor matrix corresponding to the mode-1 of $\underline{\mathbf{C}}_r$ derived from $\mathbf{F}_r^T$).

The HOPLS paper suggests $\mathbf{t}_r \leftarrow ((\underline{\mathbf{E}}_r \times_2 \mathbf{P}_r^{(1)T} \cdots \times_N \mathbf{P}_r^{(N-1)T})_{(1)} (\mathrm{vec}(\underline{\mathbf{G}}_r^{(C_r)}))^\dagger$. This ensures $\mathbf{t}_r$ is chosen considering the core $\underline{\mathbf{G}}_r^{(C_r)}$ obtained from the HOOI on $\underline{\mathbf{C}}_r$.

REQUIRE: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}, N \geq 2$ and $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$. Number of latent vectors R. Ranks $\{L_k\}_{k=2}^N$. Convergence threshold ε. Global regularisation strengths λ_X, λ_Y. Weighting exponent $\alpha > 0$.

ENSURE: $\{\mathbf{P}_r^{(n)}\}; \mathbf{Q}_{loadings}; \{\underline{\mathbf{G}}_r\}; \mathbf{D}_{coeffs}; \mathbf{T}$

1. **Initialisation:** $\underline{\mathbf{E}}_1 \leftarrow \underline{\mathbf{X}}, \mathbf{F}_{1,mat} \leftarrow \mathbf{Y}. \ \mathbf{T} \leftarrow [], \mathbf{Q}_{loadings} \leftarrow [], \mathbf{D}_{coeffs} \leftarrow [].$
2. **FOR** $r = 1$ **TO** R **DO**
    1. **IF** $\|\underline{\mathbf{E}}_r\|_F > \varepsilon$ AND $\|\mathbf{F}_{1,mat}\|_F > \varepsilon$ **THEN**
        1. $\underline{\mathbf{C}}_r \leftarrow \underline{\mathbf{E}}_r \times_1 \mathbf{F}_{1,mat}^T$
        2. Rank-$(1, L_2, \ldots, L_N)$ HOOI on $\underline{\mathbf{C}}_r$
            - (target ranks for modes from $\underline{\mathbf{E}}_r$ are $L_2, \ldots, L_N$; target rank for mode from $\mathbf{F}_{1,mat}$ is 1):
            - $\underline{\mathbf{C}}_r \approx \underline{\mathbf{G}}_r^{(C)} \times_1 \mathbf{q}_r \times_2 \mathbf{P}_r^{(1)} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)}$ *(Yields orthonormal $\mathbf{P}_r^{(n)}$, unit norm $\mathbf{q}_r$)*
        3. $\mathbf{t}_r \leftarrow ((\underline{\mathbf{E}}_r \times_2 \mathbf{P}_r^{(1)T} \cdots \times_N \mathbf{P}_r^{(N-1)T})_{(1)}(\text{vec}(\underline{\mathbf{G}}_r^{(C)}))^{\dagger})/\|\mathbf{t}_r\|$ *(normalise $\mathbf{t}_r$ like HOPLS)*
        4. Append $\mathbf{t}_r$ to $\mathbf{T}$. Append $\mathbf{q}_r$ to $\mathbf{Q}_{loadings}$.
        5. **Calculate MILR Core Tensor $\mathcal{G}_r$ for $\underline{\mathbf{X}}$:** (Same as HOPLS-MILR Tensor-Tensor algorithm)
            - $\mathcal{G}_{r,LS} \leftarrow \underline{\mathbf{E}}_r \times_1 \mathbf{t}_r^T \times_2 \mathbf{P}_r^{(1)T} \ldots \times_N \mathbf{P}_r^{(N-1)T}$
            - Construct $\underline{\mathbf{W}}_{\mathcal{G},r}$ based on indices $(idx_{j+1}/L_{j+1})^{\alpha}$ and normalise sum by $N - 1$.
            - $\mathcal{G}_r \leftarrow \mathcal{G}_{r,LS}./(1 + \lambda_X * \underline{\mathbf{W}}_{\mathcal{G},r})$
        6. **Calculate regularised scalar coefficient d_r for Y (Standard Ridge):**
            - The "core" for $\mathbf{Y}$ is the scalar d_r. It has only one element. Its "index-weight" w_j would be effectively constant (e.g., 1, or 0 if $N - 1 = 0$ like formula implies, which is fine). So it reduces to standard ridge. $\mathbf{u}_{r,vec} \leftarrow \mathbf{F}_{1,mat}\mathbf{q}_r \ d_{r,LS} \leftarrow \mathbf{t}_r^T \mathbf{u}_{r,vec} \ d_r \leftarrow \frac{1}{1+\lambda_Y} d_{r,LS}$ *(No complex w_j needed for a scalar)* Append d_r to $\mathbf{D}_{coeffs}$.
        7. **Deflation:**
            - $\underline{\mathbf{E}}_{r+1} \leftarrow \underline{\mathbf{E}}_r - [\![\underline{\mathbf{G}}_r; \mathbf{t}_r, \mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}]\!]$
            - $\mathbf{F}_{1,mat} \leftarrow \mathbf{F}_{1,mat} - d_r \mathbf{t}_r \mathbf{q}_r^T$
    2. **ELSE**
        - Break
    3. **END IF**
3. **END FOR**
4. **Return** $\{\mathbf{P}_r^{(n)}\}; \mathbf{Q}_{loadings}; \{\underline{\mathbf{G}}_r\}; \mathbf{D}_{coeffs}; \mathbf{T}$