Efficient Algorithms for Regularized Nonnegative Scale-invariant Low-rank Approximation Models

Jeremy E. Cohen[1] and Valentin Leplat[2]

[1]INSA Lyon, Universite Claude Bernard Lyon 1, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69XXX, Lyon, France
[2]Skoltech, Center for Artificial Intelligence Technology, Moscow, Russia.

Abstract

Regularized nonnegative low-rank approximations such as sparse Nonnegative Matrix Factorization or sparse Nonnegative Tucker Decomposition are an important branch of dimensionality reduction models with enhanced interpretability. However, from a practical perspective, the choice of regularizers and regularization coefficients, as well as the design of efficient algorithms, is challenging because of the multifactor nature of these models and the lack of theory to back these choices. This paper aims at improving upon these issues. By studying a more general model called the Homogeneous Regularized Scale-Invariant, we prove that the scale-invariance inherent to low-rank approximation models causes an implicit regularization with both unexpected beneficial and detrimental effects. This observation allows to better understand the effect of regularization functions in low-rank approximation models, to guide the choice of the regularization hyperparameters, and to design balancing strategies to enhance the convergence speed of dedicated optimization algorithms. Some of these results were already known but restricted to specific instances of regularized low-rank approximations. We also derive a generic Majorization Minimization algorithm that handles many regularized nonnegative low-rank approximations, with convergence guarantees. We showcase our contributions on sparse Nonnegative Matrix Factorization, ridge-regularized Canonical Polyadic decomposition and sparse Nonnegative Tucker Decomposition.

1 Introduction

1.1 Context

Unsupervised machine learning heavily relies on Low-Rank Approximation (LRA) models, which serve as fundamental tools. LRA models can be understood as dimensionality reduction techniques, exploiting the inherent multilinearity within datasets represented as matrices or tensors. One of the most commonly used LRA methods is Principal Component Analysis (PCA), which identifies orthogonal principal components within a dataset. Depending on the task at hand, the orthogonality constraint can be detrimental when a physical interpretation is to be given to the principal components [12, 26].

Over the past two decades, researchers have explored various constrained LRA models with enhanced interpretability properties, among which Nonnegative Matrix Factorization (NMF) is noteworthy. NMF replaces PCA's orthogonality constraint with an elementwise nonnegativity constraint, allowing it to represent data as a part-based decomposition [22, 37]. For tensorial data, models like Nonnegative Canonical Polyadic Decomposition (CNPD) have received significant attention. The Nonnegative Tucker Decomposition (NTD) has also gained popularity as a multilinear dictionary-learning model [48]. It is worth noting that unconstrained LRA models face interpretation challenges due to inherent rotation ambiguities. Therefore, the interpretability of LRA models largely hinges on the constraints, or regularizations, imposed on their parameters.

While the theoretical and algorithmic aspects of regularized LRA have been extensively explored, several practical challenges persist. Regularized LRA being inherently multidimensional/multi-factor problems, these challenges include selecting appropriate regularization functions and hyperparameters for each parameter matrix describing a low-dimensional subspace and devising efficient algorithms for non-Euclidean loss functions. In this work, we aim to address these two issues by studying a wider framework coined the Homogeneous Regularized Scale-Invariant problem (HRSI). In the process, we shed light on implicit regularization effects inherent to the scale invariance of explicitly regularized LRA.

Notation	Meaning
x	Vector
X	Matrix
$\mathcal{X}$	Tensor
$\mathbb{R}_+^{\times_{i=1}^d m_i}$	Set of tensors of order d with dimensions $m_1 \times \ldots \times m_d$, with real nonnegative entries
$X \times_i U$	n-mode product along the i-th mode between tensor X and matrix U
$\|X\|_F$	Frobenius norm of matrix or tensor X
$\|X\|_1$	ℓ_1 norm of matrix or tensor X
X^T	Transposed matrix X
$X[i,j]$	Element indexed by (i,j) in matrix X
$X[:,q]$	Column indexed by q in matrix X
$\{X_i\}_{i\leq n}$	Set of matrices or tensors X_i indexed by i from 1 to n
$\mathrm{dom}(g)$	Set of vectors x such that $g(x) < +\infty$
$\eta_{\mathcal{C}}$	Characteristic function of convex set $\mathcal{C}$
I_r	Identity matrix of size $r \times r$
$x \otimes y$	Tensor (outer) product
$A \odot B$	Hadamard (elementwise) product between two tensors A and B
X^α	Elementwise power of tensor X with exponent α
δ_{ii}	Sparse diagonal matrix with one nonzero at position i
x^*	Solution to a problem $\min_x f(x)$, should be clear from context

Table 1: Useful matrix and tensor notations. More details on tensor computation and multi-linear algebra can be found in [32].

In the following sections of this introduction, we formalize our study's general framework, outline its current limitations, summarize our contributions, and provide an overview of the manuscript's structure. Useful notations are summarized in Table 1.

1.2 The Homogeneous Regularized Scale-Invariant model

Let

$$\underset{\forall i\leq n,\, X_i\in\mathbb{R}^{m_i\times r}}{\operatorname{argmin}} f(\{X_i\}_{i\leq n}) + \sum_{i=1}^{n}\mu_i \sum_{q=1}^{r} g_i(X_i[:,q]) \tag{1}$$

a set of solutions to sample from, where f is a continuous map from the Cartesian product $\times_{i=1}^n \mathbb{R}^{m_i\times r} \cap \mathrm{dom}(g_i)$ to the positive reals, $\{\mu_i\}_{i\leq n}$ is a set of positive regularization hyperparameters, and $\{g_i\}_{i\leq n}$ is a set of lower semi-continuous regularization maps from $\mathbb{R}^{m_i}$ to $\mathbb{R}_+$. We assume that total cost is coercive, such that a minimizer always exists.. Furthermore, we assume the following assumptions hold:

A1 Function f is invariant to a column-wise scaling of the parameter matrices, that is for any sequence of positive diagonal matrices $\{\Lambda_i\}_{i\leq n}$,

$$f(\{X_i\Lambda_i\}_{i\leq n}) = f(\{X_i\}_{i\leq n}) \text{ if } \prod_{i=1}^{n}\Lambda_i = I_r,\ \Lambda_i > 0. \tag{2}$$

In other words, scaling any two columns $X_{j_1}[:,q]$ and $X_{j_2}[:,q]$ inversely proportionally has no effect on the value of f.

A2 Each regularization function g_i is positive homogeneous[1] of degree p_i, that is for all $i \leq n$ there exists $p_i > 0$ such that for any $\lambda > 0$ and any $x \in \mathbb{R}^{m_i}$,

$$g_i(\lambda x) = \lambda^{p_i} g_i(x) \tag{3}$$

This holds in particular for any ℓ_p norm in the ambient vector space.

[1] Our work could easily extend to homogeneous functions, but given that regularization function we consider are positive, positive homogeneity is sufficient and simplifies the subsequent derivations.

A3 Each function g_i is definite positive, so that for any $i \leq n$, $g_i(x) = 0$ if and only if x is null. [2]

We call the approximation problem (1) with assumptions A1, A2 and A3 the Homogeneous Regularized Scale-Invariant problem (HRSI).

Remark 1. *As clearly stated in Equation 1, the regularization terms are separable with respect to columns of each parameter matrix. This means that a penalization on the first matrix X_1 of the form*

$$\|X_1\|_F^2 := \sum_{q=1}^{r} \|X_1[:, q]\|_2^2$$

fits the framework, but not a regularization of the form $\|X_1\|_F$.

HRSI is quite general, but covers many instances of regularized LRA problems. To illustrate this, we note that computing the solutions to a (double) sparse NMF model with a regularization promoting sparse solutions for matrices X_1 and X_2 and the Kullback-Leibler divergence as a loss function can be formalized as sampling from

$$\underset{X_1 \geq 0, X_2 \geq 0}{\operatorname{argmin}} \ \mathrm{KL}(M, X_1 X_2^T) + \mu_1 \|X_1\|_1 + \mu_2 \|X_2\|_1 \tag{4}$$

where $M \in \mathbb{R}^{m_1 \times m_2}$ is the input data matrix. This formulation for sparse NMF fits the HRSI framework since the cost is scale-invariant and the regularizations are homogeneous and columnwise separable. More precisely, sparse NMF as defined in (4) is HRSI with $f(\{X_i\}_{i \leq n}) = \mathrm{KL}(M, X_1 X_2^T)$, $g_1(x) = \|x\|_1 + \eta_{\mathbb{R}_+^{m_1}}(x)$ and $g_2(x) = \|x\|_1 + \eta_{\mathbb{R}_+^{m_2}}(x)$ where η_C denotes the indicator function associated to the convex set C, that is $\eta_C(x) = 0$ for any $x \in C$, and $\eta_C(x) = \infty$ otherwise.

More details about regularized NMF, NCPD and NTD, and how our results apply to each of these models, are deferred to section 4.

1.3 Current limitations

Let us consider conventional variational problems, focusing the well-known LASSO problem [68]. Typically, a single regularization hyperparameter is used for this problem, although more complex models do exist [7]. Representer theorems state that solutions to such problems are combinations of extreme points within the regularization unit ball [8, 71]. It is important to note that there exists a maximum value for the regularization hyperparameter, beyond which all solutions become degenerate [69]. Additionally, homotopy methods establish a link between the regularity of solutions and the values of the regularization hyperparameters [15]. In simpler terms, the relationship between the penalty function choice, the regularization hyperparameter value, and the solution set is well-established in standard cases. However, in scale-invariant, multifactor models, this relationship is not as straightforward, and we detail here-under two crucial challenges:

- **Hyperparameter selection challenge**: Selecting hyperparameters for HRSI is not as simple as traditional cross-validation. Since HRSI exhibits scale-invariance, it has been briefly discussed in the literature [52, 56] that equivalent models arise when regularization hyperparameters are scaled, taking into account the homogeneity degree of the regularization terms. Therefore, it is not trivial to connect a specific regularization hyperparameter value to the regularity of solutions. To the best of our knowledge, there are no known maximal regularization hyperparameter values for models like sparse NMF or sparse NTD, and homotopy algorithms are not readily available. Furthermore, tuning multiple regularization hyperparameters can be challenging in practice.
- **Lack of characterisation of solutions regularity**: In the multi-factor setting, to the best of our knowledge, there is no existing Representer theorem to guide the choice for the regularization functions.

Moving on, the efficiency of algorithms for HRSI with non-Euclidean loss functions, particularly in the context of nonnegative parameter matrices and beta-divergence loss, has been underexplored. On one hand, designing efficient optimization algorithms for NMF with non-Euclidean loss is a relatively mature topic. In particular, a family of loss functions which is widely studied for both tensor and matrix nonnegative factorizations is the beta-divergence [9, 20]. Many algorithms, including the popular Majorization-Minimization (MM) algorithm, have been proposed to solve this problem [20, 37]. On the other hand, there are a few key remaining issues:

[2] This assumption limits our framework from applying to arbitrary homogeneous regularizations such as Total Variation norms or plug-and-play RELU networks.

- **MM algorithms for regularized LRA**: While MM updates are commonly used for NMF with beta-divergences as loss functions, they do not extend trivially for regularized factorization. Moreover their extension to tensor decompositions, albeit straightforward, is not readily available in the literature.
- **Challenges with normalization constraints**: Many regularized LRA models rely on normalization constraints to simplify the tuning of multiple hyperparameters and mitigate degeneracy. However, this approach can make the problem more challenging to solve due to nonconvexity issues. To the best of our knowledge, MM algorithms have no known convergence guarantees in this context.

1.4 Contributions and Structure

Our primary contribution involves providing deep insights into the selection of regularization terms g_i along with their associated hyperparameters μ_i within the HRSI framework, and the design of cost-effective balancing strategies to enhance the convergence speed of dedicated optimization algorithms. In Section 2.2 we show that regularized Low-Rank Approximation (LRA) solutions inherently benefit from implicit regularization due to their scale-invariance. This insight allows us to first unveil implicit regularization effects stemming from certain combinations of explicit regularizations. An important example: penalizing the squared Euclidean norm of all parameter matrices in a tensor CPD implies a group-LASSO regularization on the collection of rank-one terms, thus revealing the tensor CP rank. This idea was previously known for unconstrained matrix factorization [60] and explored under a Bayesian framework for tensor models [47]. We illustrate these implicit regularization effects through the lens of three models: sparse NMF (sNMF), ridge-regularized Nonnegative Canonical Polyadic Decomposition (rNCPD), and Sparse Nonnegative Tucker Decomposition (sNTD), both on synthetic data and realistic data. These findings not only simplify the difficult task of regularization and hyperparameter selection but also introduce a fresh perspective on characterizing regularized LRA model solutions.

On the front of algorithm design, we introduce a general alternating MM meta-algorithm which provably computes a stationary point of the regularized LRA problem. This approach accommodates a wide range of data fitting terms utilizing the β-divergence family and various regularization functions. Notably, a significant application of this MM algorithm is in addressing LRA with ridge regularization.

A third contribution is to show formally (on a toy problem) and experimentally on the different showcases that performing an explicit balancing of the parameters within the proposed Meta-algorithm helps with the convergence speed, which otherwise may be sublinear. This balancing procedure scales the columns of the parameter matrices so that the regularization terms are minimized, and this can be done with relatively small computational cost. In the case of sNTD where performing balancing is however nontrivial, we provide an iterative balancing strategy akin to the Sinkhorn algorithm.

These results are already partially known. Implicit regularization has been studied in several recent works, in particular in the context of matrix factorization and neural networks [60, 70]. This work generalizes these findings to higher order low-rank approximations and proposes a simple methodology to study the relationships between explicit/implicit regularizations. Some works have proposed to make use of balancing effects for practical purposes [24] albeit with a different purpose. So-called "scaling-swamps" have been moreover observed in the context of sparse NMF [52]; a slow-convergence phenomenon we describe in more details in Section 3. We review the literature in more details in Section 2.4.

The paper is structured as follows. Section 2 introduces the concept of implicit regularization induced by scale-invariance in HRSI, provides closed-form expression for the optimal balancing of parameter matrices in HRSI, and explains how the implicit regularization can inform on the effect of the regularization terms and the choice of regularization hyperparameters. Section 3 details how to build alternating algorithms for regularized low-rank approximations with a β-divergence loss using a Majorization-Minimization procedure. It also explains how to leverage practically the implicit balancing effect by enforcing an explicit balancing, and why explicit balancing can help speed up convergence theoretically on a toy example. Finally, Section 4 showcases our findings on three different models: sparse Nonnegative Matrix Factorization, ridge Canonical Polyadic Decomposition, and sparse Nonnegative Tucker Decompoition.

2 Scale invariance in HRSI leads to implicit balancing

In Section 1, we introduced HRSI and now let us study some of its properties. Firstly, we discuss the ill-posedness of HRSI, emphasizing the importance of proper regularization for each mode. After addressing this issue, we move on to our main contribution: demonstrating that the scale invariance in the cost function leads

to an optimal scaling with respect to the regularization, effectively balancing the solutions. As a result, an underlying optimization problem emerges, which is essentially equivalent[3] to HRSI and exhibits an implicit regularization effect. This implicit HRSI problem provides valuable insights into the opportunities and limitations associated with selecting appropriate regularization functions $\{g_i\}_{i\leq n}$ and regularization hyperparameters.

2.1 Ill-posedness of partially regularized homogeneous scale-invariant problems

In Equation (1) defining HRSI, we assumed that the regularization hyperparameters are nonnegative. However, it is known in the literature [41], albeit not proven in all generality to the best of our knowledge, that if the parameter μ_j for a specific index j is zero, the loss's scale-indeterminacy can result in degenerate solutions for HRSI. To explain further, matrix X_j can absorb the energy from all other matrices X_i (where $i \leq n$ and $i \neq j$), causing a significant decrease in all the regularization terms. In Proposition 1 below, we formally demonstrate that this partially regularized HRSI problem has the same optimal cost as an unregularized problem, but with unattainable solutions.

Proposition 1. *Let $\mu_j = 0$ for at least one index $j \leq n$. Then the infimum of the function*

$$\phi(\{X_i\}_{i\leq n}) := f(\{X_i\}_{i\leq n}) + \sum_{i=1}^{n} \mu_i \sum_{q=1}^{r} g_i(X_i[:, q]) \tag{5}$$

is equal to $\inf_{X_i \in \mathbb{R}^{m_i \times r}} f(\{X_i\}_{i\leq n})$. In other words, the regularization has no effect on the optimal cost value. Moreover, the infimum is not attained unless f attains its minimum at $X_i = 0$ for all $i \leq n$.

The proof is provided in Appendix A. In other words, unless all parameter matrices are regularized, the regularization functions have no effect on the minimum of the cost and solutions cannot be attained. Therefore in the following we suppose that all matrices X_i are regularized, with associated parameters $\mu_i > 0$.

2.2 Optimal balancing of parameter matrices in HRSI

Let us now enter the main discussion of this section, that is the effect of scaling invariance on regularization. Essentially, since the loss function f is scale-invariant, one can minimize the whole cost with respect to scaling of columns by only minimizing the regularization terms.

Homogeneity and positive definiteness then enables us to obtain a straightforward closed-form solution for the scaling subproblem. This solution provides valuable insights into the hidden regularizations and invariances within HRSI.

2.2.1 Optimal scaling of regularization terms is related to the weighted geometrical mean

Let $\{X_i\}_{i\leq n}$ be any set such that all X_i have nonzero columns, and $q \leq r$ a column index. To ease the notation, we pose $a_i = \mu_i g_i(X_i[:, q])$, note that $a_i > 0$ because of assumption **A3**, and consider the following minimization problem:

$$\min_{\forall i\leq n,\, \lambda_i \geq 0} \sum_{i=1}^{n} \lambda_i^{p_i} a_i \text{ such that } \prod_{i=1}^{n} \lambda_i = 1 \ . \tag{6}$$

This cost is exactly the loss in (1) where f and $g_i(X_i[:, p])$ for $p \neq q$, which are constant with respect to a scaling of the q-th columns represented by the parameters $\{\lambda_i\}_{i\leq n}$, have been removed. Our goal now is to determine the optimal scales.

One can observe that Problem (6) admits a unique optimal solution. Notably, this problem is related to an alternative definition of the weighted geometric mean of $p_i a_i$ with weights $\frac{1}{p_i}$, as also mentioned by Tibshirani in the context of neural networks weights scaling [70]. The optimal scales can provably be computed from the weighted geometrical mean β of $\{p_i a_i, \frac{1}{p_i}\}$ as

$$\lambda_i^* = \left(\frac{\beta}{p_i a_i}\right)^{\frac{1}{p_i}}, \tag{7}$$

[3]We use "essential equivalence" to feature two optimisation problems for which any solution to one is a solution to the other, up to a scaling which is compliant with assumption A1.

where $\{\lambda_i^*\}_{i\leq n}$ is the set of solutions to the scaling problem (6) and β is given as

$$\beta = \left(\prod_{i\leq n} (p_i a_i)^{\frac{1}{p_i}}\right)^{\frac{1}{\sum_{i\leq n}\frac{1}{p_i}}}. \tag{8}$$

Since at optimality the balancing procedure should have no effect, given a solution X_i^* to HRSI, it holds that $\lambda_i^* = 1$ for all $i \leq n$ with $a_i = \mu_i g_i(X_i[:,q]^*)$. In that case, we can infer that all summands in (6) must be equal up to a multiplicative factor, *i.e.* for any $i \leq j \leq n$,

$$p_i a_i = p_j a_j = \beta. \tag{9}$$

We can therefore state the following Proposition characterising the solutions of HRSI.

Proposition 2. *If $\mu_i > 0$ for all i, any solution $\{X_i^*\}_{i\leq n}$ of Problem (1) satisfies $p_i\mu_i g_i(X_i^*[:,q]) = \beta_q$ for all $i \leq n$ and $q \leq r$, where*

$$\beta_q = \left(\prod_{i\leq n} (p_i\mu_i g_i(X_i^*[:,q]))^{\frac{1}{p_i}}\right)^{\frac{1}{\sum_{i\leq n}\frac{1}{p_i}}}. \tag{10}$$

The proof of Proposition 2 is given in Appendix C. If all regularization hyperparameters μ_i are nonzero but a column $X_i[:,q]$ is null, by setting columns $X_i[:,q]$ to zero for each $i \leq n$, the penalty functions are trivially minimized with a constant cost. Therefore the optimal balancing in the presence of a null column $X_i[:,q]$ is to set the corresponding columns $X_{j\neq i}[:,q]$ to zero as well, and Proposition 2 therefore allows solutions with zero-columns.

2.2.2 Balancing algorithm and essentially equivalent scale-free implicit HRSI

We have seen in the previous section that it is possible to compute the optimal scaling of parameter matrices in order to minimize the regularization terms, while keeping the cost f constant. In simpler terms, at any specific point within the parameter space, we can reduce the cost function in (1) by applying a scaling that we will refer to as the balancing procedure. Formally, given a set of parameter matrices $\{X_i\}_{i\leq n}$ with nonzero columns, if all μ_i are positive, the balancing procedure

$$\tilde{X}_i[:,q] = \left(\frac{\beta_q}{p_i\mu_i g_i(X_i[:,q])}\right)^{1/p_i} X_i[:,q] \tag{11}$$

with β_q as defined in Proposition 2 defines a new set of parameters $\{\tilde{X}_i\}_{i\leq n}$ which solves the optimal scaling problem

$$\min_{\forall i\leq n,\ \Lambda_i\in\mathbb{R}^{r\times r}} f(\{X_i\Lambda_i\}_{i\leq n}) + \sum_{i=1}^{n}\mu_i\sum_{q=1}^{r} g_i(\Lambda_i[q,q]X_i[:,q]) \text{ where } \Lambda_i \text{ are diagonal matrices such that } \prod_{i\leq n}\Lambda_i = I_r. \tag{12}$$

Algorithm 1 Balancing algorithm

Require: A collection of columns $\{X_i[:,q]\}_{i\leq n}$ for a fixed index $q \leq r$, regularization hyperparameters $\mu_i > 0$, regularization functions g_i and their homogeneity degree p_i.

```
if X_i[:,q] = 0 for any i ≤ n then
    Set X_i[:,q] = 0 for all i ≤ n.
else
    Compute the geometric mean β_q = (∏_{i≤n} (p_i μ_i g_i(X_i[:,q]))^{1/p_i})^{1/(Σ_{i≤n} 1/p_i)}
    for i=1 : n do
        Set X̃_i[:,q] = (β_q / (p_i μ_i g_i(X_i[:,q])))^{1/p_i} X̃_i[:,q]
    end for
end if
return {X_i[:,q]}_{i≤n}
```

These equations provide an algorithmic procedure to minimize the penalizations terms in scale-invariant problems, without essentially modifying the solution since low-rank approximations are generally interpreted

up to scaling of their components. This balancing procedure will be part of the algorithm proposed in Section 3 used to tackle the prototypal class of scale-invariant regularized problems given in (1).

Another crucial observation is that a global solution $\{X_i^*\}_{i\leq n}$ of HRSI must adhere to the balancing condition, for which $p_i\mu_i g_i(X_i^*[:,q])$ remains constant for each $i \leq n$ and $q \leq r$. Consequently, we may limit the set of solutions for HRSI to that set. This leads to an interesting observation: we may transform the columnwise sum of regularization terms into a columnwise product of regularizations. Indeed, since all regularization terms are equal up to factors p_i, the regularization terms $\sum_{q=1}^r \sum_{i=1}^n \mu_i g_i(X_i[:,q])$ after optimal scaling become $\sum_{i=1}^n \frac{1}{p_i} \sum_{q=1}^r \beta_q$. The latter can be expressed as an implicit regularization term as follows

$$\tilde{\mu} \sum_{q=1}^{r} \left(\prod_{i=1}^{n} g_i(X_i[:,q])^{\frac{1}{p_i}} \right)^{\frac{1}{\sum_{i=1}^n \frac{1}{p_i}}}, \tag{13}$$

where the penalty weight, $\tilde{\mu}$, is set as $\tilde{\mu} = \left(\prod_{i=1}^n (p_i\mu_i)^{\frac{1}{p_i}}\right)^{\frac{1}{\sum_{i=1}^n \frac{1}{p_i}}} \left(\sum_{i=1}^n \frac{1}{p_i}\right)$. When all p_i-s are equal, the implicit regularization coefficient conveniently simplifies to $\tilde{\mu} = n\left(\prod_{i=1}^n \mu_i\right)^{\frac{1}{n}}$, and the regularization term boils down to $\sum_{q=1}^r \left(\prod_{i=1}^n g_i(X_i[:,q])\right)^{\frac{1}{n}}$.

The implicit regularization (13) is of significant interest in our perspective. The optimization obtained by switching out the explicit regularization with the implicit one,

$$\underset{\forall i\leq n,\, X_i\in\mathbb{R}^{m_i\times r}}{\operatorname{argmin}} f(\{X_i\}_{i\leq n}) + \tilde{\mu} \sum_{q=1}^{r} \left(\prod_{i=1}^{n} g_i(X_i[:,q])^{\frac{1}{p_i}} \right)^{\frac{1}{\sum_{i=1}^n \frac{1}{p_i}}}. \tag{14}$$

is fully scale-invariant in the sense of A1. Then, any solution to the implicit HRSI problem (14) is necessarily a solution of the explicit HRSI (1) up to a column scaling, and conversely. The proof relies on the fact that $\{\{Z_i\}_{i\leq n} \mid \forall i \leq n, Z_i = X_i\Lambda_i, \prod_{i\leq n}\Lambda_i = 1, X_i \in \mathbb{R}^{m_i\times r}\}$ is identical to $\times_{i\leq n}\mathbb{R}^{m_i\times r}$. Denoting $\phi(\{X_i\}_{i\leq n}) = f(\{X_i\}_{i\leq n}) + \sum_{i=1}^n \mu_i \sum_{q=1}^r g_i(X_i[:,q])$, this implies that

$$\inf_{\forall i\leq n,\, X_i\in\mathbb{R}^{m_i\times r}} \phi(\{X_i\}_{i\leq n}) = \inf_{\forall i\leq n,\, X_i\in\mathbb{R}^{m_i\times r},\ \prod_{i\leq n}\Lambda_i=1} \phi(\{X_i\Lambda_i\}_{i\leq n}) \tag{15}$$

We get a trivial mapping between the minimizers of HRSI (left hand side) and the implicit HRSI (right hand side).

Proposition 3. *Given regularization hyperparameters λ_i strictly positive, the HRSI problem* (1) *is essentially equivalent to the implicit HRSI problem* (14)*. In other words, every solution of HRSI is a solution to the implicit HRSI, and conversely, any solution of the implicit HRSI is, up a scaling of the solution's columns, a solution of HRSI.*

The implicit HRSI is a different formulation of the HRSI problem that could be in practice for the design of optimization algorithms. While we have not pursued this direction in this work, implicit HRSI has been the basis for the design of a recent efficient algorithm proposed for NMF [42].

We provide in Table 2 several pairs of implicit-explicit regularizations given some possible combinations of ℓ_1 and ℓ_2 norms regularizations, and detail below two particular cases of interest to the regularized LRA community.

Ridge penalization applied to all parameter matrices induces low-rankness. It has been documented in the matrix low-rank approximation literature that applying squared ℓ_2 norm penalizations on the Burer-Montero representation of the low-rank approximate matrix yields an implicit low-rank regularization [60].

Consider the matrix LRA problem with ridge penalizations, using notations from Equation (4):

$$\inf_{X_1\in\mathbb{R}^{m_1\times r}, X_2\in\mathbb{R}^{m_2\times r}} \|M - X_1X_2^T\|_F^2 + \mu\left(\|X_1\|_F^2 + \|X_2\|_F^2\right). \tag{16}$$

Our result on the effect of scale invariance on regularization can be applied to show that this problem has solutions essentially equivalent to

$$\inf_{X_1\in\mathbb{R}^{m_1\times r}, X_2\in\mathbb{R}^{m_2\times r}} \|M - X_1X_2^T\|_F^2 + 2\mu\sum_{q=1}^{r} \|X_1[:,q]\|_2\|X_2[:,q]\|_2. \tag{17}$$

Explicit Regularization	Implicit Regularization	Expected Effect
$\sum_{q=1}^{r}\sum_{i=1}^{n} l_2(X_i[:,q])^n$	$\sum_{q=1}^{r}\|L_q\|_F$	Group-sparse L_q (proved for $n=2$ [60])
$\sum_{i=1}^{n} l_1(X_i)$	$\sum_{q=1}^{r}\|L_q\|_1^{\frac{1}{n}}$	Sparse and group-sparse L_q
$l_1(X_1)^2 + l_1(X_2)^2$	$\|X_1 \otimes X_2\|_1$	unclear
$\sum_{q=1}^{r}\sum_{i=1}^{n} l_1(X_i[:,q])^n + \eta_{\mathbb{R}_+^{m_i}}(X_i[:,q])$	$\|L\|_1 + \eta_{\mathbb{R}_+^{m_1\times\ldots\times m_n}}(L)$	Sparse Nonnegative L
$\sum_{q=1}^{r} l_1(X_1[:,q])^2 + l_2(X_2)^2$	$\sum_{q=1}^{r}\sum_{j=1}^{m_1}\|L_q[j,:]\|_2$	Group-sparse rows of L_q

Table 2: Examples of Explicit-implicit regularization effects for some combinations of ℓ_1 and ℓ_2 norms. We denote $L_q = \otimes_{i=1}^{n} X_i[:,q]$ and $L = \sum_{q=1}^{r} L_q$. Derivations are explained in Appendix B.

It may be observed that the product of the ℓ_2 norms of two vectors is the ℓ_2 norm of the tensor product of these vectors. Therefore defining $L_q = X_1[:,q] \otimes X_2[:,q]$ the problem can be rewritten as

$$\inf_{L_q\in\mathbb{R}^{m_1\times m_2},\,\mathrm{rank}(L_q)\leq 1} \|M - \sum_{q=1}^{r} L_q\|_F^2 + 2\mu\sum_{q=1}^{r}\|L_q\|_2 \tag{18}$$

which is a group-LASSO regularized low-rank approximation problem. Group-LASSO regularizations are known to favor sparsity in groups of variables [73], here the rank-one terms L_q. The expected behavior of such a group-lasso regularization is a semi-automatic rank selection driven by the value of μ.

Note that these derivation are not specific to matrix problems. This means that the squared ℓ_2 norm penalties also lead to a group-norm implicit regularization on the rank-one terms of matrix or tensor low-rank approximations. This is illustrated in the case of Nonnegative CPD in Section 4.2 where ridge regularization is shown to induce low nonnegative CP-rank approximations. However because the group-lasso regularization happens on top of a rank-one constraint on each group, we did not manage to extend existing guarantees on the sparsity level of the solution. Moreover, to make the nuclear norm explicit in the matrix LRA problem and obtain the result of Srebro and co-authors [60], we need to rely on the rotation ambiguity of the ℓ_2 norm which falls outside the framework proposed here. In other words, whether there are situations where the (implicit) group-lasso penalty does not induce low-rank solutions is an open problem.

Sparse ℓ_1 penalizations may yield sparse and group-sparse low-rank approximations but individual sparsity levels cannot be tuned. In some applications of regularized LRA, it may be senseful to seek sparsity in all parameter matrices $\{X_i\}_{i\leq n}$. A seemingly natural formulation of regularized LRA to achieve this, for instance in the context of NMF when $n=2$, is the following:

$$\inf_{X_1\in\mathbb{R}_+^{m_1\times r}, X_2\in\mathbb{R}_+^{m_2\times r}} \|M - X_1X_2^T\|_F^2 + \mu_1\|X_1\|_1 + \mu_2\|X_2\|_1. \tag{19}$$

This model has been previously studied with a similar perspective [52], hence the observations we make below are known for sNMF, but carry over to any regularized LRA model.

One may expect that both estimated matrices X_1 and X_2 will be sparse, and that parameters μ_1 and μ_2 respectively control the sparsity level of X_1 and X_2. It turns out that this last assertion should be mitigated. Indeed by marginalizing the scales, and using the fact that the product of the ℓ_1 norms of vectors is the ℓ_1 norm of their tensor product, we know that solutions in (19) are essentially equivalent to minimizers in

$$\operatorname*{argmin}_{L_q\in\mathbb{R}_+^{m_1\times m_2},\,\mathrm{rank}(L_q)\leq 1} \|M - \sum_{q=1}^{r} L_q\|_F^2 + 2\sqrt{\mu_1\mu_2}\sum_{q=1}^{r}\sqrt{\|L_q\|_1}. \tag{20}$$

While it is not straightforward to prove rigorously that solutions to (20) are indeed sparse (because of the rank one constraints), we may observe that only the product of parameters μ_1 and μ_2 matters at the implicit level. Consequently, one may not tune each individual regularization hyperparameter to change the sparsity of X_1 with respect to X_2. For instance, by multiplying μ_1 by 100 and dividing μ_2 also by 100 provably does not change the sparsity level of X_1 and X_2: the solutions before and after such an hyperparameter tuning are the same up to the scale of X_1 and X_2, and this is easily seen on the implicit problem. This observation may go against the intuition that "the more one regularize, the more regularized the solutions are". This intuition does hold, but only for the product of variables, *i.e.* at the level of rank-one parameters L_q, which we confirmed

experimentally in Section 4.1. An explicit control on the sparsity level of each parameter matrix therefore requires the use of other regularizations such as the ℓ_0 count function. We discuss more about the design of the hyperparameters in the next subsection.

A final observation here is that the regularization term in the implicit sNMF formulation is a group-sparsity with a mixed $\ell_{0.5,1}$ norm. This regularization is exotic, but intuitively it should promote sparsity in the parameter matrices while also promoting group-sparsity in the rank-one components, *i.e.* perform automatic rank detection. However the two effects are not controlled individually, and in practice it is unclear which sparsity patterns will be obtained in the solutions, or if one effect will dominate.

2.3 Choosing the hyperparameters

For a given choice of μ_1 and μ_2, a proportional scaling of the hyperparameters does not change the minimum of the cost, although optimal parameter matrices will exhibit a specific column scaling. Again, what matters is only the product of the regularization hyperparameters. This observation easily extends to any number of modes n and any rank r. Thus it appears that hyperparameters $\{\mu_i\}_{i\leq n}$ may be chosen simultaneously. Let us discuss a few reasonable combinations:

- $\mu_i =: \mu$: The user provides a single hyperparameter μ and all penalizations share that value. This solution has the advantage that at optimality, for all $i < j \leq n$ and for all $q \leq r$, $p_i g_i(X_i[:,q]) = p_j g_j(X_j[:,q])$, *i.e.* solutions are balanced without dependence on μ_i. If all g_i are the same fonction (*e.g.* the ℓ_1 norm), then all parameters are penalized and scaled similarly. Moreover, when μ goes to zero, all regularizations vanish. This is practically desirable because the limit case $\mu = 0$ would be consistent with the unregularized case. Conversely, setting any hyperparameter μ_j for a given index $j \leq n$ to a fixed value causes the limit case of $\{\mu_i\}_{i\leq n, i\neq j}$ going towards zero being ill-posed, as shown in Section 2.1.
- $\mu_i = \frac{1}{m_i}\mu$: The regularization hyperparameters are scaled from a shared value by the dimensions of the parameter matrices. This can be better understood on an example, for instance with $n = 2$, setting $g_1(X) = \|X_1\|_1$ and $g_2(X_2) = \|X_2\|_F^2$. When the dimensions m_1 and m_2 are vastly different, without normalization by the matrix sizes, the balancing effect imposes that $\|X_1\|_1 = 2\|X_2\|_F^2$ which implies X_1 and X_2 entries may have very different ranges. Normalization by matrix sizes mitigates this numerical issue.

2.4 Related literature

2.4.1 Implicit regularization due to scale invariance

We are not the first to point out the implicit regularization effect of explicit regularization on parameters of a scale-invariant model. An early reference on this topic from Benichoux, Vincent and Gribonval [5] stressed out the pitfalls of ℓ_1 and ℓ_2 regularization in blind deconvolution. In particular, they showed that the solution to such regularized model are degenerate and non-informative. Their proof however is restricted to convolutive mixtures, and we did not observe this behavior with low-rank approximations. It is conjectured that this pitfall is induced by further invariances in the convolutive mixture and is not due solely to columnwise scale invariance.

Another closely related work studies the effect of scale-invariance for sparse NMF [52]. The authors obtain the same implicit problem (20) and mention the impossibility to tune each hyperparameter individually. They also notice empirically the existence of "scaling swamps" that we analyse in Section 3.3. Their analysis is however restricted to sparse NMF and they do not discuss implicit regularization effects nor generalize to other regularized low-rank approximations.

Again in the context of sparse NMF, a few works have explicited implicit formulations of similar flavor to what is introduced in this work. Marmin *et. al.* have shown the equivalence between a sNMF with normalization constraints and a scale-invariant problem, and then proceed to minimize the scale-invariant loss [42]. Tan and Févotte, by generalizing the work of Mørup [47], have studied automatic rank selection using ℓ_1 or ℓ_2 penalizations where the scales are controlled through the variance of the parameter matrices [66]. Srebro and co-authors have notoriously proposed that ridge penalization in two-factor low-rank matrix factorization is equivalent to nuclear-norm minimization, opening the research avenue on implicit regularization effects [60]. Overall, these works deal with specific models, do not capture the systemic effect of scale invariance on regularized LRA, and do not expand on the misleading intuitions of explicit regularization when confronted with the equivalent implicit formulation.

There are a few other references dealing with implicit regularization for regularized low-rank factorization problems. More closely related to tensor decompositions, Uschmajew hinted that ridge penalizations balances solutions to canonical polyadic decompsitions [72], and Roald and co-authors showed that regularizations hyperparameters can be tuned collectively in regularized low-rank approximations, given homogeneous regularizations [56]. Finally, several works have instead focused on similar implicit regularization effects in the context of deep learning. Feed forward RELU neural networks feature a similar scale-invariance, Tibshirani recalls several related results in a recent note [70], such as how group-sparse penalties on the linear layers is equivalent to a group-sparse LASSO problem [50]. Practical aspects of explicit/implicit regularizations in deep neural networks have also been studied [17, 24, 62], see this recent overview for further references [53].

2.4.2 Non-homogeneous regularizations for LRA

Within the context of regularized scale-invariant models, all parameters must be penalized to avoid degeneracy. Other solutions exist to cope with the ill-posedness which stand outside of the proposed framework, in particular within the context of sparse LRA. We quickly discuss these options below.

To induce sparsity in the parameter matrices of LRA, a first solution due to Hoyer [30] is to use what is sometimes called Hoyer sparsity:

$$g(x) = \frac{1}{\sqrt{m}-1}\frac{\sqrt{m}-\|x\|_1}{\|x\|_F} \tag{21}$$

for any $x \in \mathbb{R}^m$. This sparsity measure has the advantage of being scale invariant (homogeneous of degree zero), thus avoiding the degeneracy problem of the plain ℓ_1 norm. When used as a constraint, Hoyer sparsity also has the advantage to be intuitive to define, since $g(X) = 1$ implies X is one-sparse and $g(X) = 0$ implies X is fully dense and constant. But it suffers from a few issues:

- The proximity operator of the Hoyer sparsity is not known to the best of our knowledge. Nevertheless, there exist an efficient procedure to project on the set of vectors with bounded Hoyer sparsity [29].
- When used as a penalty, the regularization hyperparameter is harder to tune than with the ℓ_1 penalty, since maximal regularization hyperparameter values are not known and the general behavior of solutions with respect to the regularization hyperparameter is not characterized.

Note that both these issues could in principle be mitigated given sufficient research effort towards establishing the properties of the Hoyer sparsity used as a regularization penalty function.

Another solution to enforce sparsity in LRA, and arguably the most common approach, is to impose a norm constraint columnwise on the unregularized parameter matrices using the ℓ_2 or ℓ_1 norm. This is the approach often used for sparse NMF [16, 36] and the approach pursued in the first sparse NTD paper by Mørup and coauthors [48]. Normalization removes the scaling ambiguity in the cost function. In these works, empirically, imposing sparsity with the ℓ_1 norm and imposing normalization on other modes yields sparse matrices.

From the user point of view, this solution is elegant because only the ℓ_1 norm hyperparameters have to be tuned. Moreover it removes scaling ambiguity in NMF and makes the decomposition consistent in terms of the amplitudes of the parameter matrices. However from the mathematician perspective, normalization with the ℓ_2 norm is not ideal. To the best of our knowledge, MM algorithms with normalization do not have convergence guarantees. Indeed the update rules reported in [16, 36] rely on the interpretation of MM as the ratio between positive and negative gradients, which is not a maximization-minimization procedure. Moreover, some algorithms that rely on separability cannot be used with normalization constraints, such as coordinate descent which is state-of-the-art for computing solutions to the Lasso problem [45] and NMF with the Frobenius loss [22]. These updates are further more complicated to derive and implement than classical MM rules. These issues are moderately severe. However when compared to the approach presented further which does not suffer from these issues, the normalization approach seems unnecessarily complicated.

3 Algorithms

We have discussed in the previous section the implicit balancing effect of scale invariance in HRSI, and its implications for the practical design and choice of regularization functions and hyperparameters. Once regularization functions have been chosen, the practitioner is faced with a second challenge: computing solutions efficiently.

We aim to make this manuscript as practical as possible. Deriving optimization strategies for the HRSI model, which is quite general, would not serve this purpose. Therefore, in the following section, we detail the

strategy proposed to derive efficient algorithms for regularized LRA with specific loss functions rather than any HRSI model. First of all, we assume elementwise nonnegativity on each parameter matrix. This restriction allows us to propose a unified framework for our algorithm. More precisely, we detail how to compute a sparse decomposition of an input tensor or matrix with β-divergence as a loss function and mixed-sparsity regularizations on the factors. Beta-divergences are a widely used class of loss functions for nonnegative regularized LRA that covers both the Frobenius loss and the Kullback Leibler loss, arguably the two most widely used loss functions in this context.

In a nutshell, our approach relies on three building blocks:

1. Cyclic Block-coordinate Descent (BCD): in order to compute the factors of a matrix or tensor decomposition for a given input tensor, one usually has to solve a non-convex optimization problem under constraints. Most of these problems are jointly non-convex w.r.t. all the variables, but may be convex w.r.t. specific blocks of variables, or factors of the decomposition. This is why many algorithms rely on the so-called cyclic BCD approach, that is the sequential update of each block of variables of interest. We follow this principle here. Further, we refer to as "subproblem" the minimization problem w.r.t. a selected block of variables.

2. Majorization-mimization: for each subproblem, the update for one block of variables is performed by first building a simple majorizer of the objective function at the current iterate and then minimizing it exactly, in the best case in closed form. In this paper, we consider *fully separable* majorizers. Such majorizers can be expressed as a sum of one-dimensional functions, each of which depends on a single component, and therefore can be minimized w.r.t. each component separately.

3. Optimal balancing: once blocks of variables have been sequentially updated following the two previous steps, we perform a balancing step for each of them to locally minimise the regularization terms, as detailed in Section 2.2.2.

In the next sections, we discuss in more details the three blocks. Then, we present our general algorithm, dubbed as *Meta-Algorithm*, and finally discuss its convergence guarantees.

3.1 BCD: which splitting ?

Here-under we formalize the class of regularized LRA for which we design optimization algorithms:

$$\min_{\forall i \leq n, X_i \in \mathbb{R}_+^{m_i \times r}} f(\{X_i\}_{i \leq n}) + \sum_{i=1}^{n} \mu_i \sum_{q=1}^{r} g_i(X_i[:,q]), \tag{22}$$

where $f(\{X_i\}_{i \leq n})$ is the discrete, entrywise β-divergence, that is, for a matrix decomposition model, $f(X_1, X_2) = D(M|X_1X_2^T) = \sum_{k,l} d_\beta(M[k,l]|\hat{M}[k,l])$, with $\hat{M} = X_1X_2^T$ and the β-divergence between scalars $a, b \geq 0$, denoted $d_\beta(a|b)$, and equal to

$$\begin{cases} \frac{1}{\beta(\beta-1)} \left(a^\beta + (\beta-1)\, b^\beta - \beta a b^{\beta-1}\right) \text{ for } \beta \neq 0, 1, \\ a \log \frac{a}{b} - a + b \text{ for } \beta = 1, \\ \frac{a}{b} - \log \frac{a}{b} - 1 \text{ for } \beta = 0, \end{cases}$$

For $\beta = 2$, this is the standard squared Euclidean distance, that is, the squared Frobenius norm $||M - X_1X_2^T||_F^2$. For $\beta = 1$ and $\beta = 0$, the β-divergence corresponds to the Kullback-Leibler (KL) divergence and the Itakura-Saito (IS) divergence, respectively. With convention that $a \log 0 = -\infty$ for $a > 0$ and $0 \log 0 = 0$, the KL-divergence is well-defined.

We first split the variables into disjoint blocks [4] and minimize the objective function over each block sequentially. Since the updates for each block will be similar, and given the separable structure of the objective function in (22), we consider the update of the block of variables $X_i[:,q]$ (one column of X_i, dropping the i index), denoted $x \in \mathbb{R}_+^m$ to ease the notations, while the other blocks of variables are kept fixed. Therefore, we are interested in solving the following subproblem:

$$\min_{x \in \mathbb{R}_+^m} f(x) + \mu g(x) \ . \tag{23}$$

[4] their union is the initial and full set of variables

3.2 Majorization-Minimization (MM)

In order to tackle Problem (23), we follow the MM framework. Let us briefly recall the key aspects to derive updates via MM. Let us consider the general problem

$$\min_{x\in\mathbb{R}^m_+} \phi(x).$$

Given an initial iterate $\widetilde{x} \in \mathbb{R}^m_+$, MM generates a new iterate $\hat{x} \in \mathbb{R}^m_+$ that is guaranteed to decrease the objective function, that is, $\phi(\hat{x}) \leq \phi(\widetilde{x})$. In order to achieve this, MM employs the following two steps:

- Majorization: search for a function that is an upper-bound of the objective and is tight at the current iterate. This upper-bound is usually referred to as a majorizer. More formally, we are looking for a function $\bar{\phi}(x|\widetilde{x})$ such that

$$(i)\ \bar{\phi}(\widetilde{x}|\widetilde{x}) = \phi(\widetilde{x}) \quad \text{and} \quad (ii)\ \bar{\phi}(x|\widetilde{x}) \geq \phi(x) \text{ for all } x \in \mathbb{R}^m_+.$$

- Minimization: minimize the majorizer by exactly solving $\min_{x\in\mathcal{X}} \bar{\phi}(x|\widetilde{x})$ to derive the subsequent iterate $\hat{x} \in \mathcal{X}$ which is such that $(iii)\ \bar{\phi}(\hat{x}|\widetilde{x}) \leq \bar{\phi}(\widetilde{x}|\widetilde{x})$. It can be demonstrated that this approach guarantees a decrease of the objective function at each step of the iterative process, indeed we have

$$\phi(\hat{x}) \underset{(ii)}{\leq} \bar{\phi}(\hat{x}|\widetilde{x}) \underset{(iii)}{\leq} \bar{\phi}(\widetilde{x}|\widetilde{x}) \underset{(i)}{=} \phi(\widetilde{x}).$$

The updates are computed using MM, where the majorizer $\bar{\phi}$ is chosen to be separable. This means that $\bar{\phi}(x|\widetilde{x}) = \sum_{k=1} \bar{\phi}_k(x[k]|\widetilde{x}[k])$ for well-chosen univariate functions $\bar{\phi}[k]$'s. This particular choice usually allows for a closed-form solution for minimizing $\bar{\phi}$.

We now detail how to derive a separable majorizer $\bar{\phi}(x)$ for $\phi(x) := f(x)+\mu g(x)$, that is a function satisfying conditions (i) and (ii) and which can be expressed as sum of univariate functions $\bar{\phi}[k]$'s. The first term, that is $f(x)$, is majorized following the methodology introduced in [20] which consists of applying a convex-concave procedure to the d_β. In a nutshell, the convex part is majorized using Jensen's inequality and the concave part is upper-bounded by its first-order Taylor approximation. The resulting separable majorizer for $f(x)$ built at the current iterate $\widetilde{x}$ is denoted $\bar{f}(x|\widetilde{x})$.

For the second term, in the frame of this paper, we assume the following:

Assumption 1. *[39] For the smooth and definite positive regularization function $g : x \in \mathbb{R}^m_+ \to \mathbb{R}_+$, and any $\tilde{x} \in \mathbb{R}^m_+$, there exists nonnegative constants C_k with $1 \leq k \leq m$ such that the inequality:*

$$g(x) \leq \bar{g}(x|\tilde{x}) := g(\tilde{x}) + \langle \nabla g(\tilde{x}), x - \tilde{x}\rangle + \sum_{k=1}^{m} \frac{C_k}{2}(x[k] - \tilde{x}[k])^2 \tag{24}$$

is satisfied for all $x \in \mathbb{R}^m_+$.

In other words, the regularization function can be upper-bound by a *tight separable* majorizer given by Equation (24). Combining $\bar{f}(x|\widetilde{x})$ and $\bar{g}(x|\tilde{x})$, it is then easy to build a majorizer $\bar{\phi}(x|\widetilde{x})$ for $\phi(x)$ at any $\tilde{x} \in \mathcal{X}$ as follows:

$$\begin{aligned} \phi(x) &:= f(x) + \mu g(x) \\ &\leq \bar{f}(x|\widetilde{x}) + \mu g(x) \\ &\leq \bar{f}(x|\widetilde{x}) + \mu\bar{g}(x|\tilde{x}) := \bar{\phi}(x|\widetilde{x}) \quad \forall x \in \mathbb{R}^m_+, \end{aligned} \tag{25}$$

which is also separable w.r.t. each entry of x, since it is a sum of separable majorizers. Moreover, the majorizer $\bar{\phi}(x|\widetilde{x})$ given in (25) is tight at $\widetilde{x}$, that is $\bar{\phi}(\widetilde{x}|\widetilde{x}) = \phi(\widetilde{x})$. This follows directly from the definition of $\bar{f}(x|\widetilde{x})$ in [20] and $\bar{g}$ in Equation (24).

As explained earlier, given the current iterate $\tilde{x} \in \mathbb{R}^m_+$, the update for x is finally obtained as follows:

$$\hat{x} := \underset{x\in\mathbb{R}^m_+}{\operatorname{argmin}}\ \bar{\phi}(x|\widetilde{x}) \tag{26}$$

One may ask: does the minimizer $\hat{x}$ from Equation (26) exist and is unique ? It turns out that the recently introduced framework [39] provides answers to this question under mild conditions (in the case no additional equality constraints are required). In particular, two situations are covered which are summarized here-under:

1. The defined coefficients $\mu\frac{C_k}{2}$ are strictly positive for all k or $\beta \in (1,2)$,
2. some coefficients $\mu\frac{C_k}{2} = 0$ and $\beta \leq 1$

Under those conditions, [39, Proposition 1] ensures there *exists* a *unique* minimizer $\hat{x} \in (0,\infty)$. Practical illustrations of this theoretical discussion are detailed in Section 4. In the latter, among others, we consider specific tensor decomposition models, particular cases for problems (22) and we derive the exact formulae for the minimizers $\hat{x}$ given by Equation (26). Note that the closed form expression can be derived if $\beta \in \{0, 1, 3/2, 2\}$ according to [39]. Outside this set of values for β, an iterative scheme such as Newton-Raphson is required to compute the minimizer. Indeed, we will see further that the task of computing the minimizer is equivalent to finding the positive real roots of a set of univariate polynomial equations in each entry $x[k]$ of x.

3.3 Why balancing is important to improve convergence speed

A key novelty of the proposed meta-algorithm is a numerical balancing of the columns of low-rank components, following the results of Section 2. While we have shown in that section that we can easily balance columns, it can still remain unclear why this procedure could be useful in practice. To demonstrate the usefulness of the balancing procedure, we proceed in two steps. First, we show in this subsection that convergence of Alternating Least Squares (ALS) to rescale a pair of variables is sublinear when $\frac{\lambda}{y}$ vanishes. ALS is already known to feature sublinear convergence speed for several problems, see [18] and references therein, but our analysis is fairly simple and dedicated to scaling variables in regularized LRA. In short, we provide an example where we can prove the existence of "scale swamps" [52] and understand their nature. Second, we will study the effectiveness of balancing procedure in an ablation study in experiments shown later in Section 4.

For a given real value y and a regularization parameter $0 < \lambda \ll y$, consider the following scalar optimization problem

$$\min_{x_1,x_2} f(x_1,x_2) \text{ where } f(x_1,x_2) = (y - x_1x_2)^2 + \lambda(x_1^2 + x_2^2)\ . \tag{27}$$

We can use the results from Section 2 to immediately obtain that minimizers x_1^*, x_2^* must be balanced, which leads to optimal solutions $x_1^* = x_2^* = \sqrt{y-\lambda}$ (when $\lambda \leq y$). Note that λ must be non-zero for the balancing to occur, but large λ values lead to bias in the solution.

Setting aside the fact that we know the optimal solution, a classical iterative method from the regularized LRA literature to solve this problem would be the ALS algorithm. Indeed the cost is quadratic with respect to each variable but globally non-convex. Setting k as the iteration counter, the updates for the two variables are easily derived as

$$\begin{cases} x_1^{(k+1)} = \frac{x_2^{(k)} y}{x_2^{2(k)} + \lambda} \\ x_2^{(k+1)} = \frac{x_1^{(k+1)} y}{x_1^{2(k+1)} + \lambda} \end{cases} \tag{28}$$

We show in Appendix D that when λ is small with respect to y and for k large enough,

$$\frac{x_1^{k+1} - \sqrt{y-\lambda}}{x_1^k - \sqrt{y-\lambda}} \approx 1 - 4\frac{\lambda}{y}, \tag{29}$$

and similarly for x_2. Therefore ALS converges almost sub-linearly when $x_{1,2}$ are near the optimum and $\frac{\lambda}{y} \ll 1$. In fact the smaller the ratio $\frac{\lambda}{y}$, the slower the algorithm converges. There is therefore a trade-off between biais and speed. We illustrate this behavior in Figure 1.

Consequently in this problem, scaling the variables x_1, x_2 optimally is much more efficient than waiting for the scales of each variable to eventually grow closer, especially for small λ. Similar behaviors are observed in the more complex case of regularized LRA, see Section 4, which is why we propose in the Meta-Algorithm below to scale the iterates optimally after each outer iteration.

3.4 Meta-Algorithm

The proposed method for solving problem (22) is summarized in Algorithm 2, as detailed in Section 2.2.2. After performing updates of $\{X_i\}_{i\leq n}$ using MM, the second step of our method begins, which involves optimally balancing the block of variables. This balancing step is computationally inexpensive and necessarily decreases the objective function, as demonstrated in Proposition 4 in Section 3.5. Thus, it does not interfere with the convergence guaranty offered by the MM-based updates, i.e., the convergence of the sequence of objective function values. However, the convergence guarantees for the sequence of iterates will depend on the choice of the value β in Problem (30). These points will be discussed in more detail in subsection 3.5.

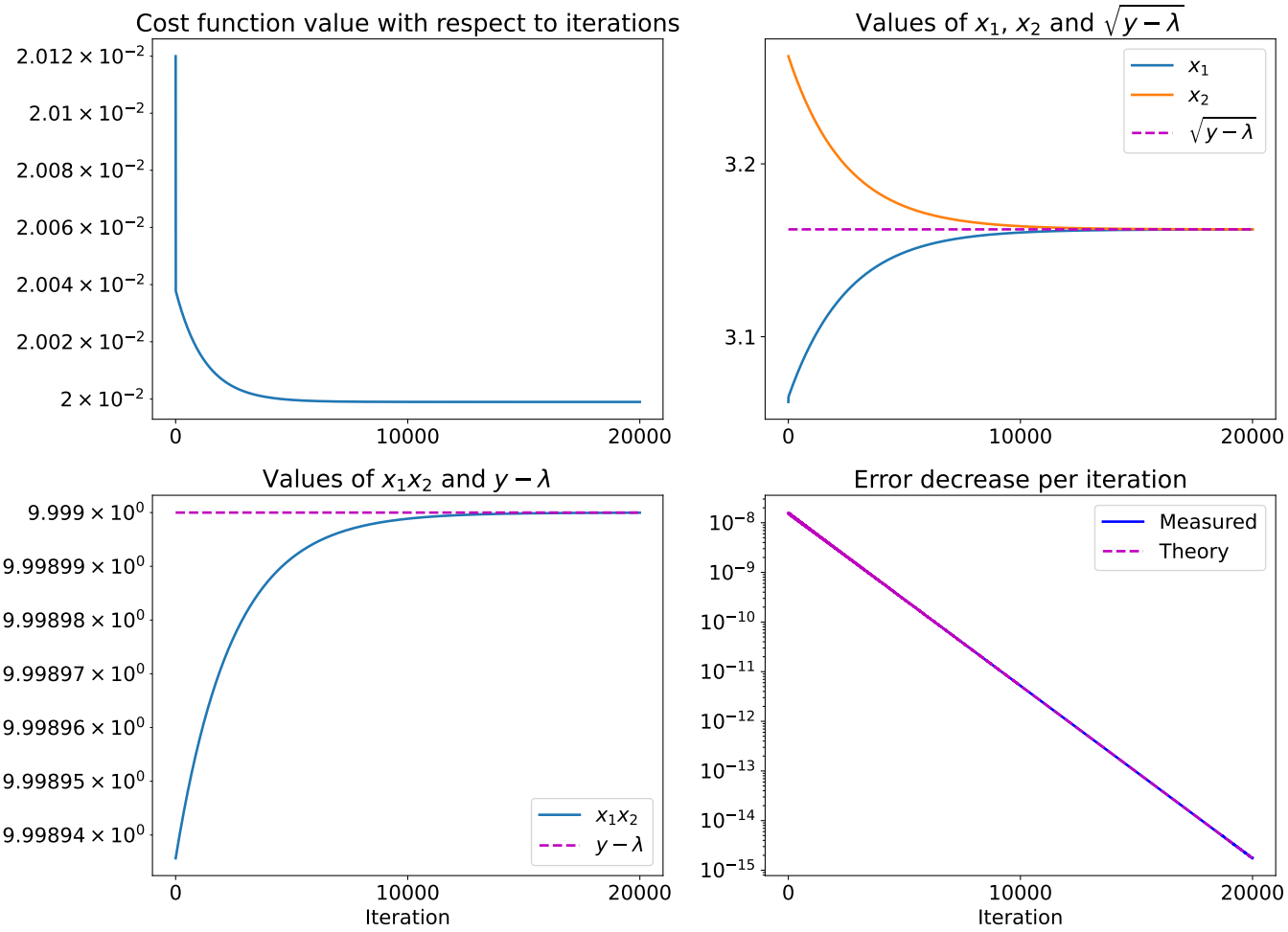

Figure 1: Study of the convergence of ALS for the toy problem (27). We choose $y = 10$ and $\lambda = 10^{-3}$. The number of iterations of ALS is set to 20000. Top left: cost function f with respect to iteration index. Top right: values of x_1 (orange), x_2 (blue) and $\sqrt{y-\lambda}$ (magenta). Bottom left: value of x_1x_2 (blue) and $y - \lambda$ (magenta). Bottom right: empirical value of the cost function variation (blue) and the theoretical value $16\frac{\lambda^2}{y}e_k^2$ (magenta). Several observations can be made: The value of x_1x_2 rapidly converges towards $y - \lambda$ (bottom left graph). However the cost converged after at least 5000 iterations, and the individual values of x_1 and x_2 are slow to converge to the optimum $\sqrt{y-\lambda}$ with respect to how fast x_1x_2 converged to $y - \lambda$. We can observe a linear convergence rate with a multiplicative constant very close to 1 in the bottom right graph, and a close match between the theoretical approximate cost decrease and the practical observation of that decrease.

Algorithm 2 Meta-Algorithm

Require: An input tensor $\mathcal{T} \in \mathbb{R}^{m_1 \times \cdots \times m_n}$, an initialization $X_i^{(0)} \in \mathbb{R}_+^{m_i \times r}$ for $i = 1, ..., n$, the factorization rank r, a maximum number of iterations, maxiter, and regularization hyperparameters μ_i.

1: *% Initialization*:

$$\eta \in \underset{\eta \geq 0}{\text{argmin}} \; f(\{\eta X_i^{(0)}\}_{i \leq n}) + \sum_{i=1}^{n} \mu_i \sum_{q=1}^{r} \eta^{p_i} g_i(X_i^{(0)}[:, q])$$

2: $X_i^{(1)} \leftarrow \eta X^{(0)}$ for all $i = 1, ..., n$.
3: **for** $i = 1 :$ q **do**
4: $\quad \{X_i^{(1)}[:, q]\}_{i \leq n}$ balanced using Algorithm 1.
5: **end for**
6: *% Main optimization loop*:
7: **for** $k = 1 :$ maxiter **do**
8: $\quad$ *%% Update of factors* $\{X_i\}_{i \leq n}$
9: $\quad$ **for** $i = 1 :$ n **do**
10: $\quad\quad$ **for** $q = 1 :$ r **do**
11: $\quad\quad\quad X_i[:, q]^{(k+1)} \leftarrow \text{argmin}_{x \in \mathcal{X}_i} \bar{\phi}(x | X_i[:, q]^{(k)})$
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: $\quad$ *%% Optimal balancing*
15: $\quad$ **for** $q = 1 :$ r **do**
16: $\quad\quad \{X_i^{(k+1)}[:, q]\}_{i \leq n}$ balanced using Algorithm 1.
17: $\quad$ **end for**
18: **end for**
19: **return** $\{X_i[:, q]\}_{i \leq n}$

Initialization In Algorithm 2, we propose a first stage of initialization for which blocks of variables are not only balanced (lines 3 to 5), but also scaled (lines 1 to 2). This scaling is crucial to avoid the "zero-locking" phenomena. To illustrate this, let us consider a matrix case $n = 2$. If we choose functions $g_1(.)$ and $g_2(.)$ as ℓ_1 norms, it is easy to show that the updates of the columns of *e.g.* factor X_1 boil down to solve a set of LASSO problems with the mixing matrix corresponding to X_2^T. For such problems, it is known that if $\|MX_2\|_\infty \leq \mu_1$, the solution is necessarily $X_1 = 0$. In this case, $(0, 0)$ being a saddle point it is very likely that Algorithm 2 will return $X_1 = X_2 = 0$. The scaling step mitigates this issue by avoiding using initial X_1 and X_2 which are unnecessarily big.

3.5 Convergence guarantees

In this section we discuss the convergence guarantees offered by Algorithm 2 w.r.t the sequence of objective functions and the sequence of iterates. Denoting $z = (\{X_i[:, q]\}_{i \leq n, q \leq r})$ (concatenation of the variables) and ϕ the objective function from Problem (22), we consider the following class of multiblock optimization problems:

$$\min_{z_i \in \mathcal{Z}_i} \phi(z_1, \ldots, z_s) \tag{30}$$

where z can be decomposed into s blocks $z = (z_1, \ldots, z_s)$, $\mathcal{Z}_i \subseteq \mathbb{R}^{m_i}$ is a closed convex set, $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \mathcal{Z}_s \subseteq \text{dom}(\phi)$, $\phi : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$, $m = \sum_{i=1}^{s} m_i$, is a continuous differentiable function over its domain, and ϕ is lower bounded and has bounded level sets.

In Theorem 1 we show that Algorithm 2 generates a sequence of iterates $\{z^i\}_{i=1}^{\infty}$ such that the sequence of objective values $\{\phi(z^i)\}$ is monotone non-increasing. Putting in the notations of (30), first, let us recall the formal definition of a tight majorizer for convenience.

Definition 1. *A function $\bar{\phi}_i : \mathcal{Z}_i \times \mathcal{Z} \to \mathbb{R}$ is said to be a tight majorizer of $\phi(z)$ if*

1. $\bar{\phi}_i(z_i, z) = \phi(z), \forall z \in \mathcal{Z}$,
2. $\bar{\phi}_i(y_i, z) \geq \phi(z_1, ..., z_{i-1}, y_i, z_{i+1}, ..., z_s), \forall y_i \in \mathcal{Z}_i, \forall z \in \mathcal{Z}$,

where z_i denotes the i-th block of variables of z with $1 \leq i \leq s$.

Using the notations from Definition 1, at iteration $k+1$, Algorithm 2 fixes the last values of block of variables $j \neq i$ and updates block z_i as follows:

$$z_i^{(k+1)} = \underset{z_i \in \mathcal{Z}_i}{\operatorname{argmin}} \ \bar{\phi}_i(z_i, z_1^{(k+1)}, ..., z_{i-1}^{(k+1)}, z_i^{(k),*}, ..., z_s^{(k),*})$$

where $z_s^{(k),*}$ denotes the s-th block of variable computed at iteration k and optimally scaled using Equations (11). After updating all the blocks as described above, Algorithm 2 performs optimal scaling on each block. We first present the Proposition 4 which will be useful for the proof of Theorem 1.

Proposition 4. *Let $\{X_i\}_{i\leq n}$ be the current decomposition, the balanced columns $X_i[:,q]^*$ computed according to Algorithm 1 necessarily decrease the objective function value of Problem* (22)*.*

Proof. The proof directly follows the results presented in Section 2.2.2, indeed we have:

$$\begin{aligned}\phi(\{X_i\}_{i\leq n}) &\underset{\substack{\forall i\leq n,\ \Lambda_i = I_{r\times r}}}{=} f(\{X_i\Lambda_i\}_{i\leq n}) + \sum_{i=1}^{n}\mu_i \sum_{q=1}^{r} g_i(\Lambda_i[q,q]X_i[:,q]) \\ &\geq f(\{X_i\Lambda_i^*\}_{i\leq n}) + \sum_{i=1}^{n}\mu_i \sum_{q=1}^{r} g_i(\Lambda_i^*[q,q]X_i[:,q]) = \phi(\{X_i^*\}_{i\leq n}).\end{aligned} \tag{31}$$

where $\Lambda_i^*[q,q] = \left(\frac{\beta_q}{p_i\mu_i g_i(X_i[:,q])}\right)^{1/p_i}$ from Equations (11) with β_q as defined in Proposition 2. The second inequality holds by definition of $\Lambda_i^*[q,q]$, that is the optimal and unique solution of the scaling Problem (12). □

Theorem 1. *Let $z^{(k)} = (\{X_i^{(k)}[:,q]\}_{i\leq n, q\leq r}) \geq 0$, and let $\bar{\phi}_i$ be tight majorizers, as per Definition 1, for $\phi : \mathcal{Z} \to \mathbb{R}_+$ from Problem* (30)*, with $\mathcal{Z} := \mathcal{Z}_1 \times \dots \mathcal{Z}_s \subseteq dom(\phi)$ and $1 \leq i \leq s = nr$. Then ϕ is monotone non-increasing under the updates of Algorithm 2. Moreover, the sequences $\{\phi(z^{(k),*})\}$ and $\{\phi(z^{(k)})\}$ converge.*

Proof. From Definition 1, we have:

$$\begin{aligned}\phi(z^{(k),*}) &= \bar{\phi}_1(z_1^{(k),*}, z_1^{(k),*}, ..., z_s^{(k),*}) \geq \bar{\phi}_1(z_1^{(k+1)}, z_1^{(k),*}, ..., z_s^{(k),*}) \\ &\geq \phi(z_1^{(k+1)}, z_2^{(k),*}, ..., z_s^{(k),*}) = \bar{\phi}_2(z_2^{(k),*}, z_1^{(k+1)}, z_2^{(k),*}, ..., z_s^{(k),*}) \\ &\geq \bar{\phi}_2(z_2^{(k+1)}, z_1^{(k+1)}, z_2^{(k),*}, ..., z_s^{(k),*}) \geq \phi(z_1^{(k+1)}, z_2^{(k+1)}, ..., z_s^{(k),*}) \geq ... \\ &\geq \phi(z_1^{(k+1)}, z_2^{(k+1)}, ..., z_s^{(k+1)}) \underset{Prop.\ 4}{\geq} \phi(z_1^{(k+1),*}, z_2^{(k+1),*}, ..., z_s^{(k+1),*})\end{aligned}$$

where $z_i^{(k+1),*}$ are computed using Equations (11) for $1 \leq i \leq s = nr$.

By assumption, ϕ from Problem (30) is a finite nonnegative sum of functions that are all bounded below onto the feasible set $\mathcal{Z}$, hence ϕ is also bounded below onto $\mathcal{Z}$. Therefore, both the non-increasing monotone sequences $\{\phi(z^{(k),*})\}$ and $\{\phi(z^{(k)})\}$ converge. □

Convergence of the sequence of iterates In the next Theorem, we establish the convergence of the iterates generated by Algorithm 2 towards coordinate-wise minimum, or critical points of Problem (30) under mild assumptions.

Theorem 2. *Let majorizers $\bar{\phi}_i(y_i, z)$, as defined in 1, be quasi-convex in y_i for $i = 1, ..., s$. Additionally, assume that Assumptions 2 are satisfied. Consider the subproblems in Equation* (118)*, and assume that both these subproblems and the optimal scaling, obtained using Equations* (11)*, have unique solutions for any point $(z_1^{(k+1)}, ..., z_{i-1}^{(k+1)}, z_i^{(k),*}, ..., z_s^{(k),*}) \in \mathcal{Z}$. Under these conditions, every limit point z^∞ of the iterates $\{z^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 2 is a coordinate-wise minimum of Problem* (30)*. Furthermore, if $\phi(\cdot)$ is regular at z^∞, then z^∞ is a stationary point of Problem* (30)*.*

Proof. See Appendix E. □

We refer the reader to Appendix E for the detailed discussion related to the convergence of iterates generated by Algorithm 2 and the proof of Theorem 2, which is an adaptation of the convergence to stationary points obtained in [54, Theroem 2] for block successive upperbound minimization (BSUM) framework.

3.6 Handling near-zero entries with the β-divergence

Among the assumptions to be satisfied to ensure the convergence of the iterates generated by Algorithm (2) to critical points of Problem (30), we have the directional differentiability of the function $\phi(.)$, see Appendix E. The latter is naturally satisfied by assuming that $\phi(.)$ is continuously differentiable onto the feasible set $\mathcal{Z}$, that we assume here to be the product of nonnegative orthants of appropriate dimension. However, in the case of the entrywise β-divergence for $f(.)$ in Problem (30), it is well known that such assumptions do not hold as soon as $\beta < 2$, see for instance the discussions in [27]. Indeed, we have

$$\frac{\partial d_\beta(a,b)}{\partial b} = b^{\beta-1} - ab^{\beta-2},$$

and it is easy to see that, for $\beta < 2$, the derivative of $d_\beta(a,b)$ w.r.t. b is not defined at zero when $a > 0$. For a matrix decomposition model, that would happen in the case $\hat{M}[i,j] = 0$ with $\hat{M} = X_1 X_2^T$ and $M[i,j] > 0$ for some i, j. To guarantee some convergence for $\{z^{(k)}\}_{k\in\mathbb{N}}$, the usual remedy is to impose the constraints $z_i \geq \epsilon$ for all $i \in [1, s]$, where $\epsilon > 0$, to Problem (30)[5]. In practice, we use the machine epsilon for ϵ, which does not influence the objective function much [27]. Using the same separable majorizers, the updates of factors can be easily generalized to the ϵ-perturbed Problem (30), and are given by

$$z_i^{(k+1)} \leftarrow \max\Big(\epsilon,\ \underset{z_i}{\operatorname{argmin}}\ \bar{\phi}_i(z_i, z_1^{(k+1)}, ..., z_{i-1}^{(k+1)}, z_i^{(k),*}, ..., z_s^{(k),*})\Big)$$

where we calculate the elementwise maximum between the updates of factors (which correspond to the closed-form expression of the minimizer of the Block Majoritzation-Minimization step at the current iteration) and ϵ, see for instance [65] where such approaches have been considered for deriving the global convergence of the multiplicative updates proposed by Lee and Seung to solve the ϵ-perturbed β-NMF problems.

The use of such approaches in Algorithm 2, that is replacing the updates of Step 11 by

$$X_i[:,q]^{(k+1)} \leftarrow \max\Big(\epsilon,\ \underset{x}{\operatorname{argmin}}\ \bar{\phi}(x|X_i[:,q]^{(k)})\Big), \tag{32}$$

is not straightforward to ensure the convergence guarantees stated in Theorem 2. The reason lies in the optimal rebalancing step of Algorithm 2. Indeed, assuming that one entry of $X_i[:,q]^{(k+1)}$ reached ϵ, then the balancing procedure may decrease this entry below this lower bound, making it non-feasible, and hence loosing the convergence of the iterates $\{z^{(k)}\}_{k\in\mathbb{N}}$ to a critical point of the ϵ-perturbed Problem (30). Here-under we discuss several strategies to deal with this issue:

- A simple and naive approach would be to compute the elementwise maximum between the balanced $X_i[:,q]^{(k+1),*}$ obtained at Step 16 of Algorithm 2 and ϵ for all $i \leq n$ and $q \leq r$. However, this approach is likely to break the monotonic behavior of the sequence of objective function values given by Theorem (1). This is because the scale-invariance of $f(.)$ may not hold, meaning that $f\Big(\{\max(\epsilon, X_i^{(k+1),*})\}_{i\leq n}\Big)$ is not necessarily equal to $f(\{X_i^{k+1}\}_{i\leq n})$. Nevertheless we anticipate that these theoretical considerations will not affect noticeably the algorithm behavior. Therefore in practice we implement a similar idea. First, if a column of a factor is elementwise smaller than ϵ, all the corresponding columns in other factors are set to ϵ as well (therefore the component is virtually set to zero). When this is not the case and at least one entry in the column is strictly larger than ϵ in each column, we first set all values smaller than ϵ to zero, optimally scale the resulting columns, and then project the outcome such that all entries are larger than ϵ.

- Another approach consists in first consider for Step 11 the updates given by Equation (32), and second monitor the evolution of $\min_{q,i} X_i^{(k+1)}[:,q]$ throughout iterations. When this value hits the predefined threshold of ϵ, the balancing steps are halted. Subsequently, for the remaining iterations of Algorithm 2, only the MM updates given by Equation (32) are taken into account. This strategy ensures the convergence of the iterates to critical points of the ϵ-perturbed Problem (30) through BSUM, with the balancing steps primarily serving to accelerate the convergence of the algorithms during the initial iterations. This approach appears to be the most suitable from a theoretical perspective, but in practice the balancing may halt at most iterations which is undesirable.

[5]This new Problem will be referred to as the ϵ-perturbed Problem (30).

We conclude this section by considering the case $\beta \geq 2$. In these scenarios, the β-divergence exhibits continuous differentiability over the feasible set, thereby ensuring that the convergence guarantees of Algorithm 2 as outlined in Theorem 2 remain applicable. However, it can still be of interest in practice to threshold the entries in parameter matrices X_i to ϵ when $\beta \geq 2$. Indeed, whenever the updates computed by MM are multiplicative, when an entry in a parameter matrix reaches zero it cannot be further modified by the algorithm. This can prevent convergence in practice when using a bad initialization or when machine precision is insufficient. In the proposed implementations of the meta-algorithm, whenever nonnegativity is imposed, in practice a small ϵ lower-bounds the entries of all factors.

3.7 Implementation of the update rules for beta divergence and ℓ_1 and ℓ_2 regularizations

The proposed Meta Algorithm (Algorithm 2) does not explicitly implement the MM update rules. However, practical implementations of the algorithm, especially in the showcases detailed in Section 4, will require these updates. The MM updates for values of β in the range of $[0, 2]$ and for regularization functions that are ℓ_p^p norms, both of which are important particular cases, are derived in Appendices F and G. In particular, the update rules for $\beta = 1$ and $\beta = 0$ with ℓ_2^2 regularizations, which have not been addressed in the existing literature, involve finding roots of low-degree polynomials and are of significant interest.

Note that MM updates for $\beta = 2$ can be derived within the proposed framework, but existing results from the literature point towards the superiority of an alternating least squares approach called HALS, which we do not recall in detail in this article since HALS is already well described in the existing literature [23, 43].

4 Showcases

In this section we propose to study three particular instances of HRSI: sparse Nonnegative Matrix Factorization (sNMF), ridge-regularized Nonnegative Canonical Polyadic Decomposition (rNCPD) and sparse Nonnegative Tucker Decomposition (sNTD). For each model, we first relate it to HRSI, explicit the balancing update and the implicit equivalent formulation, and detail an implementation of the proposed alternating MM algorithm. We then study the performance of the proposed method with and without balancing. We will purposely repeat some content introduced in earlier sections to make the showcases as self-contained as possible, while referring to the technical details when available to ease the presentation.

Along the way, in the sNTD showcase we also propose a specialized balancing algorithm. Indeed balancing the terms in the sNTD is not straightforward since the separability hypothesis used in the theoretical derivation does not hold. We propose instead a heuristic strategy reminiscent of the tensor Sinkhorn algorithm which works well in our tests.

Note that we provide an implementation of the proposed algorithm for each showcase problem, alongside reproducible experiments, in a dedidated repository[6]. Our implementation relies on tensorly [33] for multilinear and linear algebra, and package shootout[7] for running the experiments.

4.1 Sparse Nonnegative Matrix Factorization

4.1.1 Background

In the three last decades, Nonnegative Matrix Factorization [37, 51] has proved to be a powerful method for finding underlying structure within a dataset represented as a matrix. Formally, given an m-by-n nonnegative matrix $M \geq 0$ and a factorization rank r, NMF looks for two nonnegative matrices X_1 and X_2 of dimension m-by-r and n-by-r respectively such that $M \approx X_1 X_2^T$. To assess the quality of an approximation, one has to define an error measure, typically denoted $D(M, X_1 X_2^T)$ in the literature, that sum over indices i, j all the entry-by-entry errors $d(M_{i,j}, [X_1 X_2^T]_{i,j})$. NMF is usually formulated as the following optimization problem:

$$\min_{X_1 \in \mathbb{R}^{m \times r}, X_2 \in \mathbb{R}^{n \times r}} D(M, X_1 X_2^T) = \sum_{i,j} d(M_{i,j}, [X_1 X_2^T]_{i,j}), \quad \text{such that } X_1, X_2 \geq 0. \tag{33}$$

where $d(x, y) \geq 0$ for all $x, y \geq 0$ and $d(x, y) = 0$ if and only if $x = y$. The choice for $d(x, y)$ is crucial as it leads to different properties such as the differentiability and L-smoothnes (the gradient is Lipschitz-continous) of the error measure $D(M, X_1 X_2^T)$ on its active domain so that different optimization schemes are needed to tackle Problem (33), see [22, 39] and Sections 1.9.1 and 1.10 of [38] for extensive discussions on that matter. In

[6] `https://github.com/vleplat/NTD-Algorithms`
[7] `https://github.com/cohenjer/shootout`

the following, we choose for the loss function the Kullback-Leibler divergence which is a standard option for NMF [19] and corresponds to beta-divergence with $\beta = 1$.

The non-negativity constraints in Equation (33) make the representation in NMF purely additive, allowing for no subtractions. This is in contrast to other linear representations such as PCA and ICA, which can have both positive and negative coefficients. One of the most useful properties of NMF is its ability to produce a sparse representation of the data. This means that it encodes much of the information using only a few "active" components, making the factors easy to interpret. By capturing the essential features of the data with a small number of components, NMF can effectively reduce dimensionality and provide a compact representation. However, there are several well-known drawbacks associated with NMF, in particular identifiability issues: NMF decompositions are generally not unique, which means that multiple factorizations can produce similar results. This poses challenges in practical applications where uniqueness and the recovery of the true underlying factors are desired.

A possible solution to the lack of identifiability of NMF is to enforce the factors in NMF to be sparse. Under some appropriate assumptions, sparsity leads to a unique solution [11, 67]. In the example mentioned above for rich audio signals, one can introduce sparsity penalties on the activations factor X_2. This ensures that, even though there are many elements in the dictionary, only a small number will be active at the same time. This enables the model to learn a meaningful representation of the data with many more elements than what un-regularized NMF such as (33) would allow. Below we will study a sparse NMF model where both factors are sparse as it better showcases the proposed HRSI framework.

4.1.2 Relation to HRSI

In practice, the most popular technique to obtain sparser NMF solutions is to add sparsity inducing penalty terms, such as a ℓ_1-norm penalty [31], that is for instance solving the following penalized version of Problem (33):

$$\min_{X_1 \in \mathbb{R}^{m\times r}, X_2 \in \mathbb{R}^{n\times r}} F(X_1, X_2) := D(M, X_1 X_2^T) + \mu_1 \|X_1\|_1 + \mu_2 \|X_2\|_1, \quad \text{such that } X_1, X_2 \geq 0. \tag{34}$$

The ℓ_1-norm penalty term in the objective function with weight μ is added for promoting solutions where few basis vectors from X are combined at a time.

Both the regularization terms are separable columnwise as discussed in Section 2. The beta-divergence loss (see Section 3.1) is scale-invariant by columnwise scaling of the parameters. The nonnegativity constraints can be seen as parts of the regularizations. This formulation of sNMF therefore fits the proposed HRSI framework.

4.1.3 Optimal balancing and implicit formulation for sNMF

We can apply the results of Section 2 straightforwardly to obtain the following balancing updates for each column indexed by $q \leq r$:

$$X_1[:, q] \leftarrow \frac{\beta_q}{\mu_1 \|X_1[:, q]\|_1} X_1[:, q] \tag{35}$$

$$X_2[:, q] \leftarrow \frac{\beta_q}{\mu_2 \|X_2[:, q]\|_1} X_2[:, q] \tag{36}$$

where $\beta_q = \sqrt{\mu_1 \mu_2 \|X_1[:, q]\|_1 \|X_2[:, q]\|_1}$. The essentially equivalent implicit HRSI problem can be therefore cast as

$$\min_{L_q \in \mathbb{R}_+^{m_1 \times m_2},\, \mathrm{rank}(L_q) \leq 1} \mathrm{KL}(M, \sum_{q=1}^{r} L_q) + 2\sqrt{\mu_1 \mu_2} \sum_{q=1}^{r} \sqrt{\|L_q\|_1}. \tag{37}$$

which, as discussed in Section 2.2.2, shows that the individual sparsity levels cannot be tuned, and that the sparsity regularizations should also induce sparsity at the level of nonnegative rank-one components

$$L_q := X_1[:, q] X_2[:, q]^T, \text{ since } \|L_q\|_1 = \|X_1[:, q]\|_1 \|X_2[:, q]\|_1.$$

4.1.4 Meta-algorithm implementation for sNMF

The MM updates for factors X_1 and X_2 in the form of so-called "multiplicative" derived to address the ϵ-perturbed Problem (34)[8] are as follows:

[8] Here, the data fitting term is the entrywise Kullback-Leibler divergence, hence we lower-bound each factor by $\epsilon > 0$ as explained in Section 3.5

$$\hat{X}_1 = \max\left(X_1 \odot \frac{X_2 \frac{M}{X_2^T X_1}}{\mu_1 e_{m_1 \times r} + e_{m_1} \otimes X_2^T e_{m_2}}, \epsilon\right), \quad \hat{X}_2 = \max\left(X_2 \odot \frac{\frac{M^T}{X_1 X_2^T} X_1^T}{\mu_2 e_{m_2 \times r} + e_{m_2} \otimes X_1 e_r}, \epsilon\right), \tag{38}$$

where $e_\cdot$ is a vector or matrix of ones of appropriate size and the maximum is taken elementwise. The detailed computations are found in Appendix F.

4.1.5 sNMF synthetic experiments

The goal of these experiments are the following. First we want to study the impact of the balancing procedure in the meta-algorithm, compared to the same algorithm without explicit balancing. We observe the performance of both strategies in terms of loss value at the last iteration and Factor Match Score [1]. Further, we also explore a third option where balancing is performed only at initialization. Second, to validate that the individual values of μ_1 and μ_2 do not influence the relative sparsity levels of X_1 and X_2, we study the ratio $\frac{\|X_1\|_0}{\|X_2\|_0}$ with respect to the regularization hyperparameter. An entry is considered null in a matrix if it is below the threshold 2ϵ.

The setup for the synthetic data generation is the following. We set $M = X_1 X_2^T$ with $n_1 = n_2 = 30$. The rank of the model r is set to 4, and the number of components for the approximation r_e (estimated rank) is also set to 4. The noisy data matrix is sampled from $\mathcal{P}(\alpha M)$ with parameter α chosen such that the average Signal to Noise Ratio (SNR) is set to 40, and then normalized by its Frobenius norm. This corresponds to a very low noise regime given our generation setup. The true factors X_1 and X_2 are sampled elementwise from i.i.d. Uniform distributions over $[0, 1]$, and then sparsified using hard thresholding so that 30% of their entries are null.

The hyperparameters are chosen as follows. We set $\epsilon = 10^{-16}$. The regularization coefficients μ_1 and μ_2 are set on a grid $[0, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}, 5 \times 10^{-1}]$. We distinguish two study cases. Case one, regularization hyperparameters are equal, that is $\mu_1 = \mu_2$. Case two, we fix one regularization hyperparameter $\mu_1 = 1$. The first case is the recommended choice to reduce the number of hyperparameters to tune, and the second case allows to study the relative sparsity of the factors with respect to the individual values of the regularization hyperparameters (which should have no impact on sparsity as discussed earlier). Initial factors are drawn similarly to the true factors, but matrix X_1 is scaled by 100, so the parameter matrices are unbalanced to emphasize the effect of the scaling procedure. We scale the initialization optimally as described in Algorithm 2. The number of outer iterations is set to 500, and the number of inner iterations is set to 10. This experiment is repeated fifty times.

Results are shown in Figures 2 and 3. We validate several observations from the theoretical derivations:

- For small values of μ, as predicted in section 3.3, the meta-algorithm without explicit balancing struggles to converge in terms of loss value, although as shown by the FMS, the parameter matrices estimates are of similar quality than with the explicit balancing. This means that for small μ the unbalanced algorithm has found good candidate solutions X_1 and X_2 but takes time to balance these estimates. Explicit balancing minimizes the regularization terms and therefore the cost is much lower after running the algorithm. Moreover, for the sNMF problem, balancing the initial guess seems to provide similar results as balancing at each iteration. Therefore given the negligible cost of balancing, there is little reason not to perform this balancing step before any off-the-shelf algorithm for sNMF.

- We can observe that the relative sparsity levels in both cases one ($\mu_1 = \mu_2$) and two ($\mu_1 = 1$) are similar and close to 1. In other words, increasing μ_2 while fixing μ_1 did not allow for increasing the sparsity level of X_2 with respect to X_1. Further we may observe that the simulation results for cases one and two match when $\mu_1 \mu_2$ in case one is equal to μ_2 in case two, illustrating that only the product of regularization hyperparameters matters.

- Unsurprisingly, in case two, setting $\mu_2 = 0$ provides with poor results since the optimization problem is ill-posed. Case one does not exhibit this issue since an unconstrained NMF is computed in that case.

4.2 Ridge regularized Nonnegative Canonical Polyadic Decomposition

4.2.1 Background

The approximate Nonnegative Canonical Polyadic Decomposision (CPD) or PARAFAC decomposition is a low-rank approximation model where a tensor containing data is approximated by a sum of rank-one tensors.

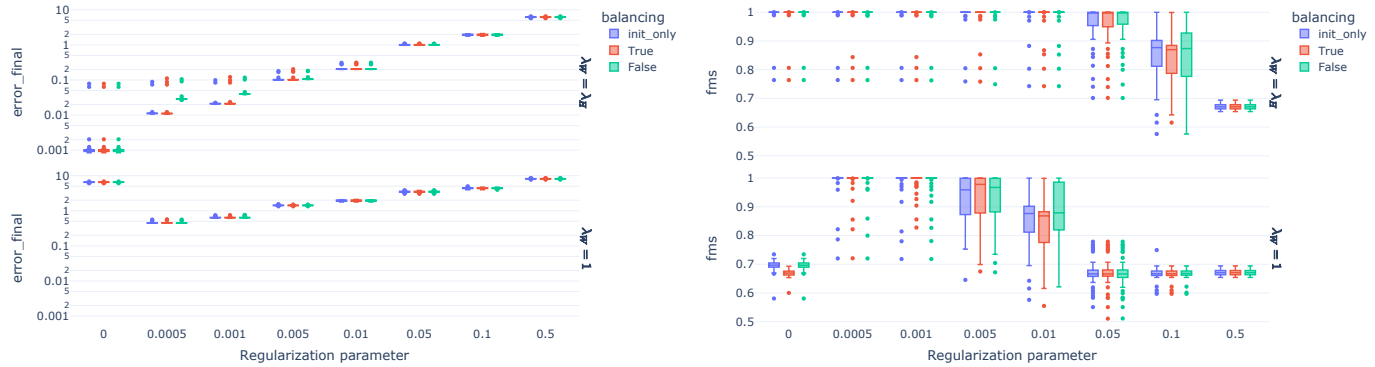

Figure 2: Results for the simulated experiment on sparse NMF. The loss after the stopping criterion is reached is shown with respect to the regularization parameters on the left plot, while the factor match score is shown in the right plot.

Figure 3: Results for the simulated experiment on sparse NMF. The ratio between the sparsity of estimated matrix X_1 and the sparsity of estimated matrix X_2 is shown in the left plot, while the sparsity level of matrices X_1 and X_2 scaled from zero (null) to one (dense) is shown in the right plot.

The number of rank-one terms is the rank of the approximate tensor and is fixed in general. A usual formulation for the third order NCPD with Frobenius loss and approximation rank r is the following:

$$\underset{X_1\in\mathbb{R}_+^{m_1\times r},\ X_2\in\mathbb{R}_+^{m_2\times r},\ X_3\in\mathbb{R}_+^{m_3\times r}}{\operatorname{argmin}} \|\mathcal{T} - I_r \times_1 X_1 \times_2 X_2 \times_3 X_3\|_F^2 \tag{39}$$

where $\mathcal{I}_r$ is a super-diagonal tensor of size $r \times r \times r$ with nonzero values set to one, $\mathcal{T} \in \mathbb{R}_+^{m_1\times m_2\times m_3}$ is the tensor of data and $\times_i$ is a multiway product [32]. This problem is not ill-posed because regularizations such as nonnegativity constraints or ridge penalties to an approximation CPD [40,72] make the loss coercive on the domain of admissible solutions.

A key issue with many low-rank approximations models, and with NCPD in particular, is the choice of the decomposition rank. In theory, computing the exact CP rank of a tensor is an NP-hard problem [28]. In practice, choosing the rank of the approximate tensor is also very challenging, since naive approaches such as deflation in general fail to produce satisfactory approximation results; in fact, in contract to the singular value decomposition, substracting the best rank-one approximation of a tensor can increase the tensor rank instead of decreasing it [61], and truncating a low-rank tensor model does not yield an optimal low-rank representation [35].

Several approaches have been described to automatically select a right number of components in the CP decomposition. A first line of work consists in penalizing the columns of factors with a group-sparse norm such as the $\ell_{1,2}$ norm [25]. However this makes the optimization problem significantly harder. Another line of work from the coupled CP decomposition literature penalizes the ℓ_1 norm of component norms [2]. Again this makes the optimization problem harder to deal with since ℓ_2 normalizations are imposed on the factors columnwise, and components weights have to be explicitly computed.

As hinted in Section 2, we propose in the following to use simple ridge penalization on all factors. From the implicit HRSI formulation, we can observe that these ridge regularization induce group sparsity on the rank-one components. This is similar in spirit to the works mentioned in the above paragraph, albeit simpler to handle algorithmically within an alternating optimization framework. The proposed ridge-regularized NCPD (rNCPD) is formulated as follows:

$$\underset{X_1\geq 0, X_2\geq 0, X_3\geq 0}{\operatorname{argmin}} \|\mathcal{T} - \mathcal{I}_r \times_1 X_1 \times_2 X_2 \times_3 X_3\|_F^2 + \mu\left(\|X_1\|_F^2 + \|X_2\|_F^2 + \|X_3\|_F^2\right) . \tag{40}$$

4.2.2 Relation to HRSI, optimal balancing and implicit formulation

It is simple to see that the rNCPD features a scale-ambiguity for each rank-one component. Indeed, the model part of rNCPD can be written as

$$\mathcal{I}_r \times_1 X_1 \times_2 X_2 \times_3 X_3 = \sum_{q=1}^{r} X_1[:,q] \otimes X_2[:,q] \otimes X_3[:,q] \tag{41}$$

where $\otimes$ is a tensor product, here the outer product. We may scale any column q by $\{\lambda_i\}_{i\leq 3}$ such that $\lambda_1\lambda_2\lambda_3 = 1$ without changing the low-rank tensor:

$$\sum_{q=1}^{r} \lambda_1\lambda_2\lambda_3 X_1[:,q] \otimes X_2[:,q] \otimes X_3[:,q] = \sum_{q=1}^{r} \lambda_1 X_1[:,q] \otimes \lambda_2 X_2[:,q] \otimes \lambda_3 X_3[:,q]. \tag{42}$$

Furthermore, the ridge penalization is columnwise separable. This means that rNCPD (40) is an instance of HRSI with three or more factors.

From Section 2, we know that solutions to rCPD will have balanced columns. The balancing procedure for the q-th column writes

$$X_1[:,q] = \sqrt{\frac{\beta_q}{2\mu\|X_1[:,q]\|_2^2}} X_1[:,q] = \left(\sqrt{2}\|X_1[:,q]\|_2^{-\frac{1}{3}} \|X_2[:,q]\|_2^{\frac{2}{3}} \|X_3[:,q]\|_2^{\frac{2}{3}}\right) X_1[:,q] \tag{43}$$

$$X_2[:,q] = \sqrt{\frac{\beta_q}{2\mu\|X_2[:,q]\|_2^2}} X_2[:,q] = \left(\sqrt{2}\|X_1[:,q]\|_2^{\frac{2}{3}} \|X_2[:,q]\|_2^{-\frac{1}{3}} \|X_3[:,q]\|_2^{\frac{2}{3}}\right) X_2[:,q] \tag{44}$$

$$X_3[:,q] = \sqrt{\frac{\beta_q}{2\mu\|X_3[:,q]\|_2^2}} X_3[:,q] = \left(\sqrt{2}\|X_1[:,q]\|_2^{\frac{2}{3}} \|X_2[:,q]\|_2^{\frac{2}{3}} \|X_3[:,q]\|_2^{-\frac{1}{3}}\right) X_3[:,q] \tag{45}$$

where $\beta_q = 4\mu \left(\|X_1[:,q]\|_F \|X_2[:,q]\|_F \|X_3[:,q]\|_F\right)^{\frac{2}{3}}$, and the underlying equivalent implicit problem is formulated as

$$\inf_{X_1 \in \mathbb{R}^{m_1 \times r},\ X_2 \in \mathbb{R}^{m_2 \times r},\ X_3 \in \mathbb{R}^{m_3 \times r}} \|\mathcal{T} - \sum_{q=1}^{r} X_1[:,q] \otimes X_2[:,q] \otimes X_3[:,q]\|_F^2 + 3\mu \sum_{q=1}^{r} \left(\|X_1[:,q]\|_2 \|X_2[:,q]\|_2 \|X_3[:,q]\|_2\right)^{\frac{3}{2}}, \tag{46}$$

or equivalently, introducing rank-one tensors $\mathcal{L}_q = X_1[:,q] \otimes X_2[:,q] \otimes X_3[:,q]$,

$$\inf_{\{\mathcal{L}_q\}_{q \leq r},\ \mathrm{rank}(\mathcal{L}_q) \leq 1} \|\mathcal{T} - \sum_{q=1}^{r} \mathcal{L}_q\|_F^2 + 3\mu \sum_{q=1}^{r} \|\mathcal{L}_q\|_F^{\frac{3}{2}}. \tag{47}$$

This implicit formulation is a variant of group-LASSO with the $\ell_2^{\frac{3}{2}}$ norm used for the group norms which is uncommon but should not be an issue for the analysis of such a model. The rank-one constraint however makes it difficult to invoke existing results regarding the sparsity of the solutions, *i.e.* the actual low-rank promotion effect of the group-sparsity regularization. Nevertheless we observe in the experiments section 4.2.4 that ridge regularization leads to low-rank solutions for the rNCPD.

4.2.3 Meta-algorithm implementation for rNCPD

In this section we derive updates for factors X_1, X_2 and X_3. Since all three subproblems have similar structure, we will only focus on the subproblem in X_1 without loss of generality.

The subproblem in X_1 is defined as follows:

$$\underset{X_1 \geq 0}{\mathrm{argmin}} \ \|T_{[1]} - X_1 (X_2 \odot X_3)^T\|_F^2 + \mu_1 \|X_1\|_F^2 \tag{48}$$

with $T_{[1]}$ the first-mode unfolding of $\mathcal{T}$ and $\odot$ the Khatri-Rao product, see [32] for more details on how to obtain this matricized optimization problem.

To solve this subproblem, we suggest to use the HALS algorithm [23] which is a fast nonnegative least squares solver for matrix problems. In the spirit of the proposed Meta-algorithm, the estimated parameter matrices are balanced after each outer iteration.

4.2.4 Numerical Experiments

rCPD simulated experiments The experiments conducted in this section are adapted from our previous work [10]. The setup is overall similar to the sNMF simulated experiment setup, but with noisy data sampled from a Gaussian distribution, rank over-estimation and no artificial unbalancing of the initial parameter matrices. We set $\mathcal{T} = \mathcal{I}_r \times_1 X_1 \times_2 X_2 \times_3 X_3$ with $n_1 = n_2 = n_3 = 30$. The rank of the model r is set to 4, and the number of components for the approximation r_e (estimated rank) is overestimated to 6. We hope that the ridge penalization will act as a rank penalization and prune two components. The noisy tensor sampled from $\mathcal{N}(0, \alpha)$ where parameter α is chosen such that the SNR is set to 200 (low noise regime). The true parameter matrices are sampled elementwise from i.i.d. Uniform distributions over $[0, 1]$.

The hyperparameters are chosen as follows. We set $\epsilon = 10^{-16}$. The regularization coefficient μ is set on a grid $[0, 10^{-3}, 10^{-2}, 10^{-1}, 2 \times 10^{-1}, 5 \times 10^{-1}, 1, 2, 5, 10]$. Initial factors are drawn similarly to the true factors. We scale the initial values optimally as described in Algorithm 2. The number of outer iterations is set to 50, and the number of inner iterations is set to 10. The experiment is ran fifty times.

The goals of these experiments are the following. First we want to study the impact of the balancing procedure in the meta-algorithm, compared to the same algorithm without explicit balancing. We also explore a third option where balancing is performed only at initialization. We measure the performance of both strategies in terms of loss value at the last iteration and Factor Match Score. Second, to validate that ridge regularization allows for efficient rank selection, we study the number of nonzero rank-one components with respect to the regularization hyperparameter μ. A component is considered null if weight associated with this component (product of ℓ_2 norms of each factor matrix along each column) is lower than $1000 \times \epsilon$ (there are $30 \times 6 = 180$ entries).

Results are shown in Figure 4 for the normalized loss function and FMS, and Figure 5 for the number of non-zero components. We validate several observations from the theoretical derivations:

Figure 4: Results for the simulated experiment on ridge CPD. The normalized loss after the stopping criterion is reached is shown with respect to the regularization parameters on the left plot, while the factor match score is shown in the right plot.

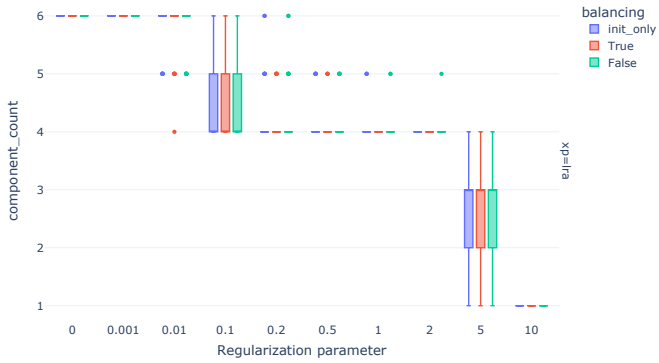

Figure 5: Results for the simulated experiment on ridge CPD. The number of components within the estimated rCPD model with respect to the regularization parameters is shown on the left plot.

- Similarly to sNMF, for small values of μ and as predicted in section 3.3, the meta-algorithm without explicit balancing struggles to converge in terms of loss value, although the parameter matrices themselves are of similar quality than with the explicit balancing. Moreover, for the rCPD problem and contrarily to the observations for sNMF, balancing the initial guess is not sufficient to ensure a proper balancing of the parameter matrices throughout the iterations.
- For a large range of regularization hyperparameter values, the number of estimated components, four, matches the true number of components. Therefore the ridge regularization has indeed performed an automatic rank selection as predicted in Section 2. The FMS is also improved when the balancing procedure is used, suggesting that for the rank selection effect of the ridge regularization benefits from the balancing procedure.

4.3 Sparse Nonnegative Tucker Decomposition

4.3.1 Background and formalism

The Nonnegative Tucker Decomposision (NTD) is another low-rank approximation model where a tensor containing data is approximated by a tensor with low multilinear ranks and nonnegative factors [48]. These multilinear ranks are hyperparameters of the model that are in general fixed. Nonnegativity constraints are essential to the NTD because they allow for some empirical identifiability properties of the parameters, however formal results on NTD identifiability are still scarce [64, 74].

In this showcase, we use the Kullback-Leibler divergence to measure discrepencies between the data tensor and the low-rank model. Computing NTD for a third order tensor then means finding a minimizer defined as follows

$$\underset{X_1\geq 0, X_2\geq 0, X_3\geq 0, \mathcal{G}\geq 0}{\operatorname{argmin}} \quad D(\mathcal{T}|\mathcal{G}\times_1 X_1\times_2 X_2\times_3 X_3) \tag{49}$$

where D is the KL divergence, $\mathcal{G} \in \mathbb{R}_+^{r_1 \times r_2 \times r_3}$ is called the core tensor and X_1, X_2, X_3 respectively in $\mathbb{R}_+^{m_i \times r_i}$ are the factor matrices. Dimensions r_i are called the multilinear ranks of the NTD[9]. We may also write the NTD model using a sum of separable terms as follows:

$$\mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3 = \sum_{q_1,q_2,q_3=1}^{r_1,r_2,r_3} \mathcal{G}[q_1, q_2, q_3] X_1[:, q_1] X_2[:, q_2] X_3[:, q_3] \tag{50}$$

Nonnegativity constraints are essential to the uniqueness of NTD but are often empirically not sufficient to guarantee the identifiability of a solution. Sparsity penalties have been documented to partially alleviate the identifiability issue, in particular when imposed on the core tensor [48, 57]. Therefore sparse NTD (sNTD) is a very commonly encountered variant of NTD.

Similarly to sNMF, sparse factors in sNTD are often regularized with the ℓ_1 norm while the dense factors are in general normalized. In this paper, we penalize the factors matrices with a ridge penalty, and the core tensor with the sparsity-inducing ℓ_1 norm. The obtained sparse NTD (sNTD) model of order three consists in computing

$$\underset{X_1 \geq 0, X_2 \geq 0, X_3 \geq 0, \mathcal{G} \geq 0}{\operatorname{argmin}} D(\mathcal{T} | \mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3) + \mu(\|\mathcal{G}\|_1 + \|X_1\|_F^2 + \|X_2\|_F^2 + \|X_3\|_F^2) \tag{51}$$

for a user-defined regularization hyperparameter $\mu \geq 0$.

4.3.2 Scaling invariance in sNTD

At first glance the sNTD does not fit in the HRSI framework. While there is a form of scale ambiguity in NTD, it is not of the form we introduced in Assumption **A1** in Section 1.2. Rather, the scaling ambiguity works pairwise between the core and each factor, for instance on the first mode,

$$(\mathcal{G} \times_1 \Lambda) \times_1 (X_1 \Lambda^{-1}) \times_2 X_2 \times_3 X_3 = \mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3 \tag{52}$$

for a diagonal matrix Λ with positive entries. This means that the core matrix has scaling ambiguities affecting its rows, columns, and fibers. It then makes little sense to talk about scaling the columns of $\mathcal{G}$ within the HRSI framework.

The fact that the scaling invariances do not fit the HRSI framework does not change the fact that the regularizations will enforce implicitly a particular choice of the parameters $\mathcal{G}$ and X_1, X_2, X_3 at optimality. Therefore it should still be valuable to design a balancing procedure to enhance convergence speed, as we have proposed for sNMF and rCPD. There are two ways to perform such a balancing, which we explore in the next two subsections.

4.3.3 Global scale-invariance: closed-form scaling

It is possible to cast the sNTD as a rank-one HRSI model, by considering a vectorized sNTD problem:

$$\underset{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, g \geq 0}{\operatorname{argmin}} f(\{x_1, x_2, x_2, g\}) + \mu(\|g\|_1 + \|x_1\|_F^2 + \|x_2\|_F^2 + \|x_3\|_F^2) \tag{53}$$

where $x_1 = \operatorname{vec}(X_1)$, $x_2 = \operatorname{vec}(X_2)$, $x_3 = \operatorname{vec}(X_3)$, $g = \operatorname{vec}(\mathcal{G})$ for a chosen arbitrary vectorization procedure, and where f is the KL divergence with vectorized inputs.

One may observe that, while in general sNTD scaling ambiguities work pairwise between the core and the factors, for a one-dimensional scale the invariance is exactly of the form assumed in HRSI: given any set of positive parameters $\{\lambda_i\}_{i \leq 4}$ with $\prod_{i \leq 4} \lambda_i = 1$, one can easily observe that $\mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3 = \mathcal{G}\lambda_4 \times_1 X_1\lambda_1 \times_2 X_2\lambda_2 \times_3 X_3\lambda_3$.

We can therefore use the results from Section 2 to perform a scalar balancing of the sNTD. Following Proposition 2, solutions to the sparse NTD must satisfy $2\mu\|X_1\|_F^2 = 2\mu\|X_2\|_F^2 = 2\mu\|X_3\|_F^2 = \mu\|\mathcal{G}\|_1 = \beta$, which yields the following scaling procedure:

$$X_1 = \sqrt{\frac{\beta}{2\mu\|X_1\|_F^2}} X_1,\ X_2 = \sqrt{\frac{\beta}{2\mu\|X_2\|_F^2}} X_2,\ X_3 = \sqrt{\frac{\beta}{2\mu\|X_3\|_F^2}} X_3,\ \mathcal{G} = \frac{\beta}{\mu\|\mathcal{G}\|_1} \mathcal{G} \tag{54}$$

with $\beta = 2^{\frac{3}{5}} \mu \left(\|X_1\|_F \|X_2\|_F \|X_3\|_F \|\mathcal{G}\|_1\right)^{2/5}$.

[9] We abusively call the r_i parameters "ranks". They are essentially the multilinear ranks of the approximation tensor, but the true rank of the factors may nevertheless be strictly smaller [3]

4.3.4 Scaling slices and the core with tensor Sinkhorn

The scalar scale invariance detailed above in Section 4.3.3 does not encompass all scaling invariances that occur in NTD. Again, in general, for each mode, the core and factor can be rescaled by inserting a diagonal matrix in-between, e.g. a matrix Λ_1 for the first mode: $\mathcal{G} \times_1 X_1 = (\mathcal{G} \times_1 \Lambda_1) \times_1 (X_1 \Lambda_1^{-1})$, where Λ_1 has strictly positive diagonal elements. The scaling on each mode affects all the slices of the core, therefore the HRSI framework does not directly apply. Nevertheless, it is possible to derive an algorithmic procedure similar to the Sinkhorn algorithm to scale the factors and the core. We provide detailed derivations for the special case of sNTD, but the proposed method extends to any regularized NTD with homogeneous regularization functions.

To understand the connection between the Sinkhorn algorithm, which aims at scaling a matrix or tensor so that the slices are on the unit simplex [34, 46, 58], and the proposed procedure, consider the following: the results of Section 2 applied to scale invariance on the first mode inform that at optimality, $\beta_{X_1}[q] := 2\|X_1^*[:, q]\|_2^2 = \|\mathcal{G}_{[1]}^*[:, q]\|_1$, where $\mathcal{G}_{[i]}$ is the mode i unfolding of the core tensor $\mathcal{G}$. This observation implies that solutions of sparse NTD must in particular be solutions to a system of equations of the form:

$$\beta_{X_1}[q] = \|\mathcal{G}_{[1]}[:, q]\|_1 \tag{55}$$

$$\beta_{X_2}[q] = \|\mathcal{G}_{[2]}[:, q]\|_1 \tag{56}$$

$$\beta_{X_3}[q] = \|\mathcal{G}_{[3]}[:, q]\|_1 \tag{57}$$

Optimally balancing sparse NTD can therefore be reduced to two operations: first, compute the optimal regularizations β_{X_i}, then transform the core and the factors so that(55)-(57) holds. Supposing we know the β_{X_i} values, computing the scalings of the slices of the core tensor in each mode is exactly what the tensor Sinkhorn algorithm is designed for [13, 63]. However, since the β_{X_i} values are not known, we cannot directly apply a tensor Sinkhorn algorithm. Tensor Sinkhorn usually involves an alternating normalization procedure, which we shall mimic in the following.

We propose a scaling heuristic which, starting from initial diagonal matrices $\Lambda_1, \Lambda_2, \Lambda_3$ set to identity matrices, alternatively balances each mode optimally:

$$\underset{\Lambda_1 \text{ diagonal}}{\operatorname{argmin}} \|\mathcal{G} \times_1 \Lambda_1 \times_2 \Lambda_2 \times_3 \Lambda_2\|_1 + \|X_1 \Lambda_1^{-1}|_F^2 \tag{58}$$

$$\underset{\Lambda_3 \text{ diagonal}}{\operatorname{argmin}} \|\mathcal{G} \times_1 \Lambda_1 \times_2 \Lambda_2 \times_3 \Lambda_3\|_1 + \|X_2 \Lambda_2^{-1}\|_F^2 \tag{59}$$

$$\underset{\Lambda_3 \text{ diagonal}}{\operatorname{argmin}} \|\mathcal{G} \times_1 \Lambda_1 \times_2 \Lambda_2 \times_3 \Lambda_3\|_1 + \|X_3 \Lambda_3^{-1}\|_F^2 \ . \tag{60}$$

This procedure computes estimates of the β_{X_i} marginals iteratively for each mode, and scales the core tensor and the factors accordingly on the fly. It can be seen as a modified tensor Sinkhorn algorithm where the β_{X_i} marginals are learned iteratively from the input factors X_1, X_2, X_3 and core $\mathcal{G}$ in order to satisfy Equations (55)-(57). Each minimization step boils down to a balancing with two factors in the spirit of what is showcased for sNMF in Section 4.1. Balancing the first mode yields

$$\mathcal{G}[q, :, :] \leftarrow \frac{\beta_{X_1}[q]}{\|\Lambda_2 \mathcal{G}[q, :, :] \Lambda_3\|_1} \mathcal{G}[q, :, :] \text{ and } X_1[:, q] \leftarrow \sqrt{\frac{\beta_{X_1}[q]}{2\|X_1[:, q]\|_2^2}} X_1[:, q] \tag{61}$$

with $\beta_{X_1} = \left(\sqrt{2}\|X_1\|_F \|\mathcal{G} \times_2 \Lambda_2 \times_3 \Lambda_3\|_1\right)^{2/3}$. The other two modes are updated in a similar way. Note that in the alternating scaling algorithm, the scales are not explicitly stored. Rather they are embedded in the core at each iteration because of the scaling update.

As the cost decreases at each iteration and is bounded below, the algorithm is guaranteed to converge. The update rule is well-defined if the factors and the core have non-zero columns/slices. Additionally, since each subproblem has a unique minimizer that is attained, and given that the objective function to minimize in Equation (58) is continuously differentiable, every limit point of this block-coordinate descent algorithm is a critical point (see [6, Proposition 2.7.1]). If the objective function is a proper closed function and is continuous over its domain (not necessarily continuously differentiable), then any limit point of the sequence is a coordinate-wise minimum (see [4, Theorem 14.3]).

While this guarantees that the scales of the factors and core will be improved, unlike for the tensor Sinkhorn, we are not able to guarantee that the proposed heuristic computes an optimal scaling although unreported experiments hint towards this direction. The proposed adaptive tensor Sinkhorn algorithm is summarized for the sNTD problem in Algorithm 3.

Finally, we handle the null components and the entries threshold ϵ for the scaling as explained in Section 3.6.

Algorithm 3 Balancing algorithm for sNTD

Require: $X_1, X_2, X_3, \mathcal{G}$, maximal number of iterations itermax.

Initialize $\beta_{X_1}[q]$, $\beta_{X_2}[q]$ and $\beta_{X_3}[q]$ respectively with $\|X_1[:,q]\|_2^2, \|X_2[:,q]\|_2^2, \|X_3[:,q]\|_2^2$ for all q.
for $i = 1$: itermax **do**
  for $q = 1{:}r_1$ **do**
    $\beta_{X_1}[q] = \left(\sqrt{2\beta_{X_1}[q]}\|\mathcal{G}[q,:,:]\|_1\right)^{\frac{2}{3}}$
    $\mathcal{G}[q,:,:] = \frac{\beta_{X_1}[q]}{\|\mathcal{G}[q,:,:]\|_1}\mathcal{G}[q,:,:]$ or $\mathcal{G}[q,:,:] = 0$ if $\beta_{X_1}[q] = 0$
  end for
  for $q = 1{:}r_2$ **do**
    $\beta_{X_2}[q] = \left(\sqrt{2\beta_{X_2}[q]}\|\mathcal{G}[:,q,:]\|_1\right)^{\frac{2}{3}}$
    $\mathcal{G}[:,q,:] = \frac{\beta_{X_2}[q]}{\|\mathcal{G}[:,q,:]\|_1}\mathcal{G}[:,q,:]$ or $\mathcal{G}[:,q,:] = 0$ if $\beta_{X_2}[q] = 0$
  end for
  for $q = 1{:}r_3$ **do**
    $\beta_{X_3}[q] = \left(\sqrt{2\beta_{X_3}[q]}\|\mathcal{G}[:,:,q]\|_1\right)^{\frac{2}{3}}$
    $\mathcal{G}[:,:,q] = \frac{\beta_{X_3}[q]}{\|\mathcal{G}[:,:,q]\|_1}\mathcal{G}[:,:,q]$ or $\mathcal{G}[:,:,q] = 0$ if $\beta_{X_3}[q] = 0$
  end for
end for
for $q = 1{:}r_1$ **do**
  $X_1[:,q] = \sqrt{\frac{\beta_{X_1}[q]}{2\|X_1[:,q]\|_F^2}}X_1[:,q]$ or $X_1[:,q] = 0$ if $\|X_1[:,q]\|_2^2 = 0$
end for
for $q = 1{:}r_2$ **do**
  $X_2[:,q] = \sqrt{\frac{\beta_{X_2}[q]}{2\|X_2[:,q]\|_F^2}}X_2[:,q]$ or $X_2[:,q] = 0$ if $\|X_2[:,q]\|_2^2 = 0$
end for
for $q = 1{:}r_3$ **do**
  $X_3[:,q] = \sqrt{\frac{\beta_{X_3}[q]}{2\|X_3[:,q]\|_F^2}}X_3[:,q]$ or $X_3[:,q] = 0$ if $\|X_3[:,q]\|_2^2 = 0$
end for
return $X_1, X_2, X_3, \mathcal{G}$

4.4 Meta-algorithm implementation for sNTD

Updates of the factors In this section we derive updates for factors X_1, X_2, X_3 with ℓ_2^2 regularizations. Since all the subproblems have similar structure, we will only focus on the subproblem in X_1 without loss of generality. For clarity, considering the matricization of the tensor $\mathcal{T}$ along the first mode and pose $T := \mathcal{T}_{[1]} \in \mathbb{R}_+^{m_1 \times m_2 m_3}$ and $U := \mathcal{G}_{[1]} (X_2 \boxtimes X_3)^T \in \mathbb{R}_+^{r_1 \times m_2 m_3}$. The subproblem in X_1 is defined as follows:

$$\underset{X_1 \geq 0}{\operatorname{argmin}}\ D(T|X_1 U) + \mu\|X_1\|_F^2 \tag{62}$$

since $\mu \sum_{q \leq r} \|X_1[:,q]\|_2^2 = \mu\|X_1\|_F^2$. For $\beta = 1$, we solve the ϵ-perturbated Problem (62) as discussed in Section 3.5. Using the derivations from Appendix F, the MM update rule for factor X_1 reads

$$\hat{X_1} = \max\left(\frac{\left[C^{.2} + S\right]^{\cdot\frac{1}{2}} - C}{2\mu}, \epsilon\right) \tag{63}$$

where $C = EU^T$ with E is a all-one matrix of size m_1-by-$m_2 m_3$ and $S = 4\mu\tilde{X_1} \odot \left(\frac{[T]}{[\tilde{X_1}U]}U^T\right)$, with $\tilde{X_1}$ the current point.

Update of the core tensor The update of the core tensor $\mathcal{G}$ is similar to the update of the sparse factors in the sNMF showcase section 4.1. For the core tensor, note the following vectorization property:

$$\operatorname{vec}(\mathcal{T}) = (X_1 \boxtimes X_2 \boxtimes X_3) \operatorname{vec}(\mathcal{G}) \tag{64}$$

Let us pose $U := (X_1 \boxtimes X_2 \boxtimes X_3) \in \mathbb{R}^{m \times r}$ with $m = m_1 m_2 m_3$, $r = r_1 r_2 r_3$, $t := \mathrm{vec}\,(\mathcal{T}) \in \mathbb{R}^m$ and $g := \mathrm{vec}\,(\mathcal{G}) \in \mathbb{R}^r$. The subproblem in g is defined as follows:

$$\underset{g \geq 0}{\operatorname{argmin}}\ D(t|Ug) + \mu \|g\|_1 \tag{65}$$

By referring to Appendix F for $\beta = 1$, the following MM update rule for g is obtained to address the ϵ-perturbed Problem (65):

$$\hat{g} = \max\left(g \odot \frac{U^T \frac{t}{Ug}}{\mu e_r + U^T e_m}, \epsilon\right) \tag{66}$$

Note that, regarding complexity and numerical costs, as opposed to factor updates, in most practical cases we cannot compute and store the matrix U explicitly since it is much larger than the dataset itself. To tackle this issue, we follow a classical approach in the tensor decomposition literature, for instance recalled in [44] that is all products Ut for any $t := \mathrm{vec}\,(\mathcal{T})$ are computed using the following identity:

$$(X_1 \boxtimes X_2 \boxtimes X_3)t = \mathrm{vec}\,(\mathcal{T} \times_1 X_1 \times_2 X_2 \times_3 X_3) \tag{67}$$

The products $U^T t$ are computed similarly since the transposition is distributive with respect to the Kronecker product. Using multilinear algebra, the update of $\mathcal{G}$ is computed as

$$\tilde{\mathcal{G}} = \max\left(\mathcal{G} \odot \frac{[\mathcal{N} \times_1 X_1^T \times_2 X_2^T \times_3 X_3^T]}{[\mu \mathcal{E}_{r_1 \times r_2 \times r_3} + \mathcal{E}_{m_1 \times m_2 \times m_3} \times_1 X_1^T \times_2 X_2^T \times_3 X_3^T]}, \epsilon\right) \tag{68}$$

with $\mathcal{E}$ an all-ones tensor and $\mathcal{N} = \frac{\mathcal{T}}{\mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3}$. Again this tensor of ones is not contracted as such in practice, instead we use summations of columns in the factors.

Algorithm 4 summarizes our method to compute sNTD (51) for the β-divergences in the particular case detailed above for $\beta = 1$ which we refer to as Sparse KL-NTD algorithm.

Algorithm 4 Sparse KL-NTD (with ℓ_1 and ℓ_2^2 regularizations resp. on the core tensor and the factors)

Require: A tensor $\mathcal{T} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$, an initialization $\mathcal{G} \in \mathbb{R}_+^{r_1 \times r_2 \times r_3}$, $X_1 \in \mathbb{R}_+^{m_1 \times r_1}$, $X_2 \in \mathbb{R}_+^{m_2 \times r_2}$ and $X_3 \in \mathbb{R}^{m_3 \times r_3}$, the factorization ranks (r_1, r_2, r_3), a maximum number of iterations, maxiter, $\epsilon > 0$, and penalty weight $\mu > 0$.

% Initialization using ℓ_2 loss:

$$\eta \in \underset{\eta \geq 0}{\operatorname{argmin}}\ \|\mathcal{T} - \eta^4 X_1 \times_1 X_2 \times_2 X_3 \times_3 \mathcal{G}\|_F^2 + \mu\left(\sum_{i=1}^{3} \eta^2 \|X_i\|_F^2 + \eta \|\mathcal{G}\|_1\right)$$

for $it = 1$: maxiter **do**
  % Update of factor X_1
  $U \leftarrow (\mathcal{G} \times_2 X_2 \times_3 X_3)_{[1]}$
  $T \leftarrow \mathcal{T}_{[1]}$
  $C \leftarrow EU^T$
  $S \leftarrow 4\mu X_1 \odot \left(\frac{[T]}{[X_1 U]} U^T\right)$
  $X_1 \leftarrow \max\left(\epsilon, \frac{\left[C^{\cdot 2} + S\right]^{\cdot \frac{1}{2}} - C}{2\mu}\right)$
  X_2 and X_3 are updated in a similar way as X_1.
  % Update of core tensor $\mathcal{G}$
  $\mathcal{N} \leftarrow (\mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3)^{\cdot(-1)} \odot \mathcal{T}$
  $\mathcal{G} \leftarrow \max\left(\epsilon, \mathcal{G} \odot \frac{[\mathcal{N} \times_1 X_1^T \times_2 X_2^T \times_3 X_3^T]}{[\mu \mathcal{E}_{r_1 \times r_2 \times r_3} + \mathcal{E}_{m_1 \times m_2 \times m_3} \times_1 X_1^T \times_2 X_2^T \times_3 X_3^T]}\right)$
  % Heuristic balancing
  Call of Algorithm 3, or Algorithm 1 on vectorized parameter matrices and core.
end for
return $\mathcal{G}$, X_1, X_2 and X_3

4.4.1 sNTD synthetic experiments

The setup for the synthetic data generation is similar to the setup for sNMF synthetic data generation in Section 4.1. We set $\mathcal{T} = \mathcal{G} \times_1 X_1 \times_2 X_2 \times_3 X_3$ with $n_1 = n_2 = n_3 = 30$. The ranks of the model r_i are set so that the true core tensor has dimensions $4 \times 4 \times 4$. However the number of components for the approximation are set so that the core tensor of the approximation has dimensions $6 \times 4 \times 4$; therefore the multilinear rank on the first mode is overestimated. The Signal to Noise Ratio (SNR) is set to 40 (low noise regime) with the noisy tensor sampled from the Poisson distibution then normalized as in the sNMF experiment. The true factors X_1, X_2 and X_3 are sampled elementwise from i.i.d. Uniform distributions over $[0, 1]$. The core tensor $\mathcal{G}$ is drawn similarly but is then sparsified using hard thresholding so that 70% of its entries are null.

The hyperparameters are chosen as follows. We set $\epsilon = 10^{-16}$. The regularization coefficient μ is set on a grid $[0, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}, 5 \times 10^{-2}, 10^{-1}, 5 \times 10^{-1}]$. Initial factors are drawn similarly to the true factors. However, the initialization is then scaled in a particular manner to emphasize the benefits of the balancing procedure. Initial matrix X_1 is scaled by 100, and we also scale the first column of initial parameter matrix X_2 by 100. We then uniformly scale the initial values optimally as described in Algorithm 2. The number of outer iterations is set to 500, and the number of inner iterations is set to 10. This experiment is repeated fifty times.

The goal of these experiments are the following. First we want to study the impact of the balancing procedure in the meta-algorithm, compared to the same algorithm without explicit balancing. Since there are two ways to perform balancing for sNTD (the scalar optimal balancing and the heuristic Sinkhorn-like balancing respectively described in Sections 4.3.3 and 4.3.4), we compare both balancing strategies against the Meta-algorithm without balancing. We observe the performance of the three strategies in terms of loss value at the last iteration and the sparsity level of the core tensor $\mathcal{G}$ in percentage of the total number of entries. We also monitor the sparisty on the slices of $\mathcal{G}$ along the first mode, to check if the true rank is detected by the algorithm. An entry is considered null in the core tensor if it is below the threshold 2ϵ.

Results are shown in Figure 6 and 7. We validate several observations from the theoretical derivations:

- For small values of μ, as predicted in section 3.3, the meta-algorithm without explicit balancing dramatically struggles to converge in terms of loss value compared to the meta-aglorithm with balancing. The scalar approach for balancing is similarly efficient at scaling the parameter matrices as the Sinkhorn approach, although the best FMS for the scalar scaling is slightly worse. This phenomenon is observed despite the scaling of the first column of the initial guess for matrix X_2, which we expected to give the upper edge to the Sinkhorn procedure. Moreover the scalar balancing is both easier to implement and faster to compute than the Sinkhorn balancing. It is therefore a good compromise between no balancing and the Sinkhorn balancing.

- The sparsity level for the core tensor in balanced methods behaves more smoothly with respect to the regularization hyperparameter than in the unbalanced method. Since all algorithms should converge to similar solutions, this means that the solution provided by the unbalanced method are less sparse than they should be. Also the true number of components on the first mode is not detected by the unbalanced method, but the balanced methods provide a reasonable pruning even though the model is not explicitly written to minimize the multilinear ranks like in the rCPD showcase.

4.4.2 sNTD for Music Segmentation

Finally we also showcase the proposed sNTD model with and without balancing in the Meta Algorithm on a music redundancy detection task. The idea of using NTD for analysing audio signals has been proposed in [59] and developed further in [43].

Given an audio recording of a song segmented into bars, a tensor spectrogramm is built by stacking time-frequency spectrogramms computed on each bar of the audio signal. The data is therefore a tensor spectrogramm $\boldsymbol{\mathcal{T}}$ of sizes $F \times T \times B$ with F the number of frequency bins and T the number of time bins, and B the number of bars in the song. By computing the NTD on the tensor spectrogramm, one can detect redundancies across the bars, and factorize the spectral and short-term temporal content into a collection of patterns. The information on the redundancy is contained in particular in the last mode parameter matrix X_3, which can be used for downstream tasks such as automatic segmentation.

Each column of matrix X_3 refers to a specific pattern (defined by the other parameters X_1, X_2 and $\mathcal{G}$, while each row maps the patterns to the bar index. In other words, we can visualize in matrix X_3 the matching

Figure 6: Results for the simulated experiment on sparse NTD. The normalized loss after the stopping criterion is reached is shown with respect to the regularization parameters on the left plot, while the factor match score is shown in the right plot.

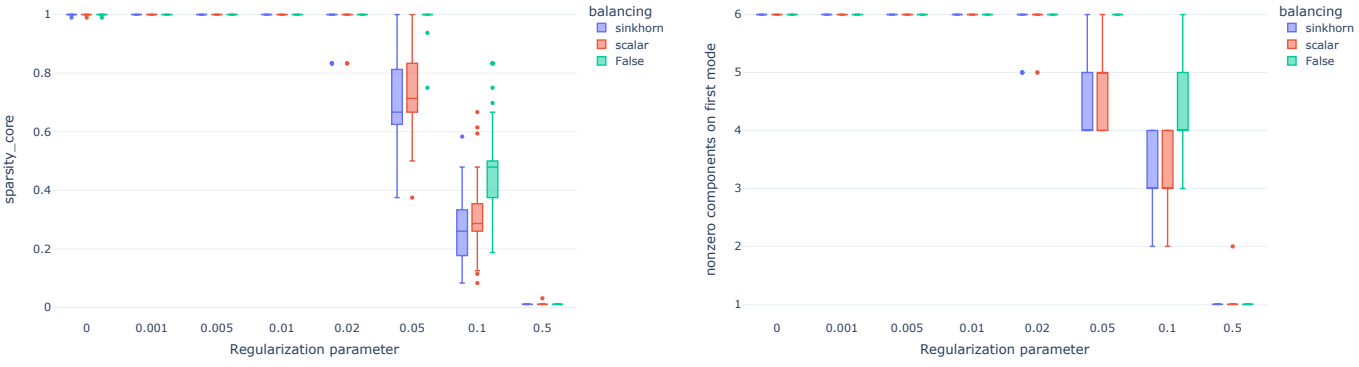

Figure 7: Results for the simulated experiment on sparse NTD. The scaled sparsity of the estimated core tensor is shown with respect to the regularization parameters on the left plot, while the number of nonzero components along the first mode of the estimated core is shown in the right plot.

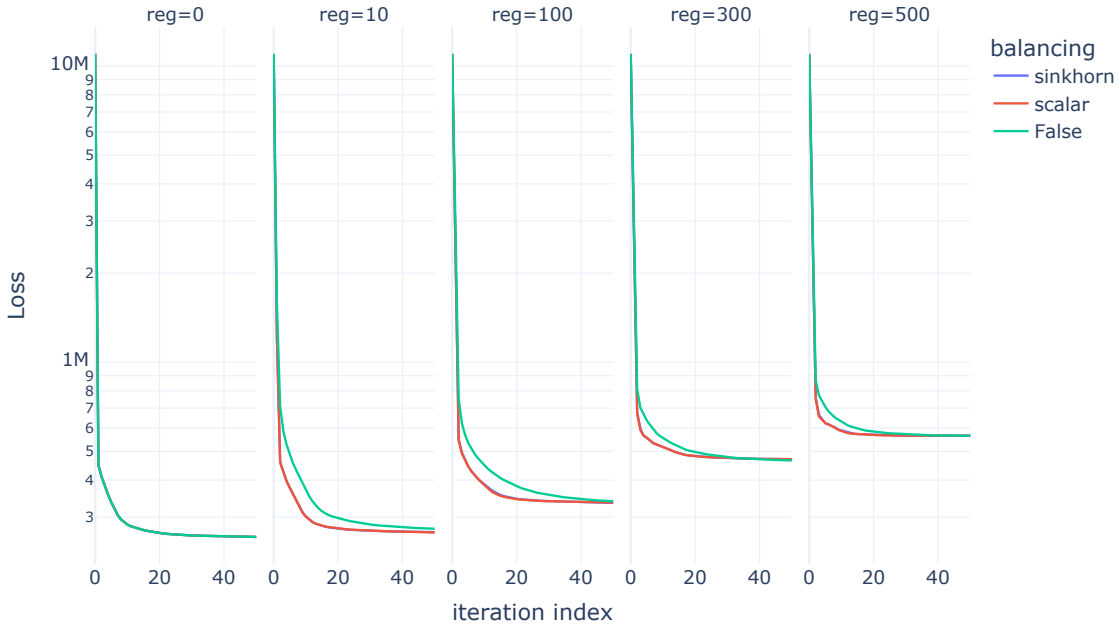

Figure 8: Results for the sNTD experiment on music redundancy detection. The loss function for each tested algorithm is shown across iterations for various values of the regularization hyperparameters. The blue and red curves are almost superposed.

between patterns and bars, and it is expected that bars with similar content will use the same patterns. Matrix X_3 should be sparse since only a few patterns are used for each bar.

Therefore, we propose here to test the sNTD with sparsity imposed on parameter matrix X_3 instead of $\mathcal{G}$ on such a task. The derivations of this variant of sNTD are omitted but straightforward. The song used for this experiment is Come Together by the Beatles, which has been shown in previous works to showcase the NTD ability to detect patterns and redundancies in music. After processing the song the same manner as in [43], we obtain a tensor of sizes $F = 80$, $T = 96$ and $B = 89$. The dimensions chosen for the core are $32 \times 12 \times 10$, which implies that the song should be factorized in at most 10 patterns. The algorithm is ran for 50 iterations with 10 inner iterations. The regularization parameter is chosen on a grid $[0, 10, 100, 300, 500]$. Initialization is carried out by performing HOSVD [14].

Figures 8 and 9 respectively show the loss function for three variants of the sNTD algorithm (no balancing, scalar balancing, Sinkhorn balancing) and the estimated parameter matrices X_3^T in each setup. An expert, ground truth segmentation of the song is represented by green vertical lines. One may observe again that the balanced algorithms converge faster than the unbalanced algorithm. Furthermore, the sparsity applied on X_3 has a clear effect on the quality of the redundancy detection. In particular for a regularization parameter of three hundred, the indices of patterns used in each bar matches well with the expert segmentation of the song There are little noticeable difference between the three variants, suggesting again that the lack of convergence of the cost was due only to scaling issues that do not affect the model interpretability.

5 Conclusions and perspectives

The design of efficient algorithms for computing low-rank approximations is an active research field, in particular in the presence of constraints and penalizations. In this work, we have shed light on the hidden effects of regularizations in nonnegative low-rank approximations, by studying a broader class of problems coined Homogeneous Regularized Scale Invariant problems. We have shown that the scale-ambiguity inherent to these problems induces a balancing effect of penalizations imposed on several parameter matrices. This balancing effect has both practical and theoretical implications. In theory, implicit balancing induces implicit regularizations effects. For instance imposing ridge penalizations on all matrices in a low-rank approximation is equivalent to using a sparsity-inducing penalization on the number of components of the model. In practice, enforcing this implicit balancing explicitly helps alternating algorithms efficiently minimizing the cost function. This is demonstrated formally on a toy problem, and practically in three different showcases, namely sparse Nonnegative Matrix Factorization, ridge Canonical Polyadic Decomposition and sparse Nonnegative Tucker

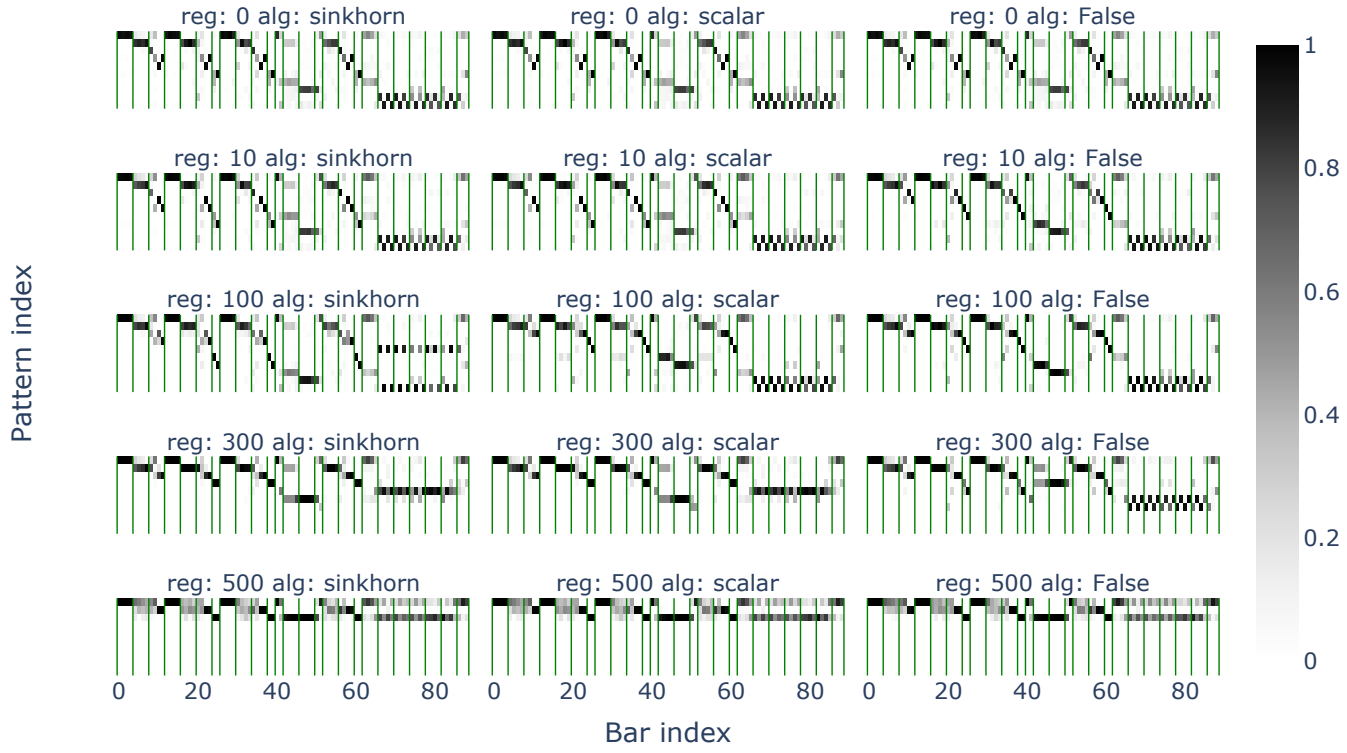

Figure 9: Results for the sNTD experiment on music redundancy detection. The reconstructed matrix X_3 is shown, transposed, for each tested algorithm and various values of the regularization hyperparameter. Green vertical bars indicate an expert segmentation of the song (Come-Together). The number of patterns (rows) is hard to estimated *a priori*, therefore ideally the estimated matrix X_3 should have sparse rows to prune unnecessary patterns. Moreover, it should be group-sparse between the green frontiers to allow for an easier segmentation by downstream methods such as dynamic programming or K-means.

Decomposition.

There are many perspectives to our findings. First, one could hope to make use of the implicit equivalent formulation of regularized low-rank approximations to derive guarantees on the solution space of these models. For instance, in general it is not clear when ℓ_1 regularized Nonnegative Matrix Factorization has zero, sparse or dense solutions. In contrast, for the LASSO problem it is known, and simple to show, that solutions are always sparse [21]. More practically, it should prove interesting to derive optimization algorithms for the implicit problem rather than the explicit problem, since it is fully scale-invariant. This has been probed in a recent work in the context of Nonnegative Matrix Factorization, with promising results [41]. Finally, our framework is restricted to homogeneous regularizations applied columnwise on parameter matrices of scale-invariant models. Positive homogeneity allows to study a large class of regularization including ℓ_p norms, but some regularization and constraints used in the literature are not positively homogeneous with respect to the columns of parameter matrices. It could be of interest to study how the proposed methodology generalizes to these problems.

Acknowledgements

The authors would like to thank Rémi Gribonval for an early assesment of this work and several pointers towards relevant literature in neural networks training. The authors also thank Cédric Févotte and Henrique De Morais Goulart for helpful discussions around the implicit and explicit formulations, as well as pointers towards relevant literature in sparse NMF. Jeremy Cohen is supported by ANR JCJC LoRAiA ANR-20-CE23-0010.

References

[1] Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 2011.

[2] Evrim Acar, Yuri Levin-Schwartz, Vince D Calhoun, and Tülay Adali. ACMTF for fusion of multi-modal neuroimaging data and identification of biomarkers. In *IEEE 25th European Signal Processing Conference*, pages 643–647, 2017.

[3] Boian Alexandrov, Derek F. DeSantis, Gianmarco Manzini, and Erik W. Skau. Nonnegative canonical tensor decomposition with linear constraints: nnCANDELINC. *Numerical Linear Algebra with Applications*, 29(6), December 2022.

[4] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[5] Alexis Benichoux, Emmanuel Vincent, and Remi Gribonval. A fundamental pitfall in blind deconvolution with sparse and shift-invariant priors. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6108–6112, 2013.

[6] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[7] Małgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J. Candès. SLOPE—adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103–1140, 2015.

[8] Claire Boyer, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric De Gournay, and Pierre Weiss. On Representer Theorems and Convex Regularization. *SIAM Journal on Optimization*, 29(2):1260–1281, 2019.

[9] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. *Csiszár's Divergences for Non-negative Matrix Factorization: Family of New Algorithms*, volume 3889. 2006.

[10] Jérémy E Cohen. Régularisation implicite des factorisations de faible rang pénalisées. *GRETSI 2023, Grenoble*, 2023.

[11] Jérémy E. Cohen and Nicolas Gillis. Identifiability of complete dictionary learning. *SIAM Journal on Mathematics of Data Science*, 1(3):518–536, 2019.

[12] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[13] I. Csiszar. i-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

[14] Lieven De Lathauwer and Joos Vandewalle. Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_n)$ reduction in multilinear algebra. *Linear Algebra and its Applications*, 391:31–55, 2004.

[15] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[16] Julian Eggert and Edgar Korner. Sparse coding and NMF. In *2004 IEEE International Joint Conference on Neural Networks*, volume 4, pages 2529–2533. IEEE, 2004.

[17] Tolga Ergen and Mert Pilanci. Implicit Convex Regularizers of CNN Architectures: Convex Optimization of Two- and Three-Layer Networks in Polynomial Time. 2020.

[18] Mike Espig and Aram Khachatryan. Convergence of Alternating Least Squares Optimisation for Rank-One Approximation to High Order Tensors, March 2015.

[19] C. Févotte, N. Bertin, and J-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[20] Cédric Févotte and Jérome Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9):2421–2456, 2011.

[21] Simon Foucart and Holger Rauhut. In *A mathematical introduction to compressive sensing*. Applied and Numeric Harmonic Analysis, Birkhauser/Springer, 2013. ISBN 978-0-8176-4948-7.

[22] Nicolas Gillis. *Nonnegative Matrix Factorization*. SIAM, Philadelphia, 2020.

[23] Nicolas Gillis and Francois Glineur. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *Neural computation*, 24(4):1085–1105, 2012.

[24] Rémi Gribonval, Theo Mary, and Elisa Riccietti. Scaling is all you need: quantization of butterfly matrix products via optimal rank-one quantization. In *29ème Colloque sur le traitement du signal et des images (GRETSI), Aug 2023, Grenoble, France.*, pages pp. 497–500, 2023.

[25] X. Han, Laurent Albera, A. Kachenoura, H. Shu, and L. Senhadji. Robust multilinear decomposition of low rank tensors. In *14th International Conference on Latent Variable Analysis and Signal Separation, LVA/ICA 2018*, volume 10891 LNCS, page 3. Springer Verlag, July 2018.

[26] Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA working papers in phonetics*, 16, 1970.

[27] L. T. K. Hien and N. Gillis. Algorithms for nonnegative matrix factorization with the kullback-leibler divergence. *Journal of Scientific Computing*, (87):93, 2021.

[28] Christopher J. Hillar and Lek-Heng Lim. Most Tensor Problems Are NP-Hard. *Journal of the ACM (JACM)*, 60(6), 2013.

[29] Patrik O. Hoyer. Non-negative sparse coding. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565. IEEE, 2002.

[30] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[31] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 05 2007.

[32] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[33] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.

[34] B. Lamond and N.F. Stewart. Bregman's balancing method. *Transportation Research Part B: Methodological*, 15(4):239–248, 1981.

[35] Lieven De Lathauwer, Baart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Ana. Appl.*, 21(4):1253–1278, 2000.

[36] Felix J. Weninger Le Roux, Jonathan and John R. Hershey. Sparse NMF–half-baked or well done? *Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Tech. Rep., no. TR2015-023*, 11:13–15, 2015.

[37] Daniel D. Lee and H. Sebastien Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[38] Valentin Leplat. *Nonnegative Matrix Factorization: Models, Optimization Problems, Algorithms and Applications*. Phd thesis, University of Mons, Januray 2021.

[39] Valentin Leplat, Nicolas Gillis, and Jérôme Idier. Multiplicative updates for nmf with beta-divergences under disjoint equality constraints. *SIAM Journal on Matrix Analysis and Applications*, 42(2):730–752, 2021.

[40] Lek-Heng Lim and Pierre Comon. Nonnegative approximations of nonnegative tensors. *Journal of chemometrics*, 23(7-8):432–441, 2009.

[41] Arthur Marmin, José Henrique de Morais Goulart, and Cédric Févotte. Majorization-Minimization for Sparse Nonnegative Matrix Factorization With the β-Divergence. *IEEE Transactions on Signal Processing*, 71:1435–1447, January 2023.

[42] Arthur Marmin, José Henrique de Morais Goulart, and Cédric Févotte. Joint Majorization-Minimization for Nonnegative Matrix Factorization with the β-divergence. *Signal Processing*, 209:109048, 2023.

[43] Axel Marmoret, Jérémy E. Cohen, Nancy Bertin, and Frédéric Bimbot. Uncovering audio patterns in music with nonnegative Tucker decomposition for structural segmentation. In *ISMIR 2020-21st International Society for Music Information Retrieval*, 2020.

[44] Axel Marmoret, Florian Voorwinden, Valentin Leplat, Jérémy E. Cohen, and Frédéric Bimbot. Nonnegative tucker decomposition with beta-divergence for music structure analysis of audio signals, 2021.

[45] Mathurin Massias, Alexandre Gramfort, and Joseph Salmon. Celer: a fast solver for the lasso with dual extrapolation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3321–3330, 2018.

[46] Bacharach Michael. *Biproportional Matrices and Input-Output Change*. Cambridge University Press, 1970.

[47] Morten Mørup and Lars Kai Hansen. Automatic relevance determination for multi-way models. *Journal of Chemometrics*, 23(7-8):352–363, 2009.

[48] Morten Mørup, Lars Kai Hansen, and Sidse M Arnfred. Algorithms for sparse nonnegative Tucker decompositions. *Neural computation*, 20(8):2112–2131, 2008.

[49] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

[50] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Workshop Track Proceedings*, 2015.

[51] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[52] Nicholas D. Sidiropoulos Papalexakis, Evangelos E. and Rasmus Bro. From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. *IEEE transactions on signal processing*, 61(2):493–506, 2013.

[53] Rahul Parhi and Robert D. Nowak. Deep Learning Meets Sparse Regularization: A signal processing perspective. *IEEE Signal Processing Magazine*, 40(6):63–74, September 2023.

[54] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[55] Edgar Rechtschaffen. Real roots of cubics: explicit formula for quasi-solutions. *The Mathematical Gazette*, (524):268–276, 2008.

[56] Marie Roald, Carla Schenker, Vince D. Calhoun, Tülay Adalı, Rasmus Bro, Jeremy E. Cohen, and Evrim Acar. An AO-ADMM approach to constraining PARAFAC2 on all modes. *SIAM Journal on Mathematics of Data Science*, 4(3):1191–1222, 2022.

[57] Roberto Rocci and Jos M. F. Ten Berge. Transforming three-way arrays to maximal simplicity. *Psychometrika*, 67(3):351–365, September 2002.

[58] Richard Dennis Sinkhorn and Paul Joseph Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 27(2):343–348, 1967.

[59] Jordan B. L. Smith and Masataka Goto. Nonnegative Tensor Factorization for Source Separation of Loops in Audio. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175, 2018.

[60] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum-Margin Matrix Factorization. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[61] Alwin Stegeman and Pierre Comon. Subtracting a best rank-1 approximation may increase tensor rank. *Linear Algebra and its Applications*, 433(7):1276–1300, December 2010.

[62] Pierre Stock, Benjamin Graham, Rémi Gribonval, and Hervé Jégou. Equi-normalization of neural networks. In *International Conference on Learning Representations*, 2019.

[63] Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Tensor balancing on statistical manifold. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3270–3279. PMLR, 06–11 Aug 2017.

[64] Yuchen Sun and Kejun Huang. Volume-Regularized Nonnegative Tucker Decomposition with Identifiability Guarantees. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023. IEEE.

[65] T. Takahashi and H. Ryota. Global convergence of modified multiplicative updates for nonnegative matrix factorization. *Computational Optimization and Applications*, (57):417–440, 2014.

[66] Vincent Y.F. Tan and Cédric Févotte. Automatic Relevance Determination in Nonnegative Matrix Factorization with the /spl beta/-Divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, July 2013.

[67] Fabian J Theis, Kurt Stadlthanner, and Toshihisa Tanaka. First results on uniqueness of sparse nonnegative matrix factorization. *2005 13th European Signal Processing Conference*, pages 1–4, 2005.

[68] Robert Tibshirani. Regression Shrinkage and Selection via The Lasso: A Retrospective. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(3):273–282, 2011.

[69] Ryan J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013.

[70] Ryan J Tibshirani. Equivalences between sparse models and neural networks. *Working Notes. URL https://www. stat. cmu. edu/ryantibs/papers/sparsitynn. pdf*, 2021.

[71] Michael Unser. A Unifying Representer Theorem for Inverse Problems and Machine Learning. *Foundations of Computational Mathematics*, 21(4):941–960, 2021.

[72] André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM J. matrix Analysis*, 33(2):639–652, 2012.

[73] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[74] Guoxu Zhou, Andrzej Cichocki, Qibin Zhao, and Shengli Xie. Efficient Nonnegative Tucker Decompositions: Algorithms and Uniqueness. *IEEE Transactions on Image Processing*, 24(12):4990–5003, December 2015.

A Proof of optimality for sparse unbalanced matrix factorization

First, let us consider without loss of generality the case $n = 2$ for Problem (5):

$$\phi(X_1, X_2) = f(X_1, X_2) + \mu_1 g(X_1) \tag{69}$$

Function ϕ is bounded below by zero because f is nonnegative, therefore it must admit an infimum over $\mathbb{R}^{m_1 \times r} \times \mathbb{R}^{m_2 \times r}$. Let us denote by q this infimum.

We are going to show that, given $l := \inf_{X_1, X_2} f(X_1, X_2)$, it holds that $q = l$. We know that $q \geq l$ since both g and μ_1 are nonnegative. To prove the converse inequality, note that from Equation (69), for any $\lambda < 1$, it holds that $\phi(X_1, X_2) > \phi(\lambda X_1, \frac{1}{\lambda} X_2)$. Therefore for any admissible (X_1, X_2),

$$q \leq \lim_{\lambda \to 0} \phi(\lambda X_1, \frac{1}{\lambda} X_2). \tag{70}$$

Now we may notice that $\phi(\lambda X_1, \frac{1}{\lambda} X_2) = f(X_1, X_2) + o(\lambda)$, and therefore prolonging by continuity the right hand side, we get that $\lim_{\lambda \to 0} \phi(\lambda X_1, \frac{1}{\lambda} X_2) = f(X_1, X_2)$. This shows that $q \leq l$, which concludes our proof that $q = l$.

One can easily note that for the infimum q to be attained, say at (X_1^*, X_2^*), one must have $\phi(X_1^*, X_2^*) = f(X_1^*, X_2^*)$ and therefore $g(X_1^*) = 0$, *i.e.* $X_1^* = 0$ and $q = f(0, X_2^*)$. By contraposition, as soon as $f(0, X_2^*)$ is not the infimum of f on the set of matrices with rank at most r, the infimum q cannot be attained.

B Proof of implicit regularization equivalences

We briefly summarize how to derive the implicit regularizations in Table 2. Two case are to be distinguished. Either the explicit regularization are columnwise separable, in which case we use the results of Section 2.2 to derive the implicit regularization, or the regularizations are not columnwise separable and we apply the same result to the whole matrix, as detailed for scalar NTD balancing in Section 4.3.3.

For instance to derive the result for ℓ_1^2 plus ℓ_2^2, consider the following cost function

$$\inf_{X_1 \in \mathbb{R}^{m_1 \times r}, X_2 \in \mathbb{R}^{m_2 \times r}} f(X_1, X_2) + \mu_1 \sum_{q \leq r} \|X_1[:, q]\|_1^2 + \mu_2 \sum_{q \leq r} \|Y[:, q]]\|_2^2 \ . \tag{71}$$

Using the scale optimality result from Section 2.2, minimizing (71) partially with respect to the scale of the q-th column of X_1 and X_2, we know that the penalizations terms with respect to the q-th column will be equal to $\beta_q(X_1, X_2)/2$ where

$$\beta_q(X_1, X_2) = \left(2(\mu_1 \mu_2)^{1/2} \|X_1[:, q]\|_1 \|X_2[:, q]\|_2\right) \ . \tag{72}$$

Moreover, the product of ℓ_1 and ℓ_2 norms further simplifies if one introduces $L_q = X_1[:, q] \otimes X_2[:, q]$:

$$\|X_1[:, q]\|_1 \|X_2[:, q]\|_2 = \left(\sum_i |X_1[i, q]|\right) \left(\sqrt{\sum_k X_2[k, q]^2}\right) = \sum_i \sqrt{\sum_k X_1[i, q]^2 X_2[k, q]^2} = \sum_i \sqrt{\sum_k L_q[i, k]^2} \tag{73}$$

we recognize the mixed $\ell_{1,2}$ norm of matrix L_q.

Other equivalences are derived in the same fashion.

C Closed-form expression to solutions of the balancing procedure

We are interested in computing the solution to the following problem:

$$\min_{\forall i \leq n,\, \lambda_i} \sum_{i=1}^{n} \lambda_i^{p_i} a_i \text{ such that } \prod_{i=1}^{n} \lambda_i = 1 \tag{74}$$

Note that we dropped the nonnegativity constraints which are not required to reach our closed-form solution. A particular solution will be naturally nonnegative as long as all products $p_i a_i$ are nonnegative. We also suppose that each constant a_i is strictly positive (see Section 2.1 for a justification).

We use the KKT conditions to compute the solutions, as is traditionally done for the related variational formulation of the geometric mean. Indeed,

$$\operatorname*{argmin}_{\{\lambda_i\}_{i \leq n}} \sum_{i=1}^{n} \lambda_i \text{ such that } \prod_{i=1}^{n} \lambda_i = \prod_{i=1}^{n} x_i \tag{75}$$

is an alternative definition of the geometric mean $\beta = \left(\prod_{i \leq n} x_i\right)^{\frac{1}{n}}$.

Let us define the Lagrangian for our problem,

$$\mathcal{L}(\{\lambda_i\}_{i \leq n}, \nu) = \sum_{i \leq n} \lambda_i^{p_i} a_i + \nu(1 - \prod_{i \leq n} \lambda_i) \ . \tag{76}$$

Optimality conditions can be written as follows:

$$\begin{cases} \nabla_{\lambda_i} \mathcal{L}(\{\lambda_i^*\}_{i \leq n}, \nu^*) = p_i a_i \lambda_i^{p_i - 1} - \nu^* \prod_{j \neq i} \lambda_j^* = 0 \\ 1 = \prod_{i \leq n} \lambda_i^* \end{cases} \tag{77}$$

Using the second equality, we can inject λ_i in the right hand side of the first equation, which yields the following necessary condition

$$\lambda_i = \left(\frac{\nu^*}{p_i a_i}\right)^{\frac{1}{p_i}} \ . \tag{78}$$

There may be a sign ambiguity depending on the power value p_i, in which case we can chose the positive solution since the original problem involved nonnegativity constraints on λ_i. Moreover, we have used the facts that $p_i a_i$ are nonzero, and that λ_i^* are also nonzero. This last condition can easily be checked since the constraint $\prod_{i \leq n} \lambda_i^* = 1$ prevents any optimal λ_i^* to be null.

To compute the optimal dual parameter, we can use again the primal feasibility condition:

$$\prod_{i \leq n} \left(\frac{\nu^*}{p_i a_i}\right)^{\frac{1}{p_i}} = 1 \tag{79}$$

which, after some simple computation, shows that ν^* is exactly the weighted geometric mean of products $p_i a_i$ with weights $\frac{1}{p_i}$,

$$\nu^* = \left(\prod_{i \leq n} (p_i a_i)^{\frac{1}{p_i}}\right)^{\frac{1}{\sum_{i \leq n} \frac{1}{p_i}}} \tag{80}$$

Note that when applying this balancing operation to an optimal solution X_i^* of HRSI, the scales λ_i must equal one, therefore at optimality,

$$p_i a_i = \nu^* \tag{81}$$

If the solution X_i^* is null, then ν^* is also null and the result holds.

D Proof of sublinear convergence rate for ALS on the scaling problem

We will study the variation of loss function when x_1 is updated, x_2 fixed, across one iteration. The problen is symmetric therefore we will generalize for all updates of either x_1 or x_2.

Let us introduce a few notations. Let k the iteration index, we set $x_i := x_i^{(k)}, \tilde{x}_1 = x_1^{(k+1)}$. We denote the residuals $r_k = y - x_1 x_2$ and $r_{k+1} = y - \tilde{x}_1 x_2$, and the regularizations $g_k = \lambda(x_1^2 + x_2^2)$ and $g_{k+1} = \lambda(\tilde{x}_1^2 + x_2^2)$.

The cost function value at iteration k is $f_k = r_k^2 + g_k$. Finally, we denote $\delta r_k = r_k - r_{k+1}$ the variation in residuals. Throughout the following derivations, we suppose that λ is much smaller than y, but also x_1^2 and x_2^2. Since both x_1^2 and x_2^2 will eventually converge towards $y - \lambda$, this is equivalent to assuming that λ is much smaller than y, and k is large enough so that x_1 and x_2 are relatively close to $\sqrt{y-\lambda}$.

We study the variation of the cost $\delta f_k = f_k - f_{k+1}$. We observe that

$$f_{k+1} = (-\delta r_k + r_k)^2 + g_{k+1} - g_k + g_k \tag{82}$$
$$= f_k - 2r_k\delta r_k + (\delta r_k)^2 + (g_{k+1} - g_k), \tag{83}$$

and therefore $\delta f_k = 2r_k\delta r_k - (\delta r_k)^2 + (g_k - g_{k+1})$. We can expend the update of the regularization terms:

$$g_k - g_{k+1} = \lambda\left(x_1^2 - \tilde{x}_1^2\right) \tag{84}$$
$$= \frac{\lambda}{x_2^2}\left((y^2 - \tilde{x}_1^2x_2^2) - (y^2 - x_1^2x_2^2)\right) \tag{85}$$
$$= \frac{\lambda}{x_2^2}\left(r_{k+1}(y + \tilde{x}_1x_2) - r_k(y + x_1x_2)\right) \tag{86}$$

Recal that $\tilde{x}_1 = \frac{y}{x_2}\frac{1}{1+\frac{\lambda}{x_2^2}}$[10]. We perform a first approximation here of $\tilde{x}_1(\lambda)$ by using its Taylor expansion around $\lambda = 0$, which hold approximately when $\frac{\lambda}{x_2^2} \ll 1$. Then we get

$$\tilde{x}_1 = \frac{y}{x_2}(1 - \frac{\lambda}{x_2^2} + \mathcal{O}(\lambda^2)) = \frac{y}{x_2} - \frac{y\lambda}{x_2^3} + \mathcal{O}(\lambda^2), \tag{87}$$

This means that

$$y + \tilde{x}_1x_2 = 2y - \frac{y\lambda}{x_2^2} + \mathcal{O}(\lambda^2). \tag{88}$$

Furthermore, under the same hypothesis, we can easily see that the residuals is of order of magnitude λ:

$$r_{k+1} = y - \tilde{x}_1x_2 \tag{89}$$
$$= \frac{y\lambda}{x_2^2} + \mathcal{O}(\lambda^2) \tag{90}$$

and therefore, we can relate the sums $y + x_1x_2$ and $y + \tilde{x}_1x_2$ to the residuals as

$$y + \tilde{x}_1x_2 = 2y - r_{k+1} + \mathcal{O}(\lambda^2) \tag{91}$$
$$y + x_1x_2 = 2y - r_k + \mathcal{O}(\lambda^2)\ . \tag{92}$$

This yields the following formula for the variation of regularization terms

$$g_k - g_{k+1} = \frac{\lambda}{x_2^2}\left(2y(r_{k+1} - r_k) + (r_k^2 - r_{k+1}^2) + \mathcal{O}(\lambda^3)\right) \tag{93}$$
$$= -2r_{k+1}\delta r_k + \mathcal{O}(\lambda^3)\ . \tag{94}$$

We have assumed that δr_k is also of the order of magnitude λ, which is shown in the derivations below. Finally, the variation in loss function can be written as

$$\delta f_k = -2r_{k+1}\delta r_k + 2r_k\delta r_k - (\delta r_k)^2 + \mathcal{O}(\lambda^3) \tag{95}$$
$$= 2\delta r_k(r_k - r_{k+1}) - (\delta r_k)^2 + \mathcal{O}(\lambda^3) \tag{96}$$
$$= (\delta r_k)^2 + \mathcal{O}(\lambda^3) \tag{97}$$

To further detail the expression of δr_k, we may first relate $\tilde{x}_1$ with x_1 since

$$\tilde{x}_1 = \frac{y}{x_2}\frac{x_2^2}{x_2^2 + \lambda} = \frac{y}{x_2}\left(1 - \frac{\lambda}{x_2^2 + \lambda}\right). \tag{98}$$

[10] ALS algorithm updates each variable as follows $\tilde{x}_1 = \underset{x_1}{\text{argmin}}\,(y - x_1x_2)^2 + \lambda(x_1^2 + x_2^2)$, solving the KKT condition leads to $\tilde{x}_1 = \frac{yx_2}{\lambda + x_2^2} = \frac{y}{x_2}\frac{1}{1+\frac{\lambda}{x_2^2}}$.

Viewing x_2 as the least squares update from x_1, we further obtain

$$\tilde{x}_1 = x_1 \frac{1 - \frac{\lambda}{x_2^2+\lambda}}{1 - \frac{\lambda}{x_1^2+\lambda}} \tag{99}$$

Therefore we get

$$\delta r_k^2 = x_2^2(x_1 - \tilde{x}_1)^2 \tag{100}$$

$$= x_1^2 x_2^2 \left(1 - \frac{1 - \frac{\lambda}{x_1^2-\lambda}}{1 - \frac{\lambda}{x_2^2+\lambda}}\right)^2 \tag{101}$$

$$= x_1^2 x_2^2 \lambda^2 \left(\frac{\frac{1}{x_2^2+\lambda} - \frac{1}{x_1^2+\lambda}}{1 - \frac{\lambda}{x_2^2+\lambda}}\right)^2 \tag{102}$$

The denominator can be neglected since it will generate terms in the order of λ^4 as soon as λ is small with respect to x_2^2, which is our working hypothesis. This yields

$$\delta r_k^2 = x_1^2 x_2^2 \lambda^2 \left(\frac{x_1^2 - x_2^2}{(x_1^2+\lambda)(x_2^2+\lambda)}\right)^2 + \mathcal{O}(\lambda^3) \tag{103}$$

$$= \frac{x_1^2 x_2^2 (x_1 + x_2)^2}{(x_1^2+\lambda)^2(x_2^2+\lambda)^2} \lambda^2 (x_1 - x_2)^2 + \mathcal{O}(\lambda^3) \tag{104}$$

Let us introduce the errors e_1, e_2 from x_1, x_2 to their target $y - \lambda$:

$$x_1 = \sqrt{y - \lambda} + e_1 \tag{105}$$

$$x_2 = \sqrt{y - \lambda} + e_2 \tag{106}$$

We may note that e_1 and e_2 are of opposite sign as soon as $k > 1$, and for simplicity of presentation, we will assume that $e_1 \approx -e_2 = e_k$. We further assume that k is large enough so that e_1, e_2 are small enough (of order of magnitude $\mathcal{O}(\lambda)$) to approximate $(x_1 + x_2)^2$ as $4y$ and $(x_1 + \lambda)^2, (x_2 + \lambda)^2$ as y up to an error $\mathcal{O}(\lambda)$. The term $x_1 - x_2$ is of order $\mathcal{O}(\lambda)$. Therefore for large enough k,

$$\delta f_k = y^2 \frac{4y}{y^2 y^2} \lambda^2 (x_1 - x_2)^2 + \mathcal{O}(\lambda^3) \tag{107}$$

$$= 4\frac{\lambda^2}{y}(x_1 - x_2)^2 + \mathcal{O}(\lambda^3) \tag{108}$$

$$= 16\frac{\lambda^2}{y} e_k^2 + \mathcal{O}(\lambda^3) \tag{109}$$

We are left to characterize the convergence speed of e_k towards zero. We can see that e_k converges linearly with however a multiplicative constant that goes to one as $\frac{\lambda}{y}$ grows smaller. Indeed,

$$e_{k+1} = \tilde{x}_1 - x^* \tag{110}$$

$$= x_1 - x^* - x_1 \left(1 - \frac{1 - \frac{\lambda}{x_2^2+\lambda}}{1 - \frac{\lambda}{x_1^2+\lambda}}\right) \tag{111}$$

$$= x_1 - x^* - x_1 \lambda \left(\frac{x_1^2 - x_2^2}{(x_1^2+\lambda)(x_2^2+\lambda)}\right) + \mathcal{O}(\lambda^2) \tag{112}$$

$$= x_1 - x^* - \frac{2\lambda}{y}(x_1 - x_2) + \mathcal{O}(\lambda^2) \tag{113}$$

where the last two lines are obtained by reusing the previous derivations for δr_k. We find that

$$e_{k+1} = e_k - e_k 4\frac{\lambda}{y} + \mathcal{O}(\lambda^2) \tag{114}$$

We see that $\frac{e_{k+1}}{e_k}$ is approximately constant, equal to $1 - 4\frac{\lambda}{y}$, which proves that ALS is a sublinear algorithm when $\frac{\lambda}{y}$ goes to zero.

E Convergence of iterates for Algorithm 2

To better analyze the convergence of the iterates generated by Algorithm 2, it is important to review some key concepts presented in [54]:

- *Algebraic structure and decomposition by blocks*: In this context, $\mathbb{R}^m$ represents a space of m-dimensional real valued vectors, which can also be represented as the product of s smaller real valued vector spaces. This can be written as:
$$\mathbb{R}^m = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times ... \times \mathbb{R}^{m_s} \tag{115}$$
where $\sum_i^s m_i = m$. In Section 3.4, $\mathcal{Z}$ was used to represent the feasible set, which is the product of s closed convex sets: $\mathcal{Z} = \mathcal{Z}_1 \times ... \times \mathcal{Z}_s$, with $\mathcal{Z}_i \subseteq \mathbb{R}^{m_i}$. Additionally, the optimization variable $z \in \mathcal{Z}$ is decomposed as $z = (z_1, z_2, ..., z_s)$, where $z_i \in \mathcal{Z}_i$ represents the i-th block of variables.

- *Directional derivative:* For a function $\phi : \mathcal{Z} \to \mathbb{R}$, where $\mathcal{Z} \subseteq \mathbb{R}^m$ is a convex set, the directional derivative of ϕ at point z along a vector d is defined as:
$$\phi^{'}(z; d) := \lim_{\lambda \to 0} \frac{\phi(z + \lambda d) - \phi(z)}{\lambda} \tag{116}$$

- *Stationary points of a function:* For a function $\phi : \mathcal{Z} \to \mathbb{R}$, where $\mathcal{Z} \subseteq \mathbb{R}^m$ is a convex set, the point $z^\star$ is considered a stationary point of $\phi(.)$ if $\phi^{'}(z^\star; d) \geq 0$ for all d such that $z^\star + d \in \mathcal{Z}$.

- *Quasi-convex function:* A function ϕ is said to be quasi-convex if it satisfies the following inequality:
$$\phi(\theta x + (1 - \theta)y) \leq \max(\phi(x), \phi(y)) \quad \forall \theta \in (0, 1), \quad \forall x, y \in \text{dom}\phi \tag{117}$$

- *Regularity of a function at a point*: The function $\phi : \mathcal{Z} \to \mathbb{R}$ is regular at the point $z \in \text{dom}\phi$ w.r.t. coordinates $m_1, ..., m_s$ if $\phi^{'}(z; d) \geq 0$ for all $d = (d_1, ..., d_s)$ with $\phi^{'}(z; d_i^0) \geq 0$, where $d_i^0 := (0, ..., d_i, ..., 0)$ and $d_i \in \mathbb{R}^{m_i}$ for all i.

- *Coordinate-wise minimum of a function* $z \in \text{dom}\phi \in \mathbb{R}^m$ is the coordinate-wise minimum of ϕ w.r.t. the coordinates in $\mathbb{R}^{m_1}, ..., \mathbb{R}^{m_s}$ if
$$\phi(z + d_i^0) \geq \phi(z) \quad \forall d_i \in \mathbb{R}^{m_i} \quad \text{with } z + d_i^0 \in \text{dom}\phi \quad \forall i$$
or equivalently:
$$\phi^{'}(z; d_i^0) \geq 0 \quad \forall d_i \in \mathbb{R}^{m_i} \quad \text{with } z + d_i^0 \in \text{dom}\phi \quad \forall i$$

- *Block Successive MM:* Algorithm 2 updates only one block of variables at a time. Specifically, in iteration $k + 1$, the selected block (referred to as block i) is updated by solving the subproblem:
$$z_i^{(k+1)} = \underset{z_i \in \mathcal{Z}_i}{\text{argmin}} \ \bar{\phi}_i(z_i, z_1^{(k+1)}, ..., z_{i-1}^{(k+1)}, z_i^{(k),*}, ..., z_s^{(k),*}) \tag{118}$$
where $\bar{\phi}_i(., z)$ is a tight majorizer of $\phi(z)$ as defined in Definition 1.

To discuss the convergence of Algorithm 2, additional assumptions about the majorizers $\bar{\phi}_i(.,.)$ are necessary. These assumptions are as follows:

Assumption 2.

1. $\bar{\phi}_i^{'}(y_i, z; d_i)|_{y_i = z_i} = \phi^{'}(z; d_i^0) \quad \forall d_i^0 = (0, ..., d_i, ..., 0)$ *such that* $z_i + d_i \in \mathcal{Z}_i$, $\forall i$. *This assumption implies that the directional derivatives of the majorizers and the objective function coincide at the current iterate for each block of variables.*

2. $\bar{\phi}_i(y_i, z)$ *is continuous in* $(y_i, z) \quad \forall i$.

Following the approach outlined in [54], the convergence outcomes of our proposed Algorithm 2 are applicable in the asymptotic regime as $k \to \infty$. This convergence relies on the aforementioned assumptions, which include the directional differentiability and the continuity of the function $\phi(\cdot)$. In addition to this, the proof leverages two critical assumptions: firstly, the quasi-convexity of the majorizers $\bar{\phi}_i(y_i, z)$, and secondly, the uniqueness of their minimizers. Here-under, we give the proof of Theorem 2 enunciated in Section 3.5.

Proof. We follow the main of steps of the general proof of BSUM [54], and slightly adapt them to the specificity of our Algorithm. First, by Theorem 1, we have:

$$\phi(z^{(0)}) \geq \phi(z^{(1)}) \geq \phi(z^{(1),*}) \geq \phi(z^{(2)}) \geq \phi(z^{(2),*}) \geq \cdots \tag{119}$$

Consider a limit point z^{∞}, by combining Equation (119) and the continuity of $\phi(.)$, we have:

$$\lim_{k\to\infty} \phi(z^{(k)}) = \phi(z^{\infty}) = \lim_{k\to\infty} \phi(z^{(k),*}) \tag{120}$$

In the ensuing discussion, we focus on the subsequence denoted as $\{z^{(k_j),*}\}$, which converges to the limit point z^{∞}. Moreover, given the finite number of variable blocks, there must exist at least one block that undergoes infinite updates within the subsequences $\{k_j\}$. Without loss of generality, we assume block s is updated infinitely often. Consequently, by constraining our attention to these subsequences, we can express this as:

$$z_s^{(k_j+1)} = \underset{z_s\in\mathcal{Z}_s}{\operatorname{argmin}} \ \bar{\phi}_s(z_s, z_1^{(k_j+1)}, z_2^{(k_j+1)}, ..., z_s^{(k_j),*}) \tag{121}$$

To establish the convergence $z^{(k_j+1)} \to z^{\infty}$ as $j \to \infty$, or equivalently, to demonstrate $z_i^{(k_j+1)} \to z_i^{\infty}$ for all i as $j \to \infty$, we consider the block $i = 1$. Assume the contrary, supposing $z_1^{(k_j+1)}$ does not converge to z_1^{∞}. Consequently, by narrowing our focus to a subsequence, there exists $\bar{\gamma} > 0$ such that:

$$\bar{\gamma} \leq \gamma^{(k_j)} = \|z_1^{(k_j+1)} - z_1^{(k_j),*}\| \quad \forall k_j$$

This implies that as $j \to \infty$, $z_1^{(k_j+1)}$ consistently deviates from $z_1^{(k_j),*}$, which converges to z_1^{∞}. Consequently, by transitivity, $z_1^{(k_j+1)}$ does not converge to z_1^{∞}.

We introduce the sequence $s^{(k_j)}$, defined as the normalized difference between $z_1^{(k_j+1)}$ and $z_1^{(k_j),*}$:

$$s^{(k_j)} := \frac{z_1^{(k_j+1)} - z_1^{(k_j),*}}{\gamma^{(k_j)}} \tag{122}$$

Notably, we observe that $\|s^{(k_j)}\| = 1$ holds for all $\{k_j\}$; thus, $s^{(k_j)}$ is a member of a compact set and consequently possesses a limit point denoted as $\bar{s}$. By further narrowing our focus to a subsequence that converges to $\bar{s}$, we can express the ensuing set of inequalities:

$$\phi(z^{(k_{j+1}),*}) \leq \phi(z^{(k_{j+1})}) \leq \phi(z^{(k_j+1),*}) \leq \phi(z^{(k_j+1)}) \tag{123}$$

which hold thanks to Equation 119 (consequence of Theorem 1) and given that $k_{j+1} \geq k_j + 1$. Moreover :

$$\begin{aligned}
\phi(z^{(k_j+1)}) &\leq \bar{\phi}_1(z_1^{(k_j+1)}, z^{(k_j),*}) \quad \text{(Using Definition 1)} \\
&= \bar{\phi}_1(z_1^{(k_j),*} + \gamma^{(k_j)} s^{(k_j)}, z^{(k_j),*}) \quad \text{(Using Equation (122))} \\
&\leq \bar{\phi}_1(z_1^{(k_j),*} + \epsilon\bar{\gamma} s^{(k_j)}, z^{(k_j),*}) \quad \text{(With } \epsilon \in [0,1] \text{ and since } z_1^{(k_j+1)} \text{ is the arg min of } \bar{\phi}_1(., z^{(k_j),*}) \text{)} \\
&:= \bar{\phi}_1(z_1^{(k_j),*} + \theta(z_1^{(k_j+1)} - z_1^{(k_j),*}), z^{(k_j),*}) \quad \text{(With } \theta = \epsilon\frac{\bar{\gamma}}{\gamma^{(k_j)}} \in [0,1]) \\
&\leq \max\left(\bar{\phi}_1(z_1^{(k_j+1)}, z^{(k_j),*}), \bar{\phi}_1(z^{(k_j),*}, z^{(k_j),*})\right) \quad \text{(Using quasi-convexity of } \bar{\phi}_1(., z^{(k_j),*})) \\
&= \bar{\phi}_1(z^{(k_j),*}, z^{(k_j),*}) \quad \text{(by definition of the arg min } z_1^{(k_j+1)}) \\
&= \phi(z^{(k_j),*}) \quad \text{(Majorizer is tight at } z^{(k_j),*} \text{)}
\end{aligned} \tag{124}$$

Letting $j \to \infty$ and combining Equations (123) and (124), we have:

$$\phi(z^{\infty}) \leq \bar{\phi}_1(z_1^{\infty} + \epsilon\bar{\gamma}\bar{s}) \leq \phi(z^{\infty}) \tag{125}$$

or equivalently:

$$\phi(z^{\infty}) = \bar{\phi}_1(z_1^{\infty} + \epsilon\bar{\gamma}\bar{s}) \tag{126}$$

Furthermore:

$$\begin{aligned}\bar{\phi}_1(z_1^{(k_{j+1}),*}, z^{(k_{j+1}),*}) = \phi(z^{(k_{j+1}),*}) \leq \phi(z^{(k_j+1),*}) &\leq \phi(z^{(k_j+1)}) \\ &\leq \bar{\phi}_1(z_1^{(k_j+1)}, z^{(k_j),*}) \\ &\leq \bar{\phi}_1(z_1, z^{(k_j),*}) \quad \forall z_1 \in \mathcal{Z}_1\end{aligned} \tag{127}$$

Thanks to the continuity of majorizers, and letting $j \to \infty$, we obtain:

$$\bar{\phi}_1(z_1^{\infty}, z^{\infty}) \leq \bar{\phi}_1(z_1, z^{\infty}) \quad \forall z_1 \in \mathcal{Z}_1$$

which implies that z_1^{∞} is the minimizer of $\bar{\phi}_1(., z^{\infty})$. By hypothesis, we assumed that the minimizer of each majorizer is unique, which is a contradiction with Equation (126). Therefore, the contrary assumption is not true, i.e., $z^{(k_j+1)} \to z^{\infty}$ for $j \to \infty$.

Since $z_1^{(k_j+1)} = \underset{z_1 \in \mathcal{Z}_1}{\text{argmin}}\ \bar{\phi}_1(z_1, z^{(k_j),*})$, then we have:

$$\bar{\phi}_1(z_1^{(k_j+1)}, z^{(k_j),*}) \leq \bar{\phi}_1(z_1, z^{(k_j),*}) \quad \forall z_1 \in \mathcal{Z}_1$$

The computation of the limit $j \to \infty$ gives:

$$\bar{\phi}_1(z_1^{\infty}, z^{\infty}) \leq \bar{\phi}_1(z_1, z^{\infty}) \quad \forall z_1 \in \mathcal{Z}_1$$

The rest of the proof is identical to [54], we detail the remaining steps for the sake of completeness. The above inequality implies that all the directional derivatives of $\bar{\phi}_1(., z^{\infty})$ w.r.t. block 1 of variables, and evaluated at z_1^{∞} are nonnegative, that is:

$$\bar{\phi}^{'}(z_1, z^{\infty}; d_1)\Big|_{z_1 = z_1^{\infty}} \geq 0 \quad \forall d_1 \in \mathbb{R}^{m_1} \text{ such that } z_1^{\infty} + d_1 \in \mathcal{Z}_1$$

Following the same rationale for the other blocks of variables, we obtain:

$$\bar{\phi}^{'}(z_i, z^{\infty}; d_i)\Big|_{z_i = z_i^{\infty}} \geq 0 \quad \forall d_i \in \mathbb{R}^{m_i} \text{ such that } z_i^{\infty} + d_i \in \mathcal{Z}_i, \forall i = 1, ..., s \tag{128}$$

Using Assumptions 2 and Equations (128), we obtain:

$$\phi^{'}(z^{\infty}; d_i^0) \geq 0 \quad \forall d_i^0 = (0, ..., d_i, ..., 0) \text{ such that } z_i^{\infty} + d_i \in \mathcal{Z}_i, \forall i = 1, ..., s \tag{129}$$

Equations (129) implies that z^{∞} is a coordinate-wise minimum of $\phi(.)$. Finally, assuming that $\phi(.)$ is regular on its domain, including z^{∞}, then we can write:

$$\phi^{'}(z^{\infty}; d) \geq 0 \quad \forall d = (d_1, ..., d_s) \text{ such that } z^{\infty} + d \in \mathcal{Z} \tag{130}$$

Under this additional regularity assumption of $\phi(.)$, Equation (130) implies that the limit point z^{∞} is a stationary point of $\phi(.)$. This concludes the proof. □

F MM update rules for the meta-algorithm

We consider a generic minimization problem of the form

$$\underset{X_1 \geq 0}{\operatorname{argmin}}\ D(T|X_1U) + \mu \sum_{q \leq r} \|X_1[:,q]\|_p^p \tag{131}$$

where $T = \mathcal{T}_{[1]} \in \mathbb{R}_+^{m_1 \times m_2 \times m_3}$, $U = \mathcal{G}_{[1]}(X_2 \boxtimes X_3)^T \in \mathbb{R}_+^{r_1 \times m_2 m_3}$ and $\mu > 0$. Notably, we considered the decomposition of a third-order tensor $\mathcal{T}$, but the approach can be generalized to any order.

In Problem (131), the regularization term is separable w.r.t. each entry of $X_1[:,q]$ for all $q \leq r$. For the data fitting term, we use the majorizer proposed in [20]. For the sake of completeness and to ease the understanding of the further developments, we briefly recall it in the following. It consists in first decomposing the β-divergence between two nonnegative scalars a and b as a sum of a convex, concave and constant terms as follows:

$$d_\beta(a|b) = \check{d}_\beta(a|b) + \hat{d}_\beta(a|b) + \bar{d}_\beta(a|b), \tag{132}$$

where $\check{d}$ is a convex function of b, $\hat{d}$ is a concave function of b and $\bar{d}$ is a constant w.r.t. b; see Table 3, and then in majorizing the convex part of the β-divergence using Jensen's inequality and majorizing the concave part by its tangent (first-order Taylor approximation).

Table 3: Differentiable convex-concave-constant decomposition of the β-divergence under the form (132) [20].

	$\check{d}(a\|b)$	$\hat{d}(a\|b)$	$\bar{d}(a)$
$\beta = (-\infty, 1) \setminus \{0\}$	$\frac{1}{1-\beta} a b^{\beta-1}$	$\frac{1}{\beta} b^\beta$	$\frac{1}{\beta(\beta-1)} a^\beta$
$\beta = 0$	ab^{-1}	$\log(b)$	$a(\log(a) - 1)$
$\beta \in [1, 2]$	$d_\beta(a\|b)$	0	0

Since $\mu \sum_{q \leq r} \|X_1[:,q]\|_p^p = \mu \|X_1\|_p^p$, the subproblem obtained after majorization (step 1 of MM, see Section 3.2) to solve has the form:

$$\underset{X_1 \geq 0}{\operatorname{argmin}}\ G(X_1|\tilde{X}_1) + \mu \|X_1\|_p^p \tag{133}$$

where $G(X_1|\tilde{X}_1)$ is the separable, tight majorizer for $D(T|X_1U)$ at $\tilde{X}_1 \geq 0$ of the following form

$$\begin{aligned} G(X_1|\tilde{X}_1) &= \sum_{i \leq m_1, k \leq r} l(X_1[i,k]|\tilde{X}_1) \\ \text{such that } l(X_1[i,k]|\tilde{X}_1) &= \sum_{j \leq m_2 m_3} \frac{\tilde{X}_1[i,k]U[k,j]}{\tilde{T}[i,j]} \check{d}(T[i,j]|\tilde{T}[i,j]\frac{X_1[i,k]}{\tilde{X}_1[i,k]}) \\ &\quad + \sum_{j \leq m_2 m_3} \Big[U[k,j]\hat{d}'(T[i,j]|\tilde{T}[i,j])\Big] X_1[i,k] + \text{cst} \end{aligned} \tag{134}$$

according to [20, Lemma 1] where $\tilde{T} = \tilde{X}_1 U$.

As explained in Section 3.2, the optimization problem (133) is a particular instance of the family of problems covered by the framework proposed in [39] in the case no additional equality constraints are required. Assuming $\mu > 0$ and the regularization terms satisfy Assumption 1 for the p value considered, therefore the minimizer of Problem (133) exists and is unique in $(0, \infty)^{m_1 \times r_1}$ as soon as $\beta < 2$.

Moreover, according to [39], it is possible to derive closed-form expressions for the minimizer of Problem (133) for the following values of β: $\beta \in \{0, 1, 3/2\}$ (if constants $C > 0$ in Assumption 1) or $\beta \in (-\infty, 1] \cup \{5/4, 4/3, 3/2\}$ (if constants $C = 0$ in Assumption 1). Outside these values for β, an iterative scheme is required to numerically compute the minimizer. We can cite the Newton's method, the Muller's method and the the procedure developed in [55] which is based on the explicit calculation of the intermediary root of a canonical form of cubic. This procedure is suited for providing highly accurate numerical results in the case of badly conditioned polynomials.

In this section, we will consider the interval $\beta \in [1, 2)$. For this setting, we can show that computing the minimizer of Problem (133) corresponds to solving, for a single element $X_1[i,k]$:

$$\begin{aligned} &\nabla_{X_1[i,k]} \Big[G(X_1|\tilde{X}_1) + \mu \|X_1\|_p^p\Big] = 0 \\ &\Leftrightarrow a X_1[i,k]^{p-1+2-\beta} + b X_1[i,k] - c = 0 \text{ since } X_1[i,k] \in (0, \infty). \end{aligned} \tag{135}$$

where the second equation has been obtained by multiplying both sides by $X_1[i,k]^{\beta-2}$, and:

- $a = p\mu > 0$ by hypothesis,
- $b = \sum_j U[k,j] \left(\frac{\tilde{T}[i,j]}{\tilde{X}_1[i,k]}\right)^{\beta-1} \geq 0$
- $c = \sum_j U[k,j]T[i,j] \left(\frac{\tilde{T}[i,j]}{\tilde{X}_1[i,k]}\right)^{\beta-2} \geq 0.$

At this stage, one may observe that finding the solution X_1 boils down to solving several polynomial equations in parallel. Nevertheless, depending on the values of β and p, we can further simplify the update.

Case β=1, p=1 Equation (135) becomes $aX_1[i,k] + bX_1[i,k] - c = 0$. The positive (real) root is computed as follows:

$$\begin{aligned} \hat{X}[i,k] &= \frac{c|_{\beta=1}}{a+b|_{\beta=1}} \\ &= \tilde{X}_1[i,k] \frac{\sum_j U[k,j] \frac{T[i,j]}{\tilde{T}[i,j]}}{\mu + \sum_j U[k,j]} \end{aligned} \tag{136}$$

which is nonnegative. In matrix format,

$$\hat{X}_1 = \tilde{X}_1 \odot \frac{\left[\frac{[T]}{[\tilde{X}_1 U]} U^\top\right]}{\left[\mu e_{r1\times m_1} + e_{m_1 \times m_2 m_3} U^\top\right]} \tag{137}$$

where $\odot$ denotes the Hadamard product, $[A]/[B]$ denotes the elementwise division between matrices A and B. Note that in [29], the regularization parameters μ is mistakenly written in the numerator. This error has been corrected in subsequent works without an explicit mention of the error [36].

Case β=1, p=2 Equation (135) becomes $aX_1[i,k]^2 + bX_1[i,k] - c = 0$. The positive (real) root is computed as follows:

$$\hat{X}_1[i,k] = \frac{\sqrt{\left(\sum_j U[k,j]\right)^2 + 8\mu \tilde{X}_1[i,k] \sum_j U[k,j] \frac{T[i,j]}{\tilde{T}[i,j]}} - \sum_k U[j,k]}{4\mu} \tag{138}$$

with $\mu > 0$. Note that although the closed-form expression in Equation (138) has a negative term in the numerator of the right-hand side, it can be easily checked that it always remains nonnegative given $T[i,j]$, $U[k,j]$ and $\tilde{X}[i,k]$ are nonnegative for all i,j,k. Furthermore, in the case the regularization term in Problem (131) is in the form $\mu \sum_{q\leq r} \frac{1}{p} \|X_1[:,q]\|_p^p$, then Equation (138) becomes:

$$\hat{X}_1[i,k] = \frac{\sqrt{\left(\sum_j U[k,j]\right)^2 + 4\mu \tilde{X}_1[i,k] \sum_j U[k,j] \frac{T[i,j]}{\tilde{T}[i,j]}} - \sum_k U[j,k]}{2\mu} \tag{139}$$

which can be expressed in matrix form as follows:

$$\hat{X}_1 = \frac{\left[C^{\cdot 2} + S\right]^{\cdot \frac{1}{2}} - C}{2\mu} \tag{140}$$

where $[A]^{\cdot\alpha}$ denotes the elementwise α-exponent, $C = eU^\top$ with e is a all-one matrix of size $m_1 \times m_2 m_3$ and $S = 4\mu\tilde{X}_1 \odot \left(\frac{[T]}{[\tilde{X}_1 U]} U^T\right)$.

Some interesting insights are discussed here-under:

- Computing the limit $\lim_{\mu\to 0} \hat{X}_1(\mu)$ makes Equation (140) tends to the original Multiplicative Updates introduced by Lee and Seung [37]. Indeed let us compute this limit in the scalar case from Equation

(138) and pose $\alpha = \sum_n u[j,k]$ and $\eta = \tilde{w}[j] \sum_j U[k,j] \frac{T[i,j]}{\tilde{T}[i,j]}$ for convenience:

$$\begin{aligned}\lim_{\mu \to 0} \frac{\sqrt{\alpha^2 + 4\mu\eta} - \alpha}{2\mu} &= \frac{\text{"0"}}{\text{"0"}} \\ &\underset{\text{H}}{=} \lim_{\mu \to 0} \frac{\frac{\partial}{\partial_\mu}\sqrt{\alpha^2 + 4\mu\eta} - \alpha}{\frac{\partial}{\partial_\mu} 2\mu} \\ &= \lim_{\mu \to 0} (\alpha^2 + 4\mu\eta)^{-\frac{1}{2}} \eta = \frac{\eta}{\alpha} = \tilde{X}_1[i,k] \frac{\sum_j U[k,j,k] \frac{T[i,j]}{\tilde{T}[i,j]}}{\sum_j U[k,j]}\end{aligned} \tag{141}$$

In matrix form we have:

$$\lim_{\mu \to 0} \hat{X}_1(\mu) = \tilde{X}_1 \odot \frac{\left[\frac{[T]}{[\tilde{X}_1 U]} U^T\right]}{[e U^T]} \tag{142}$$

which are the MU proposed by Lee and Seung for $\beta = 1$.

- Regarding the analysis of univariate polynomial equation given in (135), interestingly the existence of a unique positive real root could have been also established by using Descartes rules of sign. Indeed we can count the number of real positive roots that $p(X_1[i,k]) = aX_1[i,k]^{p-1+2-\beta} + bX_1[i,k] - c$ has (for $\beta \in [1,2]$). More specifically, let v be the number of variations in the sign of the coefficients a, b, c (ignoring coefficients that are zero). Let n_p be the number of real positive roots. Then:

  1. $n_p \leq v$,
  2. $v - n_p$ is an even integer.

  Let us consider the case $\beta = 1$ and $p = 2$ as an example: given the polynomial $p(X_1[i,k]) = aX_1[i,k]^2 + bX_1[i,k] - c$, assuming that c is positive. Then $v = 1$, so n_p is either 0 or 1 by rule 1. But by rule 2, $v - n_p$ must be even, hence $n_p = 1$. Similar conclusion can be made for any $\beta \in [1,2]$.

Finally, similar rationale can be followed to derive MM updates in closed-form for $\beta = 3/2$. The case $\beta = 2$ is less interesting to tackle with MM updates rules since each subproblem is L-smooth, allowing the use of highly efficient methods such as the Projected Accelerated Gradient Descent algorithm from [49], or the famous HALS algorithm. The derivation of the MM updates in the case $\beta = 0$ with ℓ_2^2 regularizations on the factors is detailed in Appendix G.

G Factor updates for $\beta = 0$ with ℓ_2^2 regularizations

In this section, we present the steps followed to derive MM update when the data fitting term corresponds to the well-known "Itakura-Saito" divergence, that is for $\beta = 0$, and with ℓ_2^2 regularization on $\{X_i[:,q]\}_{i \leq n}$ for all $q \leq r$. Since the update of all factors can be derived in a similar fashion, we only detail the update for $X_1 \in \mathbb{R}_+^{m_1 \times r_1}$, that is for $i = 1$, without loss of generality. The MM update for X_1 boils down to tackle the following subproblem:

$$\underset{X_1 \geq 0}{\operatorname{argmin}} \; D_0(T|X_1 U) + \mu \sum_{q \leq r} \frac{1}{2} \|X_1[:,q]\|_2^2 \tag{143}$$

where $T = \mathcal{T}_{[1]} \in \mathbb{R}_+^{m_1 \times m_2 \times m_3}$, $U = \mathcal{G}_{[1]}(X_2 \boxtimes X_3)^T \in \mathbb{R}_+^{r_1 \times m_2 m_3}$ and $\mu > 0$. We follow the methodology detailed in Sections 3.2 and 4.4. In Problem (143), the regularization terms are separable w.r.t. each entry of $X_1[:,q]$ for all $q \leq r$. For the data fitting term, we use the results from [20]. Since $\sum_{q \leq r} \frac{1}{2}\|X_1[:,q]\|_2^2 = \frac{1}{2}\|X_1\|_F^2$, this leads to solve the following problem:

$$\underset{X_1 \geq 0}{\operatorname{argmin}} \; G(X_1|\tilde{X_1}) + \frac{1}{2}\mu \|X_1\|_F^2 \tag{144}$$

where $G(X_1|\tilde{X_1})$ is the separable, tight majorizer for $D_0(T|X_1U)$ at $\tilde{X_1}$ as proposed in [20, Lemma 1].

Given that the objective function in Problem (144) is separable with respect to each entry $X_1[i,k]$ of X_1 where $1 \leq i \leq m_1$ and $1 \leq k \leq r_1$, we can update each entry separately. To find the value of $X_1[i,k]$, we solve

$$\nabla_{X_1[i,k]} \left[G(X_1|\tilde{X_1}) + \frac{1}{2}\mu ||X_1||_F^2 \right] = 0, \tag{145}$$

and check if the solution is unique and satisfies the nonnegativity constraints. Considering the discussion in Section 3.2 and the findings of [39, Proposition1], we are aware that this minimizer exists and is unique over the range of $(0, +\infty)$ due to the condition of $\mu > 0$ and the regularization function $g(.)$ $(= \frac{1}{2}||.||_F^2)$ satisfying Assumption 1 with constants $C = 1$. For now, we will put this result aside and simply compute the solution of Equation (145) and then check if it exists and is unique over the positive real line.

From Lemma [20, Lemma 1] and Table 3 for $\beta = 0$, one can show that solving (145) reduces to finding $X_1[i,k] > 0$ of the following scalar equation:

$$-\bar{a}(X_1[i,k])^{-2} + \mu X_1[i,k] + \bar{c} = 0 \tag{146}$$

where $\bar{a} = (\tilde{X}_1[i,k])^2 \sum_j U[k,j] \frac{T[i,j]}{(\tilde{T}[i,j])^2}, \geq 0$, $\tilde{T} = \tilde{X}_1 U$ and $\bar{c} = \sum_j \frac{U[k,j]}{T[i,j]}$ [11]. Posing $x = X_1[i,k]$, Equation (146) can be equivalently rewritten as follows:

$$x^3 + px^2 + qx + r = 0, \tag{147}$$

where $p = \frac{\bar{c}}{\mu} (\geq 0)$, $q = 0$ and $r = -\frac{\bar{a}}{\mu} (\leq 0)$. Equation (147) can be rewritten in the so-called normal form by posing $x = z - p/3$ as follows:

$$z^3 + az + b = 0, \tag{148}$$

where $a = \frac{1}{3}(3q - p^2) = \frac{-p^2}{3}$ and $b = \frac{1}{27}(2p^3 - 9pq + 27r) = \frac{1}{27}(2p^3 + 27r)$. Under the condition $\frac{b^2}{4} + \frac{a^3}{27} > 0$, Equation (148) has one real root and two conjugate imaginary roots. This condition boils down to $\frac{1}{4}(2p^3 + 27r)^2 - p^6 > 0$. By definition of p (non-negative) and r (non-positive) above, this condition holds if $p > 0$ and $r < -\frac{4p^3}{27}$. The first condition on p is satisfied assuming that $\bar{c}$ is positive. The second condition on r is satisfied by further assuming $\mu > \sqrt{\frac{4}{27}\frac{(\bar{c})^3}{\bar{a}}}$ [12]. Therefore, the positive real root has the following closed-form expression:

$$z = \sqrt[3]{-\frac{b}{2} + \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}} + \sqrt[3]{-\frac{b}{2} - \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}}, \tag{149}$$

and hence

$$X_1[i,k] = z - \frac{\bar{c}}{3\mu}, \tag{150}$$

With a bit of technicalities, one can show this root is positive, under the assumptions $\bar{c} > 0$ and $\mu > \sqrt{\frac{4}{27}\frac{(\bar{c})^3}{\bar{a}}}$. In matrix form, we obtain the following update

$$X_1 = Z - \frac{[\bar{C}]}{[3\mu]},$$

where $Z = \left[\frac{-B}{2} + \left[\frac{[B]^{\cdot 2}}{4} + \frac{[A]^{\cdot 3}}{27}\right]^{\cdot (1/2)}\right]^{\cdot (1/3)} + \left[\frac{-B}{2} - \left[\frac{[B]^{\cdot 2}}{4} + \frac{[A]^{\cdot 3}}{27}\right]^{\cdot (1/2)}\right]^{\cdot (1/3)}$, $A = -\frac{1}{3}\left[\frac{[\bar{C}]}{[\mu]}\right]^{\cdot 2}$, $B = \frac{1}{27}\Big(2\left[\frac{[\bar{C}]}{[\mu]}\right]^{\cdot 3} + 27\frac{[-\bar{A}]}{[\mu]}\Big)$, $\bar{A} = [\tilde{X}_1]^{\cdot 2} \odot \Big(\frac{[T]}{[\tilde{X}_1 U]^{\cdot 2}} U^\top\Big)$ and $\bar{C} = \frac{[E]}{[\tilde{X}_1 U]} U^\top$, with E denoting a matrix full of ones of size $m_1 \times m_2 m_3$.

Note that we skipped in the discussion above the following cases:

1. $\frac{b^2}{4} + \frac{a^3}{27} = 0$, that is when $r = -\frac{4p^3}{27}$, or equivalently $\mu = \sqrt{\frac{4}{27}\frac{(\bar{c})^3}{\bar{a}}}$: here, Equation (148) has one nonnegative real root given by $z = 2\sqrt{-\frac{a}{3}} = \frac{2}{3}\frac{\bar{c}}{\mu}$ (if $b < 0$) and hence $X_1[i,k] = z - \frac{\bar{c}}{3\mu} = \frac{\bar{c}}{3\mu} \geq 0$. If $b > 0$, the root is given by $z = \frac{\bar{c}}{3\mu}$, and then $X_1[i,k] = 0$.

2. $\frac{b^2}{4} + \frac{a^3}{27} < 0$, that is when $\mu < \sqrt{\frac{4}{27}\frac{(\bar{c})^3}{\bar{a}}}$: then Equation (148) has one nonnegative real root given among $z = \frac{2}{3}\frac{\bar{c}}{\mu}\cos\left(\frac{\Phi}{3} + \frac{2k\pi}{3}\right)$ with $k = 0, 1, 2$ and where $\cos\Phi = \frac{1}{2}\frac{|2p^3 + 27r|}{p^3}$.

Those cases are treated in the implementation of the algorithm.

[11] The solution $X_1[i,k] = 0$ only exists if $\bar{c} = 0$, that is, by definition of $\bar{c}$, when the column $U[k,:]$ is zero.

[12] In practice, this condition means that μ should not be chosen too small.