Majorization-minimization for Sparse Nonnegative Matrix Factorization with the β-divergence

Arthur Marmin, José Henrique de Morais Goulart, Cédric Févotte

Majorization-minimization for Sparse Nonnegative Matrix Factorization with the β-divergence

Arthur Marmin, José Henrique de Morais Goulart, and Cédric Févotte,

***Abstract*—This article introduces new multiplicative updates for nonnegative matrix factorization with the β-divergence and sparse regularization of one of the two factors (say, the activation matrix). It is well known that the norm of the other factor (the dictionary matrix) needs to be controlled in order to avoid an ill-posed formulation. Standard practice consists in constraining the columns of the dictionary to have unit norm, which leads to a nontrivial optimization problem. Our approach leverages a reparametrization of the original problem into the optimization of an equivalent scale-invariant objective function. From there, we derive block-descent majorization-minimization algorithms that result in simple multiplicative updates for either ℓ_1-regularization or the more "aggressive" log-regularization. In contrast with other state-of-the-art methods, our algorithms are universal in the sense that they can be applied to any β-divergence (i.e., any value of β) and that they come with convergence guarantees. We report numerical comparisons with existing heuristic and Lagrangian methods using various datasets: face images, an audio spectrogram, hyperspectral data, and song play counts. We show that our methods obtain solutions of similar quality at convergence (similar objective values) but with significantly reduced CPU times.**

***Index Terms*—Nonnegative matrix factorization (NMF), beta-divergence, majorization-minimization method (MM), sparse regularization**

I. Introduction

Nonnegative matrix factorization (NMF) consists in decomposing a data matrix $\mathbf{V}$ with nonnegative entries into the products $\mathbf{WH}$ of two nonnegative matrices [1], [2]. When the data samples are arranged in the columns of $\mathbf{V}$, the first factor $\mathbf{W}$ can be interpreted as a dictionary of basis vectors (or atoms). The second factor $\mathbf{H}$, termed activation matrix, contains the expansion coefficients of each data sample onto the dictionary. NMF has found many applications such as feature extraction in image processing and text mining [2], audio source separation [3], blind unmixing in hyperspectral imaging [4], [5], and user recommendation [6]. For a thorough presentation of NMF and its applications, see [7]–[9].

NMF is usually cast as the minimization of a well-chosen measure of fit between $\mathbf{V}$ and $\mathbf{WH}$. A widespread choice for the measure of fit is the β-divergence, a family of divergences parametrized by a single shape parameter $\beta \in \mathbb{R}$. This family notably includes the squared Frobenius norm (quadratic loss) as well as the Kullback-Leibler (KL) and Itakura-Saito (IS) divergences [10], [11]. NMF is well-known to favor part-based representations that *de facto* produce a sparse representation of the input data (because of the sparsity of either $\mathbf{W}$ or $\mathbf{H}$) [2]. This is a consequence of the nonnegativity constraints that produce zeros on the border of the admissible domain of $\mathbf{W}$ and $\mathbf{H}$. However, it is sometimes desirable to accentuate or control the sparsity of the factors by regularizing NMF with specific sparsity-promoting terms. This can improve the interpretability or suitability of the resulting representation as illustrated in the seminal work of Hoyer [12], [13].

Sparse regularization using penalty terms on the factors is the most common approach to induce sparsity. A classic penalty term is the ℓ_1 norm (the sum of the entries of the nonnegative factor), used for example in [12], [14]–[19]. Other ℓ_p norms such as the $\ell_{1/2}$ norm have also been considered, e.g., [20], [21]. Other works have considered log-regularization (i.e., penalizing the sum of the logarithms of the entries of the factor) which leads to more "aggressive" sparsity, e.g., [22]–[24]. Sparse regularization with information measures is also considered in [25], [26]. Another approach to induce sparsity consists in applying hard constraints to the factors (rather than mere penalization), using ℓ_0 constraints [27], [28] or using the sparseness measure introduced in [13]. Regularization of NMF with group-sparsity has also been a very active topic, see, e.g., early references [22], [23], [29].

In this paper, we assume without loss of generality that the sparse regularization is applied to $\mathbf{H}$ (our results apply equally as well to $\mathbf{W}$ by transposing $\mathbf{V}$ and exchanging of the roles of $\mathbf{W}$ and $\mathbf{H}$). NMF with sparse regularization of $\mathbf{H}$ requires controlling the norm of $\mathbf{W}$. Indeed, the measure of fit only depends on the product of $\mathbf{W}$ and $\mathbf{H}$ while the regularization term solely depends on $\mathbf{H}$: it is then possible to arbitrarily decrease the overall objective function by decreasing the scale of $\mathbf{H}$ and increasing the scale of $\mathbf{W}$ (this will be made more precise in Section II-B). A standard approach to avoid this issue is to constrain the norms (either ℓ_1 or ℓ_2) of the individual columns of $\mathbf{W}$ to be less or equal to 1. This has been enforced in various ways in the literature as we now explain.

A. State of the art

A first strategy, employed in [12], [17], consists in using projected gradient descent for the update of $\mathbf{W}$. This procedure works well with the quadratic loss function and unit ℓ_2 norm constraint that is used in those papers.

A second strategy consists in reparametrizing $\mathbf{W}$ as $\mathbf{W}\,\mathrm{Diag}^{-1}(\|\mathbf{w}_1\|,\dots,\|\mathbf{w}_K\|)$, where $\mathbf{w}_k$ denotes the k-th column of $\mathbf{W}$, and optimizing over the new rescaled variable. This approach was proposed in [14] for NMF with

This work is supported by the European Research Council (ERC FACTORY-CoG-6681839).

A. Marmin and C. Févotte are with IRIT, Université de Toulouse, CNRS, Toulouse, France (email: arthur.marmin@irit.fr; cedric.fevotte@irit.fr).

J. H. de M. Goulart is with IRIT, Université de Toulouse, Toulouse INP, Toulouse, France (e-mail: henrique.goulart@irit.fr).

the quadratic loss and was extended to NMF with the β-divergence (referred to as β-NMF in the following) in [30]. The proposed update for $\mathbf{W}$ is heuristic (it will be presented in Section III-B); while successful in practice, it lacks a proof of convergence (in particular, it does not ensure non-increasingness of the objective function as it will be illustrated in Section VI-C1).

A third strategy consists in following a Lagrangian approach and minimizing an augmented objective function that includes the desired constraints on the columns of $\mathbf{W}$. This is the approach pursued in [31] for β-NMF and described in Section III-A. Unfortunately, practical updates are only obtained for $\beta \leq 1$ and specific values $\beta \in \{\frac{5}{4}, \frac{4}{3}, \frac{3}{2}\}$. This excludes most of the interval $\beta \in (1, 2)$ which is of applicative interest. For admissible values of β, this method offers theoretical guarantees and good experimental performance. However, the update of Lagrangian multipliers requires a numerical procedure that can be costly. In the special case $\beta = 1$, the Lagrangian multipliers have a closed-form expression that simplifies the updates (see, e.g., [32]).

Finally, a last strategy consists in rewriting sparse NMF as the optimization of an equivalent scale-invariant objective function.

In this approach, detailed in Section IV, the rows of $\mathbf{H}$ are multiplied by the norms of the columns of $\mathbf{W}$ and the new objective function can be optimized without norm constraints. In this scheme, the columns of $\mathbf{W}$ can be normalized at the end of the optimization (and the rows of $\mathbf{H}$ rescaled accordingly). This approach was applied to NMF with the IS divergence and log-regularization in [22] and to NMF with the KL divergence and a Markov regularization of the rows of $\mathbf{H}$ in [33]. In these two cases, a block-descent Majorization-Minimization (MM) algorithm was proposed. The approach is well-posed and does not rely on any heuristic. The MM algorithm results in simple multiplicative updates that ensure non-increasingness of the objective function.

B. Contributions

In this paper, we generalize the approach of [22], [33] to NMF with every possible β-divergence, i.e., for all $\beta \in \mathbb{R}$ and not merely $\beta = 0$ and $\beta = 1$. More precisely, we first design a universal block-descent MM algorithm for β-NMF with ℓ_1-regularization of $\mathbf{H}$, and unit ℓ_1 norm constraint on the columns of $\mathbf{W}$. This algorithm extends [33], that was specifically designed for the KL divergence and a different regularization term. Then, we design another universal block-descent MM algorithm for β-NMF with log-regularization of $\mathbf{H}$, and unit ℓ_1 norm constraint on the columns of $\mathbf{W}$. The algorithm extends [22] that was aimed at the IS divergence solely. In both cases, the block-descent MM approach leads to alternating multiplicative updates that are free of tuning parameters. They are easy to implement and enjoy linear complexity per iteration. By design, the MM framework ensures the non-increasingness and thus the convergence of the objective function. We further show the convergence of the iterates to the set of stationary points of the problem using the theoretical framework of [19].

Then we demonstrate the practical advantages of our method with extensive simulations using datasets arising from various applications: face images, audio spectrogram, hyperspectral data and song play-counts. We compare our MM algorithm for ℓ_1-regularized β-NMF with the heuristic presented in [30] and also with the Lagrangian method from [31]. Additionally, we adapt the heuristic of [30] for β-NMF with log-regularization and compare it with our MM algorithm. In all cases, we show that our MM algorithms obtain solutions whose quality is similar to that of existing algorithms at convergence (similar objective values) but often with significantly reduced CPU times. Moreover, our algorithms overcome some of the limitations of these other approaches, namely that the heuristic [30] has no convergence guarantees and that the Lagrangian approach [31] cannot be applied to any value of β.

C. Outline

The rest of this article is organized as follows. Section II introduces β-NMF with ℓ_1-regularization of the activation matrix $\mathbf{H}$. It explains the necessity of controlling the norm of $\mathbf{W}$ to formulate a well-posed optimization problem. Section III details the state of the art for the latter problem, and more precisely the heuristic method [30] and the Lagrangian method [31]. Section IV presents our universal block-descent MM algorithm for β-NMF with ℓ_1-regularization. The derivations lead to multiplicative updates with convergence guarantees. Section V extends the methodology of Section IV to β-NMF with log-regularization of $\mathbf{H}$. Experimental results are presented in Section VI and Section VII concludes.

D. Notation

The set $\mathbb{N}$ denotes the set of natural numbers while $[\![1, N]\!]$ denotes its subset containing natural numbers from 1 to N. The set $\mathbb{R}_+$ denotes the set of nonnegative real numbers. Bold upper case letters denote matrices, bold lower case letters denote vectors, and lower case letters denote scalars. The notation $[\mathbf{M}]_{ij}$ and m_{ij} both stand for the element of $\mathbf{M}$ located at the i^{th} row and the j^{th} column. The operators $\odot$ and $/$, and $\cdot^{\cdot\alpha}$ applied to matrices denote the entry-wise multiplication, division and power α, respectively. For a matrix $\mathbf{M}$, the notation $\mathbf{M} \geq 0$ denotes entry-wise nonnegativity. The vector $\mathbf{1}_N$ and the matrix $\mathbf{1}_{F \times N}$ are the vector of dimension N and the matrix of dimension $F \times N$ composed solely of 1 respectively.

II. NMF WITH β-DIVERGENCE AND ℓ_1 REGULARIZATION

A. Objective

Our goal is to factorize an $F \times N$ nonnegative data matrix $\mathbf{V}$ into the product $\mathbf{WH}$ of two nonnegative factor matrices of dimensions $F \times K$ and $K \times N$ respectively. The inner rank K is assumed to be a fixed parameter of the problem. Placing a ℓ_1-regularization term on $\mathbf{H}$, we aim at solving the following problem

$$\inf_{\mathbf{W},\mathbf{H} \geq 0} \quad \mathcal{J}(\mathbf{W}, \mathbf{H}) \stackrel{\text{def}}{=} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \alpha \|\mathbf{H}\|_1 , \qquad (1)$$

where $\|\mathbf{H}\|_1 = \sum_{k,n} |h_{kn}| = \sum_{k,n} h_{kn}$ and α is a nonnegative hyperparameter that governs the degree of sparsity of $\mathbf{H}$. The data-fitting term D_β is defined as

$$D_\beta(\mathbf{V} \mid \mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d_\beta(v_{fn} \mid [\mathbf{WH}]_{fn}),$$

where d_β is the β-divergence [10], [11] given by

$$d_\beta(x \mid y) = \begin{cases} x \log \frac{x}{y} - x + y & \text{if } \beta = 1 \\ \frac{x}{y} - \log \frac{x}{y} - 1 & \text{if } \beta = 0 \\ \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{x y^{\beta-1}}{\beta-1} & \text{otherwise.} \end{cases}$$

The choice of β can be made in accordance with the application or the assumed noise model for $\mathbf{V}$ [11]. Common values for β are 0, 1, and 2; they corresponds to the IS divergence, KL divergence and squared Frobenius norm, respectively. The range $\beta \in [0, 2]$ is the one with largest practical interest. Values of $\beta \in [0, 0.5]$ are, for example, customary in audio spectral decomposition [34], [35]. Values of $\beta \in [1, 2]$ have proven efficient in hyperspectral unmixing [36]. Note that the latter interval is important because the β-divergence $d_\beta(x|y)$ is convex with respect to (w.r.t.) y when $\beta \in [1, 2]$. In that case, the optimization subproblems in $\mathbf{W}$ and $\mathbf{H}$ are separately convex, though $\mathcal{J}(\mathbf{W}, \mathbf{H})$ is always jointly non-convex.

B. Well-posed constrained formulation

An important observation is that D_β suffers from a scaling ambiguity since it only depends on the product $\mathbf{WH}$ and not on $\mathbf{W}$ and $\mathbf{H}$ separately. As a consequence, Problem (1) is ill-posed: for any solution $(\mathbf{W}^*, \mathbf{H}^*)$, there is always a solution $(\tau\mathbf{W}^*, \frac{1}{\tau}\mathbf{H}^*)$, with $\tau > 1$ a positive real constant, that yields a better minimizer, i.e., $\mathcal{J}(\tau\mathbf{W}^*, \frac{1}{\tau}\mathbf{H}^*) < \mathcal{J}(\mathbf{W}^*, \mathbf{H}^*)$. Hence, the function $\mathcal{J}$ is not coercive: for any feasible point $(\mathbf{W}, \mathbf{H})$, we can find a real number $\tau > 1$ such that the sequence $\left\{\tau^k \mathbf{W}, \frac{1}{\tau^k}\mathbf{H}\right\}_{k\in\mathbb{N}}$ remains feasible but diverges in norm while the sequence $\left\{\mathcal{J}(\tau^k \mathbf{W}, \frac{1}{\tau^k}\mathbf{H})\right\}_{k\in\mathbb{N}}$ is decreasing and bounded. Therefore, the infimum of (1) is never attained and there exists no minimizer $(\mathbf{W^*}, \mathbf{H^*})$.

A natural way of tackling the ill-posedness of (1) is to control the norm of $\mathbf{W}$. This can be done by adding a penalizing term $\|\mathbf{W}\|$ (for a chosen norm) to $\mathcal{J}(\mathbf{W}, \mathbf{H})$ [15], [19]. Alternatively, we may minimize $\mathcal{J}(\mathbf{W}, \mathbf{H})$ subject to the additional constraint $\|\mathbf{W}\| \leq \theta$ where θ is a positive hyperparameter (it can be easily shown that this returns solutions such that $\|\mathbf{W}\| = \theta$) [28], [37]. We choose here to minimize $\mathcal{J}(\mathbf{W}, \mathbf{H})$ subject to the additional constraint that the individual columns of $\mathbf{W}$ have unit norm. This is a rather natural option for dictionary learning, as it makes sense to retrieve atoms (the columns of $\mathbf{W}$) that have equal norm. Instead, a constraint on the norm of the full matrix $\mathbf{W}$ can return a solution with atoms of different weights. We choose for practical commodity to constrain the ℓ_1 norm of the columns of $\mathbf{W}$. In the end, we address in Sections III and IV the following problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \ \mathcal{J}(\mathbf{W}, \mathbf{H}) \quad \text{s.t. } (\forall k \in [\![1, K]\!], \ \|\mathbf{w}_k\|_1 = 1). \quad (2)$$

Section III presents the state-of-the-art methods for solving the above problem [30], [31] while Section IV introduces our block-descent MM algorithm.

III. State of the art

A. Lagrangian method

A standard method to solve optimization problems with constraints is the method of Lagrange multipliers. This method has been suggested by [31] to solve β-NMF under a wide possibility of linear equality constraints on either $\mathbf{W}$ or $\mathbf{H}$. The Lagrangian associated to Problem (2) can be written as

$$\mathcal{L}(\mathbf{W}, \mathbf{H}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \alpha \|\mathbf{H}\|_1 - \sum_{k=1}^{K} \nu_k (\|\mathbf{w}_k\|_1 - 1), \quad (3)$$

where $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_K]^\top \in \mathbb{R}^K$ is the vector of Lagrangian multipliers.

The saddle points of $\mathcal{L}(\mathbf{W}, \mathbf{H}, \boldsymbol{\nu})$ subject to $\mathbf{W}, \mathbf{H} \geq 0$ yield solutions of Problem (2). As such, the authors of [31] describe a block-coordinate algorithm that alternately updates the blocks $\mathbf{W}$, $\mathbf{H}$, and $\boldsymbol{\nu}$. Given $\boldsymbol{\nu}$, the individual updates of $\mathbf{W}$, $\mathbf{H}$ are handled with one step of block-descent MM, using the methodology of [11], [38] (see also Section IV-B). [1] This leads to the following updates when $\beta \leq 1$

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^\top \mathbf{S}_\beta}{\mathbf{W}^\top \mathbf{T}_\beta + \alpha} \right)^{\cdot \frac{1}{2-\beta}} \quad (4)$$

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \left(\frac{\mathbf{S}_\beta \mathbf{H}^\top}{\mathbf{T}_\beta \mathbf{H}^\top - \mathbf{1}_F \boldsymbol{\nu}^\top} \right)^{\cdot \frac{1}{2-\beta}}, \quad (5)$$

where the matrices $\mathbf{S}_\beta$ and $\mathbf{T}_\beta$ are defined as

$$\mathbf{S}_\beta = \mathbf{V} \odot (\mathbf{WH})^{\cdot(\beta-2)}, \quad (6)$$

$$\mathbf{T}_\beta = (\mathbf{WH})^{\cdot(\beta-1)}. \quad (7)$$

More intricate updates can be obtained for $\beta \in \{\frac{5}{4}, \frac{4}{3}, \frac{3}{2}\}$. In every case, only the update of $\mathbf{W}$ depends on $\boldsymbol{\nu}$, which we highlight next with the abusive notation $\mathbf{W}(\boldsymbol{\nu})$. The k-th multiplier ν_k must ensure that $\mathbf{W}(\boldsymbol{\nu})$ given by (5) satisfies $\|\mathbf{w}_k(\nu_k)\|_1 = 1$, i.e.,

$$\sum_f w_{fk}(\nu_k) = 1.$$

The latter equation involves finding the root of a rational function and has no closed-form solution. However, the authors of [31] show that it has a unique solution that can be estimated with a standard Newton-Raphson procedure in about 10 to 100 subiterations. The solution is also shown to ensure that the denominator in (5) remains positive so that the update is well-defined and preserves nonnegativity.

Overall, the Lagrangian method [31] is conceptually well-grounded and elegant. It ensures that $\mathbf{W}$ satisfies the desired

[1] To be accurate, [31] uses a slightly different formulation. Indeed, instead of optimizing the Lagrangian (3) associated to Problem (2), they optimize the Lagrangian associated with the problem of minimizing a majorizer of $C(\mathbf{W}) = D_\beta(\mathbf{V}|\mathbf{WH})$ subject to unit norm constraints. The resulting updates turn out to be the same.

norm constraint at every iteration and also ensures non-increasingness of $\mathcal{J}(\mathbf{W},\mathbf{H})$. However it applies to specific values of β and requires a numerical subroutine for the estimation of the Lagrange multipliers.

B. Heuristic multiplicative updates

Another way to solve Problem (2) was proposed in [14] for NMF with the quadratic loss and extended to β-NMF in [30]. It consists first in formulating (2) as an unconstrained problem based on a reparametrization of the factor $\mathbf{W}$. More precisely, the factor $\mathbf{W}$ is replaced by the normalized factor $\mathbf{W}\mathbf{\Lambda}^{-1}$, where $\mathbf{\Lambda}$ is a $K \times K$ diagonal matrix with entries $\lambda_k = \|\mathbf{w}_k\|$ and $\|.\|$ is some chosen norm. The original papers [14], [30] consider ℓ_2 normalization. We adapt their methodology to ℓ_1 normalization for fair comparison with the other methods considered in this paper. This results in the following minimization problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \quad \tilde{\mathcal{J}}(\mathbf{W},\mathbf{H}) \stackrel{\text{def}}{=} D_\beta\left(\mathbf{V} \mid \mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{H}\right) + \alpha \left\|\mathbf{H}\right\|_1 . \quad (8)$$

The authors of [14], [30] then propose to solve Problem (8) using a block-alternating algorithm that updates $\mathbf{W}$ and $\mathbf{H}$ in turns. Multiplicative updates for each factor are obtained by employing a heuristic commonly used in NMF, see [34], [39]. Looking at the update of $\mathbf{W}$, the heuristic first consists in decomposing the gradient $\nabla_{\mathbf{W}}\tilde{\mathcal{J}}(\mathbf{W},\mathbf{H})$ of $\tilde{\mathcal{J}}(\mathbf{W},\mathbf{H})$ w.r.t. $\mathbf{W}$ into the difference of two nonnegative functions, i.e., $\nabla_{\mathbf{W}}\tilde{\mathcal{J}} = \nabla^+_{\mathbf{W}}\tilde{\mathcal{J}} - \nabla^-_{\mathbf{W}}\tilde{\mathcal{J}}$. Such a decomposition does exist for the considered function, though it might not exist for other problems. Then, given a current iterate of $\mathbf{H}$, a multiplicative update of $\mathbf{W}$ is constructed as

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \frac{\nabla^-_{\mathbf{W}}\tilde{\mathcal{J}}(\mathbf{W},\mathbf{H})}{\nabla^+_{\mathbf{W}}\tilde{\mathcal{J}}(\mathbf{W},\mathbf{H})}. \quad (9)$$

The motivating principle of update (9) is as follows. Assume that $[\nabla_{\mathbf{W}}\tilde{\mathcal{J}}]_{fk} > 0$ for a given coefficient w_{fk}, then the ratio in (9) is lower than 1 and the multiplicative update decreases w_{fk} as it can be expected from a descent algorithm. Likewise, update (9) increases the value of w_{fk} when the gradient is negative. The heuristic works well in practice but does not come with any guarantee. In particular, it does not ensure that $\tilde{\mathcal{J}}$ decreases at every iteration (and it turns out that $\tilde{\mathcal{J}}$ sometimes increases). Applying the heuristic to $\mathbf{W}$ and $\mathbf{H}$ in Problem (8) leads to the following updates

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top\mathbf{S}_\beta}{\mathbf{W}^\top\mathbf{T}_\beta + \alpha} \quad (10)$$

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \frac{\mathbf{S}_\beta\mathbf{H}^\top + \mathbf{1}_{F\times F}(\mathbf{W}\odot\mathbf{T}_\beta\mathbf{H}^\top)}{\mathbf{T}_\beta\mathbf{H}^\top + \mathbf{1}_{F\times F}(\mathbf{W}\odot\mathbf{S}_\beta\mathbf{H}^\top)} \quad (11)$$

$$\mathbf{W} \longleftarrow \mathbf{W}\mathbf{\Lambda}^{-1}, \quad (12)$$

where $\mathbf{S}_\beta$ and $\mathbf{T}_\beta$ are defined in (6) and (7).

Note that the updates (4) and (10) coincide up to the exponent $1/(2-\beta)$. As a matter of fact, when the other variables are fixed, the problems of minimizing $\mathcal{L}$ and $\tilde{\mathcal{J}}$ w.r.t. $\mathbf{H}$ are identical, but solved differently. The MM update (4) can be generalized to all values of β as shown in [23, Supplementary Material], [19]. This means that we could use a theoretically-grounded MM update of $\mathbf{H}$ instead of the heuristic (10). However, omitting the exponent often results in a beneficial acceleration in practice and we stick to the formulation of [30] given by (10) for fair comparison.

In summary, the approach proposed by [30] is intuitive, easy to implement, and applicable in principle for all values of β. Unfortunately, it lacks theoretical support. In the next section, we present a theoretically sound algorithm that results in equally simple multiplicative updates with equal or better performance.

IV. A unified block-descent MM algorithm for β-NMF with ℓ_1 regularization

In this section, we first reformulate (2) into a well-posed optimization problem that is free of norm constraints. This is similar to the approach of [30] except that we use a different reformulation. The reformulated problem allows to derive a block-descent MM algorithm that results in simple multiplicative updates for both $\mathbf{W}$ and $\mathbf{H}$. By design, the algorithm ensures that the objective function values are non-increasing and convergent. Additionally, we show that the sequence of iterates produced by the algorithm also converges.

A. Equivalent scale-invariant objective function

1) Reformulation without norm constraints: Let us introduce the following problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \quad \check{\mathcal{J}}(\mathbf{W},\mathbf{H}) \stackrel{\text{def}}{=} D_\beta(\mathbf{V} \mid \mathbf{W}\mathbf{H}) + \alpha \left\|\mathbf{\Lambda}\mathbf{H}\right\|_1 , \quad (13)$$

where $\mathbf{\Lambda}$ is defined like in Section III-B, i.e., $\mathbf{\Lambda} = \mathrm{Diag}\left(\|\mathbf{w}_1\|_1, \ldots, \|\mathbf{w}_K\|_1\right)$. Let us denote by $\mathbb{F}$ the feasible set of Problem (2), i.e.

$$\mathbb{F} = \left\{(\mathbf{W},\mathbf{H}) \in \mathbb{R}_+^{F\times K} \times \mathbb{R}_+^{K\times N} | \forall k \in [\![1,K]\!],\ \|\mathbf{w}_k\|_1 = 1\right\}.$$

Then for any $(\mathbf{W},\mathbf{H}) \in \mathbb{F}$, we have $\check{\mathcal{J}}(\mathbf{W},\mathbf{H}) = \mathcal{J}(\mathbf{W},\mathbf{H})$. The following lemmas show that Problem (13) and Problem (2) are equivalent.

Lemma 1. *Let $(\mathbf{W}^*,\mathbf{H}^*) \geq 0$ be a solution of Problem (13). Let us define their renormalized equivalents by $\bar{\mathbf{W}}^* = \mathbf{W}^*\mathbf{\Lambda}^{*-1}$ and $\bar{\mathbf{H}}^* = \mathbf{\Lambda}^*\mathbf{H}^*$ where $\mathbf{\Lambda}^* = \mathrm{Diag}\left(\|\mathbf{w}_1^*\|_1, \ldots, \|\mathbf{w}_K^*\|_1\right)$. Then, $(\bar{\mathbf{W}}^*,\bar{\mathbf{H}}^*)$ is a solution of Problem (2).*

Proof. Assume that $\bar{\mathbf{W}}^*$, $\bar{\mathbf{H}}^*$ is not a solution of Problem (2). Then, there exists $(\bar{\mathbf{W}}^+,\bar{\mathbf{H}}^+) \in \mathbb{F}$ such that $\mathcal{J}(\bar{\mathbf{W}}^+, \bar{\mathbf{H}}^+) < \mathcal{J}(\bar{\mathbf{W}}^*, \bar{\mathbf{H}}^*)$. By design, we have $\mathcal{J}(\bar{\mathbf{W}}^*,\bar{\mathbf{H}}^*) = \check{\mathcal{J}}(\mathbf{W}^*,\mathbf{H}^*)$. Furthermore, $\mathcal{J}(\bar{\mathbf{W}}^+, \bar{\mathbf{H}}^+) = \check{\mathcal{J}}(\bar{\mathbf{W}}^+, \bar{\mathbf{H}}^+)$. It follows that $\check{\mathcal{J}}(\bar{\mathbf{W}}^+,\bar{\mathbf{H}}^+) < \check{\mathcal{J}}(\mathbf{W}^*,\mathbf{H}^*)$, which contradicts the assumption that $(\mathbf{W}^*,\mathbf{H}^*)$ is a solution of Problem (13). □

Lemma 2. *Let $(\bar{\mathbf{W}}^*,\bar{\mathbf{H}}^*) \in \mathbb{F}$ be a solution of Problem (2). Then $(\bar{\mathbf{W}}^*,\bar{\mathbf{H}}^*)$ is a solution of Problem (13).*

Proof. This follows from $\check{\mathcal{J}}(\bar{\mathbf{W}},\bar{\mathbf{H}}) = \mathcal{J}(\bar{\mathbf{W}},\bar{\mathbf{H}})$ when $(\bar{\mathbf{W}},\bar{\mathbf{H}}) \in \mathbb{F}$. □

Thanks to Lemma 1, we can solve Problem (13) without norm constraints and renormalize the solution to obtain a solution to Problem (2).

2) Symmetry of the roles of $\mathbf{W}$ *and* $\mathbf{H}$*:* Note that the penalty term $\|\mathbf{\Lambda H}\|_1$ in (13) now depends on $\mathbf{W}$. Interestingly, it can be expanded and written as follows

$$\|\mathbf{\Lambda H}\|_1 = \sum_{k,n} \|\mathbf{w}_k\|_1 h_{k,n} = \sum_{f,k,n} w_{fk} h_{kn} = \sum_{f,k} \|\underline{\mathbf{h}}_k\|_1 w_{fk},$$

where $\underline{\mathbf{h}}_k$ denotes the k^{th} row of $\mathbf{H}$, and the indices f, k, n run from 1 to F, K, N, respectively. This shows that the updates of $\mathbf{H}$ and $\mathbf{W}$ in alternating mininization correspond to equivalent problems: the roles of $\mathbf{H}$ and $\mathbf{W}$ can be exchanged by transposition of $\mathbf{V}$. Next, we describe a block-descent MM algorithm for Problem (13). We start by recalling the principle of MM.

B. Principle of majorization-minimization

MM is a two-step iterative optimization method with a long history and renewed interest, see recent overviews [40], [41]. Let $C(\mathbf{X})$ be a real-valued function to minimize over its domain $\mathbb{E}$, where $\mathbf{X}$ is a matrix variable of arbitrary size. Let $\tilde{\mathbf{X}} \in \mathbb{E}$ be a current iterate. The majorization step of MM consists of building an *auxiliary function* $G(\mathbf{X}|\tilde{\mathbf{X}})$ which is an upper bound of C that is locally tight at $\tilde{\mathbf{X}}$. Mathematically, it must satisfy the two following properties

$$\forall \mathbf{X} \in \mathbb{E}, \quad G(\mathbf{X} \mid \tilde{\mathbf{X}}) \geq C(\mathbf{X}) \tag{14}$$

$$G(\tilde{\mathbf{X}} \mid \tilde{\mathbf{X}}) = C(\tilde{\mathbf{X}})\,. \tag{15}$$

The minimization step of MM consists in minimizing $G(\mathbf{X}|\tilde{\mathbf{X}})$ w.r.t. $\mathbf{X}$, or at least finding an update $\hat{\mathbf{X}}$ such that $G(\hat{\mathbf{X}}|\tilde{\mathbf{X}}) \leq G(\tilde{\mathbf{X}}|\tilde{\mathbf{X}})$. This results in the following descent lemma

$$C(\hat{\mathbf{X}}) \leq G(\hat{\mathbf{X}} \mid \tilde{\mathbf{X}}) \leq G(\tilde{\mathbf{X}} \mid \tilde{\mathbf{X}}) = C(\tilde{\mathbf{X}})\,. \tag{16}$$

As such, MM ensures by design that the objective function C is non-increasing at every iteration. Convergence of the iterates of $\mathbf{X}$ is not straightforward and usually involves problem-dependent assumptions, see, e.g., [19] for NMF. Next, we apply MM to alternating mininization of $\mathbf{W}$ and $\mathbf{H}$ for Problem (13).

C. Construction of an auxiliary function for sparse NMF

In this section we are interested in the minimization of the functions $\mathbf{H} \mapsto \check{\mathcal{J}}(\mathbf{W}, \mathbf{H})$ (with fixed $\mathbf{W}$) and $\mathbf{W} \mapsto \check{\mathcal{J}}(\mathbf{W}, \mathbf{H})$ (with fixed $\mathbf{H}$). As explained at the end of Section IV-A, these two optimization problems are essentially the same and we will only address the first one. Given $\mathbf{W}$, our strategy to build an auxiliary function $G(\mathbf{H}|\tilde{\mathbf{H}})$ for $C(\mathbf{H}) = D_\beta(\mathbf{V}|\mathbf{WH}) + \alpha \|\mathbf{\Lambda H}\|_1$ consists in majorizing the data-fitting and regularization terms separately, and adding up the resulting functions.

1) Majorization of the data-fitting term: Producing an auxiliary function for the data-fitting term $\mathbf{H} \mapsto D_\beta(\mathbf{V}|\mathbf{WH})$ is a well-known problem and we use the existing results of [11], [38], [42]. For all values of β, the data-fitting term can be decomposed into the sum of a convex function and a concave function. The convex term may be majorized using Jensen's inequality while the concave term may be majorized with the tangent inequality. Adding up the two resulting functions results in the auxiliary function $G_\beta(\mathbf{H}|\tilde{\mathbf{H}})$ given in Table I. This procedure has been used in many NMF papers, including [31], and the details can be found in [11].

TABLE I: Expression of the auxiliary function $G_\beta(\mathbf{H}|\tilde{\mathbf{H}})$ for the data-fitting term $\mathbf{H} \mapsto D_\beta(\mathbf{V}|\mathbf{WH})$, up to an additive constant (from [11]). We use the following notations: $\tilde{p}_{kn}$ denotes the elements of the matrix $\mathbf{W}^\top \tilde{\mathbf{S}}_\beta$, where $\tilde{\mathbf{S}}_\beta$ is computed from (6) with $\mathbf{H} = \tilde{\mathbf{H}}$. Similarly, $\tilde{q}_{kn}$ denotes the elements of $\mathbf{W}^\top \tilde{\mathbf{T}}_\beta$.

	$G_\beta\left(\mathbf{H}\middle\|\tilde{\mathbf{H}}\right)$
$\beta < 1$	$\sum_{k,n} \left[\tilde{q}_{kn} h_{kn} - \frac{1}{\beta - 1} \tilde{p}_{kn} \tilde{h}_{kn} \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right)^{\beta - 1} \right]$
$\beta = 1$	$\sum_{k,n} \left[\tilde{q}_{kn} h_{kn} - \tilde{p}_{kn} \tilde{h}_{kn} \log \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right) \right]$
$\beta \in (1, 2]$	$\sum_{k,n} \left[\frac{1}{\beta} \tilde{q}_{kn} \tilde{h}_{kn} \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right)^{\beta} - \frac{1}{\beta - 1} \tilde{p}_{kn} \tilde{h}_{kn} \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right)^{\beta - 1} \right]$
$\beta > 2$	$\sum_{k,n} \left[\frac{1}{\beta} \tilde{q}_{kn} \tilde{h}_{kn} \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right)^{\beta} - \tilde{p}_{kn} \tilde{h}_{kn} \right]$

2) Majorization of the regularization term: We now address the majorization of $S(\mathbf{H}) = \|\mathbf{\Lambda H}\|_1 = \sum_{k,n} \lambda_k h_{kn}$, where we recall that $\lambda_k = \|\mathbf{w}_k\|_1$. We need to distinguish two cases: when $\beta \leq 1$, no majorization of S is actually needed. Indeed, in that case we may use

$$G(\mathbf{H} \mid \tilde{\mathbf{H}}) = G_\beta(\mathbf{H} \mid \tilde{\mathbf{H}}) + \alpha S(\mathbf{H})\,, \tag{17}$$

because $G(\mathbf{H}|\tilde{\mathbf{H}})$ has a simple closed-form minimizer, given in Section IV-D. When $\beta > 1$, this property is not longer true. In that case, we need to majorize $S(\mathbf{H})$ as well. Following [38], we use the following inequality that holds for $h, \tilde{h} > 0$ and $\beta > 1$

$$h \leq \frac{\tilde{h}}{\beta} \left(\frac{h}{\tilde{h}} \right)^{\beta} + \tilde{h} \left(1 - \frac{1}{\beta} \right)\,. \tag{18}$$

Note that the inequality is tight when $h = \tilde{h}$. Applied term to term to $S(\mathbf{H})$, this leads to the following majorizer of the regularization term

$$G_S(\mathbf{H} \mid \tilde{\mathbf{H}}) = \sum_{k,n} \lambda_k \frac{\tilde{h}_{kn}}{\beta} \left(\frac{h_{kn}}{\tilde{h}_{kn}} \right)^{\beta} + \text{cst}\,,$$

where cst contains terms that are constant w.r.t. h_{kn}. [2] In the end, when $\beta > 1$, we use

$$G(\mathbf{H} \mid \tilde{\mathbf{H}}) = G_\beta(\mathbf{H} \mid \tilde{\mathbf{H}}) + \alpha G_S(\mathbf{H} \mid \tilde{\mathbf{H}})\,,$$

which admits a simple closed-form solution given in the next section. The extra majorization step (18) essentially allows $G(\mathbf{H}|\tilde{\mathbf{H}})$ to be composed of monomials of only two different orders, β and $\beta - 1$, hence allowing for closed-form minimization.

[2] We will use the same notation cst in different places to avoid cluttering, though the constants might be different.

D. Minimization of the auxiliary function

The second step of MM consists of minimizing $G(\mathbf{H}|\tilde{\mathbf{H}})$ w.r.t. $\mathbf{H}$. By design, G is smooth, separable and strictly convex and thus we only need to set its gradient to zero (w.r.t. $\mathbf{H}$). This step involves standard calculus and leads in the end to the following update

$$h_{kn} = \tilde{h}_{kn}\left(\frac{\tilde{p}_{kn}}{\tilde{q}_{kn} + \alpha \left\|\mathbf{w}_k\right\|_1}\right)^{\gamma(\beta)},$$

where $\tilde{p}_{kn}$ and $\tilde{q}_{kn}$ are defined in Table I and

$$\gamma(\beta) = \begin{cases} \frac{1}{2-\beta} & \text{if } \beta < 1 \\ 1 & \text{if } \beta \in [1,2] \\ \frac{1}{\beta-1} & \text{if } \beta > 2 \end{cases}.$$

As explained before, a similar update can be derived for w_{fk} by exchanging the roles of $\mathbf{W}$ and $\mathbf{H}$. In the end, this leads to the following multiplicative matrix updates

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^\top \mathbf{S}_\beta}{\mathbf{W}^\top(\mathbf{T}_\beta + \alpha \mathbf{1}_{F\times N})}\right)^{\cdot\gamma(\beta)} \tag{19}$$

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \left(\frac{\mathbf{S}_\beta \mathbf{H}^\top}{(\mathbf{T}_\beta + \alpha \mathbf{1}_{F\times N})\mathbf{H}^\top}\right)^{\cdot\gamma(\beta)}. \tag{20}$$

A pseudo-code of the resulting procedure, coined MM-SNMF-ℓ_1, is given in Algorithm 1. A few comments are in order. First, as in standard NMF practice, only one step of MM is applied to $\mathbf{H}$ and $\mathbf{W}$ in each iteration. Applying several sub-iterations brings no benefit in practice. Like $\mathcal{J}$, $\tilde{\mathcal{J}}$ or $\mathcal{L}$, the objective function $\check{\mathcal{J}}$ is non-convex w.r.t. $\mathbf{W}$ and $\mathbf{H}$. When $\beta \in [1,2]$, the individual sub-problems in $\mathbf{W}$ and $\mathbf{H}$ are convex, but $\check{\mathcal{J}}$ is still jointly non-convex. As such, initialization matters and we will present average performance results over several random initializations in Section VI. Several data-dependent initialization schemes are presented in [9]. In the next section we discuss the convergence of the iterates produced by Algorithm 1.

Algorithm 1 MM-SNMF-ℓ_1

Input: Nonnegative matrix $\mathbf{V}$ and initialization $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$.
Output: Nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{V} \approx \mathbf{WH}$ with sparse $\mathbf{H}$.

1: Initialize i to 0.
2: Initialize $(\mathbf{W}_i, \mathbf{H}_i)$ to $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$.
3: **repeat**
4: Update $\mathbf{H}_i$ using (19):

$$\tilde{\mathbf{V}} \leftarrow \mathbf{W}_i\mathbf{H}_i$$

$$\mathbf{H}_{i+1} \leftarrow \mathbf{H}_i \odot \left(\frac{\mathbf{W}_i^\top\left(\mathbf{V} \odot \tilde{\mathbf{V}}^{\cdot(\beta-2)}\right)}{\mathbf{W}_i^\top(\tilde{\mathbf{V}}^{\cdot(\beta-1)} + \alpha \mathbf{1}_{F\times N})}\right)^{\cdot\gamma(\beta)}$$

5: Update $\mathbf{W}_i$ using (20):

$$\tilde{\mathbf{V}} \leftarrow \mathbf{W}_i\mathbf{H}_{i+1}$$

$$\mathbf{W}_{i+1} \leftarrow \mathbf{W}_i \odot \left(\frac{\left(\mathbf{V} \odot \tilde{\mathbf{V}}^{\cdot(\beta-2)}\right)\mathbf{H}_i^\top}{(\tilde{\mathbf{V}}^{\cdot(\beta-1)} + \alpha \mathbf{1}_{F\times N})\mathbf{H}_i^\top}\right)^{\cdot\gamma(\beta)}$$

6: Increment i.
7: **until** stopping criterion is met
8: Rescale $\mathbf{W}_i$ and $\mathbf{H}_i$:

$$\mathbf{\Lambda} \leftarrow \operatorname{Diag}\left(\left(\left\|\mathbf{w}_k\right\|_1\right)_{k\in[\![1,K]\!]}\right)$$

$$(\mathbf{W}_i, \mathbf{H}_i) \leftarrow (\mathbf{W}_i\mathbf{\Lambda}^{-1}, \mathbf{\Lambda}\mathbf{H}_i)$$

9: **return** $(\mathbf{W}_i, \mathbf{H}_i)$

E. Convergence

By construction, the sequence of objective values produced by Algorithm 1 in non-increasing. Because $\check{\mathcal{J}}$ is bounded below by zero, the sequence thus converges. The following theorem additionally states the convergence of the iterates.

Theorem 1. *For any data matrix $\mathbf{V} \in \mathbb{R}_+^{F\times N}$, rank $K \in \mathbb{N}$ and regularization parameter $\alpha > 0$, the sequence of iterates $\{\mathbf{W}_i, \mathbf{H}_i\}_{i\in\mathbb{N}}$ generated by Algorithm 1 converges to the set of stationary points of Problem* (13). [3]

Proof. The authors in [19] prove the convergence of the iterates of a block-descent MM algorithm constructed like Algorithm 1 for the following problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \quad D_\beta(\mathbf{V} \mid \mathbf{WH}) + \alpha_1 \left\|\mathbf{H}\right\|_1 + \alpha_2 \left\|\mathbf{W}\right\|_1.$$

[3]Due to the coercivity and the continuity of $\check{\mathcal{J}}$, the sequence $\{\mathbf{W}_i, \mathbf{H}_i\}_{i\in\mathbb{N}}$ has at least one limit point. As such, the convergence of $\{\mathbf{W}_i, \mathbf{H}_i\}_{i\in\mathbb{N}}$ to the set of stationary points means that every limit point of $\{\mathbf{W}_i, \mathbf{H}_i\}_{i\in\mathbb{N}}$ is a stationary point. A stationary point is defined as a feasible point for which the necessary optimality condition given by Euler's inequality holds.

As a matter of fact, their proof of convergence can be applied step by step to our own block-descent MM approach for solving Problem (13). Indeed, the auxiliary functions that we derived for $\mathbf{H}$ and equivalently for $\mathbf{W}$ satisfy the five properties required in [19, Definition 2] to establish convergence. Using $G(\mathbf{H}|\tilde{\mathbf{H}})$ for exposition, the five properties are the following.

- Property 1 and 2 correspond to Equations (14) and (15) that define a valid auxiliary function.
- Property 3 dictates that $\nabla_{h_{kn}} G(\mathbf{H}|\tilde{\mathbf{H}})$ is a function of $h_{kn}/\tilde{h}_{kn}$.
- Property 4 dictates that the directional derivatives of $G(\mathbf{H}|\tilde{\mathbf{H}})$ and $\check{\mathcal{J}}(\mathbf{H})$ coincide at $\mathbf{H} = \tilde{\mathbf{H}}$.
- Property 5 dictates that $G(\mathbf{H}|\tilde{\mathbf{H}})$ is strictly convex w.r.t. $\mathbf{H}$.

Standard calculus and convex analysis show that Properties 3–5 are satisfied by $G(\mathbf{H}|\tilde{\mathbf{H}})$, whereas Properties 1–2 are satisfied by construction. The results of [19] also require $\check{\mathcal{J}}$ to be coercive which is satisfied thanks to the regularization term $\|\mathbf{\Lambda H}\|_1$. □

V. Extension to NMF with the β-divergence and log-regularization

In this section, we extend the methodology of Section III-B and Section IV to sparse β-NMF with log-regularization. More precisely, we are interested in solving

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \mathcal{J}_{\log}(\mathbf{W}, \mathbf{H}) \quad \text{s.t. } (\forall k \in [\![1,K]\!], \left\|\mathbf{w}_k\right\|_1 = 1), \tag{21}$$

where

$$\mathcal{J}_{\log}(\mathbf{W},\mathbf{H}) \stackrel{\text{def}}{=} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \alpha \sum_{k,n} \log(h_{kn}+\epsilon)\,. \quad (22)$$

The log-regularization term $\psi(x) = \log(|x|+\epsilon)$ used in (22) was popularized by [43] for sparse linear regression ($x \in \mathbb{R}$). For small positive ϵ, this function is much sharper at the origin than the ℓ_1 norm. As such, it accentuates the sparsity of the solutions, which can be necessary or desired in practice. In the context of NMF, it was considered in [22]–[24]. Following the discussion in Section II-B, the unit-norm constraints in (21) ensure that the minimization problem is well-posed.

Changing the regularization term in $\mathcal{J}(\mathbf{W},\mathbf{H})$ only influences the update of $\mathbf{H}$ in the Lagrangian and heuristic methods described in Sections III-A and III-B. Indeed, the update of $\mathbf{W}$ is unchanged given $\mathbf{H}$. This is not true with our approach described in Section IV because the regularization term in the reformulated scale-invariant objective function depends on both $\mathbf{W}$ and $\mathbf{H}$. Furthermore, under the log-regularization term, the minimization problems w.r.t. $\mathbf{W}$ and $\mathbf{H}$ are not exchangeable anymore but can still be handled in the MM framework.

In Section V-A, we first extend the method of [30] to Problem (21) and derive heuristic multiplicative updates that appear to work in practice. Then, we derive our principled block-descent MM algorithm in Section V-B. The Lagrangian method from [31] could be also extended to Problem (21) for values of $\beta \leq 1$ by combining the update of $\mathbf{W}$ in Section III-A with our update of $\mathbf{H}$ in Section V-B. However this does not change the conclusions of Section III-A about the limitations of the Lagrangian approach and we chose to omit this method in our experimental comparisons when considering log-regularization.

A. Heuristic multiplicative updates

We adapt the approach of [30] by replacing $\mathbf{W}$ with $\mathbf{W}\mathbf{\Lambda}^{-1}$ in (22) and using alternating multiplicative updates of $\mathbf{W}$ and $\mathbf{H}$ derived using the heuristic (9). By standard calculus, this results in the following updates

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top \mathbf{S}_\beta}{\mathbf{W}^\top \mathbf{T}_\beta + \frac{\alpha}{\mathbf{H}+\epsilon}} \quad (23)$$

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \frac{\mathbf{S}_\beta \mathbf{H}^\top + \mathbf{1}_{F\times F}(\mathbf{W} \odot \mathbf{T}_\beta \mathbf{H}^\top)}{\mathbf{T}_\beta \mathbf{H}^\top + \mathbf{1}_{F\times F}(\mathbf{W} \odot \mathbf{S}_\beta \mathbf{H}^\top)} \quad (24)$$

$$\mathbf{W} \longleftarrow \mathbf{W}\mathbf{\Lambda}^{-1}, \quad (25)$$

where $\mathbf{S}_\beta$ and $\mathbf{T}_\beta$ are defined in (6) and (7). As stated before, only the update of $\mathbf{H}$ is changed when compared to the updates derived in Section III-B for β-NMF with ℓ_1 regularization.

B. Block-descent majorization-minimization algorithm

We now apply the methodology of Section IV to Problem (21). Following Section IV-A we can show that Problem (21) is equivalent to

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \quad \check{\mathcal{J}}_{\log} \stackrel{\text{def}}{=} D_\beta(\mathbf{V} \mid \mathbf{WH}) + \alpha \sum_{k,n} \psi\left(\lambda_k h_{k,n}\right)\,, \quad (26)$$

where we recall that $\lambda_k = \|\mathbf{w}_k\|_1$. As opposed to $\check{\mathcal{J}}$, the roles of $\mathbf{W}$ and $\mathbf{H}$ are not exchangeable anymore in $\check{\mathcal{J}}_{\log}$ and we now proceed to derive separate MM updates for the two factors.

1) Update of $\mathbf{H}$*:* Given $\mathbf{W}$, we start by constructing an auxiliary function $G(\mathbf{H}|\tilde{\mathbf{H}})$ for the function $C(\mathbf{H}) = D_\beta(\mathbf{V}|\mathbf{WH}) + \alpha S(\mathbf{H})$, where $S(\mathbf{H}) = \sum_{k,n} \psi\left(\lambda_k h_{k,n}\right)$. We use the same notations C, S and G as in Section IV in order to avoid cluttering. We majorize the data-fitting term with the same function $G_\beta(\mathbf{H}|\tilde{\mathbf{H}})$ than before, given in Table I. We now turn to the majorization of $S(\mathbf{H})$.

By concavity of the logarithm, the individual summands of $S(\mathbf{H})$ can be majorized locally at $\tilde{\mathbf{H}}$ with the tangent inequality

$$\psi(\lambda_k h_{kn}) \leq \psi(\lambda_k \tilde{h}_{kn}) + \lambda_k \psi'(\lambda_k \tilde{h}_k)(h_{kn} - \tilde{h}_{kn})\,, \quad (27)$$

where $\psi'(x) = 1/(x+\epsilon)$ for all $x \in \mathbb{R}_+$. From there, we need to distinguish two cases like in Section IV-C.

When $\beta \leq 1$, we may simply apply (27) to and use the following auxiliary function for $S(\mathbf{H})$

$$G_S(\mathbf{H}|\tilde{\mathbf{H}}) = \sum_{k,n} \frac{h_{kn}}{\tilde{h}_{kn} + \frac{\epsilon}{\lambda_k}} + \text{cst}\,,$$

where cst contains terms that are constant w.r.t. h_{kn}. This leads to an auxiliary function $G(\mathbf{H}|\tilde{\mathbf{H}}) = G_\beta(\mathbf{H}|\tilde{\mathbf{H}}) + \alpha G_S(\mathbf{H}|\tilde{\mathbf{H}})$ that has a simple closed-form minimizer when $\beta \leq 1$. The minimization is infeasible when $\beta > 1$ and we need to resort to an additional majorization step, using again (18). This leads to the following auxiliary function for $S(\mathbf{H})$

$$G_S(\mathbf{H},\tilde{\mathbf{H}}) = \frac{1}{\beta} \sum_{k,n} \frac{\tilde{h}_{kn}}{\tilde{h}_{kn} + \frac{\epsilon}{\lambda_k}} \left(\frac{h_{kn}}{\tilde{h}_{kn}}\right)^\beta + \text{cst}\,.$$

In the end, for all $\beta \in \mathbb{R}$, G is a smooth, separable and strictly convex function that is easily minimized by setting its gradient to zero. This leads to the following multiplicative update

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^\top \mathbf{S}_\beta}{\mathbf{W}^\top \mathbf{T}_\beta + \frac{\alpha}{\mathbf{H}+\frac{\epsilon}{\mathbf{\Upsilon}}}}\right)^{\cdot\gamma(\beta)}, \quad (28)$$

where $\mathbf{\Upsilon} = \mathbf{W}^\top \mathbf{1}_{F\times N}$.

2) Update of $\mathbf{W}$*:* A very similar strategy can be employed for the update of $\mathbf{W}$. Given $\mathbf{H}$, we now need to build an auxiliary function $F(\mathbf{W}|\tilde{\mathbf{W}})$ for the minimization of $B(\mathbf{W}) = D_\beta(\mathbf{V}|\mathbf{WH}) + \alpha R(\mathbf{W})$, where $R(\mathbf{W}) = \sum_{k,n} \psi(h_{kn} \|\mathbf{w}_k\|_1)$. The data-fitting term can be majorized by switching the roles of $\mathbf{W}$ and $\mathbf{H}$ in Table I; slightly abusing the notations again we denote the resulting auxiliary function by $G_\beta(\mathbf{W}|\tilde{\mathbf{W}})$. Let us now address the majorization of $R(\mathbf{W})$. Given the current update $\tilde{\mathbf{W}}$, we may again invoke the concavity of $\psi(x)$ to form the following inequality

$$\begin{aligned}\psi(h_{kn} \|\mathbf{w}_k\|_1) &\leq \psi(h_{kn} \|\tilde{\mathbf{w}}_k\|_1) + \frac{\|\mathbf{w}_k\|_1 - \|\tilde{\mathbf{w}}_k\|_1}{\|\tilde{\mathbf{w}}_k\|_1 + \frac{\epsilon}{h_{kn}}} \\ &= \frac{1}{\|\tilde{\mathbf{w}}_k\|_1 + \frac{\epsilon}{h_{kn}}} \sum_f w_{fk} + \text{cst}\,. \quad (29)\end{aligned}$$

From there, we use a path that is similar to the update of $\mathbf{H}$. When $\beta \leq 1$, we may simply apply inequality (29) to the summands of $R(\mathbf{W})$ and use the following majorizer

$$F_R(\mathbf{W}|\tilde{\mathbf{W}}) = \sum_{f,k} \left(\sum_n \frac{1}{\|\tilde{\mathbf{w}}_k\|_1 + \frac{\epsilon}{h_{kn}}} \right) w_{fk} + \text{cst}\,.$$

When $\beta > 1$ we need to further majorize the terms w_{fk} using (18). In the end, the minimization of $F(\mathbf{W}|\tilde{\mathbf{W}}) = G_\beta(\mathbf{W}|\tilde{\mathbf{W}}) + \alpha F_R(\mathbf{W}|\tilde{\mathbf{W}})$ leads to the following multiplicative update of $\mathbf{W}$

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \left(\frac{\mathbf{S}_\beta \mathbf{H}^\top}{\mathbf{T}_\beta \mathbf{H}^\top + \mathbf{1}_{F\times N}\left(\frac{\alpha}{\boldsymbol{\Upsilon} + \frac{\epsilon}{\mathbf{H}}}\right)^\top} \right)^{\cdot\gamma(\beta)} . \quad (30)$$

3) Resulting algorithm and convergence: Our resulting MM algorithm to address (26) is displayed in Algorithm 2 and is referred to as MM-SNMF-log. By design, the algorithm ensures that the sequence of objective values $(\check{\mathcal{J}}_{\log}(\mathbf{W}_i, \mathbf{H}_i))_{i\in\mathbb{N}}$ is non-increasing and convergent. The convergence of the iterates can also be proven, using the same rationale as the proof of Theorem 1. We simply need G and F to verify the five properties listed in the proof, which is easily checked. Properties 1–3 hold by construction of the auxiliary functions, Property 4 can be verified using standard calculus, and Property 5 results from convexity properties. In the end, we have derived the first universal algorithm for sparse β-NMF with log-regularization. The algorithm is simple to implement, has linear complexity per iteration and can be applied for any value of $\beta \in \mathbb{R}$. It is free of tuning parameters and enjoys strong convergence properties.

Algorithm 2 MM-SNMF-log

Input: Nonnegative matrix $\mathbf{V}$ and initialization $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$.
Output: Nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{V} \approx \mathbf{WH}$ with sparse $\mathbf{H}$.

1: Initialize i to 0.
2: Initialize $(\mathbf{W}_i, \mathbf{H}_i)$ to $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$.
3: **repeat**
4: $\quad \boldsymbol{\Upsilon} \leftarrow \mathbf{W}_i^\top \mathbf{1}_{F\times N}$
5: $\quad$ Update $\mathbf{H}_i$ using (28):

$$\tilde{\mathbf{V}} \leftarrow \mathbf{W}_i \mathbf{H}_i$$
$$\mathbf{H}_{i+1} \leftarrow \mathbf{H}_i \odot \left(\frac{\mathbf{W}_i^\top \left(\mathbf{V} \odot \tilde{\mathbf{V}}^{\cdot(\beta-2)} \right)}{\mathbf{W}_i^\top \tilde{\mathbf{V}}^{\cdot(\beta-1)} + \frac{\alpha}{\mathbf{H}_i + \frac{\epsilon}{\boldsymbol{\Upsilon}}}} \right)^{\cdot\gamma(\beta)}$$

6: $\quad$ Update $\mathbf{W}_i$ using (30):

$$\tilde{\mathbf{V}} \leftarrow \mathbf{W}_i \mathbf{H}_{i+1}$$
$$\mathbf{W}_{i+1} \leftarrow \mathbf{W}_i \odot \left(\frac{\left(\mathbf{V} \odot \tilde{\mathbf{V}}^{\cdot(\beta-2)} \right) \mathbf{H}_{i+1}^\top}{\tilde{\mathbf{V}}^{\cdot(\beta-1)} \mathbf{H}_{i+1}^\top + \mathbf{1}\left(\frac{\alpha}{\boldsymbol{\Upsilon} + \frac{\epsilon}{\mathbf{H}_{i+1}}} \right)^\top} \right)^{\cdot\gamma(\beta)}$$

7: $\quad$ Increment i.
8: **until** stopping criterion is met
9: Rescale $\mathbf{W}_i$ and $\mathbf{H}_i$:

$$\boldsymbol{\Lambda} \leftarrow \text{Diag}\left((\|\mathbf{w}_k\|_1)_{k\in[\![1,K]\!]} \right)$$
$$(\mathbf{W}_i, \mathbf{H}_i) \leftarrow (\mathbf{W}_i \boldsymbol{\Lambda}^{-1}, \boldsymbol{\Lambda} \mathbf{H}_i)$$

10: **return** $(\mathbf{W}_i, \mathbf{H}_i)$

VI. Experimental Results

In this section, we compare our MM methods against the Lagrangian and the heuristic methods presented in Section III on four different datasets. We first give an example showing that the heuristic is not a descent algorithm in contrast with our MM method. We then compare MM-SNMF-ℓ_1 described by Algorithm 1 for solving Problem (2) with its Lagrangian and heuristic counterparts presented in Sections III-A and III-B. We refer to the latter two methods as L-SNMF-ℓ_1 and H-SNMF-ℓ_1 respectively. We finally compare our algorithm MM-SNMF-log described by Algorithm 2 with H-SNMF-log which is the variant of the heuristic method given in Section V-A and aimed at solving Problem (21).

A. Description of the datasets and hyperparameter choices

To compare the algorithms under realistic conditions, we select four different datasets coming from various applications that are described below.

- The Olivetti dataset from AT&T Laboratories Cambridge [44] containing 400 greyscale images of faces with dimensions 64×64 that are vectorized and stored as the columns of $\mathbf{V}$. From these images, NMF can be used to learn part-based features that are represented by the dictionary of features $\mathbf{W}$ [2]. The factor $\mathbf{H}$ then contains the activation encodings of the features for each image of the collection.
- An audio magnitude spectrogram generated from an excerpt of the original recording of the song "Four on Six" by Wes Montgomery. The signal corresponds to the first five seconds of the song sampled at $44.1\,\text{kHz}$. The spectrogram is then computed with a Hamming window of length 1024 (23ms) and with an overlap of 50%. The use of NMF in this context consists in extracting elementary audio time-frequency patterns represented in $\mathbf{W}$ with their temporal activations given by $\mathbf{H}$ [3].
- A hyperspectral image with resolution 50×50 pixels over 189 spectral bands acquired over Moffett Field in 1997 by the Airborne Visible Infrared Imaging Spectrometer [45]. Using NMF on such an image allows extracting a dictionary $\mathbf{W}$ of individual spectra representing the different materials, as well as their relative proportions stored in $\mathbf{H}$ [5].
- The TasteProfile dataset [46] containing counts of songs played by users of a music streaming service. In this context, NMF may extract the user preferences represented by the matrix $\mathbf{W}$ as well as the different song attributes represented by the matrix $\mathbf{H}$ [6]. We apply a preprocessing to the dataset similarly to [47] and many other papers using this dataset: we keep only users and songs with more than twenty interactions. The latter preprocessing still resulting in a large and highly sparse

TABLE II: Dimensions of the datasets used in our experiments

	F	N	K	β	α_{ℓ_1}	$\alpha_{\log}$
Olivetti	4 096	400	10	1	0.01	5
Spectrogram	513	858	10	$\{0, 0.5\}$	$\{600, 5\}$	$\{0.5, 5\}$
TasteProfile	16 301	12 118	50	1	5000	0.5
Moffett	189	2 500	3	$\{1.3, 2\}$	$\{1000, 0.05\}$	$\{0.5, 0.02\}$

dataset.

Table II displays the dimensions of each dataset together with the values of β and α that we have used in our experiments. The value of α_{ℓ_1} is used for Problem (2) while the value of $\alpha_{\log}$ is used for Problem (21). The constant ϵ in the log-regularization is set to 0.01. For the spectrogram and Moffett dataset, we perform tests with two different values of β and thus use different values of α accordingly. Note that we have tested several values of the regularization parameter α and we have chosen one that yields representative results. In particular, for the chosen values, the regularization does not become negligible in comparison with the data term and conversely.

The values of β have been chosen according to standard practice, see, e.g., [11] and references in Section II-A. Values of β in the $[0, 0.5]$ interval are recommended for audio spectra as they give more importance to small-energy coefficients. The value $\beta = 1$ produces a data-fitting term that corresponds to the log-likelihood of a Poisson model; this fits well with integer-valued data such as counts (TasteProfile) or RGB images (Olivetti). Values of β in the $[1, 2]$ interval offer a good compromise between Poisson and additive Gaussian noise assumptions, which suits well to hyperspectral data (Moffett).

B. Set-up

All the simulations presented in this section have been conducted in Matlab 2021a running on an Intel i7-8650U CPU with a clock cycle of 1.90GHz shipped with 16GB of memory.[4]

For each dataset, we compare the factorization obtained by the different methods from 50 different initializations. The elements of $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$ are drawn randomly according to a half-normal distribution obtained by folding a centered Gaussian distribution of standard deviation equal to 5.

1) Stopping criterion: The following stopping criterion has been used for all the algorithms

$$\frac{|\mathcal{J}(\mathbf{W}^-, \mathbf{H}^-) - \mathcal{J}(\mathbf{W}, \mathbf{H})|}{|\mathcal{J}(\mathbf{W}, \mathbf{H})|} \leq \delta \,, \tag{31}$$

where δ is a tolerance set to 10^{-5}, $\mathbf{W}$ and $\mathbf{H}$ are the current iterates while $\mathbf{W}^-$ and $\mathbf{H}^-$ are the previous ones. The regularization term in $\mathcal{J}$ depends on the context and is either the ℓ_1 or the log-regularization. The absolute value at the denominator is necessary in the case of the log-regularization since the logarithm function, and thus $\mathcal{J}$, could be negative. If the convergence is not reached after $5,000$ iterations, we stop the algorithm and return the current estimated factor matrices.

[4]Matlab code will be made publicly available at the time of publication.

2) Implementation: The pseudo-code for MM-SNMF-ℓ_1 and MM-SNMF-log is shown in Algorithms 1 and 2 respectively. The implementation of H-SNMF-ℓ_1 and H-SNMF-log is similar except that the multiplicative rules are replaced by the ones given by (10), (11) and (23), (24) respectively, together with the renormalization (12).

The implementation of L-SNMF-ℓ_1 follows the one of MM-SNMF-ℓ_1 but uses the multiplicative updates given by (4) and (5) instead of (19) and (20). Furthermore, an additional step consisting in computing the optimal Lagrangian multipliers has to be performed between the steps 4 and 5 of Algorithm 1.

The β-divergence $d_\beta(x|y)$ is not always well defined when x or y takes the value zero due to the presence of quotients and logarithms. Consequently, we use in practice $D_\beta(\mathbf{V}+\kappa|\mathbf{WH}+\kappa)$ with a small constant κ instead of the objective function $D_\beta(\mathbf{V}|\mathbf{WH})$ for numerical stability. For the heuristic method, this leads to replace $\mathbf{WH}$ by $\mathbf{WH}+\kappa$ in the expression of the gradient and thus in the updates (10) and (11). For our method and the Lagrangian one (which both rely on MM), we can also safely replace $\mathbf{WH}$ by $\mathbf{WH}+\kappa$ in the multiplicative updates. This can be proven by treating κ as a $(K+1)^{\text{th}}$ constant component in the derivations like in [36].

3) Performance evaluation: We compare the different algorithms with two metrics: their computational efficiency (CPU time) and the quality of the returned solutions. The latter is assessed by the value of the normalized objective function $\mathcal{J}(\mathbf{W}, \mathbf{H})/FN$ at the solution returned by the algorithms.

C. Results

1) Descent property: We illustrate in this section that H-SNMF-ℓ_1 is not a descent algorithm unlike MM-SNMF-ℓ_1. To this end, we generate a random data matrix $\mathbf{V}$ of dimension 50×40 by drawing its elements according to the same half-normal distribution used for drawing the elements of $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$. We apply both H-SNMF-$\ell_1$ and MM-SNMF-ℓ_1 with $K = 3$, $\beta = -0.5$, and $\alpha = 5$. Then, we plot the values of the normalized objective function for the first fifty iterations in Figure 1. We use the same initialization for both methods but do not plot the value of the objection function at the initialization (iteration 0) for the sake of clarity. We observe that the blue curve corresponding to MM-SNMF-ℓ_1 is non-increasing whereas the red curve representing H-SNMF-ℓ_1 is oscillating. This example demonstrates the theoretical advantage to use MM-SNMF-ℓ_1 over H-SNMF-ℓ_1. Furthermore, we observe on this example that H-SNMF-ℓ_1 requires more iterations than MM-SNMF-ℓ_1 to reach a given value of the objective function close to a local optimum. This observation will be verified in the experiments on the datasets in the next section.

2) Performance comparison for Problem (2)*:* We now run H-SNMF-ℓ_1, L-SNMF-ℓ_1, and MM-SNMF-ℓ_1 on the four datasets. The average values of the normalized objective function $\mathcal{J}/FN$ at the solutions returned by the three algorithms are given in the top part of Table III. We observe first that optimal values of the objective function do not vary much with the initialization for the three methods. Moreover, we notice

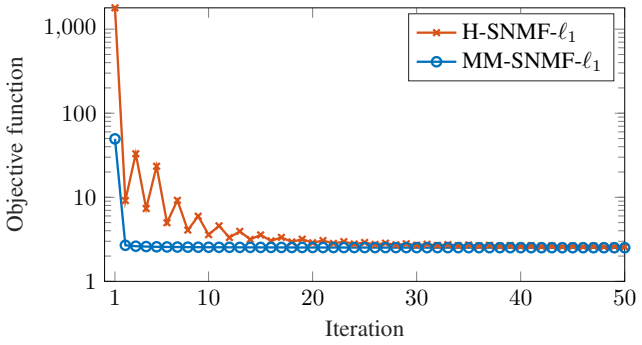


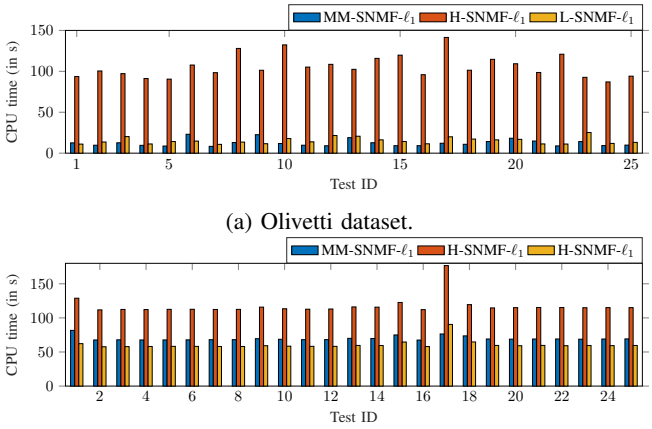
Fig. 1: Values of the normalized objective function through the first hundred of iterations. Results obtained on synthetic data matrix $\mathbf{V}$ with parameters $(F, N) = (50, 40)$, $K = 3$, $\beta = -0.5$.

that H-SNMF-ℓ_1 and MM-SNMF-ℓ_1 yield solutions with a similar quality while L-SNMF-ℓ_1 may return higher-quality solutions.

On Figures 2, 3, and 4, we show the CPU time used by each method before convergence. For the sake of clarity, we only show the first 25 realizations, the behaviour of the others is similar. We observe that the efficiency of the methods in term of CPU time depends on the dataset, on the value of β and on the initialization (e.g., for Moffett or the spectrogram datasets). However, we notice some general trends: for example, H-NMF-ℓ_1 is always the slowest on Olivetti and TasteProfile datasets. The middle part of Table III exposes this trend by displaying the corresponding average run times together with their standard deviations shown within parentheses. We observe that H-SNMF-ℓ_1 is the slowest in average for every dataset while MM-SNMF-ℓ_1 is the fastest one except for TasteProfile for which L-SNMF-ℓ_1 is 13% faster.

The three algorithms do not follow the same path in the optimization space. In particular, we can observe in the bottom part of Table III that MM-SNMF-ℓ_1 converges in less iterations than H-SNMF-ℓ_1. Since the complexity per iteration of these methods are similar—compare the multiplicative updates (10) with (19) and (11) with (20)—this explains the observed difference in CPU time. Furthermore, one can see that L-SNMF-ℓ_1 converges in a smaller number of iterations than MM-SNMF-ℓ_1 for some datasets. However, its iterations are more expensive due to the update of the Lagrangian multipliers through a Newton-Raphson iterative method.

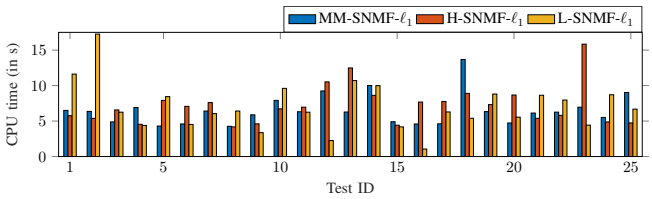
3) Performance comparison for Problem (21)*:* We now compare H-SNMF-log with MM-SNMF-log. Similarly to the previous section, the statistics on the values of the objective function, on the CPU times, and on the numbers of iterations are shown in Table IV while Figures 5, 6, and 7 display the CPU time for the first 25 Monte-Carlo realizations. Results similar to the ℓ_1-regularization can be observed: both methods return solutions of nearly same quality whereas MM-SNMF-ℓ_1 is significantly faster for all datasets except for the spectrogram, for which both methods use in average the same CPU time. This difference in CPU time is explained by the number of iterations before convergence as shown in Table IV: MM-SNMF-ℓ_1 takes on average about 20% to 50% less iterations than H-SNMF-ℓ_1 to converge. The difference in CPU time is particularly significant for the large scale dataset TasteProfile.



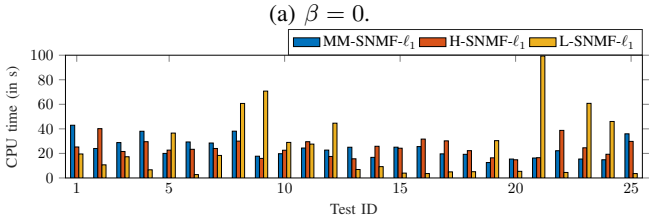
(a) Olivetti dataset.

(b) TasteProfile dataset.

Fig. 2: Comparative performance with Olivetti and TasteProfile datasets using the ℓ_1-regularization ($\beta = 1$).

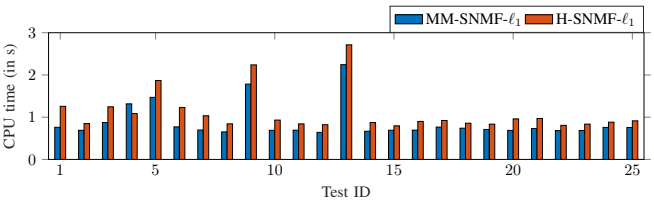


(a) $\beta = 0$.

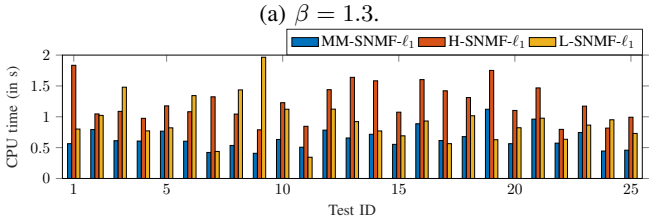


(b) $\beta = 0.5$.

Fig. 3: Comparative performance with a spectrogram using the ℓ_1-regularization.



(a) $\beta = 1.3$.

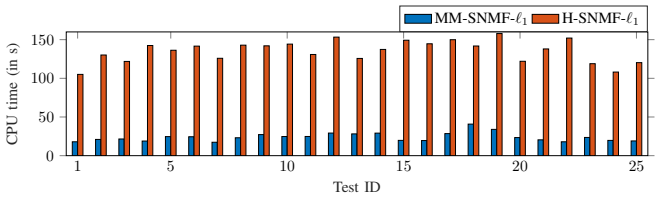


(b) $\beta = 2$.

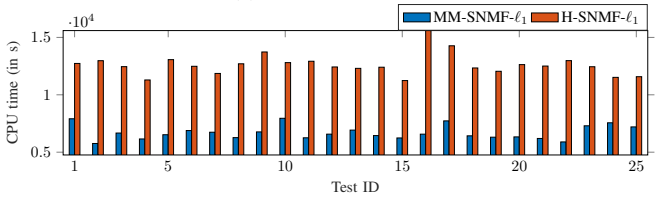
Fig. 4: Comparative performance with Moffett dataset using the ℓ_1-regularization.

TABLE III: Statistics for the three algorithms designed to solve Problem (2). The top section shows the average values of the objective function $\mathcal{J}/FN$ at the returned solutions. The middle section gives the average CPU times with the lowest ones highlighted in bold. The bottom section yields the corresponding average number of iterations. Standard deviations are given within parentheses.

	L-SNMF-ℓ_1	H-SNMF-ℓ_1	MM-SNMF-ℓ_1
		Objective function	
Olivetti	3.16 (±7E-3)	3.16 (±9E-3)	3.16 (±6E-3)
Spectrogram ($\beta = 0$)	0.88 (±3E-2)	20.3 (±3E-2)	20.3 (±3E-2)
Spectrogram ($\beta = 0.5$)	0.60 (±3E-2)	2.98 (±7E-3)	2.98 (±6E-3)
TasteProfile	0.76 (±7E-6)	9.15 (±5E-6)	9.15 (±5E-6)
Moffet ($\beta = 1.3$)	—	0.17 (±2E-5)	0.17 (±2E-4)
Moffet ($\beta = 2$)	4.1E-3 (±7E-3)	4.6E-3 (±1E-2)	4.6E-3 (±7E-3)
		CPU time	
Olivetti	14.1s (± 3.8)	103.4s (±16.8)	**11.2s** (±3.5)
Spectrogram ($\beta = 0$)	6.5s (± 3.4)	6.4s (± 2.2)	**6.3s** (±1.8)
Spectrogram ($\beta = 0.5$)	23.5s (± 2.5)	24.5s (± 6.3)	**22.5s** (±7.2)
TasteProfile	**60.6s** (± 6.5)	117.5s (±12.9)	69.8s (±3.4)
Moffet ($\beta = 1.3$)	—	1.2s (± 0.6)	**0.9s** (±0.4)
Moffet ($\beta = 2$)	0.8s (± 0.3)	1.1s (± 0.3)	**0.6s** (±0.1)
		Number of iterations	
Olivetti	763 (±112)	947 (± 99)	767 (±111)
Spectrogram ($\beta = 0$)	219 (±109)	160 (± 46)	239 (± 55)
Spectrogram ($\beta = 0.5$)	197 (± 93)	144 (± 37)	183 (± 55)
TasteProfile	14 (± 0)	23 (± 0)	23 (± 0)
Moffet ($\beta = 1.3$)	—	12 (± 6)	11 (± 6)
Moffet ($\beta = 2$)	137 (± 25)	133 (± 49)	95 (± 19)



(a) Olivetti dataset.

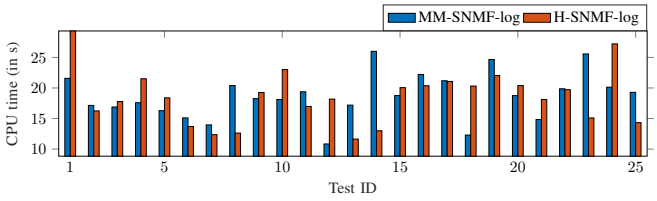


(b) TasteProfile dataset.

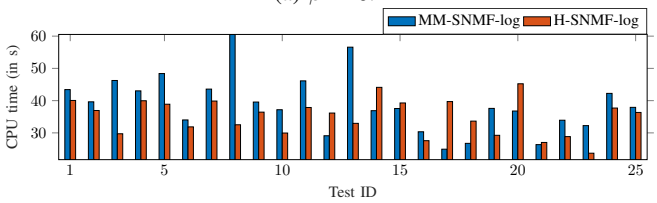
Fig. 5: Comparative performance with Olivetti and TasteProfile datasets using the log-regularization ($\beta = 1$).

VII. Conclusion

We have presented a block-descent MM algorithm for β-NMF with ℓ_1-regularization or log-regularization on one factor and unit ℓ_1-norm constraint on the columns of the other. Our algorithm takes the form of iterative multiplicative updates with are simple and efficient to compute. In contrast with state-of-the-art methods, our resulting algorithm can be applied to every β-divergence and owns desirable theoretical properties such as non-increasingness, convergence of the objective function as well as the convergence of its iterates to the set of stationary points of the problem. Furthermore, we have observed experimentally that our MM algorithm estimates factors with competitive quality and leads in many cases to a significant decrease of CPU time when compared to state-of-the-art methods.

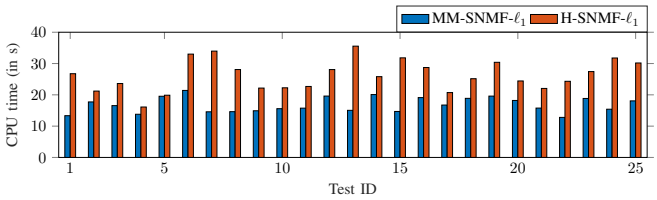


(a) $\beta = 0$.

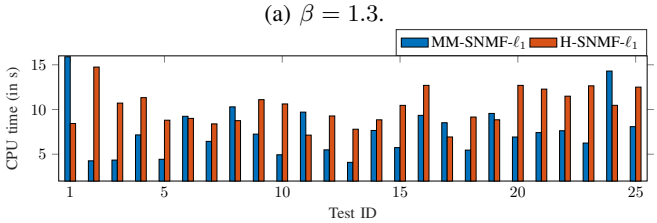


(b) $\beta = 0.5$.

Fig. 6: Comparative performance with a spectrogram using the log-regularization.



(a) $\beta = 1.3$.



(b) $\beta = 2$.

Fig. 7: Comparative performance with Moffett dataset using the log-regularization.

References

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun. 1994.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[3] P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 66–75, May 2014.

[4] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 155–173, Sep. 2007.

[5] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

TABLE IV: Statistics for the three algorithms designed to solve Problem (21). The top section shows the average values of the objective function $\mathcal{J}/FN$ at the returned solutions. The middle section gives the average CPU times with the lowest ones highlighted in bold. The bottom section yields the corresponding average number of iterations. Standard deviations are given within parentheses. Average numbers of iterations to solve Problem (21).

	H-SNMF-log	MM-SNMF-log
	Objective function	
Olivetti	1.96 (±9E-3)	1.96 (±7E-3)
Spectrogram ($\beta = 0$)	5.16E-1 (±4E-3)	5.09E-1 (±4E-3)
Spectrogram ($\beta = 0.5$)	-6.58E-3 (±7E-2)	-5.68E-2 (±9E-3)
TasteProfile	1.77E-2 (±4E-5)	1.76E-2 (±2E-5)
Moffet ($\beta = 1.3$)	1.13E-3 (±6E-5)	1.08E-2 (±5E-5)
Moffet ($\beta = 2$)	1.03E-3 (±1E-5)	1.02E-2 (±1E-5)
	CPU time	
Olivetti	132s (± 15)	**23s** (± 6)
Spectrogram ($\beta = 0$)	**17s** (± 4)	**17s** (± 4)
Spectrogram ($\beta = 0.5$)	**34s** (± 6)	39s (± 9)
TasteProfile	12613s (±927)	**6704s** (±603)
Moffet ($\beta = 1.3$)	25s (± 4)	**16s** (± 2)
Moffet ($\beta = 2$)	10s (± 2)	**7s** (± 3)
	Number of iterations	
Olivetti	1180 (± 130)	920 (±112)
Spectrogram ($\beta = 0$)	512 (± 86)	611 (±132)
Spectrogram ($\beta = 0.5$)	198 (± 33)	300 (± 73)
TasteProfile	1900 (± 131)	929 (± 83)
Moffet ($\beta = 1.3$)	1001 (± 240)	801 (±206)
Moffet ($\beta = 2$)	1235 (± 282)	959 (±286)

[6] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE Int. Conf. Data Mining*. IEEE, Dec. 2008.

[7] A. Cichocki, R. Zdunek, and A. H. Phan, *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons Inc, 2009.

[8] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: identifiability, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 59–80, Mar. 2019.

[9] N. Gillis, *Nonnegative matrix factorization*. Society for Industrial and Applied Mathematics, Jan. 2020.

[10] A. Cichocki, S. Cruces, and S. ichi Amari, "Generalized Alpha-Beta divergences and their application to robust nonnegative matrix factorization," *Entropy*, vol. 13, no. 1, pp. 134–170, jan 2011.

[11] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Comput.*, vol. 23, no. 9, pp. 2421–2456, Sep. 2011.

[12] P. O. Hoyer, "Non-negative sparse coding," in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*. IEEE, 2002.

[13] ——, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, p. 1457–1469, Dec. 2004.

[14] J. Eggert and E. Körner, "Sparse coding and NMF," in *Proc. Int. Joint Conf. Neur. Netw.* IEEE, Jul. 2004, pp. 2529–2533.

[15] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, May 2007.

[16] A. Cichoki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Trans. Fund. Electron. Comm. Comput. Sci.*, vol. E92-A, no. 3, pp. 708–721, 2009.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 10–60, 2010.

[18] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.

[19] R. Zhao and V. Y. F. Tan, "A unified convergence analysis of the multiplicative update algorithm for regularized nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 129–138, Jan. 2018.

[20] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "L1/2 sparsity constrained nonnegative matrix factorization for hyperspectral unmixing," in *Proc. International Conference on Digital Image Computing: Techniques and Applications*. IEEE, Dec. 2010.

[21] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral unmixing with ℓ_q regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6793–6806, Nov. 2014.

[22] A. Lefèvre, F. Bach, and C. Févotte, "Itakura-Saito nonnegative matrix factorization with group sparsity," in *Proc. Int. Conf. Acoust. Speech Signal Process.* IEEE, May 2011.

[23] V. Y. F. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the β-divergence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1592–1605, Jul. 2013.

[24] C. Peng, Y. Zhang, Y. Chen, Z. Kang, C. Chen, and Q. Cheng, "Log-based sparse nonnegative matrix factorization for data representation," *Knowledge-Based Syst.*, vol. 251, p. 109127, Sep. 2022.

[25] M. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete latent variable decomposition of counts data," in *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20. Curran Associates, Inc., 2007.

[26] C. Joder, F. Weninger, D. Virette, and B. Schuller, "A comparative study on sparsity penalties for NMF-based speech separation: Beyond LP-norms," in *Proc. Int. Conf. Acoust. Speech Signal Process.* IEEE, May 2013.

[27] R. Peharz and F. Pernkopf, "Sparse nonnegative matrix factorization with ℓ_0-constraints," *Neurocomputing*, vol. 80, pp. 38–46, Mar. 2012.

[28] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Programm.*, vol. 146, no. 1-2, pp. 459–494, Jul. 2013.

[29] J. Kim, R. D. C. Monteiro, and H. Park, "Group sparsity in nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*. Society for Industrial and Applied Mathematics, apr 2012.

[30] J. Le Roux and F. W. J. R. Hershey, "Sparse NMF – half-baked or well done?" Mitsubishi Electric Research Laboratories, Tech. Rep., Mar. 2015.

[31] V. Leplat, N. Gillis, and J. Idier, "Multiplicative updates for NMF with β-divergences under disjoint equality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 42, no. 2, pp. 730–752, Jan. 2021.

[32] L. Filstroff, O. Gouvert, C. Févotte, and O. Cappé, "A comparative study of Gamma Markov chains for temporal non-negative matrix factorization," *IEEE Trans. Signal Process.*, vol. 69, pp. 1614–1626, 2021.

[33] S. Essid and C. Févotte, "Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 415–425, Feb. 2013.

[34] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito Divergence: With application to music analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830, Mar. 2009.

[35] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[36] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4810–4819, Dec. 2015.

[37] X. Fu, K. Huang, and N. D. Sidiropoulos, "On identifiability of nonnegative matrix factorization," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 328–332, mar 2018.

[38] Z. Yang and E. Oja, "Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1878–1891, Dec. 2011.

[39] A. Cichocki, R. Zdunek, and S. ichi Amari, "Csiszár's divergences for non-negative matrix factorization: Family of new algorithms," in *Independent Component Analysis and Blind Signal Separation*. Springer Berlin Heidelberg, 2006, pp. 32–39.

[40] K. Lange, *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Jul. 2016.

[41] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[42] M. Nakano, H. Kameoka, J. L. Roux, Y. Kitano, N. Ono, and S. Sagayama, "Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β-divergence," in *IEEE Int. Workshop Mach. Learn. Signal Process.* IEEE, Sep. 2010.

[43] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, Oct. 2008.
[44] F. S. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. Workshop on Applications of Computer Vision*. IEEE Comput. Soc. Press, 1994.
[45] Jet Propulsion Lab (JPL). (2006) Aviris free data. California Inst. Technol., Pasadena, CA. [Online]. Available: http://aviris.jpl.nasa.gov/html/aviris.freedata.html
[46] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. Int. Conf. Music Inform. Retrieval (ISMIR)*, 2011.
[47] O. Gouvert, T. Oberlin, and C. Févotte, "Ordinal non-negative matrix factorization for recommendation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3680–3689.