Exact Multiplicative Factor Updates for Convolutional Beta-NMF in 2D

Pedro J. Villasana T.
ATG Sound Technology Research
Dolby Sweden
Stockholm, Sweden
Pedro.Villasana@dolby.com

Stanislaw Gorlow
ATG Sound Technology Research
Dolby Sweden
Stockholm, Sweden
Stanislaw.Gorlow@dolby.com

***Abstract*—In this paper we extend the convolutional NMF with the beta-divergence as cost function to two dimensions and derive exact multiplicative updates for its factors. Our updates correct and generalize the nonnegative matrix factor deconvolution, as proposed by Schmidt and Mørup. We prove that the cost is non-increasing under the new updates for beta between 0 and 2. By numerical simulation we confirm that both the cost's mean and standard deviation are monotonically decreasing in a consistent manner across the most common values for beta.**

***Index Terms*—Nonnegative matrix factorization, multiplicative updates, beta-divergence, convolution, 2D**

I. Introduction

Nonnegative matrix factorization (NMF) finds applications in the field of machine learning and in connection with inverse problems. NMF became popular after Lee and Seung derived multiplicative factor updates for gradient descent that lead to a faster convergence without violating nonnegativity of the data [1]. In [2], they further gave proof of their convergence to a stationary point, using the squared Euclidean distance and the generalized Kullback–Leibler divergence as cost function. The factorization's origins can be traced back to [3], [4].

A convolutional variant of the factorization was introduced in [5]. There, the basic idea is to model temporal relations in the neighborhood of a point in the time-frequency plane. The corresponding factor updates are taken from [2] and result in a biased factorization. To provide a remedy, multiple activation matrices are updated in [6], one for each translation, and the final update is made by taking the average over all activation matrices. The exact same principles are applied in [7]. There, the authors combine the updates from [2] with the averaging from [6] in an efficient manner. Why these updates are inexact is explained in [8]. A nonnegative matrix factor deconvolution in 2D that uses either the squared Euclidean distance or the Kullback–Leibler divergence as the cost can be found in [9]. It is worth pointing out that the update rule for the activation matrix is different from those in [5]–[7]. Convolutional NMF has been deployed with arguable success, e.g., to extract sound objects [5], to separate speakers [6], to detect onsets [7], to transcribe music [9], and recently to enhance speech [10] or to discover recurrent patterns in neural data [11].

In this paper, we continue and extend our previous work on the convolutional NMF under β-divergence (β-CNMF) [8] to two dimensions and derive exact multiplicative updates for its factors. The updates generalize the factor deconvolution, as it was introduced in [9], to the family of β-divergences. As we provide a summary of the derivation, it can easily be shown that one of the update rules in [9] is incorrect when the cost is set equal to the Kullback–Leibler divergence. We show that our updates lead to a monotonically decreasing β-divergence [12] in terms of the mean and the standard deviation and that the corresponding convergence curves are consistent across the most common values for β. A formal proof is further given to validate the numerical results. Note that in [13] it was shown that the β-divergence allows to construct an estimator that is more robust to outliers than the Kullback–Leibler divergence.

II. Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) is an umbrella term for a low-rank matrix approximation of the form

$$\mathbf{V} \simeq \mathbf{W}\,\mathbf{H} = \mathbf{U} \tag{1}$$

with $\mathbf{V} \in \mathbb{R}_{\geqslant 0}^{K\times N}$, $\mathbf{W} \in \mathbb{R}_{\geqslant 0}^{K\times I}$, and $\mathbf{H} \in \mathbb{R}_{\geqslant 0}^{I\times N}$, where I is the predetermined rank of the factorization. The letters above help distinguish between visible (v) and hidden variables (h) that are put in relation through weights (w). The factorization is usually formulated as a minimization problem with an associated cost function C according to

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}}\; C(\mathbf{W},\mathbf{H}) \qquad \text{subject to } w_{ki}, h_{in} \geqslant 0 \tag{2}$$

with

$$C(\mathbf{W},\mathbf{H}) \equiv \mathcal{L}(\mathbf{V},\mathbf{U}), \tag{3}$$

where $\mathcal{L}$ is a loss function that assesses the error between $\mathbf{V}$ and its low-rank approximation $\mathbf{U}$.

A. β-Divergence

The loss from (3) can be expressed by means of a contrast or distance function between the elements of $\mathbf{V}$ and $\mathbf{U}$. Due to its robustness with respect to outliers for certain values of the input parameter $\beta \in \mathbb{R}$ [13], we resort to the β-divergence

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

[12] as a subclass of the Bregman divergence [14], [15], which for the points $p > 0$ and $q > 0$ is given by [15]

$$d_\beta(p, q) = \begin{cases} p\,\dfrac{p^{\beta-1} - q^{\beta-1}}{\beta - 1} - \dfrac{p^\beta - q^\beta}{\beta} & \text{if } \beta \notin \{0, 1\}, \\ p \log \dfrac{p}{q} - p + q & \text{if } \beta = 1, \\ \dfrac{p}{q} - \log \dfrac{p}{q} - 1 & \text{if } \beta = 0. \end{cases} \tag{4}$$

Accordingly, the β-divergence between two matrices, $\mathbf{V}$ and $\mathbf{U}$, is defined entrywise as

$$D_\beta(\mathbf{V} \parallel \mathbf{U}) \overset{\text{def}}{=} \sum_{k=1}^{K} \sum_{n=1}^{N} d_\beta(v_{kn}, u_{kn}) \tag{5a}$$

with

$$u_{kn} = \sum\nolimits_i w_{ki}\, h_{in}. \tag{5b}$$

Note that the D_β has a single global minimum in $v_{kn} = \sum_i w_{ki}\, h_{in}$, $\forall k, n$, although strict convexity for d_β is granted only in the second argument for $\beta \in [1, 2]$ [15], [16].

B. Multiplicative Factor Updates

Given that (4) is continuously differentiable and that the first derivative is monotonically decreasing or increasing if $q < p$ or $q > p$, respectively, we can use gradient descent to find the minimum of (5). Holding $\mathbf{W}$ or $\mathbf{H}$ fixed, the iterative update of the variable factor $\mathbf{X}$ ($\mathbf{H}$ or $\mathbf{W}$) at iteration t reads

$$\mathbf{X}^{t+1} = \mathbf{X}^t - \mu\, \nabla C\big(\mathbf{X}^t, \cdot^t\big), \quad t \geqslant 0. \tag{6}$$

Splitting the gradient in components with opposite signs,

$$\nabla C\big(\mathbf{X}^t, \cdot^t\big) = \nabla C_+\big(\mathbf{X}^t, \cdot^t\big) - \nabla C_-\big(\mathbf{X}^t, \cdot^t\big), \tag{7}$$

and extending the step size μ to a matrix that changes with t,

$$\mu^t \overset{\text{def}}{=} \mathbf{X}^t \circ \big[\nabla C_+\big(\mathbf{X}^t, \cdot^t\big)\big]^{\circ -1}, \tag{8}$$

(6) can be converted to a multiplicative form [1], [2]:

$$\mathbf{X}^{t+1} = \mathbf{X}^t \circ \big[\nabla C_+\big(\mathbf{X}^t, \cdot^t\big)\big]^{\circ -1} \circ \nabla C_-\big(\mathbf{X}^t, \cdot^t\big), \tag{9}$$

where $\circ$ denotes the Hadamard, i.e. entry-wise product, and $\cdot^{\circ -1}$ stands for the entry-wise inverse. The step size is chosen in such a way as to ensure nonnegativity of the factor updates on the assumption that they were initialized with nonnegative values [9], [16].

C. Discrete Convolution in 2D

As can be seen from (5b), the weight w_{ki} for the ith variable h_i in column n is applied using the scalar product. Should h_i evolve with n, we can assume that the current state (or value) of h_i is correlated with its past and future states. We can take this into account by replacing the scalar product in our model by a convolution. Postulating causality and letting the weight w_{ki} have finite support of cardinality M, convolution along n writes

$$\sum_{m=0}^{M-1} w_{kim}\, h_{i,n-m} \overset{\text{def}}{=} \left(\mathbf{w}_{ki} * \mathbf{h}_i\right)_n \tag{10a}$$

with

$$\mathbf{w}_{ki} = \begin{bmatrix} w_{ki,0} & w_{ki,1} & \cdots & w_{ki,M-1} \end{bmatrix} \tag{10b}$$

and

$$\mathbf{h}_i = \begin{bmatrix} h_{i,n} & h_{i,n-1} & \cdots & h_{i,n-M+1} \end{bmatrix}. \tag{10c}$$

The operation can be converted to a matrix multiplication by lining up the states $\mathbf{h}_i^\mathsf{T}$ for $n = 0, 1, \ldots, N-1$ in a truncated Toeplitz matrix:

$$\mathbf{H}_i = \begin{bmatrix} h_{i,0} & h_{i,1} & \cdots & h_{i,N-1} \\ 0 & h_{i,0} & \cdots & h_{i,N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{i,N-M} \end{bmatrix}. \tag{11}$$

Using (10) and (11), $\mathbf{V}$ can now be approximated as

$$\mathbf{U} = \sum_{i=1}^{I} \left(\mathbf{W}_i * \mathbf{h}_i\right)_n = \sum_{i=1}^{I} \mathbf{W}_i\, \mathbf{H}_i \tag{12a}$$

with

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{w}_{1,i}^\mathsf{T} & \mathbf{w}_{2,i}^\mathsf{T} & \cdots & \mathbf{w}_{K,i}^\mathsf{T} \end{bmatrix}^\mathsf{T}. \tag{12b}$$

In practice, I can be quite large and M is usually small. It is therefore convenient to rewrite (12) as, see [5], [6]:

$$\mathbf{U} = \sum_{m=0}^{M-1} \mathbf{W}_m\, \mathbf{H}_{\xrightarrow{m}} \qquad \text{with } \mathbf{W}_m = \begin{bmatrix} w_{ki\cdot} \end{bmatrix}_m, \tag{13}$$

where $\cdot_{\xrightarrow{m}}$ is a column-wise right-shift operation that shifts all the columns of $\mathbf{H}$ by m positions to the right, and fills the vacant positions with zeros. The operation is size-preserving. It can be seen that the convolutional NMF (CNMF) has M times as many weights as (1), whereas the number of hidden variables is equal.

The convolution can be augmented by another dimension [9], which can be formulated as

$$\sum_{l=0}^{L-1} \sum_{m=0}^{M-1} w_{k-l,im}\, h_{li,n-m} \overset{\text{def}}{=} \left(\mathbf{W}_i ** \mathbf{H}_i\right)_{kn} \tag{14a}$$

with

$$\mathbf{W}_i = \begin{bmatrix} w_{k,i,0} & \cdots & w_{k,i,M-1} \\ w_{k-1,i,0} & \cdots & w_{k-1,i,M-1} \\ \vdots & \ddots & \vdots \\ w_{k-L+1,i,0} & \cdots & w_{l-L+1,i,M-1} \end{bmatrix} \tag{14b}$$

and

$$\mathbf{H}_i = \begin{bmatrix} h_{0,i,n} & \cdots & h_{0,i,n-M+1} \\ h_{1,i,n} & \cdots & h_{1,i,n-M+1} \\ \vdots & \ddots & \vdots \\ h_{L-1,i,n} & \cdots & h_{L-1,i,n-M+1} \end{bmatrix}. \tag{14c}$$

Using the notation from (13), the convolutional data model for

(14) in two dimensions can be written as

$$\mathbf{U} = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} {}^{l}\!\!\downarrow\!\mathbf{W}_m \, \mathbf{H}_{l \,\xrightarrow{m}} \quad \text{with } \mathbf{H}_l = \big[h_{\cdot in}\big]_l \tag{15}$$

and $\mathbf{W}_m$ as in (13). From (15) one can see that the CNMF in two dimensions has L times as many hidden variables as (13). Analogous to the right-shift operator, ${}^{l}\!\!\downarrow\!\cdot$ is a row-wise down-shift operator.

D. Uniqueness and Normalization

It is understood that the factorization is not unique. This can be shown easily by the equivalence

$$\mathbf{U} \equiv \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} {}^{l}\!\!\downarrow\!\mathbf{W}_m \, \mathbf{B} \, \mathbf{B}^{-1} \, \mathbf{H}_{l \,\xrightarrow{m}} \tag{16}$$

with $\mathbf{W}_m \leftarrow \mathbf{W}_m \, \mathbf{B}$ and $\mathbf{H}_l \leftarrow \mathbf{B}^{-1} \, \mathbf{H}_l$, for any $\mathbf{B} \in \mathbb{R}^{I \times I}$ that has an inverse. Nonnegativity still holds for $\mathbf{W}_m$ and $\mathbf{H}_l$ if $\mathbf{B}$ is a nonnegative diagonal matrix. The property is usually used to enforce the same p-norm on the matrices $\{\mathbf{W}_i\}$:

$$\mathbf{B} = \operatorname{diag}\left(\|\mathbf{W}_1\|_p^{-1}, \|\mathbf{W}_2\|_p^{-1}, \ldots, \|\mathbf{W}_I\|_p^{-1} \right) \tag{17}$$

with

$$\|\mathbf{W}_i\|_p \overset{\text{def}}{=} \left(\sum_{k=1}^{K} \sum_{m=1}^{M} w_{kim}^{p} \right)^{1/p} . \tag{18}$$

III. β-CNMF in 2D

Following up the considerations from Section II, we adopt the data model from (15) and derive multiplicative updates with the entrywise β-divergence from (5) as the loss function. The outcome is a β-CNMF [8] in two dimensions. A summary of the derivation follows.

A. Derivation

With $u_{kn} = \sum_{l,i,m} w_{k-l,im} \, h_{li,n-m}$, $p \in \{1, 2, \ldots, K\}$, $q \in \{1, 2, \ldots, I\}$, and $r \in \{0, 1, \ldots, M-1\}$:

$$\begin{aligned} \frac{\partial D_\beta(\mathbf{V} \parallel \mathbf{U})}{\partial w_{pqr}} &= \sum\nolimits_{k,n} \frac{\partial d_\beta(v_{kn}, u_{kn})}{\partial u_{kn}} \cdot \frac{\partial u_{kn}}{\partial w_{pqr}} \\ &= \sum\nolimits_{k,n} \left(u_{kn}^{\beta-1} - v_{kn} \, u_{kn}^{\beta-2} \right) \sum\nolimits_{l} \delta(p-k+l) \, h_{lq,n-r} \\ &= \sum\nolimits_{l,n} \left(u_{p+l,n}^{\beta-1} - v_{p+l,n} \, u_{p+l,n}^{\beta-2} \right) h_{lq,n-r}, \end{aligned} \tag{19}$$

where δ is the Kronecker delta function. Choosing μ in (6) as (8) and using (19) in (9) leads to the update rule for $\mathbf{W}_m$:

$$\begin{aligned} \mathbf{W}_m^{t+1} = \mathbf{W}_m^{t} \circ \Big[\sum\nolimits_{l} {}_{l}\!\uparrow\!\mathbf{U}^{t\,\circ(\beta-1)} \, \mathbf{H}_{l \,\xrightarrow{m}}^{t\,\mathsf{T}} \Big]^{\circ -1} \\ \circ \sum\nolimits_{l} \Big[{}_{l}\!\uparrow\!\mathbf{V} \circ {}_{l}\!\uparrow\!\mathbf{U}^{t\,\circ(\beta-2)} \Big] \, \mathbf{H}_{l \,\xrightarrow{m}}^{t\,\mathsf{T}}, \end{aligned} \tag{20a}$$

where ${}_{l}\!\uparrow\!\cdot$ is the up-shift operator. The update rule for $\mathbf{H}_l$ can be derived in similar fashion [17], resulting in

$$\begin{aligned} \mathbf{H}_l^{t+1} = \mathbf{H}_l^{t} \circ \Big[\sum\nolimits_{m} {}^{l}\!\!\downarrow\!\mathbf{W}_m^{t}{}^{\mathsf{T}} \, \mathbf{U}^{t\,\circ(\beta-1)}_{\xleftarrow{m}} \Big]^{\circ -1} \\ \circ \sum\nolimits_{m} {}^{l}\!\!\downarrow\!\mathbf{W}_m^{t}{}^{\mathsf{T}} \Big[\mathbf{V}_{\xleftarrow{m}} \circ \mathbf{U}^{t\,\circ(\beta-2)}_{\xleftarrow{m}} \Big], \end{aligned} \tag{20b}$$

where $\cdot_{\xleftarrow{m}}$ is the left-shift operator, respectively.

B. Relation to Schmidt and Mørup's work

In [9], multiplicative updates are derived for a CNMF in 2D with either the (squared) Euclidean distance or the Kullback–Leibler divergence as cost function. In the dimension of time, their updates are the same as ours for $\beta = 2$. For $\beta = 1$, there is however the difference that the $\mathbf{U}$-matrix in the first line of (20a) and (20b) is not shifted, neither up nor to the left. The error is that δ in (19) is 1 in their derivation, which is true if and only if $k = p + l$.

C. Proof of Convergence

To prove that the (entrywise) β-divergence is nonincreasing under the convolutional update rules given in (20), we resort to the methodology used in [2], [16].

Definition 1: $G \colon \mathbb{R}_{\geqslant 0}^{L \times I \times N} \times \mathbb{R}_{\geqslant 0}^{L \times I \times N} \to \mathbb{R}_{\geqslant 0}$ is an auxiliary function for $F \colon \mathbb{R}_{\geqslant 0}^{L \times I \times N} \to \mathbb{R}_{\geqslant 0}$ if and only if

$$G(\mathbf{H}, \mathbf{H}) = F(\mathbf{H}) \tag{21a}$$

and

$$G(\mathbf{H}', \mathbf{H}) \geqslant F(\mathbf{H}'). \tag{21b}$$

At iteration $t+1$, any $\mathbf{H}^{t+1}$ satisfying

$$G\big(\mathbf{H}^{t+1}, \mathbf{H}^{t}\big) \leqslant G\big(\mathbf{H}^{t}, \mathbf{H}^{t}\big) \tag{22}$$

also satisfies

$$F\big(\mathbf{H}^{t+1}\big) \leqslant F\big(\mathbf{H}^{t}\big), \tag{23}$$

because

$$F\big(\mathbf{H}^{t+1}\big) \leqslant G\big(\mathbf{H}^{t+1}, \mathbf{H}^{t}\big) \leqslant G\big(\mathbf{H}^{t}, \mathbf{H}^{t}\big) = F\big(\mathbf{H}^{t}\big). \tag{24}$$

Theorem 1: Let $w_{kim} > 0$ and $h_{lin} > 0$, where $u_{kn} = \sum_{l,i,m} w_{k-l,im} \, h_{li,n-m}$. Then,

$$\begin{aligned} G(\mathbf{H}', \mathbf{H}) = \\ \sum_{k,n} \sum_{l,i,m} \frac{w_{k-l,im} \, h_{li,n-m}}{u_{kn}} \, d_{\smile}\!\left(v_{kn}, u_{kn} \frac{h'_{li,n-m}}{h_{li,n-m}} \right) \\ + d_{\frown}(v_{kn}, u_{kn}) \\ + \dot{d}_{\frown}(v_{kn}, u_{kn}) \sum_{l,i,m} w_{k-l,im} \left(h'_{li,n-m} - h_{li,n-m} \right) \\ + d_{-}(v_{kn}, u_{kn}) \end{aligned} \tag{25}$$

with $\dot{d}_{\frown} \overset{\text{def}}{=} \partial d_{\frown} / \partial u_{kn}$ is an entrywise auxiliary function for

$$F(\mathbf{H}) = \sum_{kn} d_\beta \left(v_{kn}, \sum_{l,i,m} w_{k-l,im} \, h_{li,n-m} \right), \tag{26}$$

where

$$d_\beta(p, q) = d_{\smile}(p, q) + d_{\frown}(p, q) + d_{-}(p, q) \tag{27}$$

represents a convex-concave-constant decomposition of the β-divergence w.r.t. q.

It is trivial to show that (21a) is met. Given (27), (26) can be decomposed as

$$F(\mathbf{H}) = F_{\smile}(\mathbf{H}) + F_{\frown}(\mathbf{H}) + F_{-}(\mathbf{H}). \tag{28}$$

Since $G_{-}(\mathbf{H}', \mathbf{H}) = F_{-}(\mathbf{H}')$ for any $\mathbf{H}$, what is left to do is to prove (21b) for the convex and the concave component.

Proof: First, comparing (25) and (28), define

$$\begin{aligned} &G_{\smile}(\mathbf{H}', \mathbf{H}) = \\ &\qquad \sum_{k,n} \sum_{l,i,m} \frac{w_{k-l,im}\, h_{li,n-m}}{u_{kn}} \\ &\qquad\quad \cdot d_{\smile}\left(v_{kn}, u_{kn} \frac{h'_{li,n-m}}{h_{li,n-m}}\right) \end{aligned} \tag{29}$$

as an entrywise auxiliary function for $F_{\smile}(\mathbf{H}')$. Introduce

$$a_{lim} = \frac{w_{k-l,im}\, h_{li,n-m}}{u_{kn}}, \tag{30}$$

so that $\sum_{l,i,m} a_{lim} = 1$. Using Jensen's inequality, it is easy to show that

$$\begin{aligned} &G_{\smile}(\mathbf{H}', \mathbf{H}) \\ &\quad = \sum_{k,n} \sum_{l,i,m} a_{lim}\, d_{\smile}\left(v_{kn}, \frac{w_{k-l,im}\, h'_{li,n-m}}{a_{lim}}\right) \\ &\quad \geqslant \sum_{k,n} d_{\smile}\left(v_{kn}, \sum_{l,i,m} a_{lim} \frac{w_{k-l,im}\, h'_{li,n-m}}{a_{lim}}\right) \\ &\quad = \sum_{k,n} d_{\smile}\left(v_{kn}, \sum_{l,i,m} w_{k-l,im}\, h'_{li,n-m}\right) \\ &\quad = F_{\smile}(\mathbf{H}'). \end{aligned} \tag{31}$$

□

Now, consider the first-order Taylor series of $F_{\frown}(\mathbf{H}')$ about $\mathbf{H}$ as an (entrywise) auxiliary function $G_{\frown}(\mathbf{H}', \mathbf{H})$ for $F_{\frown}(\mathbf{H}')$,

$$G_{\frown}(\mathbf{H}', \mathbf{H}) = F_{\frown}(\mathbf{H}) + \sum_{l,i,n} \frac{\partial F_{\frown}(\mathbf{H})}{\partial h_{lin}} \,(h'_{lin} - h_{lin}). \tag{32}$$

Eq. (32) fulfills condition (21a) by definition and also (21b), because the tangent to $F_{\frown}(\mathbf{H}')$ at $\mathbf{H}$ is an upper bound of $F_{\frown}(\mathbf{H}')$. Given that

$$\begin{aligned} &\frac{\partial F_{\frown}(\mathbf{H})}{\partial h_{liq}} = \\ &\quad \sum_{k,n} \dot{d}_{\frown}(v_{kn}, u_{kn}) \sum_{m} w_{k-l,im}\, \delta(q - n + m), \end{aligned} \tag{33}$$

where δ is the Kronecker delta function, the auxiliary function for the concave component reads

$$\begin{aligned} &G_{\frown}(\mathbf{H}', \mathbf{H}) = \\ &\qquad \sum_{k,n} d_{\frown}(v_{kn}, u_{kn}) + \sum_{k,n} \dot{d}_{\frown}(v_{kn}, u_{kn}) \\ &\qquad\quad \cdot \sum_{l,i,m} w_{k-l,im}\, \big(h'_{li,n-m} - h_{li,n-m}\big). \end{aligned} \tag{34}$$

□

Having shown that $G(\mathbf{H}', \mathbf{H})$ is an auxiliary function for the entrywise β-divergence $F(\mathbf{H}')$ associated with a convolutional data model, using (25) we can now go about showing that the update rules in (20) satisfy (23) via (22).

Lemma 1: $G(\mathbf{H}', \mathbf{H})$ is an entrywise auxiliary function for $F(\mathbf{H}')$ that for all $\beta \in \mathbb{R}$ can be written as

$$G(\mathbf{H}', \mathbf{H}) = \sum_{l,i,n} G(h'_{lin}, h_{lin}) + \mathbf{const} \tag{35}$$

with

$$\begin{aligned} &G(h'_{lin}, h_{lin}) = \\ &\quad h_{lin} \sum_{k,m} \frac{w_{k-l,im}}{u_{k,n+m}}\, d_{\smile}\left(v_{k,n+m}, u_{k,n+m} \frac{h'_{lin}}{h_{lin}}\right) \\ &\qquad + (h'_{lin} - h_{lin}) \sum_{k,m} w_{k-l,im}\, \dot{d}_{\frown}(v_{k,n+m}, u_{k,n+m}). \end{aligned} \tag{36}$$

Theorem 2: For any $\beta \in [0, 2]$, the convolutional update rules in (20) satisfy (22).

Proof: Using the decomposition of the β-divergence from [16, Table 1] in (36), we can express the difference between $G(h_{inl}, h_{inl})$ and $G(h'_{inl}, h_{inl})$ for h'_{inl} according to (20b) piecewise as

$$\begin{aligned} &G(h_{lin}, h_{lin}) - G(h'_{lin}, h_{lin}) = d_{\smile}(h'_{lin}, h_{lin}) \\ &\qquad - d_{\smile}(h'_{lin}, h'_{lin}) - \dot{d}_{\frown}(h'_{lin}, h_{lin})\,(h'_{lin} - h_{lin}) \\ &\quad = \begin{cases} \frac{1}{1-\beta}\, h^{\beta}_{lin} \left[1 - (1 - \eta_{lin})\, \beta - \eta^{\beta}_{lin}\right] & \text{if } \beta \in [0, 1), \\ h_{lin} \left[\eta_{lin} \log \eta_{lin} + (1 - \eta_{lin})\right] & \text{if } \beta = 1, \\ \frac{1}{\beta\,(\beta - 1)}\, h^{\beta}_{lin} \left[\eta^{\beta}_{lin} - 1 + (1 - \eta_{lin})\, \beta\right] & \text{if } \beta \in (1, 2], \end{cases} \end{aligned} \tag{37}$$

where

$$\eta_{lin} = \frac{h'_{lin}}{h_{lin}} = \frac{\sum_{k,m} w_{k-l,im}\, v_{k,n+m}\, u^{\beta-2}_{k,n+m}}{\sum_{k,m} w_{k-l,im}\, u^{\beta-1}_{k,n+m}}. \tag{38}$$

Evaluating the expression in the square brackets of each piece, one can show that

$$\begin{cases} 1 - (1 - \eta_{lin})\, \beta - \eta^{\beta}_{lin} \geqslant 0 & \text{if } \beta \in [0, 1), \\ \eta_{lin} \log \eta_{lin} + (1 - \eta_{lin}) \geqslant 0 & \text{if } \beta = 1, \\ \eta^{\beta}_{lin} - 1 + (1 - \eta_{lin})\, \beta \geqslant 0 & \text{if } \beta \in (1, 2], \end{cases} \tag{39}$$

and thus

$$G(h_{lin}, h_{lin}) - G(h'_{lin}, h_{lin}) \geqslant 0 \quad \text{if } \beta \in [0, 2], \tag{40}$$

from which (22) follows directly via (35). □

Reversing the roles of $\mathbf{W}$ and $\mathbf{H}$, it can also be shown that $F(\mathbf{W}')$ is nonincreasing under the updates.

IV. Simulation

In this section, we simulate and assess the convergence of the newly derived updates for 1×10^3 iterations. To that end, we generate 1×10^2 distinct $\mathbf{V}$-matrices from M χ^2-distributed $\mathbf{W}_m$-matrices,

$$w_{kim} = \sum_{p=1}^{2} w^2_{kimp} \sim \chi^2_2 \qquad w_{kimp} \sim \mathcal{N}(0, 1), \tag{41}$$

and L uniformly distributed $\mathbf{H}_l$-matrices,

$$h_{lin} \sim \mathcal{U}(0, 1). \tag{42}$$

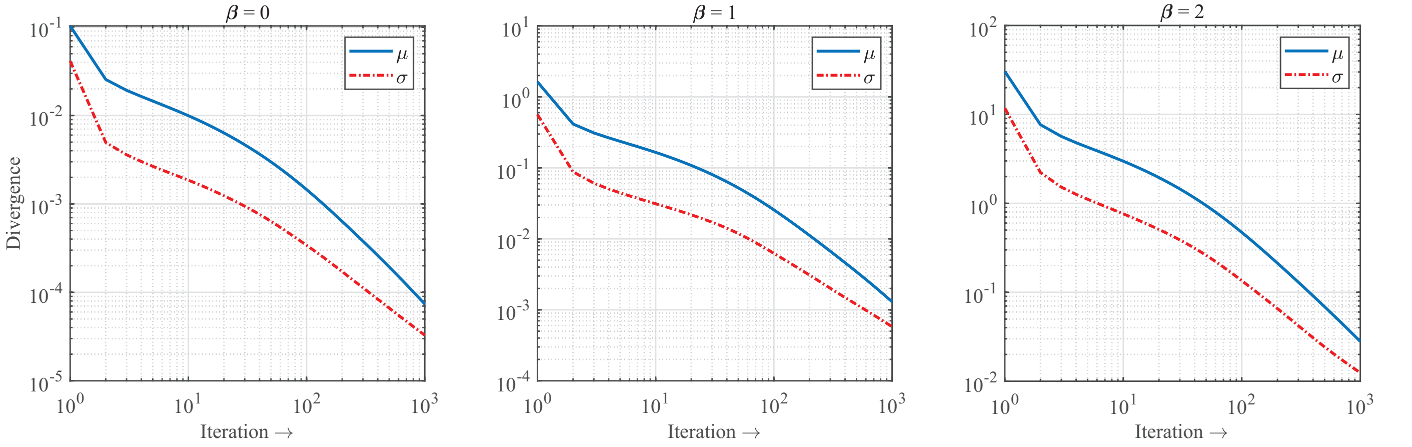

Fig. 1. Simulation results showing the mean and the standard deviation of the divergence between $\mathbf{V}$ and $\mathbf{U}$.

We select $M = L = 2$. The factorization is repeated 1×10^1 times, using random initializations of $\{\mathbf{W}_m^{t=0}\}$ and $\{\mathbf{H}_l^{t=0}\}$ with non-zero entries. So, the curves in Fig. 1 were computed over ensembles of 1×10^3 costs at each iteration (step). The number of visible variables and observations is $K = 1 \times 10^1$ and $N = 2.5 \times 10^1$, while the number of hidden variables I is 5×10^0.

As can be seen from Fig. 1, the multiplicative updates are stable (the entry-wise divergence is monotonically decreasing w.r.t. both the mean and the standard deviation) and they also are consistent across different values of β. The difference in scale is because

$$d_\beta(p, q) \equiv p^\beta \, d_\beta\left(1, \frac{q}{p}\right), \tag{43}$$

which evinces that only the Itakura–Saito divergence ($\beta = 0$) is scale invariant. In addition, we measured the run time as a function of the β-value on an Intel Xeon E5-2637 v3 CPU at 3.5 GHz with 16 GB of RAM. For $\beta = 0$, one iteration takes about 1.41 times longer than for $\beta = 2$, whereas for $\beta = 1$ an iteration takes only a factor of 1.05 longer. The convergence curves have a similar trajectory for different values of K, N, and I. Our reference code can be downloaded from [18].

V. Conclusion

In summary, this paper extends our previous work on the β-CNMF to two dimensions. The β-CNMF in 2D corrects and generalizes the (2D) nonnegative matrix factor deconvolution by Schmidt and Mørup. By a formal proof and via numerical simulation it was validated that the new updates are stable and that their convergence behavior is consistent.

References

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[2] ——, "Algorithms for non-negative matrix factorization," in *NIPS 2001*, 2001, pp. 556–562.

[3] 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, 1994.

[4] P. Paatero, "Least squares formulation of robust non-negative factor analysis," *Chemom. Intell. Lab. Syst.*, vol. 37, no. 1, pp. 23–35, 1997.

[5] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *ICA 2004*, 2004, pp. 494–499.

[6] ——, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 1, pp. 1–12, 2007.

[7] W. Wang, A. Cichocki, and J. A. Chambers, "A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2858–2864, 2009.

[8] P. J. Villasana T., S. Gorlow, and A. T. Hariraman, "Multiplicative updates for convolutional NMF under β-divergence," *Optim. Lett.*, May 2019. [Online]. Available: https://doi.org/10.1007/s11590-019-01434-9

[9] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *ICA 2006*, 2006, pp. 700–707.

[10] M. Sun, Y. Li, J. F. Gemmeke, and X. Zhang, "Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback–Leibler divergence," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 7, pp. 1233–1242, 2015.

[11] E. L. Mackevicius, A. H. Bahle, A. H. Williams, S. Gu, N. I. Denisenko, M. S. Goldman, and M. S. Fee, "Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience," *eLife*, vol. 8, Feb 2019. [Online]. Available: https://doi.org/10.7554/eLife.38471

[12] A. Basu, I. R. Harris, N. L. Hjort, and M. C. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[13] M. Mihoko and S. Eguchi, "Robust blind source separation by beta divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1859–1886, 2002.

[14] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. & Math. Phys*, vol. 7, no. 3, pp. 200–217, 1967.

[15] A. Cichocki and S.-i. Amari, "Families of alpha- beta- and gamma-divergences: Flexible and robust measures of similarities," *Entropy*, vol. 12, no. 6, pp. 1532–1568, 2010.

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

[17] P. J. Villasana T., "New variants of nonnegative matrix factorization with application to speech coding and speech enhancement," Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2019. [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1324307

[18] P. Villasana and S. Gorlow, "MATLAB code for beta-convolutional nonnegative matrix factorization in 2D," Mar 2019. [Online]. Available: https://doi.org/10.24433/CO.7116855.v1