# Sparse Non-negative Matrix Factor 2-D Deconvolution

**Morten Mørup** MM@IMM.DTU.DK
**Mikkel N. Schmidt** MNS@IMM.DTU.DK
*Informatics and Mathematical Modelling*
*Technical University of Denmark*
*Richard Petersens Plads, Building 321*
*DK-2800 Kgs. Lyngby, Denmark*

**Editor:** n/a

## Abstract

We introduce the non-negative matrix factor 2-D deconvolution (NMF2D) model, which decomposes a matrix into a 2-dimensional convolution of two factor matrices. This model is an extension of the non-negative matrix factor deconvolution (NMFD) recently introduced by Smaragdis (2004). We derive and prove the convergence of two algorithms for NMF2D based on minimizing the squared error and the Kullback-Leibler divergence respectively. Next, we introduce a sparse non-negative matrix factor 2-D deconvolution model that gives easy interpretable decompositions and devise two algorithms for computing this form of factorization. The developed algorithms have been used for source separation (Schmidt and Mørup, 2005) and music transcription (Schmidt and Mørup, 2006).

**Keywords:** Non-negative Matrix Factorization (NMF), Sparse Decomposition, NMFD, translation invariant NMF, NMF2D/SNMF2D.

## 1. Introduction

In matrix decomposition techniques such as principal component analysis (PCA), independent component analysis (ICA), and non-negative matrix factorization (NMF) the matrix $\mathbf{V}$ is explained by an instantaneous mixing of the sources $\mathbf{H}$ with the mixing matrix $\mathbf{W}$

$$\begin{aligned} \mathbf{V} &= \mathbf{W}\mathbf{H}, \\ \mathbf{V}_{i,j} &= \sum_d \mathbf{W}_{i,d}\mathbf{H}_{d,j}. \end{aligned}$$

In convolutive matrix decompositions the mixing is not instantaneous but a convolutive mixture of the sources $\mathbf{H}$ (Parra et al., 1998; Nguyen Thi and Jutten, 1995)

$$\begin{aligned} \mathbf{V} &= \sum_\tau \mathbf{W}^\tau \overrightarrow{\mathbf{H}}^\tau, \\ \mathbf{V}_{i,j} &= \sum_{d,\tau} \mathbf{W}^\tau_{i,d}\mathbf{H}_{d,j-\tau}, \end{aligned}$$

where $\overrightarrow{\mathbf{H}}^\tau$ denotes shifting each column in $\mathbf{H}$, $\tau$ positions to the right. Since each row in $\mathbf{V}$ is a convolutive mixture of the rows of $\mathbf{H}$, we refer to the estimation of $\mathbf{W}$ and $\mathbf{H}$ as matrix factor deconvolution.

©2006 Morten Mørup and Mikkel N. Schmidt.

Matrix factor deconvolution is currently a topic of great interest — one of the main problems is finding efficient algorithms (Dyrholm et al., 2006). Smaragdis (2004) has introduced a fast convolutive non-negative matrix factorization with multiplicative updates based on Kullback-Leibler (KL) divergence minimization. A similar algorithm based on least squares (LS) minimization using a transformation matrix to form the convolution was derived by Eggert et al. (2004). Despite the growing attention given to convolutive models no attention has to our knowledge been given to models that are convolutive in both the mixing and source matrices. We extend the models of Smaragdis and Eggert to form a non-negative matrix factor 2-D deconvolution (NMF2D). We derive and prove the convergence of two algorithms for NMF2D based on LS-minimization and KL-divergence respectively. Both algorihtms are extensions of the algorithms for non-negative matrix factorization introduced by Lee and Seung (2000).

Since the NMF2D decomposition in general is not unique and possibly overcomplete, we further introduce two algorithms for sparse non negative matrix factor double deconvolution (SNMF2D). As for NMF2D, the SNMF2D algorithms are completely based on multiplicative updates extending the approach of Eggert and Korner (2004). We conjecture that both the SNMF2D algorithms converge to a local minimum of the cost function.

This paper establishes the NMF2D and SNMF2D methods and evaluates the proposed algorithms. The developed algorithms for NMF2D decomposition have proven useful for source separation (Schmidt and Mørup, 2005) and the SNMF2D algorithms have been useful for music transcription (Schmidt and Mørup, 2006).

The paper is structured as follows: In Section 2 we give an introduction to NMF2D. Then, we derive iterative algorithms based on LS and KL-divergence minimization. This is followed by a derivation of the sparse non-negative matrix factor 2-D deconvolution algorithms. In Section 3 we demonstrate the algorithms on toy examples. Finally, in Section 4 we elaborate on the proposed algorithms. A MATLAB implementation of the algorithms as well as a demo containing the analysed datasets can be downloaded from http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4521/zip/imm4521.zip.

## 2. Non-negative Matrix Factor 2-D Deconvolution

Consider the non-negative matrix factorization (NMF) problem (Lee and Seung, 2000):

$$\mathbf{V} \approx \mathbf{WH}$$

where $\mathbf{V} \in \mathbb{R}^{I \times J}$,$\mathbf{W} \in \mathbb{R}^{I \times D}$, and $\mathbf{H} \in \mathbb{R}^{D \times J}$ are non-negative matrices. Lee and Seung (2000) devise two algorithms to find $\mathbf{W}$ and $\mathbf{H}$: For the least squared error and the KL divergence they proved that the following recursive updates converge to a local minimum

$$\text{Least Squares}: \quad \mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\mathbf{V}\mathbf{H}^{\mathrm{T}}}{\mathbf{W}\mathbf{H}\mathbf{H}^{\mathrm{T}}}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{W}^{\mathrm{T}}\mathbf{V}}{\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H}},$$

$$\text{KL divergence}: \quad \mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}\mathbf{H}^{\mathrm{T}}}{\mathbf{1} \cdot \mathbf{H}^{\mathrm{T}}}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{W}^{\mathrm{T}}\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}}{\mathbf{W}^{\mathrm{T}} \cdot \mathbf{1}},$$

where $A \bullet B$ and $\frac{A}{B}$ denotes element-wise multiplication and division respectively. These algorithms can be derived by minimizing the cost function using a gradient based search

choosing the step size appropriately to yield multiplicative updates. Compared to principal component analysis (PCA) and independent component analysis (ICA), NMF gives a more sparse/part based decomposition (Lee and Seung, 1999). Furthermore, the decomposition is unique under certain conditions (Donoho and Stodden, 2003), making it unnecessary to impose constraints in the form of orthogonality or independence. These properties have resulted in a great interest in NMF lately.

The NMF2D model extends the NMF model to be a 2-dimensional convolution of $\mathbf{W}$ and $\mathbf{H}$:

$$\begin{aligned} \mathbf{V} \approx \mathbf{\Lambda} &= \sum_{\tau,\phi} \overset{\downarrow\phi}{\mathbf{W}}^{\tau} \overset{\rightarrow\tau}{\mathbf{H}}^{\phi}, \\ \mathbf{\Lambda}_{i,j} &= \sum_{\tau,\phi,d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau}, \end{aligned}$$

where $\overset{\downarrow\phi}{\mathbf{W}}$ denotes the downward shift operator which moves each element in the matrix $\phi$ rows down, and $\overset{\rightarrow\tau}{\mathbf{H}}$ denotes the right shift operator which moves each element in the matrix $\tau$ columns to the right, i.e.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad \overset{\downarrow 2}{\mathbf{A}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 3 \end{pmatrix}, \quad \overset{\rightarrow 1}{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 0 & 7 & 8 \end{pmatrix}.$$

We note that the non-negative matrix factor deconvolution model introduced by Smaragdis (2004) is a special case of the NMF2D model where $\phi = \{0\}$. For illustrations of the NMF2D model see Figure 1 and 2.

In the following derivation of the algorithms for NMF2D and SNMF2D, the derivative of a given element of $\mathbf{\Lambda}$ with respect to a given element of $\mathbf{W}^{\tau}$ and $\mathbf{H}^{\phi}$ is needed

$$\frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \mathbf{W}^{\tau'}_{i',d'}} = \frac{\partial \sum_{\tau,\phi,d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau}}{\partial \mathbf{W}^{\tau'}_{i',d'}} = \mathbf{H}^{i-i'}_{d',j-\tau'},$$

$$\frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \mathbf{H}^{\phi'}_{d',j'}} = \frac{\partial \sum_{\tau,\phi,d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau}}{\partial \mathbf{H}^{\phi'}_{d',j'}} = \mathbf{W}^{j-j'}_{i-\phi',d'}.$$

### 2.1 Least Squares Cost Function

First, we consider the least squares cost function

$$C_{LS} = \frac{1}{2} ||\mathbf{V} - \mathbf{\Lambda}||^2_F = \frac{1}{2} \sum_{i,j} (\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j})^2.$$

Minimizing the squared error corresponds to maximizing the likelihood of a homoscedatic Gaussian noise model. The derivative of $C_{LS}$ with respect to a given element in $\mathbf{W}^{\tau}$ is given

by

$$\begin{aligned}\frac{\partial C_{LS}}{\partial \mathbf{W}^{\tau'}_{i',d'}} &= \frac{\partial}{\partial \mathbf{W}^{\tau'}_{i',d'}} \frac{1}{2} \sum_{i,j} \left(\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j}\right)^2 \\ &= -\sum_{i,j} \left(\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j}\right) \mathbf{H}^{i-i'}_{d',j-\tau'} \\ &= -\sum_{\phi,j} \left(\mathbf{V}_{i'+\phi,j} - \mathbf{\Lambda}_{i'+\phi,j}\right) \mathbf{H}^{\phi}_{d',j-\tau'}.\end{aligned}$$

Similarly for a given element in $\mathbf{H}^{\phi}$ we find:

$$\frac{\partial C_{LS}}{\partial \mathbf{H}^{\phi'}_{d',j'}} = -\sum_{\tau,i} \left(\mathbf{V}_{i,j'+\tau} - \mathbf{\Lambda}_{i,j'+\tau}\right) \mathbf{W}^{\tau}_{i-\phi',d'}$$

The factors $\mathbf{W}$ and $\mathbf{H}$ can be found by minimizing the cost function using a gradient based search. Consequently, the recursive update for an element in $\mathbf{W}$ is given by

$$\mathbf{W}^{\tau'}_{i',d'} \leftarrow \mathbf{W}^{\tau'}_{i',d'} - \eta \frac{\partial C_{LS}}{\partial \mathbf{W}^{\tau'}_{i',d'}}. \tag{1}$$

Similar to the approach of Lee and Seung (2000), we can choose the step size $\eta$ to cancel the first term in equation 1

$$\eta = \frac{\mathbf{W}^{\tau'}_{i',d'}}{\sum_{\phi,j} \mathbf{\Lambda}_{i'+\phi,j}\ \mathbf{H}^{\phi}_{d',j-\tau'}}$$

which gives us the following simple multiplicative update

$$\mathbf{W}^{\tau'}_{i',d'} \leftarrow \mathbf{W}^{\tau'}_{i',d'} \frac{\sum_{\phi,j} \mathbf{V}_{i'+\phi,j}\ \mathbf{H}^{\phi}_{d',j-\tau'}}{\sum_{\phi,j} \mathbf{\Lambda}_{i'+\phi,j}\ \mathbf{H}^{\phi}_{d',j-\tau'}}.$$

A similar step size can be found for $\mathbf{H}$ also giving a simple multiplicative update. In matrix notation the updates of $\mathbf{W}^{\tau}$ and $\mathbf{H}^{\phi}$ can be written as:

$$\mathbf{W}^{\tau} \leftarrow \mathbf{W}^{\tau} \bullet \frac{\sum_{\phi} \overset{\uparrow\phi\ \ \rightarrow\tau}{\mathbf{V}}\ \mathbf{H}^{\phi}}{\sum_{\phi} \overset{\uparrow\phi\ \ \rightarrow\tau}{\mathbf{\Lambda}}\ \mathbf{H}^{\phi}}, \qquad \mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau} \overset{\downarrow\phi\ \ \leftarrow\tau}{\mathbf{W}^{\tau}}\ \mathbf{V}}{\sum_{\tau} \overset{\downarrow\phi\ \ \leftarrow\tau}{\mathbf{W}^{\tau}}\ \mathbf{\Lambda}}.$$

Alternating between the updates for $\mathbf{W}$ and $\mathbf{H}$ forms the algorithm for least squares NMF2D minimization. A proof of the convergence of the algorithm is given in Appendix A.

### 2.2 KL Divergence Cost Function

Consider the Kullback-Leibler divergence given by:

$$C_{KL} = \sum_{i,j} \mathbf{V}_{i,j} \log \frac{\mathbf{V}_{i,j}}{\mathbf{\Lambda}_{i,j}} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}.$$

While least square minimization attempts to retain as much of the variance as posible in the data, minimizing the KL divergence corresponds to assuming a multinomial noise model. Following the same steps as for the derivation of the algorithm for the least squares cost function, we get the following update equations for the KL divergence cost function

$$\mathbf{W}^\tau \leftarrow \mathbf{W}^\tau \bullet \frac{\sum_\phi \left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\uparrow\phi} \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}{\sum_\phi \mathbf{1}\ \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}, \qquad \mathbf{H}^\phi \leftarrow \mathbf{H}^\phi \bullet \frac{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi}\ \left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\leftarrow\tau}}{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi}\ \mathbf{1}}.$$

A proof of the convergence of the algorithm is given in Appendix B.

The two algorithms for NMF2D are summarized in Table 1.

**NMF2D Least Squares**

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ randomly.
2. $\mathbf{\Lambda} = \sum_{\tau,\phi} \mathbf{W}^\tau_{\downarrow\phi} \mathbf{H}^\phi_{\rightarrow\tau}$
3. $\mathbf{H}^\phi \leftarrow \mathbf{H}^\phi \bullet \frac{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi} \mathbf{V}^{\leftarrow\tau}}{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi} \mathbf{\Lambda}^{\leftarrow\tau}}$
4. $\mathbf{\Lambda} = \sum_{\tau,\phi} \mathbf{W}^\tau_{\downarrow\phi} \mathbf{H}^\phi_{\rightarrow\tau}$
5. $\mathbf{W}^\tau \leftarrow \mathbf{W}^\tau \bullet \frac{\sum_\phi \mathbf{V}^{\uparrow\phi} \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}{\sum_\phi \mathbf{\Lambda}^{\uparrow\phi} \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}$
6. Repeat from 2 until convergence.

**NMF2D KL-divergence**

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ randomly.
2. $\mathbf{\Lambda} = \sum_{\tau,\phi} \mathbf{W}^\tau_{\downarrow\phi} \mathbf{H}^\phi_{\rightarrow\tau}$
3. $\mathbf{H}^\phi \leftarrow \mathbf{H}^\phi \bullet \frac{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi} \left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\leftarrow\tau}}{\sum_\tau \mathbf{W}^{\tau\,\mathrm{T}}_{\downarrow\phi} \mathbf{1}}$
4. $\mathbf{\Lambda} = \sum_{\tau,\phi} \mathbf{W}^\tau_{\downarrow\phi} \mathbf{H}^\phi_{\rightarrow\tau}$
5. $\mathbf{W}^\tau \leftarrow \mathbf{W}^\tau \bullet \frac{\sum_\phi \left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\uparrow\phi} \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}{\sum_\phi \mathbf{1}\ \mathbf{H}^{\phi\,\mathrm{T}}_{\rightarrow\tau}}$
6. Repeat from 2 until convergence.

Table 1: Algorithms for Non-negative Matrix Factor 2-D Deconvolution.

### 2.3 Sparse Non-negative Matrix Factor 2-D Deconvolution

In the following we will extend the two NMF2D algorithms to give sparse decompositions, i.e. form a sparse NMF2D (SNMF2D). There are several reasons why a SNMF2D is of interest.

- The NMF2D is not in general unique. If the data does not span the positive octant adequately, a rotation of $\mathbf{W}$ and opposite rotation of $\mathbf{H}$ can give the same result (as for NMF, see Donoho and Stodden, 2003). By imposing sparseness the solution being the sparsest (which is often also the most interpretable) can be found.

- For double convolutive models the structure of a factor in $\mathbf{H}$ can to some extend be put into the signature of the same factor in $\mathbf{W}$ and vice versa (see also Figure 2). By imposing sparseness on $\mathbf{H}$ this ambiquity can be relieved by forcing the structure onto $\mathbf{W}$

- Let T and $\Phi$ denote the total numbers of $\tau$ and $\phi$ shifts respectively. Then the free parameters of the NMF2D model is $(I \cdot T + J \cdot \Phi) \cdot D$ where the size of the analyzed data is $I \cdot J$. Consequently, the NMF2D model tends to be overcomplete when allowing many shifts and components. In these overcomplete situations imposing sparseness is known to give good representations (Olshausen, 1996).

Consequently, the main problem of NMF2D is component ambiguity, i.e. lack of uniqueness in the decompositions. To improve uniqueness constraints in the form of sparsity has proven usefull (Hoyer, 2002, 2004; Eggert and Korner, 2004). While Hoyer (2002) uses the $L_1$-norm to penalize $\mathbf{H}$, we here derive in line with the approach of Eggert and Korner (2004) algorithms based on multiplicative updates using a general sparsity penalty term on $\mathbf{H}$ given by $f(\mathbf{H})$. This yields the following cost functions for the SNMF2D

$$C_{SLS} = \frac{1}{2}\sum_{i,j}(\mathbf{V}_{i,j} - \widetilde{\mathbf{\Lambda}}_{i,j})^2 + \beta f(\mathbf{H})$$

$$C_{SKL} = \sum_{i,j}\mathbf{V}_{i,j}\log\frac{\mathbf{V}_{i,j}}{\widetilde{\mathbf{\Lambda}}_{i,j}} - \mathbf{V}_{i,j} + \widetilde{\mathbf{\Lambda}}_{i,j} + \beta f(\mathbf{H}).$$

Here, $\widetilde{\mathbf{\Lambda}}$ is the model computed with factor wise normalized $\mathbf{W}$ such that the sparsity term $f(\mathbf{H})$ can't be minimized simply by letting $\mathbf{H}$ go to zero while $\mathbf{W}$ goes to infinity

$$\widetilde{\mathbf{\Lambda}} = \sum_{\tau,\phi}\overset{\downarrow\phi}{\widetilde{\mathbf{W}}}{}^{\tau}\ \overset{\rightarrow\tau}{\mathbf{H}}{}^{\phi} \quad \text{where} \quad \widetilde{\mathbf{W}}^{\tau}_{i,d} = \frac{\mathbf{W}^{\tau}_{i,d}}{\sqrt{\sum_{\tau,i}\left(\mathbf{W}^{\tau}_{i,d}\right)^2}} = \frac{\mathbf{W}^{\tau}_{i,d}}{\|\mathbf{W}_d\|_2}.$$

The parameter $\beta$ weights the importance of the sparsity term to the reconstruction. Following the same steps for minimizing these cost function as for the previously discussed algorithms, we get the following updates for SNMF2D summarized in Table 2.

Since the derivative of $f(\mathbf{H})$ is used in the updates a negative derivative can potentially result in negative updates. Consequently, $f(\mathbf{H})$ can be any function with positive derivative.

**SNMF2D Least Squares**

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ randomly.
2. Define $\widetilde{\mathbf{W}}^{\tau}_{i,d} = \frac{\mathbf{W}^{\tau}_{i,d}}{\|\mathbf{W}_d\|_2}$
3. $\widetilde{\mathbf{\Lambda}} = \sum_{\tau}\sum_{\phi} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau} \overset{\rightarrow\tau}{\mathbf{H}}^{\phi}$
4. $\mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau\mathrm{T}} \overset{\leftarrow\tau}{\mathbf{V}}}{\sum_{\tau} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau\mathrm{T}} \overset{\leftarrow\tau}{\widetilde{\mathbf{\Lambda}}} + \beta \frac{\partial f(\mathbf{H})}{\partial \mathbf{H}^{\phi}}}$
5. $\widetilde{\mathbf{\Lambda}} = \sum_{\tau}\sum_{\phi} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau} \overset{\rightarrow\tau}{\mathbf{H}}^{\phi}$
6. $\mathbf{W}^{\tau} \leftarrow \widetilde{\mathbf{W}}^{\tau} \bullet \frac{\sum_{\phi} \overset{\uparrow\phi\rightarrow\tau}{\mathbf{V}}^{\mathrm{T}} \mathbf{H}^{\phi} + \widetilde{\mathbf{W}}^{\tau} diag(\sum_{\tau} \mathbf{1}((\overset{\uparrow\phi\rightarrow\tau}{\mathbf{\Lambda}'}^{\mathrm{T}} \mathbf{H}^{\phi}) \bullet \widetilde{\mathbf{W}}^{\tau}))}{\sum_{\phi} \overset{\uparrow\phi\rightarrow\tau}{\widetilde{\mathbf{\Lambda}}}^{\mathrm{T}} \mathbf{H}^{\phi} + \widetilde{\mathbf{W}}^{\tau} diag(\sum_{\tau} \mathbf{1}((\overset{\uparrow\phi\rightarrow\tau}{\mathbf{V}}^{\mathrm{T}} \mathbf{H}^{\phi}) \bullet \widetilde{\mathbf{W}}^{\tau}))}$
7. Repeat from 2 until convergence.

**SNMF2D KL-divergence**

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ randomly.
2. Define $\widetilde{\mathbf{W}}^{\tau}_{i,d} = \frac{\mathbf{W}^{\tau}_{i,d}}{\|\mathbf{W}_d\|_2}$
3. $\widetilde{\mathbf{\Lambda}} = \sum_{\tau}\sum_{\phi} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau} \overset{\rightarrow\tau}{\mathbf{H}}^{\phi}$
4. $\mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau\mathrm{T}} \left(\frac{\overset{\leftarrow\tau}{\mathbf{V}}}{\widetilde{\mathbf{\Lambda}}}\right)}{\sum_{\tau} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau\mathrm{T}} \cdot \mathbf{1} + \beta \frac{\partial f(\mathbf{H})}{\partial \mathbf{H}^{\phi}}}$
5. $\widetilde{\mathbf{\Lambda}} = \sum_{\tau}\sum_{\phi} \overset{\downarrow\phi}{\widetilde{\mathbf{W}}}^{\tau} \overset{\rightarrow\tau}{\mathbf{H}}^{\phi}$
6. $\mathbf{W}^{\tau} \leftarrow \widetilde{\mathbf{W}}^{\tau} \bullet \frac{\sum_{\phi} \left(\frac{\overset{\uparrow\phi}{\mathbf{V}}}{\widetilde{\mathbf{\Lambda}}}\right) \overset{\rightarrow\tau}{\mathbf{H}}^{\phi\mathrm{T}} + \widetilde{\mathbf{W}}^{\tau} diag(\sum_{\tau} \mathbf{1}((\mathbf{1} \overset{\rightarrow\tau}{\mathbf{H}^{\phi}}^{T}) \bullet \widetilde{\mathbf{W}}))}{\sum_{\phi} (\mathbf{1} \cdot \overset{\rightarrow\tau}{\mathbf{H}^{\phi}}^{\mathrm{T}} + \widetilde{\mathbf{W}}^{\tau} diag(\sum_{\tau} \mathbf{1}((\left(\frac{\overset{\uparrow\phi}{\mathbf{V}}}{\widetilde{\mathbf{\Lambda}}}\right) \overset{\rightarrow\tau}{\mathbf{H}^{\phi}}^{\mathrm{T}}) \bullet \widetilde{\mathbf{W}}^{\tau}))}$
7. Repeat from 2 until convergence.

Table 2: Algorithms for Sparse Non-negative Matrix Factor 2-D Deconvolution. Here, diag$(\cdot)$ denotes a matrix with the argument on the diagonal.

For example $f(\mathbf{H})$ can be the $L_\alpha$-norm ($\alpha > 0$) given by

$$f(\mathbf{H}) = \|\mathbf{H}\|_\alpha = \left( \sum_{\phi,d,j} |\mathbf{H}_{d,j}^{\phi}|^\alpha \right)^{1/\alpha} \quad \text{then} \quad \frac{\partial f(\mathbf{H})}{\partial \mathbf{H}^{\phi}} = \frac{\mathbf{H}^{\phi^{\bullet(\alpha-1)}}}{\|\mathbf{H}\|_\alpha^{\alpha-1}}$$

Where $\mathbf{H}^{\phi^{\bullet(\alpha-1)}}$ denotes raising each element in $\mathbf{H}^{\phi}$ to the power $(\alpha - 1)$.

While the convergence of the updates of $\mathbf{H}$ using least square minimization with the $L_1$ norm follows straight forward from the proof of convergence of the $\mathbf{H}$ update given in Hoyer (2002), see also Appendix A the convergence of the $\mathbf{H}$ update is also easily proven for the KL-algorithm, see Appendix B. We haven't been able to prove the convergence of $\mathbf{W}$ for any of the two algorithms. Eggert and Korner (2004) conjectured their sparse algorithm for NMF least square minimization is convergent. We will conjecture the SNMF2D based on LS as well as KL-minimization are also convergent since extensive tests of the SNMF2D algorithms showed no signs of divergence.

## 3. Examples

The NMF2D based on LS and KL-divergence minimization were tested in their ability to find the components of two different simulated data sets. The first dataset was created to have little ambiguity between $\mathbf{H}$ and $\mathbf{W}$ while the second data set had a high degree of ambiguity.

As shown in Figure 1 both methods give a good reconstruction of the first simulated data set correctly identifying the three circles and crosses in $\mathbf{W}$. Since the amount of $\phi$ shifts were less than the size of the structure in $\mathbf{W}$ the data could not be reconstructed by putting the structure in $\mathbf{W}$ onto $\mathbf{H}$.

In the second data set the shifts $\phi$ was in the range of the two components in $\mathbf{W}$ now only consisting of one circle and a cross. Consequently, the data could be reasonable well explained by letting $\mathbf{H}$ draw some or all of the circles and crosses as seen in Figure 2. As shown in the figure imposing sparseness relieves this ambiguity between $\mathbf{W}$ and $\mathbf{H}$.

## 4. Discussion

From the first simulated data set it is seen that when no ambiguity between $\mathbf{W}$ and $\mathbf{H}$ is present, both NMF2D algorithms correctly identifies the components. However, when ambiguity is present as in the second simulated data set the NMF2D algorithms failed in always identifying the correct components. However, the SNMF2D algorithms here using the $L_1 - norm$ could resolve the ambiguity by forcing all structure in $\mathbf{H}$ onto $\mathbf{W}$ giving the correct components. However, the sparseness constraint is not guaranteed to always remove the cost function from a suboptimal solution.

Notice how only half the cross is recovered as the algorithm converges to the Local Minimum 2 in figure 2. Once the algorithm has placed the components in $\mathbf{W}$ it cannot always relocate the component as it becomes better identified. The NMF2D model suffers from a shift redundancy between the factors of $\mathbf{W}$ and $\mathbf{H}$, such that a factor in $\mathbf{W}$ can be shifted in any direction as long as the corresponding factor in $\mathbf{H}$ is shifted conversely, if we disregard edge effects. In order to relive this shift ambiguity $\mathbf{W}$ and $\mathbf{H}$ can be aligned to

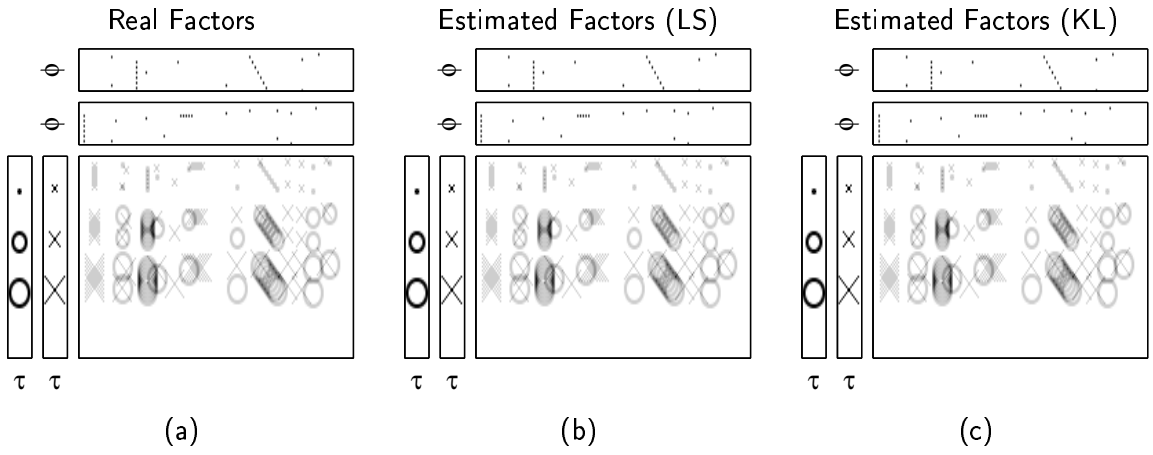


Figure 1: NMF2D on a toy problem: **(a)** The simulated data. $\mathbf{W}$ consists of three crosses of varying size in the first factor and three circles of varying size in the second. These signatures are convolved with $\mathbf{H}$ given in the top of the figure to yield the data matrix $\mathbf{V}$ which is a mix of both components. **(b)** The Result when analyzing the simulated data $\mathbf{V}$ using NMF2D based on least square minimization. Clearly, the algorithm successfully identifies the two components. **(c)** The result when analyzing the simulated data $\mathbf{V}$ using NMF2D based on the KL-divergence minimization. Again the two components are successfully identified.

meet specific criteria, for example by shifting the factors such that the mean value of the column coefficients in each component in $\mathbf{W}$ are set to attain the maximum at the center column of $\mathbf{W}$ and likewise for the rows of $\mathbf{H}$. Aligning $\mathbf{W}$ and $\mathbf{H}$ would have circumvented this local minimum.

While the LS-algorithm very often ended in suboptimal solutions the KL algorithm almost always identifyed the correct components in the data set having ambiguities. Consider the model error $\mathbf{E}_{i,j} = \mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j}$. We then have $C_{LS} = \sum_{ij} \mathbf{E}_{i,j}^2$ while $C_{KL} = Constant + \sum_{ij} -\mathbf{V}_{i,j} log(\mathbf{V}_{i,j} - \mathbf{E}_{i,j}) - \mathbf{E}_{i,j}$. Consequently, outliers are weighted relatively stronger for the LS than the KL algorithm. The stronger focus on outliers might result in more local minimas in the LS cost function since the improvement (reduction) of some of the residuals $\mathbf{E}_{i,j}$ has to counteract potential increase in error of other residuals $\mathbf{E}_{i',j'}$. This could be the reason why the KL algorithm more often attained the global optimum. However, this issue needs further investigation.

Choosing $f(\mathbf{H})$ as well as the weight of sparseness to the reconstruction, $\beta$, significantly impact the solutions found. In general choosing $f(\mathbf{H})$ to be the $L_\alpha$-norm where $\alpha < 1$ will penalize small values relatively more than large values of $\mathbf{H}$ and vice versa for $\alpha > 1$. It is worth noting that norms less than 1 are non-convex and consequently results in cost functions suffering of many local suboptimal solutions. Furthermore, the $L_1$-norm corresponds to translating the data towards the x-axis which results in a form of thresholding of the values in $\mathbf{H}$. By the algorithm devised for SNMF2D the sparsity penalty $f(\mathbf{H})$ the most adequate for the given data can be chosen.

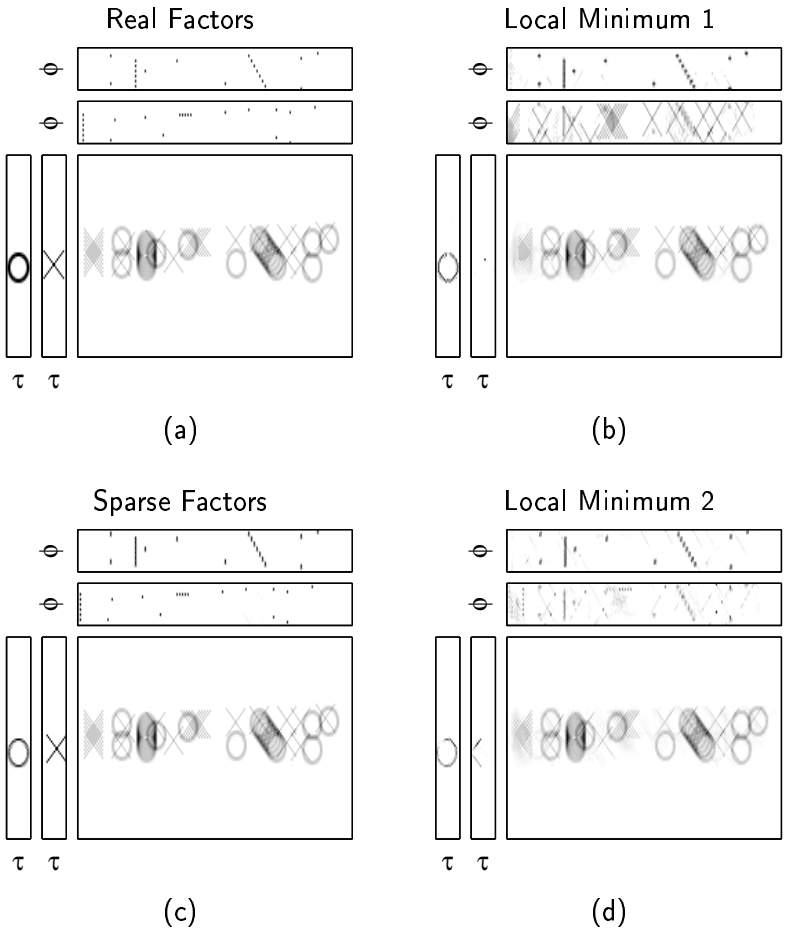


Figure 2: **(a)** The simulated data now with the **W** matrix only having one circle and a cross . **(b)**, **(d)** Examples of results ending in suboptimal solutions when analyzing the data. The ambiguity between **W** and **H** is so large that all of the crosses in b and part of the crosses and circles in d are described by drawing them in **H** instead of **W**. **(c)** When imposing sparseness ($f(\mathbf{H})$ given by the $L_1 - norm$) this ambiquity is circumvented. Primarily, the suboptimal solutions were reached by the LS algorithm while the KL algorithm almost always correctly identified the components.

Let T and $\Phi$ denote the total numbers of $\tau$ and $\phi$ shifts respectively. Then all the algorithms presented here are $\mathcal{O}(I \cdot J \cdot D \cdot T \cdot \Phi)$ i.e. for a given $D \cdot T \cdot \Phi$ the algorithms grow linear with the size of $V = I \cdot J$. Consequently, even for large problems the algorithms efficiently fit convolutive ($\phi = 0$ or $\tau = 0$) and double convolutive models. Furthermore, as the convolutive non-negative matrix factorization introduced by Smaragdis (2004) is a special case of the double convolutive non-negative matrix factorization presented here. The convergence of Smaragdis single convolutive factorization follows from the convergence of the corresponding double convolutive algorithm by setting either $\tau$ or $\phi$ to 0.

The algorithms above were derived for non-negative decomposition yielding multiplicative updates. However, the derived gradients can be used to form a gradient based algorithms for a general matrix factor 2-D deconvolution where $\mathbf{W}$ and $\mathbf{H}$ is not assumed non-negative.

## 5. Conclusion

We gave two algorithms for NMF2D and proved their convergence. We further gave two algorithms we conjecture convergent for sparse NMF2D (SNMF2D). While NMF2D successfully identified the components of data where the structure of the components was greater than the possible shifts, the SNMF2D was capable of correctly identify the components of data having ambiguity between $\mathbf{W}$ and $\mathbf{H}$. The NMF2D and SNMF2D have proven useful in separation (Schmidt and Mørup, 2005) and transcription of music (Schmidt and Mørup, 2006). It is our strong belief that the NMF2D and SNMF2D model will be useful in a wide range of signal analysis primarily where NMF already has been applied.

## Appendix A. Convergence of the LS Algorithm

The proof is based on the use of an auxiliary function and follows closely the proofs for the convergence of NMF algorithm of Lee and Seung (2000). Briefly stated, an auxiliary function G to the function F is defined by: $G(H, H^t) \geq F(H)$ and $G(H, H) = F(H)$. If G is an auxiliary function then F is non-increasing under the update $H = \arg\min_H G(H, H^t)$.

The proof of convergence of the least squares updates follows essentially the proof of the least squares NMF updates of Lee and Seung (2000). We start by defining:

$$F(\mathbf{H}) = \frac{1}{2} \sum_{i,j} \left( \mathbf{V}_{i,j} - \sum_{\tau,\phi,d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau} \right)^2$$

Notice that F is just the regular least square cost function $C_{LS}$. Define the vector $\mathbf{h}_a$ as $\mathbf{h}_a = \mathbf{H}^{\phi}_{d,k}$. This vector is simply a vectorization of $\mathbf{H}$ where $a$ indexes all combinations of $\phi$, $d$, and $k$ . The gradient vector $\nabla F_a$ and Hessian matrix $\mathbf{Q}_{a,b}$ found by differentiating F with respect to the element $\mathbf{H}^{\phi}_{d,k}$ and $\mathbf{H}^{\phi'}_{d',k'}$ denoted by $a$ and $b$, gives:

$$\begin{aligned} \nabla F_a = \frac{\partial C_{LS}}{\partial \mathbf{H}^{\phi'}_{d',k'}} &= -\sum_{\tau,i} \left( \mathbf{V}_{i,k'+\tau} - \mathbf{\Lambda}_{i,k'+\tau} \right) \mathbf{W}^{\tau}_{i-\phi',d'} \\ \mathbf{Q}_{a,b} = \frac{\partial F(\mathbf{H})^2}{\partial \mathbf{H}^{\phi'}_{d',k'} \partial \mathbf{H}^{\phi}_{d,k}} &= \sum_{\tau,i} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{W}^{k-k'+\tau}_{i-\phi',d'} \end{aligned}$$

Since $F(\mathbf{H})$ is a quadratic function it is completely described by a second order Taylor expansion here expressed in terms of $\mathbf{h}$ as:

$$F(\mathbf{h}) = F(\mathbf{h}^t) + (\mathbf{h} - \mathbf{h}^t)^{\mathrm{T}} \nabla F(\mathbf{h}^t) + \frac{1}{2}(\mathbf{h} - \mathbf{h}^t)^{\mathrm{T}} \mathbf{Q} (\mathbf{h} - \mathbf{h}^t)$$

Now let $K(\mathbf{h}^t)$ be a diagnoal matrix defined by

$$K(\mathbf{h}^t)_{ab} = \delta_{ab} (\mathbf{Q}\mathbf{h}^t)_a / (\mathbf{h}^t)_a.$$

Further, define the auxiliary function

$$G(\mathbf{h}, \mathbf{h}^t) = F(\mathbf{h}^t) + (\mathbf{h} - \mathbf{h}^t) \nabla F(\mathbf{h}^t) + \frac{1}{2}(\mathbf{h} - \mathbf{h}^t)^{\mathrm{T}} K(\mathbf{h}^t)(\mathbf{h} - \mathbf{h}^t).$$

Clearly $G(\mathbf{h}, \mathbf{h}) = F(\mathbf{h})$. Finding $G(\mathbf{h}, \mathbf{h}^t) \geq F(\mathbf{h}^t)$ corresponds to

$$(\mathbf{h} - \mathbf{h}^t)^{\mathrm{T}} (K(\mathbf{h}^t) - \mathbf{Q})(\mathbf{h} - \mathbf{h}^t) \geq 0$$

This requires the matrix $(K(\mathbf{h}^t) - \mathbf{Q})$ to be positive semidefinite (Lee and Seung, 2000).

The rest of the proof follows closely the convergence proof of the regular NMF (Lee and Seung, 2000). Define the matrix $\mathbf{M}_{a,b}(\mathbf{h}^t) = \mathbf{h}^t_a (K(\mathbf{h}^t) - \mathbf{Q})_{a,b} \mathbf{h}^t_b$. This is just a rescaling of the elements in $(K(\mathbf{h}^t) - \mathbf{Q})$. Then $(K(\mathbf{h}^t) - \mathbf{Q})$ is semipositive definite if and only if $\mathbf{M}$ is

$$\begin{aligned}
\nu^t \mathbf{M} \nu &= \sum_{ab} \nu^t_a \mathbf{M}_{a,b} \nu_b \\
&= \sum_{ab} \nu^t_a (\mathbf{h}^t_a (\delta_{ab} (\mathbf{Q}\mathbf{h}^t)_a / (\mathbf{h}^t)_a - \mathbf{Q})_{a,b} \mathbf{h}^t_b) \nu_b \\
&= \sum_{ab} \mathbf{h}^t_a \mathbf{Q}_{a,b} \mathbf{h}^t_b \nu^2_a - \nu_a \mathbf{h}^t_a \mathbf{Q} \mathbf{h}^t_b \nu_b \\
&= \sum_{ab} \mathbf{Q}_{a,b} \mathbf{h}^t_a \mathbf{h}^t_b (\frac{1}{2}\nu^2_a + \frac{1}{2}\nu^t_b - \nu_a \nu_b) \\
&= \frac{1}{2} \sum_{ab} \mathbf{Q}_{a,b} \mathbf{h}^t_a \mathbf{h}^t_b (\nu_a - \nu_b)^2 \geq 0
\end{aligned}$$

all that is left to prove is that minimizing G yield the least square updates

$$\begin{aligned}
\frac{\partial G(\mathbf{h}, \mathbf{h}^t)}{\partial \mathbf{h}} &= 0 \\
\Leftrightarrow \mathbf{h} &= \mathbf{h}^t - K(\mathbf{h}^t)^{-1} \nabla F(\mathbf{h}^t) \\
\Leftrightarrow \mathbf{h}_a &= \mathbf{h}^t_a - \frac{(\mathbf{h}^t)_a}{(\mathbf{Q}\mathbf{h}^t)_a} \nabla F(\mathbf{h}^t)_a.
\end{aligned}$$

Changing the indexing $a$ to be of the parameters $\phi$, $d$, and $j$, we get

$$(\mathbf{Q}\mathbf{h}^t)_a = \sum_{\tau, i} \mathbf{W}^{\tau}_{i-\phi,d} \sum_{j',d',\phi'} \mathbf{W}^{j-j'+\tau}_{i-\phi',d'} \mathbf{H}^{\phi'}_{d',j'} = \sum_{\tau,i} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{\Lambda}_{i,j+\tau}.$$

Consequently

$$\begin{aligned}\mathbf{H}^{\phi}_{d,k} &= \mathbf{H}^{\phi^t}_{d,k} + \frac{\mathbf{H}^{\phi^t}_{d,k}\sum_{\tau,i}\mathbf{W}^{\tau}_{i-\phi,d}(\mathbf{V}_{i,\tau+k}-\mathbf{\Lambda}_{i,\tau+k})}{\sum_{\tau,i}\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{\Lambda}_{i,k+\tau}} \\ &= \mathbf{H}^{\phi^t}_{d,k}\frac{\sum_{\tau,i}\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{V}_{i,\tau+k}}{\sum_{\tau,i}\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{\Lambda}_{i,\tau+k}},\end{aligned}$$

which concludes the proof. By the symmetry of $\mathbf{W}$ and $\mathbf{H}$ the proof of convergence of the $\mathbf{W}$ update is achieved by interchanging the roles of $\mathbf{W}$ and $\mathbf{H}$ in the above.

For the SNMF2D algorithm based on LS the convergence of the $\mathbf{H}$ update is easily proven for $L_1$ defining $K(\mathbf{h}^t)_{ab} = \delta_{ab}\frac{(\mathbf{Q}\mathbf{h}^t)_a+\beta}{(\mathbf{h}^t)_a}$ as proposed by Hoyer (2002) for regular NMF. The convergence of $\mathbf{H}$ for any $\alpha \neq 1$ is however not easy to prove since the auxiliary function is neither fully explained by a second order Taylor expansion, nor can a simple $K(\mathbf{h}^t)_{ab}$ be defined to give the correct updates. We were also unable to proof the updates of $\mathbf{W}$ since the normalization of the factors $\mathbf{W}$ results in the cost function not being explained fully by any finite Taylor expansion.

## Appendix B. Convergence of the KL Algorithm

This proof follows straightforward the proof of the convergence for NMF given by Lee and Seung (2000) again the proof is based on the use of auxiliary functions, see B. The main ingredient is to note than by the convexity of the log function, we have:

$$-\log\sum_a x_a \leq -\sum_a \alpha_a \log\frac{x_a}{\alpha_a} \quad \text{if} \quad \sum_a \alpha_a = 1$$

Define $\alpha_a$ by the three indices $d, \phi$ and $\tau$:

$$\alpha_a = \frac{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H^t}^{\phi}_{d,j-\tau}}{\mathbf{\Lambda^t}_{i,j}}$$

Where $\mathbf{\Lambda}^t_{i,j}$ is the reconstruction found using $\mathbf{H}^t$. Notice $\sum_{d,\phi,\tau}\alpha_a = 1$. We now have:

$$\begin{aligned}F(\mathbf{H}) &= \sum_{i,j}\left(\mathbf{V}_{i,j}\log\frac{\mathbf{V}_{i,j}}{\mathbf{\Lambda}_{i,j}} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}\right) \\ &= \sum_{i,j}\left(\mathbf{V}_{i,j}\log\mathbf{V}_{i,j} - \mathbf{V}_{i,j}\log\mathbf{\Lambda}_{i,j} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}\right) \\ &= \sum_{i,j}\left(\mathbf{V}_{i,j}\log\mathbf{V}_{i,j} - \mathbf{V}_{i,j}\log\sum_{\phi d\tau}\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H}^{\phi}_{d,j-\tau} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}\right) \\ &\leq \sum_{i,j}\left(\mathbf{V}_{i,j}\log\mathbf{V}_{i,j} - \mathbf{V}_{i,j}\sum_{\phi d\tau}\frac{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H^t}^{\phi}_{d,j-\tau}}{\mathbf{\Lambda}^t_{i,j}}\log\frac{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H}^{\phi}_{d,j-\tau}}{\frac{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H^t}^{\phi}_{d,j-\tau}}{\mathbf{\Lambda^t}_{i,j}}} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}\right) \\ &= G(\mathbf{H},\mathbf{H}^{\mathrm{T}})\end{aligned}$$

Since $G(\mathbf{H},\mathbf{H}) = F(\mathbf{H})$ which is easy to see from the above - we only need to prove that minimizing G yields the KL-updates:

$$\begin{aligned}
\frac{\partial G(\mathbf{H},\mathbf{H}^{\mathrm{t}})}{\mathbf{H}^{\phi}_{d,k}} &= \frac{\partial}{\mathbf{H}^{\phi}_{d,k}} \sum_{i,j} \left( -\mathbf{V}_{i,j} \sum_{\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}} \log \frac{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H}^{\phi}_{d,j-\tau}}{\frac{\mathbf{W}^{\tau}_{i-\phi,d}{\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}}} + \mathbf{\Lambda}_{i,j} \right) \\
&= \frac{\partial}{\mathbf{H}^{\phi}_{d,k}} \sum_{i,j} \left( -\mathbf{V}_{i,j} \sum_{\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}} \log \mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H}^{\phi}_{d,j-\tau} + \mathbf{\Lambda}_{i,j} \right) \\
&= \sum_{i,\tau} \left( -\mathbf{V}_{i,k+\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}} \frac{1}{\mathbf{W}^{\tau}_{i-\phi,d}\mathbf{H}^{\phi}_{d,k}} \mathbf{W}^{\tau}_{i-\phi,d} + \mathbf{W}^{\tau}_{i-\phi,d} \right) \\
&= \frac{1}{\mathbf{H}^{\phi}_{d,k}} \sum_{i,\tau} -\mathbf{V}_{i,k+\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}} + \sum_{i,\tau} \mathbf{W}^{\tau}_{i-\phi,d}
\end{aligned}$$

Equating this gradient to zero gives:

$$\begin{aligned}
\mathbf{H}^{\phi}_{d,k} &= \frac{\sum_{i,\tau} \mathbf{V}_{i,k+\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}}}{\sum_{i,\tau} \mathbf{W}^{\tau}_{i-\phi,d}} \\
&= \frac{\sum_{i,\tau} \mathbf{V}_{i,k+\tau} \frac{\mathbf{W}^{\tau}_{i-\phi,d} {\mathbf{H}^{\mathrm{t}}}^{\phi}_{d,j-\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,j}}}{\sum_{i,\tau} \mathbf{W}^{\tau}_{i-\phi,d}} \\
&= \mathbf{H}^{t^{\phi}}_{d,k} \frac{\sum_{i,\tau} \mathbf{W}^{\tau}_{i-\phi,d} \frac{\mathbf{V}_{i,k+\tau}}{{\mathbf{\Lambda}^{\mathbf{t}}}_{i,k+\tau}}}{\sum_{i,\tau} \mathbf{W}^{\tau}_{i-\phi,d}}
\end{aligned}$$

Which concludes the proof. By the symmetry of $\mathbf{W}$ and $\mathbf{H}$ the proof of convergence of the $\mathbf{W}$ update is achieved by interchanging the roles of $\mathbf{W}$ and $\mathbf{H}$ in the above.

For the SNMF2D algorithm based on KL the proof of the $\mathbf{H}$ updates follows by replacing $\mathbf{\Lambda}$ with $\widetilde{\mathbf{\Lambda}}$ and $\mathbf{W}$ with $\widetilde{\mathbf{W}}$ in the above while defining $\widetilde{F}(\mathbf{H}) = F(\mathbf{H}) + \beta f(\mathbf{H})$ and $\widetilde{G}(\mathbf{H},\mathbf{H}^t) = G(\mathbf{H},\mathbf{H}^t) + \beta f(\mathbf{H})$. Clearly $\widetilde{G}(\mathbf{H},\mathbf{H}^t)$ is then an auxiliary function of $\widetilde{F}(\mathbf{H})$ while equating the gradient of $\widetilde{G}(\mathbf{H},\mathbf{H}^t)$ to zero gives the $\mathbf{H}$ update.

Contrary to the $\mathbf{H}$ update we were unable to prove the convergence of the $\mathbf{W}$ update. Defining the auxiliary function $\widetilde{G}(\mathbf{W},\mathbf{W}^t)$ and equating the derivative of this function to zero we were unable to isolate $\mathbf{W}$ as a function of $\mathbf{W}^t$.

## References


David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *NIPS*, 2003.

M. Dyrholm, S. Makeig, and L. K. Hansen. Model structure selection in convolutive mixtures. In *6th International Conference on Independent Component Analysis and Blind Source Separation*, 2006.

J. Eggert and E. Korner. Sparse coding and nmf. In *Neural Networks*, volume 4, pages 2529–2533, 2004.

J. Eggert, H. Wersing, and E. Korner. Transformation-invariant representation and nmf. In *Neural Networks*, volume 4, pages 2535– 2539, 2004.

P.O. Hoyer. Non-negative sparse coding. *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565, 2002.

P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 2004.

Daniel D Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91, 1999. ISSN 00280836.

Hoang-Lan Nguyen Thi and Christian Jutten. Blind source separation for convolutive mixtures. *Signal Processing*, 45(2):209–229, 1995. ISSN 01651684.

Field D.J. Olshausen, B. A. Emergence of simple-cell receptiove field propertises by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

L. Parra, C. Spence, and B. De Vries. Convolutive blind source separation based on multiple decorrelation. *Neural Networks for Signal Processing VIII, 1998. Proceedings of the 1998 IEEE Signal Processing Society Workshop*, pages 23 –32, 1998.

M. N. Schmidt and M. Mørup. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. In *ICA2006*, 2005.

M.N. Schmidt and M. Mørup. Sparse non-negative matrix factor 2-d deconvolution for automatic transcription of polyphonic music. *Technical Report, Technical University of Denmark*, 2006.

Paris Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs. *International Symposium on Independent Component Analysis and Blind Source Separation (ICA)*, 3195:494, sep 2004.