# Non-negative mixtures

Plumbley, MD; Cichocki, A; Bro, R

Accepted preprint of:
M. D. Plumbley, A. Cichocki and R. Bro. Non-negative mixtures. In P. Comon and C. Jutten (Ed), Handbook of Blind Source Separation: Independent Component Analysis and Applications. Chapter 13, pp 515-547. Academic Press, Feb 2010. ISBN-13: 978-0-12-374726-6. doi 10.1016/B978-0-12-374726-6.00018-7

# Chapter 1

# Non-negative mixtures

M. D. Plumbley, A. Cichocki and R. Bro

## 1.1 Introduction

Many real-world unmixing problems involve inherent non-negativity constraints. Most physical quantities are non-negative: lengths, weights, amounts of radiation, and so on. For example, in the field of air quality, the amount of a particulate from a given source in a particular sample must be non-negative; and in musical audio signal processing, each musical note contributes a non-negative amount to the signal power spectrum. This type of non-negativity constraint also arises in, e.g. hyperspectral image analysis for remote sensing, positron emission tomography (PET) image sequences in medical applications, or semantic analysis of text documents.

Often we lose this non-negativity constraint when, for example, we subtract the mean from the data, such as when we perform the usual pre-whitening process for independent component analysis (ICA). However, we need to be aware that doing this may lose us important information that could help find the solution to our unmixing problem. Even where the non-negativity constraint is not inherently part of the problem, analogies with biological information processing systems suggest that this is an interesting direction to investigate, since information in neural systems is typically communicated using spikes, and the spike rate is a non-negative quantity.

In this chapter we discuss some algorithms for the use of non-negativity constraints in unmixing problems, including *positive matrix factorization* (PMF) [71], *non-negative matrix factorization* (NMF), and their combination with other unmixing methods such as *non-negative ICA* and sparse non-negative matrix factorization. The 2-D models can be naturally extended to multiway array (tensor) decompositions, especially Non-negative Tensor Factorization (NTF) and Non-negative Tucker Decomposition (NTD).

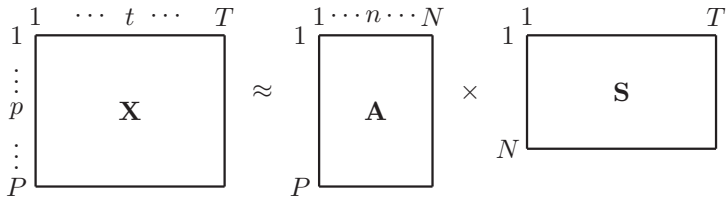


Figure 1.1: Basic NMF model $\mathbf{X} \approx \mathbf{AS}$

## 1.2 Non-negative Matrix Factorization

Suppose that our sequence of observation vectors $\mathbf{x}_t$, $1 \leqslant t \leqslant T$ is approximated by a linear mixing model

$$\mathbf{x}_t \approx \mathbf{A}\mathbf{s}_t = \sum_n \mathbf{a}_n s_{nt}$$

or in matrix notation

$$\mathbf{X} \approx \mathbf{AS} = \mathbf{A}\,\mathbf{V}^T = \sum_n \mathbf{a}_n \mathbf{v}_\mathbf{n}^\mathbf{T} \tag{1.1}$$

where $\mathbf{X} = [x_{pt}]$ is a data matrix of observations $x_{pt}$ for the $p$-th source at the $t$-th sample, $\mathbf{A} = [a_{pn}] = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N] \in \mathbb{R}^{P \times N}$ is a mixing matrix giving the contribution of the $n$-th source to the $p$ observation, and $\mathbf{S} = [s_{nt}]$ is a source matrix giving the value for the $n$-th mixture at the $t$-th sample (Fig. 1.1) and for convenience we use $\mathbf{V} = \mathbf{S}^T = [\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_N}] \in \mathbb{R}^{\mathbf{T} \times \mathbf{N}}$.

In this chapter, we are interested in the conditions where the sources $\mathbf{S}$ and/or the mixing contributions $\mathbf{A}$ are *non-negative*. The problem of finding $\mathbf{A}$ and $\mathbf{S}$ given only the observed mixtures $\mathbf{X}$ when both $\mathbf{A}$ and $\mathbf{S}$ and are non-negative first analyzed by Leggett [59] under the name *curve-resolution* and later by Paatero and Tapper [71] as the *positive matrix factorization* (PMF). Although the method was commonly used in certain fields, it was later re-invented and popularized by Lee and Seung as the *non-negative matrix factorization* (NMF) [56]. In the ten years since the Lee and Seung paper appeared in *Nature*, there have been hundreds of papers describing algorithms and applications of NMF[1].

In "plain" NMF we only assume non-negativity of $\mathbf{A}$ and $\mathbf{S}$. Unlike blind source separation methods based on independent component analysis (ICA) we do not assume that the sources $s_n$ are independent, although we will introduce other assumptions or constraints on $\mathbf{A}$ or $\mathbf{S}$ later. We notice that this symmetry of assumptions leads to a symmetry in the factorization: for (1.1) we could just as easily write

$$\mathbf{X}^\mathrm{T} \approx \mathbf{S}^\mathrm{T}\mathbf{A}^\mathrm{T} \tag{1.2}$$

[1]While the terms *curve-resolution* and *PMF* pre-date NMF, we will prefer *NMF* in this chapter due to its widespread popular use in the source separation literature

so the meaning of "source" and "mixture" are somewhat arbitrary.

The standard NMF model has been extended in various ways, including Semi-NMF, Multi-layer NMF, Tri-NMF, Orthogonal NMF, Non-smooth NMF and Convolutive NMF. We shall explore some of these extensions later (Section 1.3).

### 1.2.1 Simple gradient descent

Let us first develop a simple alternating gradient descent method to solve the standard NMF problem (1.1) for $\mathbf{A}$ and $\mathbf{S}$ given the observations $\mathbf{X}$. Consider the familiar Euclidean distance cost function

$$J_{\mathrm{E}} = D_{\mathrm{E}}(\mathbf{X}; \mathbf{AS}) = \tfrac{1}{2}\|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2 = \tfrac{1}{2}\sum_{pt}(x_{pt} - [\mathbf{AS}]_{pt})^2 \tag{1.3}$$

where $[\mathbf{M}]_{pt}$ is the $(p,t)$-th element of the matrix $\mathbf{M}$. For a simple gradient descent step for $\mathbf{S}$, we wish to update $\mathbf{S}$ according to

$$\mathbf{S} \leftarrow \mathbf{S} - \eta\frac{\partial J_{\mathrm{E}}}{\partial \mathbf{S}} \tag{1.4}$$

where $\eta$ is a small update factor and $[\partial J_{\mathrm{E}}/\partial \mathbf{S}]_{nt} = \partial J_{\mathrm{E}}/\partial s_{nt}$, or as individual terms

$$s_{nt} \leftarrow s_{nt} - \eta_{nt}\frac{\partial J_{\mathrm{E}}}{\partial s_{nt}} \tag{1.5}$$

where we now allow $\eta_{nt}$ to take on different values for each combination of $(n,t)$.

In order to calculate the partial derivative, consider that our cost function

$$J_{\mathrm{E}} = \tfrac{1}{2}\|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2 = \tfrac{1}{2}\operatorname{trace}((\mathbf{X} - \mathbf{AS})^{\mathrm{T}}(\mathbf{X} - \mathbf{AS})) \tag{1.6}$$

obtains an infinitesimal change $J_{\mathrm{E}} \leftarrow J_{\mathrm{E}} + \partial J_{\mathrm{E}}$ due to an infinitesimal change to $\mathbf{S}$,

$$\mathbf{S} \leftarrow \mathbf{S} + \partial \mathbf{S}. \tag{1.7}$$

Differentiating (1.6) w.r.t. this infinitesimal change $\partial \mathbf{S} = [\partial s_{nt}]$ we get

$$\begin{aligned}
\partial J_{\mathrm{E}} &= -\operatorname{trace}((\mathbf{X} - \mathbf{AS})^{\mathrm{T}}\mathbf{A}\partial \mathbf{S}) && (1.8)\\
&= -\operatorname{trace}((\mathbf{A}^{\mathrm{T}}\mathbf{X} - \mathbf{A}^{\mathrm{T}}\mathbf{AS})^{\mathrm{T}}\partial \mathbf{S}) && (1.9)\\
&= -\sum_{nt}[\mathbf{A}^{\mathrm{T}}\mathbf{X} - \mathbf{A}^{\mathrm{T}}\mathbf{AS}]_{nt}\partial s_{nt} && (1.10)
\end{aligned}$$

and hence

$$\frac{\partial J_{\mathrm{E}}}{\partial s_{nt}} = -[\mathbf{A}^{\mathrm{T}}\mathbf{X} - \mathbf{A}^{\mathrm{T}}\mathbf{AS}]_{nt} = -([\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt} - [\mathbf{A}^{\mathrm{T}}\mathbf{AS}]_{nt}). \tag{1.11}$$

Substituting (1.11) into (1.5) we get

$$s_{nt} \leftarrow s_{nt} + \eta_{nt}([\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt} - [\mathbf{A}^{\mathrm{T}}\mathbf{AS}]_{nt}) \tag{1.12}$$

or gradient update step for $s_{nt} = [\mathbf{S}]_{nt}$. Due to the symmetry between $\mathbf{S}$ and $\mathbf{A}$, a similar procedure will derive

$$a_{pn} \leftarrow a_{pn} + \eta_{pn}([\mathbf{X}\mathbf{S}^{\mathrm{T}}]_{pn} - [\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn}) \tag{1.13}$$

as the gradient update step for $a_{pn} = [\mathbf{A}]_{pn}$. A simple gradient update algorithm would therefore be to alternate between applications of (1.12) and (1.13) until convergence, while maintaining the non-negativity of the elements $a_{pn}$ and $s_{nt}$, i.e. we would actually apply

$$s_{nt} \leftarrow [s_{nt} + \eta_{nt}([\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt} - [\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt})]_{+} \tag{1.14}$$

where $[s]_{+} = \max(0, s)$ is the rectification function, and similarly for $a_{pn}$.

### 1.2.2 Multiplicative updates

While gradient descent is a simple procedure, convergence can be slow, and the convergence can be sensitive to the step size. In an attempt to overcome this, Lee and Seung [57] applied *multiplicative update rules*, which have proved particularly popular in NMF applications since then.

To construct a multiplicative update rule for $s_{nt}$, we can choose $\eta_{nt}$ such that the first and third terms on the RHS of (1.12) cancel, i.e. $s_{nt} = \eta_{nt}[\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt}$ or $\eta_{nt} = s_{nt}/[\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt}$. Substituting this back into (1.12) we get

$$s_{nt} \leftarrow s_{nt}\frac{[\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt}}{[\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt}} \tag{1.15}$$

which is now in the form of a multiplicative update to $s_{nt}$. Repeating the process for $a_{pn}$ we get the update rule pair

$$a_{pn} \leftarrow a_{pn}\frac{[\mathbf{X}\mathbf{S}^{\mathrm{T}}]_{pn}}{[\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn}} \qquad s_{nt} \leftarrow s_{nt}\frac{[\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt}}{[\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt}}. \tag{1.16}$$

An alternative pair of update rules can be derived by starting from the (generalized) Kullback-Leibler divergence,

$$J_{\mathrm{KL}} = D_{\mathrm{KL}}(\mathbf{X}; \mathbf{A}\mathbf{S}) = \sum_{pt}\left(x_{pt}\log\frac{x_{pt}}{[\mathbf{A}\mathbf{S}]_{pt}} - x_{pt} + [\mathbf{A}\mathbf{S}]_{pt}\right) \tag{1.17}$$

which reduces to the usual KL divergence between probability distributions when $\sum_{pt} x_{pt} = \sum_{pt}[\mathbf{A}\mathbf{S}]_{pt} = 1$. Repeating the derivations above for this (1.17) we obtain the gradient descent update rules

$$a_{pn} \leftarrow \left[a_{pn} + \eta_{pn}(\sum_{t} s_{nt}\, x_{pt}/[\mathbf{A}\mathbf{S}]_{pt} - \sum_{t} s_{nt})\right]_{+} \tag{1.18}$$

$$s_{nt} \leftarrow \left[s_{nt} + \eta_{nt}(\sum_{p} a_{pn}\, x_{pt}/[\mathbf{A}\mathbf{S}]_{pt} - \sum_{p} a_{pn})\right]_{+} \tag{1.19}$$

and the corresponding multiplicative update rules

$$a_{pn} \leftarrow a_{pn} \frac{\sum_t s_{nt}\, x_{pt}/[\mathbf{AS}]_{pt}}{\sum_t s_{nt}} \qquad s_{nt} \leftarrow s_{nt} \frac{\sum_p a_{pn}\, x_{pt}/[\mathbf{AS}]_{pt}}{\sum_p a_{pn}}. \tag{1.20}$$

(In practice, a small positive $\epsilon$ is added to the denominator of each of these updates in order to avoid divide-by-zero problems.)

In fact we can obtain even simpler update equations if we introduce a sum-to-1 constraint on the columns of $\mathbf{A}$

$$\lambda_n \triangleq \sum_p a_{pn} = 1. \tag{1.21}$$

We can always obtain this from any factorization $\mathbf{AS}$ by mapping $\mathbf{A}' \leftarrow \mathbf{A}\Lambda$, $\mathbf{S}' \leftarrow \mathbf{S}\Lambda^{-1}$ where $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_N)$ is the $N \times N$ diagonal matrix with the sums of the columns of $\mathbf{A}$ as its diagonal entries.

We can impose this constraint after (1.20) with a further update step

$$a_{pn} \leftarrow \frac{a_{pn}}{\sum_p a_{pn}} \tag{1.22}$$

which in turn makes the division by $\sum_t s_{nt}$ in (1.20) redundant, since it will appear inside both the numerator and denominator of (1.22). So, using this together with the constraint $\sum_p a_{pn} = 1$ in the right hand equation in (1.20), we get the simpler update equations

$$\begin{aligned} a_{pn} &\leftarrow a_{pn} \sum_t s_{nt}(x_{pt}/[\mathbf{AS}]_{pt}) \\ a_{pn} &\leftarrow \frac{a_{pn}}{\sum_p a_{pn}} \\ s_{nt} &\leftarrow s_{nt} \sum_p a_{pn}(x_{pt}/[\mathbf{AS}]_{pt}) \end{aligned} \tag{1.23}$$

which is the algorithm presented in [56].

These multiplicative update rules have proved to be attractive since they are simple, do not need the selection of an update parameter $\eta$, and their multiplicative nature and non-negative terms on the RHS ensure that the elements cannot become negative. They do also have some numerical issues, including that it is possible for the denominators to become zero, so practical algorithms often add a small offset term to prevent divide-by-zero errors [2]. There are also now a number of alternative algorithms available which are more efficient, and we shall consider some of these later.

### 1.2.3 Alternating Least Squares (ALS)

Rather than using a gradient descent direction to reduce the Euclidean cost function $J_{\mathrm{E}}$ in (1.3), we can use a Newton-like method to find alternately the $\mathbf{S}$ and $\mathbf{A}$ that directly minimizes $J_{\mathrm{E}}$.

Let us first consider the update to $\mathbf{S}$ for a fixed $\mathbf{A}$. Writing the derivative in (1.11) in matrix form we get

$$\frac{\partial J_{\mathrm{E}}}{\partial \mathbf{S}} = -(\mathbf{A}^{\mathrm{T}}\mathbf{X} - \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}) \tag{1.24}$$

which must be zero at the minimum, i.e. the equation

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})\mathbf{S} = \mathbf{A}^{\mathrm{T}}\mathbf{X} \tag{1.25}$$

must hold at the $\mathbf{S}$ that minimizes $J_{\mathrm{E}}$. We can therefore solve (1.25) for $\mathbf{S}$, either using $\mathbf{S} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}$, or through more efficient linear equation solver methods such as the Matlab function `linsolve`. Similarly for $\mathbf{A}$ we minimize $J_{\mathrm{E}}$ by solving $(\mathbf{S}\mathbf{S}^{\mathrm{T}})\mathbf{A}^{\mathrm{T}} = \mathbf{S}\mathbf{X}^{\mathrm{T}}$ for $\mathbf{A}$.

Now these least squared solutions do not themselves enforce the non-negativity of $\mathbf{S}$ and $\mathbf{A}$. The simplest way to do this is to project the resulting optimal values into the positive orthant, producing the resulting sequence of steps:

$$\mathbf{S} \leftarrow [(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{+} \tag{1.26}$$

$$\mathbf{A} \leftarrow [\mathbf{X}\mathbf{S}^{\mathrm{T}}(\mathbf{S}\mathbf{S}^{\mathrm{T}})^{-1}]_{+} \tag{1.27}$$

where $[\mathbf{M}]_{+}$ sets all negative values of the matrix to zero. While the removal of the negative values by projection onto the positive orthant means that there are no theoretical guarantees on its performance [49], this procedure has been reported to perform well in practice [91, 2].

Rather than using ad hoc truncation of least squares solutions it is also possible to use the NNLS (non-negativity constrained least squares) algorithm of Hanson and Lawson [32]. This is an active-set algorithm which in a finite number of steps will give the least squares solution subject to the non-negativity constraints. In the context of the ALS algorithm, the original algorithm can be speeded up substantially by using the current active set as a starting point. In practice, the active set does not change substantially during iterations, so the cost of using the NNLS algorithm in this way is typically less than unconstrained least squares fitting. Further speed-up is possible by exploiting the structure of the ALS updates [7].

Recently algorithms have been introduced to reduce the computational complexity of these ALS algorithms by performing block-wise or separate row/column updates instead of updating the whole matrices of the whole factor matrices $\mathbf{A}$ and $\mathbf{S}$ each step [15, 16, 21]. We will return to these large-scale NMF algorithms in Section 1.4.3.

## 1.3 Extensions and Modifications of NMF

The basic NMF method that we have introduced in the previous section has been modified in many different ways, either through the introduction of costs and/or penalties on the factors, inclusion of additional structure, or extension to multi-factor and tensor factorization.

### 1.3.1 Constraints and Penalties

It is often useful to be able to modify the standard NMF method by imposing certain constraints or penalties to favour particular types of solutions. For example, in (1.21) we have already seen that Lee and Sung [56] included sum-to-1 constraint on the columns $\mathbf{a}_n$ of $\mathbf{A}$

$$\sum_p a_{pn} = 1$$

as an option as part of their method, to remove the scaling redundancy between columns $\mathbf{a}_n$ of $\mathbf{A}$ and the rows of $\mathbf{S}$. Since all the elements $a_{pn}$ are non-negative, $a_{pn} \geqslant 0$, we notice also that $\sum_p a_{pn} = \sum_p |a_{pn}| \equiv \|\mathbf{a}_n\|_1$, so this also imposes a unit $\ell_1$ norm on each of the columns of $\mathbf{A}$.

#### 1.3.1.1 Sparseness

Hoyer [42] introduced a modification to the NMF method to include a *sparseness* penalty on the elements of $\mathbf{S}$, which he called *non-negative sparse coding*. He modified the Euclidean cost function (1.3) to include an additional penalty term:

$$D_{\mathrm{ESS}}(\mathbf{X}; \mathbf{AS}) = \tfrac{1}{2}\|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2 + \lambda \sum_{nt} s_{nt} \tag{1.28}$$

for some weight $\lambda \geqslant 0$. Hoyer also required a unit $\ell_1$ norm on the columns of $\mathbf{A}$, $\|\mathbf{a}_n\|_1 = 1$.

From a probabilistic perspective, Hoyer and Hyvärinen [44] pointed out that (1.28) is equivalent to a maximum log-likelihood approach where we assume that the noise $\mathbf{E} = \mathbf{X} - \mathbf{AS}$ has a normal distribution, while the sources have an exponential distribution, $p(s_{nt}) \propto \exp(-s_{nt})$.

Hoyer showed that this new cost function was nonincreasing under the $\mathbf{S}$ update rule

$$s_{nt} \leftarrow s_{nt} \frac{[\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt}}{[\mathbf{A}^{\mathrm{T}}\mathbf{AS}]_{nt} + \lambda} \tag{1.29}$$

which is a very simple modification of the original Lee-Sung multiplicative update rule (1.15). A similar rule was not available for the update to $\mathbf{A}$, so he instead suggested a projected gradient method

$$\begin{aligned} a_{pn} &\leftarrow [a_{pn} - \eta([\mathbf{ASS}^{\mathrm{T}}]_{pn} - [\mathbf{XS}^{\mathrm{T}}]_{pn})]_+ \\ a_{pn} &\leftarrow a_{pn}/\|\mathbf{a}_n\|_2 \end{aligned} \tag{1.30}$$

so that the complete algorithm is to repeat (1.30) and (1.29) until convergence.

Hoyer and Hyvärinen [44] demonstrated that NMF with this sparsity penalty can lead to learning of higher-level contour coding from complex cell outputs [44]. Sparsity constraints are also useful for text mining applications [74].

As an alternative way to include sparseness constraints in the NMF method, Hoyer [43] also introduced the idea of maintaining a fixed level of sparseness for

the columns of $\mathbf{A}$ and rows of $\mathbf{S}$, where this is defined as

$$\text{sparseness}(\mathbf{u}) = \frac{\sqrt{N} - \|\mathbf{u}\|_1/\|\mathbf{u}\|_2}{\sqrt{N}-1} \tag{1.31}$$

where $N$ is the number of elements of the vector $\mathbf{u}$. This measure of sparseness (1.31) is defined so that a vector $\mathbf{u}_\text{S}$ with a single non-zero element has $\text{sparseness}(\mathbf{u}_\text{S}) = 1$, and a vector $\mathbf{u}_\text{NS}$ with all $N$ components equal (disregarding sign changes) has $\text{sparseness}(\mathbf{u}_\text{NS}) = 0$.

The idea of the method is to iteratively update $\mathbf{A}$ and $\mathbf{S}$ while maintaining fixed levels of sparseness, specifically $\text{sparseness}(\mathbf{a}) = S_\mathbf{A}$ for the columns of $\mathbf{A}$, and $\text{sparseness}(\mathbf{s}) = S_\mathbf{S}$ for the rows of $\mathbf{S}$. (An additional unity $\ell_1$ norm constraint on the rows of $\mathbf{S}$, $\|s_n\|_2 = 1$, is used to avoid scaling ambiguities.)

Updating with these sparseness constraints is achieved with a sequence of projected gradient updates

$$\mathbf{a}_n \leftarrow P_\mathbf{A}\left[\mathbf{a}_n - \eta_\mathbf{A}([\mathbf{A}\mathbf{S}\mathbf{S}^\text{T}]_{\bullet n} - [\mathbf{X}\mathbf{S}^\text{T}]_{\bullet n})\right] \tag{1.32}$$

$$\mathbf{s}_n \leftarrow P_\mathbf{S}\left[\mathbf{s}_n - \eta_\mathbf{S}([\mathbf{A}^\text{T}\mathbf{A}\mathbf{S}]_{n\bullet} - [\mathbf{A}^\text{T}\mathbf{X}]_{n\bullet})\right] \tag{1.33}$$

where $[\mathbf{M}]_{\bullet n}$ is the $n$-th column vector of $\mathbf{M}$, $[\mathbf{M}]_{n\bullet}$ is the $n$-th row vector of $\mathbf{M}$, and $P_\mathbf{A}[\cdot]$ and $P_\mathbf{S}[\cdot]$ are special projection operators for columns of $\mathbf{A}$ and rows of $\mathbf{S}$ respectively which impose the required level of sparseness. The projection operator $P_\mathbf{A}[\mathbf{a}]$ projects the column vector $\mathbf{a}$ so that it is (a) non-negative, (b) has the same $\ell_1$ norm $\|\mathbf{a}\|_2$, and (c) has the required sparseness level, $\text{sparseness}(\mathbf{a}) = S_\mathbf{A}$. Similarly, the projection operator $P_\mathbf{S}[\mathbf{s}]$ projects the row vector $\mathbf{s}$ so that it is (a) non-negative, (b) has unit $\ell_1$ norm $\|\mathbf{s}\|_2 = 1$, and (c) has the required sparseness level, $\text{sparseness}(\mathbf{s}) = S_\mathbf{S}$. These projection operators are implemented by an iterative algorithm which solves this joint constraint problem: for details see [43].

Hoyer demonstrated that this method was able to give parts-based representations of image data, even when the images were not so well aligned, and where the original NMF algorithm would give a global representation [43].

#### 1.3.1.2 "Smoothness"

Another common penalty term is so-called "smoothness" constraint, obtained by penalizing the (squared) Frobenius norm of e.g. $\mathbf{A}$ [76]:

$$\|\mathbf{A}\|_\text{F}^2 = \sum_{pn} a_{pn}^2. \tag{1.34}$$

The name "smoothness" is perhaps a little misleading: it does not refer to any "blurring" or "smoothing" between e.g. neighbouring pixels in an image, it merely refers to the penalization of large values $a_{pn}$, so the resulting matrix is less "spiky" and hence more "smooth".

If we add this non-smoothness penalty (1.34) into the Euclidean cost function (1.3) we obtain a new cost function

$$J = D(\mathbf{X};\mathbf{A}\mathbf{S}) = \tfrac{1}{2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_\text{F}^2 + \tfrac{1}{2}\alpha\sum_{pn} a_{pn}^2 \tag{1.35}$$

which will act to reduce the tendency to produce large elements in $\mathbf{A}$. From a probabilistic perspective we can regard this as imposing a Gaussian prior on the elements $a_{pn}$ of $\mathbf{A}$. This modifies the derivative of $J$ w.r.t. $\mathbf{A}$, giving

$$\frac{\partial J}{\partial a_{pn}} = -([\mathbf{X}\mathbf{S}^{\mathrm{T}}]_{pn} - [\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn}) + \alpha a_{pn} \tag{1.36}$$

giving a new gradient update step of

$$a_{pn} \leftarrow a_{pn} + \eta_{pn}([\mathbf{X}\mathbf{S}^{\mathrm{T}}]_{pn} - [\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn} - \alpha a_{pn}) \tag{1.37}$$

and again using $\eta_{pn} = a_{pn}/[\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn}$ we obtain the multiplicative update

$$a_{pn} \leftarrow a_{pn} \frac{[\mathbf{X}\mathbf{S}^{\mathrm{T}}]_{pn} - \alpha a_{pn}}{[\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{pn}} \tag{1.38}$$

for which $J$ in (1.35) is non-increasing [76]. To ensure $a_{pn}$ remains non-negative in this multiplicative update, we can set negative values to a small positive $\epsilon$. (If we were simply to set negative elements to zero, the multiplicative update would never be able make that element non-zero again if required.)

Similarly, we can separately or alternatively apply such a non-smoothness penalty to $\mathbf{S}$, obtaining a similar adjustment to the update steps for $s_{nt}$.

#### 1.3.1.3 Continuity

In the context of audio source separation, Virtanen [94] proposed a *temporal continuity* objective along the rows ($t$-direction) of $\mathbf{S}$ (or alternatively, along the columns of $\mathbf{A}$, as in Virtanen's original paper [94]). This temporal continuity is achieved by minimizing a total variation (TV) cost to penalize changes in the values of $s_{nt}$ in the $t$ ("time") direction

$$C_{\mathrm{TV}t}(\mathbf{S}) = \tfrac{1}{2} \sum_{nt} |s_{n,(t-1)} - s_{n,t}| \tag{1.39}$$

where $t$ is summed from 2 to $T$. Total variation has also been applied for image reconstruction in the Compressed Sensing literature, where it is used in a 2-dimensional form [62], and an earlier approach for smoothness (in this sense) was developed and showcased in spectroscopy [5].

The derivative of $C_{\mathrm{TV}t}$ is straightforward:

$$\frac{\partial s_{nt} C_{\mathrm{TV}t}(\mathbf{S})}{\partial s_{nt}} = \begin{cases} -1 & \text{if } s_{n,t} < s_{n,(t-1)} \text{ and } s_{n,t} < s_{n,(t+1)}, \\ +1 & \text{if } s_{n,t} > s_{n,(t-1)} \text{ and } s_{n,t} > s_{n,(t+1)}, \\ 0 & \text{otherwise.} \end{cases} \tag{1.40}$$

(apart from the boundary cases $t = 1$ and $t = T$) so this can be incorporated into a steepest-descent update method for $\mathbf{S}$.

Chen and Cichocki [11] introduced a different smoothness measure based on the difference between $s_{nt}$ and a "temporally smoothed" (low-pass-filtered) version

$$\bar{s}_n(t) = \alpha \bar{s}_n(t-1) + \beta s_n(t) \tag{1.41}$$

where $\beta = 1 - \alpha$, and we write $s_n(t) \equiv s_{nt}$ to clarify the time dimension. We can write this in matrix notation for the rows $\mathbf{s}_n$ of $\mathbf{S}$ as

$$\bar{\mathbf{s}}_n = \mathbf{T}\mathbf{s}_n, \qquad \mathbf{T} = \begin{bmatrix} \beta & 0 & \cdots & 0 \\ \alpha\beta & \beta & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \alpha^{T-1}\beta & \dots & \alpha\beta & \beta \end{bmatrix} \tag{1.42}$$

where $\mathbf{T}$ is a $T \times T$ Toeplitz matrix that we can simplify to retain only e.g. the diagonal and first 4 subdiagonals by neglecting terms in $\alpha^k\beta$ for $k > 4$.

By incorporating a cost

$$R = \frac{1}{T}\|\mathbf{s}_n - \bar{\mathbf{s}}_n\|_2^2 = \|(\mathbf{I} - \mathbf{T})\mathbf{s}_n\|_2^2 \tag{1.43}$$

and a unit-variance (fixed $\ell_1$-norm) constraint on the rows $\mathbf{s}_n$, they obtain a modification to the Euclidean cost (1.3)

$$J = \tfrac{1}{2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2 + \frac{\lambda}{2T}\sum_n \|(\mathbf{I} - \mathbf{T})\mathbf{s}_n\|_2^2 \tag{1.44}$$

where $\lambda$ is a regularization coefficient, and hence a new multiplicative update step for $\mathbf{S}$ as

$$s_{nt} \leftarrow s_{nt}\frac{[\mathbf{A}^{\mathrm{T}}\mathbf{X}]_{nt}}{[\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}]_{nt} + \lambda[\mathbf{S}\mathbf{Q}]_{nt}} \tag{1.45}$$

where $\mathbf{Q} = \frac{1}{T}(\mathbf{I} - \mathbf{T})^{\mathrm{T}}(\mathbf{I} - \mathbf{T})$.

### 1.3.2 Relaxing the non-negativity constraints

We can consider relaxing or replacing the non-negativity constraints on the factors. For example, if we remove all non-negativity constraints from (1.1) and instead impose an orthogonality and unit norm constraint on the columns of $\mathbf{A}$, minimizing the mean squared error (1.3) will find the *principal subspace*, i.e. the subspace spanned by the principal components of $\mathbf{S}$.

#### 1.3.2.1 Semi-NMF

In *Semi-NMF* [23] we assume that only one factor matrix $\mathbf{A}$ or $\mathbf{S}$ is non-negative, giving for example $\mathbf{X} \approx \mathbf{A}\mathbf{S}$ where $\mathbf{S}$ is non-negative, but $\mathbf{A}$ can be of mixed sign.

To achieve uniqueness of factorization we need to impose additional constraints such as mutual independence, sparsity or semi-orthogonality. This leads, for example to non-negative ICA, non-negative sparse coding, or non-negative PCA.

#### 1.3.2.2 Non-negative ICA

Suppose that we relax the non-negativity on $\mathbf{A}$, and instead suppose that the rows $\mathbf{s}_n$ of $\mathbf{S}$ are sampled from $N$ independent non-negative sources $s_1, \cdots, s_N$. In other words, we suppose we have an independent component analysis (ICA) model, with an additional constraint of non-negativity on the sources $s_n$: we refer to this as *non-negative independent component analysis* (NNICA).

If we wish, we can always solve NNICA using classical ICA approaches, then change the sign of any negative sources [14]. However, we can also consider the NNICA model directly. Suppose we whiten the observation vectors $\mathbf{x}$ to give

$$\mathbf{z} = \mathbf{W}\mathbf{x} \tag{1.46}$$

so that $\mathbf{z}$ has identity covariance $\mathbb{E}\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\} = \mathbf{I}$, but do this whitening without subtracting the mean $\overline{\mathbf{z}}$ of $\mathbf{z}$. Then to find the independent components (factors) $\mathbf{s}$ it is sufficient to look for an orthonormal rotation matrix $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{I}$ such that the resulting output $\mathbf{y} = \mathbf{Q}\mathbf{z} = \mathbf{Q}\mathbf{W}\mathbf{x}$ is non-negative [78]. This leads to simple algorithms such as a *non-negative PCA* method [79, 81] related to the nonlinear PCA rule for standard ICA [67], as well as constrained optimization approaches based on the Lie Group geometry of the set of orthonormal matrices [80].

### 1.3.3 Structural factor constraints

In certain applications, the factors $\mathbf{A}$ and $\mathbf{S}$ may have a natural structure that should be reflected in the parametrizations of the factors. For example, Smaragdis [87, 88] and Virtanen [95] introduced a *Convolutive NMF* model, whereby our model becomes

$$x_{pt} \approx \sum_{n,u} a_{pn}(u) s_{n,t-u} \tag{1.47}$$

which we can write in a matrix convolution form as (Fig. 1.2)

$$\mathbf{X} = \sum_{u=0}^{U-1} \mathbf{A}(u) \overset{u\rightarrow}{\mathbf{S}} \tag{1.48}$$

where the $\overset{u\rightarrow}{\cdot}$ matrix notation indicates that the contents of the matrix are shifted $u$ places to the right

$$[\overset{u\rightarrow}{\mathbf{S}}]_{nt} = [\mathbf{S}]_{n,t-u}. \tag{1.49}$$

Finding non-negative $\mathbf{A}(u)$ and $\mathbf{S}$ from (1.47) is also known as *non-negative matrix factor deconvolution* (NMFD).

Schmidt and Mørup [86] extended the convolutive model to a 2-dimensional convolution

$$x_{pt} \approx \sum_{n,q,u} a_{p-q,n}(u) s_{n,t-u}(q) \tag{1.50}$$

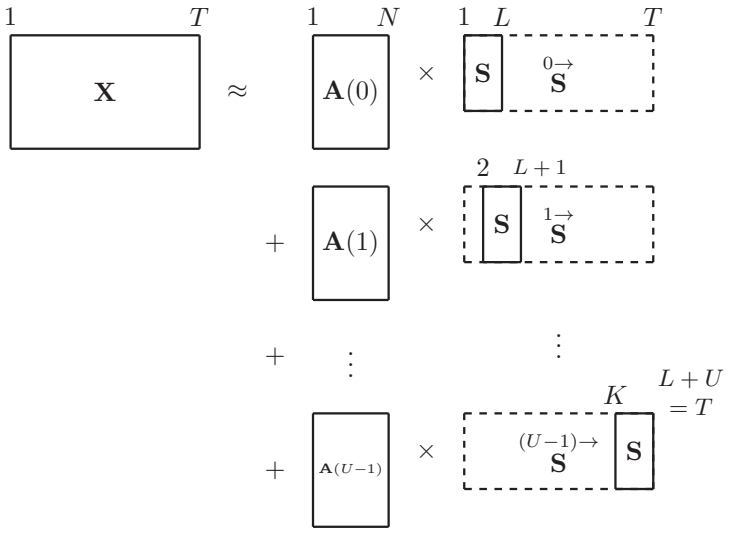


Figure 1.2: Convolutive NMF model for Non-negative Matrix Factor Deconvolution (NMFD)

which we can write in a matrix convolution form as (Fig. 1.3)

$$\mathbf{X} = \sum_{q=0}^{Q-1} \sum_{u=0}^{U-1} \overset{q\downarrow}{\mathbf{A}}(u) \overset{u\rightarrow}{\mathbf{S}}(q) \tag{1.51}$$

where the $\overset{q\downarrow}{\cdot}$ matrix notation indicates that the contents of the matrix are shifted $q$ places down

$$[\overset{q\downarrow}{\mathbf{A}}]_{pn} = [\mathbf{A}]_{p-q,n}. \tag{1.52}$$

Alternatively, if we change notation a little to write

$$a_n(p-q,u) \equiv a_{p-q,n}(u) \qquad s_n(q,t-u) \equiv s_{n,t-u}(q) \tag{1.53}$$

we could write (1.51) as

$$\mathbf{X} = \sum_{n=1}^{N} \mathbf{X}_n \tag{1.54}$$

where

$$[\mathbf{X}_n]_{pt} = \sum_{q=0}^{Q-1} \sum_{u=0}^{U-1} a_n(p-q,t-u) s_n(q,u) \equiv a_n(p,t) * s_n(p,t) \tag{1.55}$$

with $*$ as a 2-D convolution operator. So this can be viewed as a sum of $N$ elementary 2D "objects" $s_n(p,t)$ convolved with "filters" $a_n(p,t)$, or vice-versa.

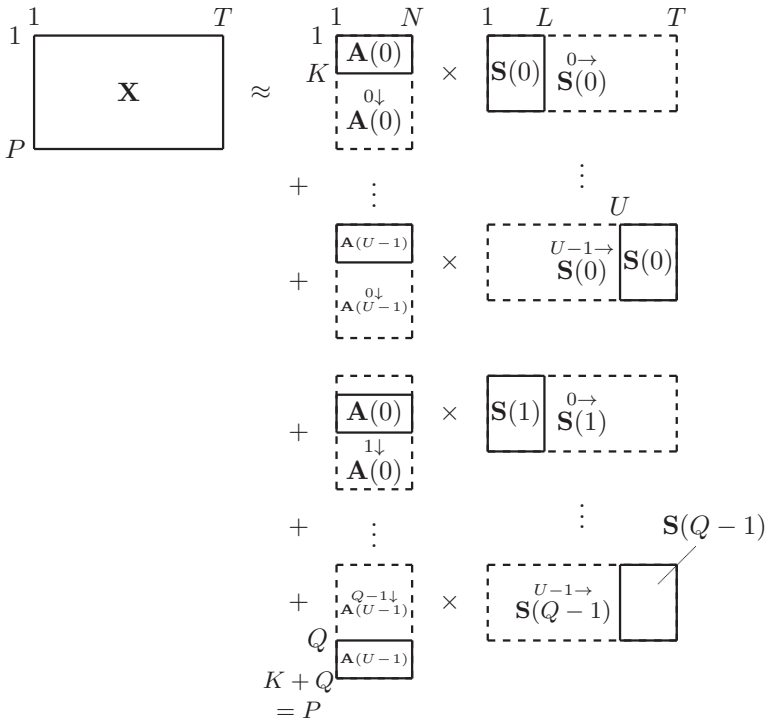


Figure 1.3: Two-dimensional convolutive NMF model (NMF2D)

This type of 1-D and 2-D convolutive model has been applied to the analysis of audio spectrograms. For example, Smaragdis [87] used the 1-D model to analyze drum sounds, on the basis that drum sounds produce a characteristic time-frequency pattern that repeats whenever the drum is "hit". On the other hand, Schmidt and Mørup [86] applied the 2-D model to analysis of spectrograms of pitched sounds on a log-frequency scale. Here a time shift ($u$) corresponds to onset time of the note, while the frequency shift ($q$) corresponds to adding a constant log-frequency offset, or multiplying all pitches in the "object" by a constant factor.

In a more general case, we can consider transform-invariant factorization [97]

$$\mathbf{X} = \sum_u \mathbf{A}^{(u)} \mathbf{T}^{(u)}(\mathbf{S}) \tag{1.56}$$

where $\{\mathbf{T}^{(u)}, u = 1, \ldots, U\}$ is a set of matrix transformation functions. This can include 1-D and 2-D convolutions (if $u$ ranges over a 2-D set) but could represent more general transforms.

As a further generalization, Schmidt and Laurberg [85] introduce the idea that the matrices $\mathbf{A}$ and $\mathbf{S}$ can be determined by underlying parameters. Their model is given by

$$\mathbf{X} \approx \mathbf{A}(\mathbf{a})\mathbf{S}(\mathbf{s}) \tag{1.57}$$

where $\mathbf{a}$ and $\mathbf{s}$ are parameters which determine the generation of the matrix-valued functions $\mathbf{A}(\mathbf{a})$ and $\mathbf{S}(\mathbf{s})$. In their paper they model $\mathbf{a}$ and $\mathbf{s}$ as Gaussian processes.

### 1.3.4 Multi-Factor and Tensor Models

The standard NMF model (1.1) is sometimes known as *Two-Way Factor Model*, being a product of two matrices. There are many different ways to extend this to models with three or more factors, or to models which include tensors as factors, i.e. where each element has more than two indices. For example, we could have order 3 tensors, which have elements $x_{ijk}$ with 3 indices, instead of the usual matrices which have elements $x_{ij}$ with 2 indices (i.e. our usual matrices are order 2 tensors) [36].

#### 1.3.4.1 Multi-layer NMF

In multi-layer NMF the basic matrix $\mathbf{A}$ is replaced by a set of cascaded (factor) matrices. Thus, the model can be described as [17, 13]

$$\mathbf{X} \approx \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_K \mathbf{S}. \tag{1.58}$$

Since the model is linear, all the matrices $\mathbf{A}_k$ $(k = 1, 2, \ldots, K)$ can be merged into a single matrix $\mathbf{A}$ if no any additional constraints are imposed upon the individual matrices $\mathbf{A}_k$. However, we impose usually sparsity constraints for each individual matrix $\mathbf{A}_k$ and then multi-layer NMF can be used to considerably improve the performance of standard NMF algorithms due to distributed

structure and alleviating the problem of local minima. To improve the performance of the NMF algorithms (especially, for ill-conditioned and badly-scaled data) and to reduce the risk of getting stuck in local minima of a cost function due to non-convex alternating minimization, we use multi-stage procedure combined with a multi-start initialization, in which we perform a sequential decomposition of non-negative matrices as follows. In the first step, we perform the basic approximate decomposition $\mathbf{X} \approx \mathbf{A}_1\mathbf{S}_1$ using any available NMF algorithm with sparsity constraint imposed to matrix $\mathbf{A}_1$. In the second stage, the results obtained from the first stage are used to build up a new input data matrix $\mathbf{X} \leftarrow \mathbf{S}_1$, that is, in the next step, we perform a similar decomposition $\mathbf{S}_1 \approx \mathbf{A}_2\mathbf{S}_2$, using the same or different update rules. We continue our decomposition taking into account only the last obtained components. The process can be repeated for an arbitrary number of times until some stopping criteria are satisfied. Physically, this means that we build up a distributed system that has many layers or cascade connections of $K$ mixing subsystems. The key point in this approach is that the update process to find parameters of matrices $\mathbf{S}_k$ and $\mathbf{A}_k$ $(k = 1, 2, \ldots, K)$ is performed sequentially, i.e. layer-by-layer, where each layer is randomly initialized with different initial conditions.

Tri-NMF also called the three factor NMF can be considered as a special case of the multi-layer NMF and can take the following general form [23]:

$$\mathbf{X} \approx \mathbf{AMS} \tag{1.59}$$

where non-negativity constraints are imposed to all or to the selected factor matrices. Note that if we do not impose any additional constraints to the factors (besides non-negativity), the three-factor NMF can be reduced to the standard (two-factor) NMF by imposing the following mapping $\mathbf{A} \leftarrow \mathbf{AM}$ or $\mathbf{S} \leftarrow \mathbf{MS}$.

However, the three-factor NMF is not equivalent to the standard NMF if we apply additional constraints or conditions. For example, in orthogonal Tri-NMF we impose additional orthogonality constraints upon the matrices $\mathbf{A}$ and $\mathbf{S}$, $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ and $\mathbf{S}\mathbf{S}^T = \mathbf{I}$, while the matrix $\mathbf{M}$ can be an arbitrary unconstrained matrix (i.e., it has both positive and negative entries). For uni-orthogonal Tri-NMF only one matrix $\mathbf{A}$ or $\mathbf{S}$ is orthogonal. Non-smooth NMF (nsNMF) was proposed by Pascual-Montano et al. [73] and is a special case of the three-factor NMF model in which the matrix $\mathbf{M}$ is fixed and known, and is used for controlling the sparsity or smoothness of the factor matrix $\mathbf{S}$ and/or $\mathbf{A}$.

#### 1.3.4.2 Non-negative Tensor Factorization

In early work on matrix factorization without non-negativity constraints, Kruskal [55] considered "three way arrays" (order 3 tensors) of the form (Fig. 1.4)

$$x_{ptq} = \sum_n a_{pn} s_{nt} d_{qn} = \sum_n a_{pn} v_{tn} d_{qn} = \tag{1.60}$$

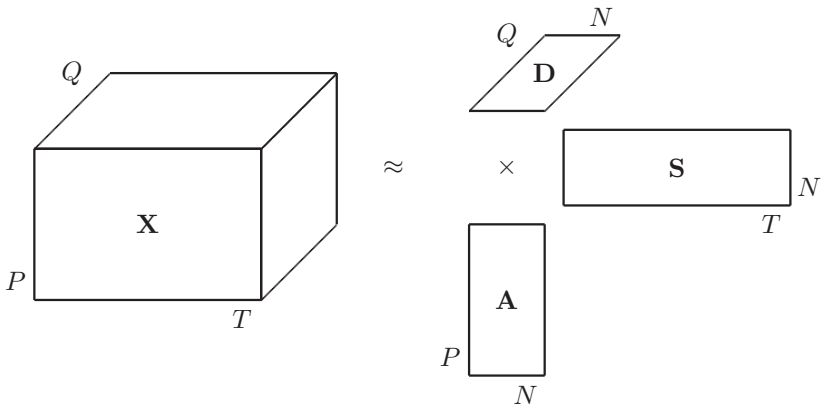


Figure 1.4: Three-Way PARAFAC Factor model

which can be written in matrix notation using the frontal slices of data tensor as

$$\mathbf{X}_q \approx \mathbf{A}\mathbf{D}_q\mathbf{S} = \mathbf{A}\mathbf{D}_q\mathbf{V}^T \tag{1.61}$$

where $[\mathbf{X}_q]_{pt} = x_{ptq}$ represent frontal slices of $mX$ and $\mathbf{D}_q$ is the $N \times N$ diagonal matrix with elements $[\mathbf{D}_q]_{nn} = d_{nq}$. This model is known as the *PARAFAC* or *CANDECOMP* (CANonical DECOMPosition) model [36]. A non-negative version of PARAFAC was first introduced by Carroll et al. [9] and Krijnen & ten Berge [54]. Later, more efficient approaches were developed by Bro (1997) [4] based on the modified NNLS mentioned earlier and Paatero [70] who generalized his earlier 2-way positive matrix factorization (PMF) method to the 3-way PARAFAC model, referring to the result as *PMF3* (3-way positive matrix factorization). The non-negatively constrained PARAFAC is also sometimes called *non-negative tensor factorization* (NTF). In some cases NTF methods may increase the number of factors and add complexity. However, in many contexts they do not lead to an increase in the number of factors, (they maintain them) and quite often they lower the complexity - because NNLS is cheaper than LS in iterative algorithms. In addition, this approach can result in a reduced number of active parameters yielding a clearer "parts-based" representation [63]. Non-negatively constrained PARAFAC has been used in numerous applications in environmental analysis, food studies, pharmaceutical analysis and in chemistry in general [6].

Later Welling and Weber [96] also discussed a factorization of an order $R$ tensor $x_{i_1,\ldots,i_R}$ into a product of $r$ order 2 tensors

$$x_{p_1,\ldots,p_R} \approx \sum_{n=1}^{N} a^{(1)}_{p_1,n} a^{(2)}_{p_2,n} \cdots a^{(R)}_{p_R,n} \tag{1.62}$$

subject to the constraint that the parameters are non-negative. They called the result *positive tensor factorization* (PTF) or *non-negative tensor factorization*

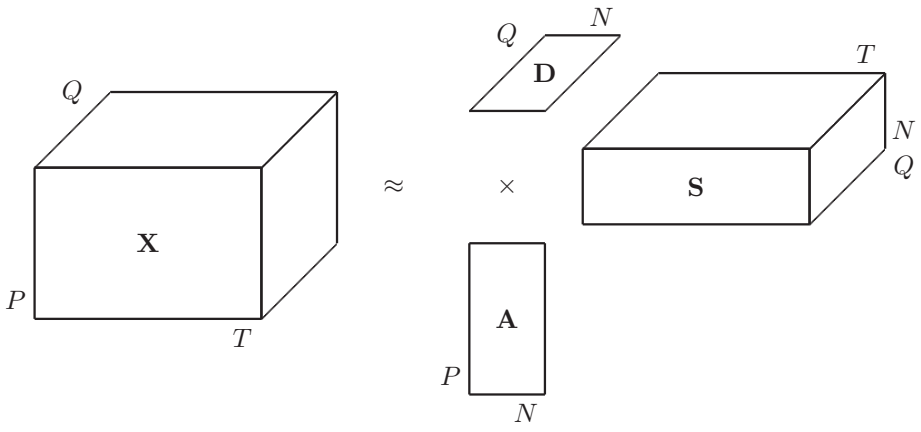


Figure 1.5: PARAFAC2/NTF2 factor model

(NTF). NTF can be presented in vector-matrix form as follows

$$\mathbf{X} \approx \sum_{n=1}^{N} \mathbf{a}_n^{(1)} \circ \mathbf{a}_n^{(2)} \circ \cdots \circ \mathbf{a}_n^{(R)} = \mathbf{I} \times_{\mathbf{1}} \mathbf{A}^{(\mathbf{1})} \times_{\mathbf{1}} \mathbf{A}^{(\mathbf{2})} \cdots \times_{\mathbf{R}} \mathbf{A}^{(\mathbf{R})} \tag{1.63}$$

where $\circ$ denotes outer product and $\times_r$ denotes $r$-mode multiplication of tensor via matrix and $\mathbf{I}$ is $R$-order identity tensor (with one on the superdiagonal). Welling and Weber develop update rules for NTF which are analogous to the Lee and Sung [57] multiplicative update rules.

Ding et al [24] also considered adding orthogonality constraints to the 3-way factor model (1.60). They showed that this additional constraint leads to a clustering model, and demonstrated its application to document clustering.

A further extension of these tensor models is to allow one or more of the factors to also be a higher-order tensor. For example, the *PARAFAC2* model [35, 48] includes an order 3 tensor in the factorization (Fig. 1.5):

$$x_{ptq} \approx \sum_n a_{pn} s_{ntq} d_{nq}. \tag{1.64}$$

In matrix notation we can write (1.64) as

$$\mathbf{X}_q \approx \mathbf{A}\mathbf{D}_q\mathbf{S}_q \tag{1.65}$$

with $\mathbf{X}_q$ and $\mathbf{D}_q$ as for the PARAFAC/PMF3 model above, and $[\mathbf{S}_q]_{nt} = s_{ntq}$. In addition to eqn. (1.64), the PARAFAC2 model includes extra constraints on the $\mathbf{S}_q$ matrices to obtain a unique solution. The first non-negative algorithm for PARAFAC2 was introduced in [5]. Cichocki et al. [20, 19] call the model in eqn. (1.64) *NTF2* to distinguish it from the PARAFAC-based non-negative tensor factorization (NTF) model (1.60).

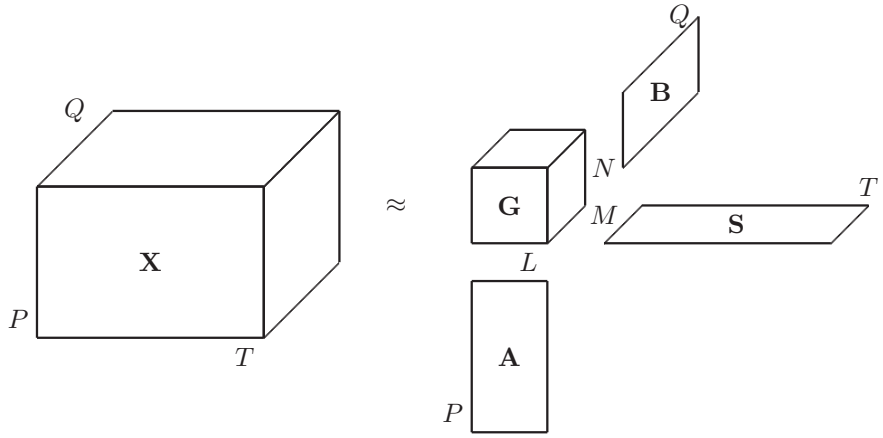


Figure 1.6: Three-way Tucker model

Fitzgerald et al [26] combined convolutive NMF models (NMFD/NMF2D) with tensor factorization, leading to shift-invariant non-negative tensor factorization. They applied this to musical audio source separation, where the tensor $\mathbf{X}$ is of order 3, representing spectrograms with frequency $p$, time $t$ and channel $q$.

Another multi-way model is the *Tucker model* (Fig. 1.6)

$$x_{pqt} \approx \sum_{lmn} g_{lmn} a_{pl} s_{tm} b_{qn} \tag{1.66}$$

which in its general form is

$$x_{p_1,p_2,\ldots,p_R} \approx \sum_{n_1,\ldots,n_R} g_{n_1,\ldots,n_R} a_{p_1,n_1} \times \cdots \times a_{p_R,n_R} \tag{1.67}$$

where the *Tucker core* $g_{n_1,\ldots,n_r}$ controls the interaction between the other factors. Tucker models have also been implemented in non-negative versions, where it is sometimes called Non-negative Tucker Decomposition (NTD). The first implementations of non-negative Tucker as well as a number of other constraints were given in [47] and in [5]. Several researchers have recently applied non-negative Tucker models to EEG analysis, classifications and feature extractions, and have demonstrated encouraging results [63, 50, 51, 75].

### 1.3.5 ALS Algorithms for Non-negative Tensor Factorization

The almost all existing NMF algorithms can be relatively easily extended for $R$-order non-negative tensor factorization by using the concept of matricizing or unfolding. Generally speaking, the unfolding of an $R$-th order tensor can be

understood as process of the construction of a matrix containing all the $r$-mode vectors of the tensor. The order of the columns is not unique and in this book it is chosen in accordance with Kolda and Bader [53]. The mode-$r$ unfolding of tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_R}$ is denoted by $\mathbf{X}_{(r)}$ and arranges the mode-r fibers into columns of a matrix.

Using the concept of unfolding an $R$-order NTF can represented as set of the following non-negative matrix factorizations

$$\mathbf{X}_{(r)} \approx \mathbf{A}^{(r)} \mathbf{Z}_{(-r)}, \qquad (r = 1, 2, \ldots, R) \tag{1.68}$$

where $\mathbf{X}_{(r)} \in \mathbb{R}_+^{I_r \times I_1 \cdots I_{r-1} I_{r+1} \cdots I_R}$ is $r$-mode unfolded matrix of the $R$-order tensor $\mathbf{X} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_R}$ and

$$\mathbf{Z}_{(-r)} = \left[ \mathbf{A}^{(R)} \odot \cdots \odot \mathbf{A}^{(r+1)} \odot \mathbf{A}^{(r-1)} \odot \cdots \odot \mathbf{A}^{(1)} \right]^T \in \mathbb{R}_+^{N \times I_1 \cdots I_{r-1} I_{r+1} \cdots I_R} \tag{1.69}$$

where $\odot$ denotes of Khatri-Rao product [53].

Using this model we can drive a standard (global) ALS update rules:

$$\mathbf{A}^{(r)} \leftarrow \left[ \mathbf{X}_{(r)} \mathbf{Z}_{(-r)}^T \left( \mathbf{Z}_{(-r)}^T \mathbf{Z}_{(-r)} \right)^{-1} \right]_+, \quad (r = 1, 2, \ldots, R). \tag{1.70}$$

By defining the residual tensor as

$$\begin{aligned} \mathbf{X}^{(n)} &= \mathbf{X} - \sum_{j \neq n} \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \cdots \circ \mathbf{a}_j^{(R)} \\ &= \mathbf{X} - \sum_{j=1}^{N} \left( \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \cdots \circ \mathbf{a}_j^{(R)} \right) + \left( \mathbf{a}_n^{(1)} \circ \mathbf{a}_n^{(2)} \circ \cdots \circ \mathbf{a}_n^{(R)} \right), \\ &= \mathbf{X} - \widehat{\mathbf{X}} + \left( \mathbf{a}_n^{(1)} \circ \mathbf{a}_n^{(2)} \circ \cdots \circ \mathbf{a}_n^{(R)} \right), \qquad (n = 1, 2, \ldots, N) \end{aligned} \tag{1.71}$$

we can derive local ALS updates rules [75]:

$$\mathbf{a}_n^{(r)} \leftarrow \left[ \mathbf{X}_{(r)}^{(n)} \left( \mathbf{a}_n^{(R)} \odot \cdots \odot \mathbf{a}_n^{(r+1)} \odot \mathbf{a}_n^{(r-1)} \odot \cdots \odot \mathbf{a}_n^{(1)} \right) \right]_+, \tag{1.72}$$

for $r = 1, 2, \ldots, R$ and $n = 1, 2, \ldots, N$ and with normalization (scaling) $\mathbf{a}_n^{(r)} \leftarrow ||\mathbf{a}_n^{(r)} / \mathbf{a}_n^{(r)}||_2$ for $r = 1, 2, \ldots, R-1$. The local ALS update can be expressed in equivalent tensor notation:

$$\mathbf{a}_n^{(r)} \leftarrow \left[ \mathbf{X}^{(n)} \times_1 \mathbf{a}_n^{(1)} \cdots \times_{r-1} \mathbf{a}_n^{(r-1)} \times_{r+1} \mathbf{a}_n^{(r+1)} \cdots \times_R \mathbf{a}_n^{(R)} \right]_+, \tag{1.73}$$

$$(r = 1, 2, \ldots, R) \quad (n = 1, 2, \ldots, N). \tag{1.74}$$

In similar way we can derive global and local ALS updates rules for Non-negative Tucker Decomposition [21, 75].

## 1.4 Further Non-negative Algorithms

In Section 1.2 we briefly developed three simple and popular algorithms for NMF. It is arguably the very simplicity of these algorithms, and in particular the Lee-Seung multiplicative algorithms (1.16) and (1.23) which have led to the popularity of the NMF approach.

Nevertheless, in recent years researchers have gained an improved understanding of the properties and characteristics of these NMF algorithms. For example, while Lee and Seung [57] claimed that their multiplicative algorithm (1.16) converges to a stationary point, this is now disputed [31], and in any case Lin [61] also points out that a stationary point is not necessarily a minimum. For more on these alternative approaches, see e.g. [12, 90, 2, 61, 19].

In addition, there has previously been interest in the effect of non-negative constraints in neural network learning (e.g. [29, 34, 89]), Another approach is the use of geometric constraints, based on looking for the edges or bounds of the scattering matrix [3, 38, 1]. Recent work has also investigated alternative algorithms specifically designed for large-scale NMF problems [15, 21]. In this section we will investigate at some of these alternative approaches.

### 1.4.1 Neural Network approaches

Given an input $\mathbf{X} = [x_{pt}]$, representing a sequence of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_T$, we can construct a simple linear "neural network" model

$$\mathbf{Y} = \mathbf{B}\mathbf{X} \tag{1.75}$$

where $\mathbf{B}$ is a $Q \times P$ linear weight matrix and $\mathbf{Y} = [y_{qt}]$ is the output from neuron $q$ for sample $t$. We can write (1.75) in its pattern-by-pattern form as

$$\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t) \qquad t = 1, 2, \ldots \tag{1.76}$$

Without any non-negativity constraints, the network (1.75) has been widely studied for the task of principal component analysis (PCA) or PCA subspace analysis (PSA): see e.g. [66, 41]. For example, Williams [98] described his *Symmetric Error Correction* (SEC) network, based on the idea of reducing the mean squared error reconstruction. A similar method was suggested independently by Oja and Karhunen [68] to find the principal subspace of a matrix. For the learning algorithm in the SEC network, the weight matrix $\mathbf{B}$ is updated on a pattern-by-pattern basis according to

$$\mathbf{B}(t+1) = \mathbf{B}(t) + \eta(t)[\mathbf{x}(t) - \widehat{\mathbf{x}}(t)]\mathbf{y}^{\mathrm{T}}(t) \tag{1.77}$$

where $\widehat{\mathbf{x}}(t) = \mathbf{B}^{\mathrm{T}}\mathbf{y}(t)$ is considered to be an approximate reconstruction of the input $\mathbf{x}$ using the weights $\mathbf{B}$. Alternatively, the following batch update rule can be used:

$$\mathbf{B}(t+1) = \mathbf{B}(t) + \eta(t)[\mathbf{X} - \widehat{\mathbf{X}}]\mathbf{Y}^{\mathrm{T}} \tag{1.78}$$

where $\widehat{\mathbf{X}} = \mathbf{B}^{\mathrm{T}}\mathbf{Y}$ is the approximate reconstruction. With $m \leqslant n$ outputs, and without any non-negativity constraints, update rule (1.77) finds the minimum of the mean squared reconstruction error

$$J_{\mathrm{E}} = D_{\mathrm{E}}(\mathbf{X}; \widehat{\mathbf{X}}) = \tfrac{1}{2}\|\mathbf{X} - \widehat{\mathbf{X}}\|_{\mathrm{F}}^2 \tag{1.79}$$

and hence finds the principal subspace of the input, i.e. the space spanned by the principal eigenvectors of $\mathbf{X}\mathbf{X}^T$ [99].

Harpur and Prager [34] suggested modifying this network to include a non-negativity constraint on the output vector $\mathbf{y}$, so that its activity is determined by

$$y_q(t) = [\mathbf{b}_q^{\mathrm{T}}\mathbf{x}(t)]_+ \tag{1.80}$$

where $\mathbf{b}_q = (b_{q1}, \ldots, b_{qP})^{\mathrm{T}}$, and use this non-negative $\mathbf{Y}$ to form the reconstruction $\widehat{\mathbf{X}}$ in (1.78). They showed that this *recurrent error correction* (REC) network, with the non-negativity constraint on the output, could successfully separate out individual horizontal and vertical bars from images in the 'bars' problem introduced by Földiák [28], while the network without the non-negativity constraint would not.

Harpur noted that this *recurrent error correction* (REC) network might be under-constrained when fed with a mixture of non-negative sources, illustrating this for $n = m = 2$ [33, p68]. He suggests that this uncertainty could be overcome by starting learning with weight vectors inside the 'wedge' formed by the data, but points out that this would be susceptible to any noise on the input. Plumbley [77] attempted to overcome this uncertainty by incorporating anti-Hebbian lateral inhibitory connections between the output units, a modification of Földiák's Hebbian/anti-Hebbian network [27].

Charles and Fyfe [10], following on from earlier work of Fyfe [29], investigated a range of non-negative constraints on the weights and/or outputs of a PCA network. Their goal was to find a *sparse coding* of data, with most values are zero or near zero [69]. With non-negative constraints on the outputs, they noted that update equation (1.78) is a special case of the nonlinear PCA algorithm [46], and so their learning algorithm minimizes the residual error at the input neurons. They also tested their network on the 'bars' problem, using various nonlinearities (threshold linear, sigmoid and exponential) as well as pre-processing to equalize the input variances $E(x_i^2)$. They found that performance was most reliable with non-negative constraint on weights $b_{qp}$ as well as the outputs $y_q(t)$.

## 1.4.2 Geometrical Methods

### 1.4.2.1 Edge Vectors

Several non-negative methods have been inspired by a geometric approach to the problem. Much of the earliest work in NMF in the seventies and eighties was based on such approaches (see e.g. [3] and references therein). Consider the 2-dimensional case $P = N = 2$. If the sources $s_{nt}$ are non-negative, we can often see this clearly on a scatter plot of $x_{1t}$ against $x_{2t}$ (Fig 1.7). This scatter

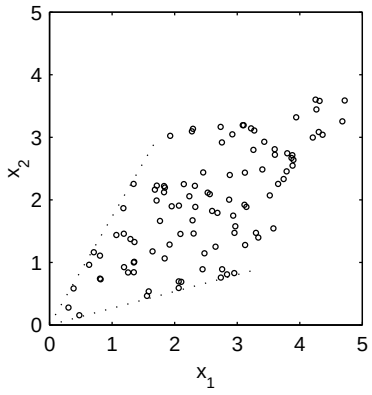


Figure 1.7: Scatter plot for observations of weighted non-negative sources.

plot suggests that we could identify the underlying generating factors by looking for the *edges* in the distribution [38]. For example, suppose sample $t'$ of source $p' = 2$ were zero, i.e. $s_{2,t'} = 0$. Then we immediately have

$$x_{pt'} = a_{p,1} s_{1,t'} \tag{1.81}$$

meaning that we can solve for the *basis vector* $\mathbf{a}_1 = (a_{1,1}, a_{2,1})$ apart from a scaling ambiguity [37]. This condition occurs along the edge of the scatter plot, so if we find observed vectors $\mathbf{x}_t$ on both of these edges, so-called *edge vectors*, then we can estimate the original mixing matrix $\mathbf{A}$, and hence the source matrix $\mathbf{S}$.

This approach has been generalized to more than two dimensions using the concept of an *extremal polyhedral cone*, finding a few spanning vectors that fix the edges of the data [93] (see also the review by Henry [39]) and Henry [40] introduces a related *extreme vector algorithm* (EVA) that searches for $N$-dimensional edges in the data. The geometrical can also give insights into issues of uniqueness of NMF, which has been investigated by Donoho and Stodden [25] and Klingenberg et al [52].

#### 1.4.2.2 Bounded pdf approaches

Some geometrical algorithms have also been introduced for cases where the sources have an additional constraint of being bounded from above as well as bounded from below (as in the non-negative case).

Puntonet et al [83, 84] developed separation algorithms for sources with such a bounded pdf. Their algorithm operates as each data vector arrives, updating the weights to minimize an *angular proximity*, and they also consider adjustments to their algorithm to cope with noise, which might give rise to observed data vectors which lie outside the basis vectors [84, 82] In contrast to normal ICA-based measures, which require independent sources, they found

that their approach can be used to separate non-independent bounded sources. For good separation, Puntonet et al. [84] note that it is important to obtain *critical vectors* that map to the edges of the hyperparallelpiped, analogous to the *edge vectors* in the geometrical NMF/PMF methods.

Yamaguchi, Hirokawa and Itoh [101, 45] independently propose a similar approach for bounded data. They proposed an algebraic method for ICA of images pairs, based on the extremum points on a scatter diagram. This uses the upper- and lower-boundedness of source image values, and does not use independence. They also note that the algorithm relies on critical vectors at the apexes of the scatter diagram, so signals with low pdf at their extrema will be more difficult to separate.

Finally, Basak and Amari [1] considered the special case of bounded source signals with uniform pdf. After pre-whitening, the data will fills a hypercube. The hypercube is rigidly rotated using a matrix exponential $\mathbf{B} = \exp(\eta\mathbf{Z})$ to generate special (determinant 1) orthogonal matrices $\mathbf{B} \in SO(N)$ with a local learning rule used to bring data points into the unit hypercube by minimizing a 1-norm distance outside of this unit hypercube. This leads to a type of nonlinear PCA-type learning rule [46] with nonlinearity $g(y) = \text{sgn}(y)$ if $y$ is outside the hypercube.

### 1.4.3 Algorithms for large-scale NMF problems

For large scale NMF problems, where the data matrix $\mathbf{X}$ is very large, the computation complexity and memory required for standard NMF algorithms can become very large. Recently new algorithms have been introduced which reduce these through e.g. block-wise or row/column-wise updates.

#### 1.4.3.1 ALS for large-scale NMF

If the data matrix $\mathbf{X}$ is of large dimension ($P \gg 1$ and $T \gg 1$), but where the number of non-negative components $N$ is relatively small, ($N \ll P$ and $N \ll T$), we can reduce the computational complexity and memory allocation by taking a block-wise approach, where we select only very few rows and columns of the data matrix $\mathbf{X}$. In this approach, instead of performing a single large-scale factorization $\mathbf{X} \approx \mathbf{AS}$ we sequentially perform two (much smaller dimensional) non-negative matrix factorizations:

$$\mathbf{X}_R \approx \mathbf{A}_R\, \mathbf{S} \tag{1.82}$$

$$\mathbf{X}_C \approx \mathbf{A}\, \mathbf{S}_C \tag{1.83}$$

where $\mathbf{X}_R \in \mathbb{R}^{R\times T}$ and $\mathbf{X}_C \in \mathbb{R}^{P\times C}$ are data matrices constructed from the preselected rows and columns of the data matrix $\mathbf{X} \in \mathbb{R}^{P\times T}$, respectively. Analogously, we can construct the reduced matrices: $\mathbf{A}_R \in \mathbb{R}^{R\times N}$ and $\mathbf{S}_C \in \mathbb{R}^{N\times C}$ by using the same indices for the columns and rows as those used for the construction of the data sub-matrices $\mathbf{X}_R$ and $\mathbf{X}_C$, respectively.

There are several strategies to choose the columns and rows of the input data matrix. The simplest scenario is to randomly select rows and columns

from a uniform distribution. Another heuristic option is to choose those rows and columns that provide the largest $l_p$-norm, especially the Chebyshev-norm, $p = \infty$.

This approach can be applied to any NMF algorithm. In the special case, for squared Euclidean distance (Frobenius norm), instead of alternately minimizing the cost function $J_E = ||\mathbf{X} - \mathbf{A}\ \mathbf{S}||_F^2$, we can minimize sequentially set of two cost functions:

$$J_{ES} = ||\mathbf{X}_R - \mathbf{A}_R\ \mathbf{S}||_F^2 \quad \text{for fixed} \quad \mathbf{A}_R \tag{1.84}$$

$$J_{EA} = ||\mathbf{X}_C - \mathbf{A}\ \mathbf{S}_C||_F^2 \quad \text{for fixed} \quad \mathbf{S}_C \tag{1.85}$$

This leads to the following ALS updates rules for large-scale NMF [15, 21]

$$\mathbf{S} \leftarrow [(\mathbf{A}_R^{\mathrm{T}}\mathbf{A}_R)^{-1}\mathbf{A}_R^{\mathrm{T}}\mathbf{X}_R]_+ \tag{1.86}$$

$$\mathbf{A} \leftarrow [\mathbf{X}_C\mathbf{S}_C^{\mathrm{T}}(\mathbf{S}_C\mathbf{S}_C^{\mathrm{T}})^{-1}]_+. \tag{1.87}$$

#### 1.4.3.2 Hierarchical ALS

An alternative fast local ALS algorithm, called *Hierarchical ALS* (HALS), sequentially estimates the individual columns $\mathbf{a}_n$ of $\mathbf{A}$ and rows $\mathbf{s}_n$ of $\mathbf{S}$ instead of directly computing the whole factor matrices $\mathbf{A}$ and $\mathbf{S}$ in each step[2]. The HALS algorithm is often used for multi-layer models (see Section 1.3.4.1) in order to improve performance.

The basic idea is to define the residual matrix [5, 18, 30]:

$$\mathbf{X}^{(n)} = \mathbf{X} - \sum_{j \neq n} \mathbf{a}_j\ \mathbf{s}_j^T = \mathbf{X} - \mathbf{A}\ \mathbf{S} + \mathbf{a}_n\mathbf{s}_n^T, \quad (n = 1, 2, \ldots, N) \tag{1.88}$$

and to minimize the set of squared Euclidean cost functions:

$$J_{EA}^{(n)} = ||\mathbf{X}^{(n)} - \mathbf{a}_n\ \mathbf{s}_n^T||_F^2 \quad \text{for fixed} \quad \mathbf{s}_n \tag{1.89}$$

$$J_{EB}^{(n)} = ||\mathbf{X}^{(n)} - \mathbf{a}_n\ \mathbf{s}_n^T||_F^2 \quad \text{for fixed} \quad \mathbf{a}_n \tag{1.90}$$

subject to constraints $\mathbf{a}_n \geq 0$ and $\mathbf{s}_n \geq 0$ for $n = 1, 2, \ldots, N$. In order to estimate the stationary points, we simply compute the gradients of the above local cost functions with respect to the unknown vectors $\mathbf{a}_n$ and $\mathbf{s}_n$ (assuming that other vectors are fixed) and equalize them to zero:

$$\frac{\partial J_{EA}^{(n)}}{\partial \mathbf{a}_n} = \mathbf{a}_n\ \mathbf{s}_n^T\ \mathbf{s}_n - \mathbf{X}^{(n)}\ \mathbf{s}_n = 0 \tag{1.91}$$

$$\frac{\partial J_{EB}^{(n)}}{\partial \mathbf{s}_n} = \mathbf{a}_n\ \mathbf{a}_n^T\ \mathbf{s}_n - \mathbf{X}^{(n)\,T}\ \mathbf{a}_n = 0. \tag{1.92}$$

[2]The HALS algorithm is "Hierarchical" since we sequentially minimize a set of simple cost functions which are hierarchically linked to each order via residual matrices which approximate rank-one bilinear decomposition.

Hence, we obtain the local ALS algorithm:

$$\mathbf{a}_n \leftarrow \frac{1}{\mathbf{s}_n^T \mathbf{s}_n} \left[\mathbf{X}^{(n)} \mathbf{s}_n\right]_+ \tag{1.93}$$

$$\mathbf{s}_n \leftarrow \frac{1}{\mathbf{a}_n^T \mathbf{a}_n} \left[\mathbf{X}^{(n)\,T} \mathbf{a}_n\right]_+ . \tag{1.94}$$

In practice, we usually normalize the column vectors $\mathbf{a}_n$ and $\mathbf{s}_n$ to unit length vectors (in $l_2$-norm sense) at each iteration step. In such case the above updates local ALS updates rules can be further simplified by ignoring the denominators and imposing a vector normalization after each iterative step, to give a simplified scalar form of the HALS updated rules:

$$a_{pn} \leftarrow \left[\sum_t v_{tn} x_{pt}^{(n)}\right]_+ , \qquad a_{pn} \leftarrow a_{pn}/||\mathbf{a}_n||_2^2 \tag{1.95}$$

$$v_{tn} \leftarrow \left[\sum_p a_{pn} x_{pt}^{(n)}\right]_+ \tag{1.96}$$

where $x_{pt}^{(n)} = x_{pt} - \sum_{j \neq n} a_{pj}\, b_{tj}$. The above updates rules are extremely simple and quite efficient and can be further optimized for large scale NMF [15, 16, 21].

## 1.5 Applications

NMF has been applied to a very wide range of tasks such as air quality analysis, text document analysis, and image processing. While it would be impossible to fully survey every such application here, we will select a few here to illustrate the possibilities, and as pointers for further information.

### 1.5.1 Air Quality and Chemometrics

As discussed by Henry [39] in the field of air quality, $s_{jk}$ represents the amount of a particulate from source $j$ in sample $k$, and so must be non-negative. Similarly, $a_{ij}$ is the mass fraction of chemical constituent (or *species*) $i$ in source $j$, which again must be positive. This leads to an interpretation of (1.1) as a chemical mass balance equation, where $x_{ik}$ are the total amount of species $i$ observed in sample $k$. This is known as a *multivariate receptor model* [37] where $a_{ij}$ are called the *source compositions*, and $s_{jk}$ are called the *source contributions*.

In geochemistry, this model could also represent the composition of geological samples modelled as a mixture of $N$ pure components. In chemeometrics, the spectrum of a mixture is represented as a linear combination of the spectra or pure components. Again, the nature of the physical process leading to the observations require that all of these quantities are non-negative [39].

### 1.5.2 Text analysis

Text mining usually involves the classification of text documents into groups or clusters according to their similarity in semantic characteristics. For example, a web search engine often returns thousands of pages in response to a broad query, making it difficult for users to browse or to identify relevant information. Clustering methods can be used to automatically group the retrieved documents into a list of meaningful topics. The NMF approach is attractive for document clustering, and usually exhibits better discrimination for clustering of partially overlapping data than other methods such as Latent Semantic Indexing (LSI).

Preprocessing strategies for document clustering with NMF are very similar to those for LSI. First, the documents of interest are subjected to stop-word removal and word streaming operations. Then, for each document a weighted term-frequency vector is constructed that assigns to each entry the occurrence frequency of the corresponding term. Assuming $P$ dictionary terms and $T$ documents, the sparse term-document matrix $\mathbf{X} \in \mathbb{R}^{P \times T}$ is constructed from weighted term-frequency vectors, that is

$$x_{pt} = f_{pt} \log \left( \frac{T}{T_p} \right) \tag{1.97}$$

where $f_{pt}$ is the frequency of occurring the $p$-th term in the $t$-th document, and $T_p$ is the number of documents containing the $p$-th term. The entries of $\mathbf{X}$ are always non-negative and equal to zero when either the $p$-th term does not appear in the $t$-th document or appears in all the documents.

The aim is to factorize the matrix $\mathbf{X}$ into the non-negative basis matrix $\mathbf{A}$ and the non-negative topic-document matrix $\mathbf{X} \in \mathbb{R}_+^{N \times T}$ where $N$ denotes the number of topics. The position of the maximum value in each column-vector in $\mathbf{S}$ informs us to which topic a given document can be classified. The columns of $\mathbf{A}$ refer to the cluster centres, and the columns in $\mathbf{S}$ are associated with the cluster indicators. A more general scheme for simultaneous clustering both with respect to terms and documents can be modeled by Tri-NMF.

The application of NMF to document clustering has also been discussed by many researchers. For example B. Xu et al. [100] propose to use orthogonality constraints in their Constrained NMF algorithm, where the orthogonality of lateral components is enforced by the additional penalty terms added to the KL I-divergence and controlled by the penalty parameters.

In language modelling, Novak and Mammone [65] used non-negative matrix factorization as an alternative to Latent Semantic Analysis for language modelling in an application directed at automatic speech transcription of biology lectures. Tsuge et al [92] also applied (NMF) to dimensionality reduction of document vectors applied to document retrieval of MEDLINE data. They minimize either Euclidean distance or Kullback-Leibler divergence of the reconstruction, showing that NMF gave better performance than the conventional vector space model.

### 1.5.3 Image processing

Image analysis often includes non-negativity, corresponding to e.g. the non-negative amount of light falling on a surface and a non-negative reflectance of an illuminated surface. In their now-classic paper, Lee and Seung [56] showed that NMF could discover a "parts-based" representations of face images. The found parts like the eyes and mouth would be represented by different NMF basis images, unlike other analysis approaches such as PCA which would tend to produce global basis images which covered the whole face image. However, this parts-based representation may be strongly dependent on the background and content colour, and may not always be obtained [43].

The non-negativity constraint also arises in, for example, hyperspectral image analysis for remote sensing [72, 60, 64] where $\mathbf{A}$ is considered to model the amount of substances at each pixel, with $\mathbf{S}$ the spectral signatures of those substances.

Buchsbaum and Bloch [8] also applied NMF to Munsell colour spectra, which are widely used in colour naming studies. The basis functions that emerged corresponded to spectra representing familiar colour names, such as "Red", "Blue", and so on.

NMF has also been applied to sequences of images. Lee et al [58] applied NMF to dynamic myocardial PET (positron emission tomography) image sequences. They were able to extract basis images that corresponded to major cardiac components, together with time-activity curves with shapes that were similar to those observed in other studies.

### 1.5.4 Audio analysis

While audio signals take both positive and negative samples when represented as a raw time series of samples, non-negativity constraints arise when represented as a power or magnitude spectrogram. Due to the time-shift-invariant nature of audio signals, convolutive NMF models (Section 1.3.3) are suitable for these. The have been used to discover e.g. drum sounds in an audio stream [87], and for separation of speech [88] and music [95]. To allow for pitch-invariant basis functions, Schmidt and Mørup [86] extended the convolutive model to a 2-dimensional convolution using a spectrogram with a log-frequency scale, so that changes in fundamental frequency become shifts on the log-frequency axis.

### 1.5.5 Gene expression analysis

NMF has also been increasingly used recently in analysing DNA microarrays. Here the rows of $\mathbf{X}$ represent the expression levels of genes, while the columns represent the different samples. NMF is then used to search for "metagenes", helping for example to identify functionally related genes. For a recent review of this area, see e.g. [22]

## 1.6 Conclusions

In this chapter we have briefly presented basic models and associated learning algorithms for non-negative matrix and tensor factorizations. Currently the most efficient and promising algorithms seem to be those based on the alternating least squares (ALS) approach: these implicitly exploit the gradient and Hessian of the cost functions and provide high convergence speed if they are suitably designed and implemented. Multiplicative algorithms are also useful where the data matrix and factor matrices are very sparse. We have also explored a range of generalizations and extensions of these models, and alternative approaches and algorithms that also enforce non-negativity constraints, including special algorithms designed to handle large scale problems. Finally we touched on a few applications of non-negative methods, including chemometrics, text processing, image processing and audio analysis.

With non-negativity constraints found naturally in many real-world signals, and with the improved theoretical understanding and practical algorithms produced by recent researchers, we consider that the non-negative methods we have discussed in this chapter are a very promising direction for future research and applications.

## Acknowledgements

MP is supported by EPSRC Leadership Fellowship EP/G007144/1 and EU FET-Open project FP7-ICT-225913 “Sparse Models, Algorithms, and Learning for Large-scale data (SMALL)”.

## Bibliography

[1] J. Basak and S.-I. Amari, *Blind separation of a mixture of uniformly distributed source signals: A novel approach*, Neural Computation, 11 (1999), pp. 1011–1034.

[2] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis, 52 (2007), pp. 155–173.

[3] O. S. Borgen and B. R. Kowalski, *An extension of the multivariate component-resolution method to three components*, Anal. Chim. Acta, 174 (1985), pp. 1–26.

[4] R. Bro, *PARAFAC: Tutorial and applications*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 149–171.

[5] ——, *Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications*, PhD thesis, University of Amsterdam, The Netherlands, 1998.

[6] ——, *Review on multiway analysis in chemistry-2000-2005*, Critical Reviews In Analytical Chemistry, 36 (2006), pp. 279–293.

[7] R. Bro and S. de Jong, *A fast non-negativity-constrained least squares algorithm*, J. Chemom., 11 (1997), pp. 393–401.

[8] G. Buchsbaum and O. Bloch, *Color categories revealed by non-negative matrix factorization of Munsell color spectra*, Vision Research, 42 (2002), pp. 559–563.

[9] J. D. Carroll, G. de Soete, and S. Pruzansky, *Fitting of the latent class model via iteratively reweighted least squares CANDECOMP with nonnegativity constraints*, in Multiway data analysis, R. Coppi and S. Bolasco, eds., Elsevier, Amsterdam, 1989, pp. 463–472.

[10] D. Charles and C. Fyfe, *Modelling multiple-cause structure using rectification constraints*, Network: Computation in Neural Systems, 9 (1998), pp. 167–182.

[11] Z. Chen and A. Cichocki, *Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints*. Paper preprint, 2005.

[12] M. Chu, F. Diele, R. Plemmons, and S. Ragni, *Optimality, computation, and interpretation of nonnegative matrix factorizations*, 18 Oct. 2004. Preprint.

[13] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, *Extended SMART algorithms for non-negative matrix factorization*, Springer, LNAI-4029, 4029 (2006), pp. 548–562.

[14] A. Cichocki and P. Georgiev, *Blind source separation algorithms with matrix constraints*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E86–A (2003), pp. 522–531.

[15] A. Cichocki and A. Phan, *Fast local algorithms for large scale nonnegative matrix and tensor factorizations*, IEICE (invited paper), (2009).

[16] A. Cichocki, A. Phan, and C. Caiafa, *Flexible HALS algorithms for sparse non-negative matrix/tensor factorization*, in Proc. of 18-th IEEE workshops on Machine Learning for Signal Processing, Cancun, Mexico, 16–19, October 2008.

[17] A. Cichocki and R. Zdunek, *Multilayer nonnegative matrix factorization*, Electronics Letters, 42 (2006), pp. 947–948.

[18] A. Cichocki, R. Zdunek, and S. Amari, *Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization*, in Lecture Notes in Computer Science, LNCS-4666, 2007, pp. 169–176.

[19] A. Cichocki, R. Zdunek, and S.-i. Amari, *Nonnegative matrix and tensor factorization*, IEEE Signal Processing Magazine, 25 (2008), pp. 142–145.

[20] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. ichi Amari, *Novel multi-layer non-negative tensor factorization with sparsity constraints*, in Adaptive and Natural Computing Algorithms, 2007, pp. 271 – 280.

[21] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations*, Wiley, Chichester, 2009.

[22] K. Devarajan, *Nonnegative matrix factorization: An analytical and interpretive tool in computational biology*, PLoS Comput Biol, 4 (2008), p. e1000029.

[23] C. Ding, T. Li, and M. I. Jordan, *Convex and semi-nonnegative matrix factorizations*. Accepted to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.

[24] C. Ding, T. Li, W. Peng, and H. Park, *Orthogonal nonnegative matrix tri-factorizations for clustering*, in KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2006, ACM, pp. 126–135.

[25] D. Donoho and V. Stodden, *When does non-negative matrix factorization give a correct decomposition into parts?*, in Advances in Neural Information Processing Systems 16, S. Thrun, L. Saul, and B. Schölkopf, eds., MIT Press, Cambridge, MA, 2004.

[26] D. FitzGerald, M. Cranitch, and E. Coyle, *Extended nonnegative tensor factorisation models for musical sound source separation*, Computational Intelligence and Neuroscience, 2008 (2008). Article ID 872425.

[27] P. Földiák, *Adaptive network for optimal linear feature extraction*, in Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, IJCNN-89, vol. 1, Washington D.C., 18–22 June 1989, pp. 401–405.

[28] ——, *Forming sparse representations by local anti-Hebbian learning*, Biological Cybernetics, 64 (1990), pp. 165–170.

[29] C. Fyfe, *Positive weights in interneurons*, in Neural Computing: Research and Applications II. Proceedings of the Third Irish Neural Networks Conference, Belfast, Northern Ireland, 1-2 Sept 1993, G. Orchard, ed., Irish Neural Networks Association, Belfast, NI, 1994, pp. 47–58.

[30] N. Gillis and F. Glineur, *Nonnegative matrix factorization and underapproximation*, in 9th International Symposium on Iterative Methods in Scientific Computing, Lille, France, 2008.

[31] E. F. Gonzalez and Y. Zhang, *Accelerating the Lee-Seung algorithm for nonnegative matrix factorization*, Tech. Report TR05-02, Dept. of Computational and Applied Mathematics, Rice University, 3 Mar. 2005.

[32] R. J. Hanson and C. L. Lawson, *Solving least squares problems*, Prentice-Hall, Inc,, Englewood Cliffs, 1974.

[33] G. F. Harpur, *Low Entropy Coding with Unsupervised Neural Networks*, PhD thesis, Department of Engineering, University of Cambridge, February 1997.

[34] G. F. Harpur and R. W. Prager, *Development of low entropy coding in a recurrent network*, Network: Computation in Neural Systems, 7 (1996), pp. 277–284.

[35] R. A. Harshman, *PARAFAC2: Mathematical and technical notes*, UCLA Working Papers in Phonetics, 22 (1972), pp. 30–47.

[36] R. A. Harshman and M. E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H. G. Law, J. C. W. Snyder, J. Hattie, and R. P. McDonald, eds., Praeger, New York, 1984, pp. 122–215.

[37] R. C. Henry, *History and fundamentals of multivariate air quality receptor models*, Chemometrics and Intelligent Laboratory Systems, 37 (1997), pp. 37–42.

[38] R. C. Henry, *Receptor model applied to patterns in space (RMAPS) part I: Model description*, J. Air Waste Manage. Assoc., 47 (1997), pp. 216–219.

[39] R. C. Henry, *Multivariate receptor models–current practice and future trends*, Chemometrics and Intelligent Laboratory Systems, 60 (2002), pp. 43–48.

[40] ——, *Multivariate receptor modeling by n-dimensional edge detection*, Chemometrics and Intelligent Laboratory Systems, 65 (2003), pp. 179–189.

[41] K. Hornik and C.-M. Kuan, *Convergence analysis of local feature extraction algorithms*, Neural Networks, 5 (1992), pp. 229–240.

[42] P. O. Hoyer, *Non-negative sparse coding*, in Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing), Martigny, Switzerland, 2002, pp. 557–565.

[43] P. O. Hoyer, *Non-negative matrix factorization with sparseness constraints*, Journal of Machine Learning Research, 5 (2004), pp. 1457–1469.

[44] P. O. Hoyer and A. Hyvärinen, *A multi-layer sparse coding network learns contour coding from natural images*, Vision Research, 42 (2002), pp. 1593–1605.

[45] K. Itoh, *Blind signal separation by algebraic independent component analysis*, in Proceedings of the 13th Annual Meeting of the IEEE Lasers and Electro-Optics Society (LEOS 2000), vol. 2, Rio Grande, Puerto Rico, 13-16 November 2000, pp. 746–747 vol.2.

[46] J. Karhunen and J. Joutsensalo, *Representation and separation of signals using nonlinear PCA type learning*, Neural Networks, 7 (1994), pp. 113–127.

[47] H. A. L. Kiers and A. K. Smilde, *Constrained three-mode factor analysis as a tool for parameter estimation with second-order instrumental data*, Journal of Chemometrics, 12 (1998), pp. 125–147.

[48] H. A. L. Kiers, J. M. F. ten Berge, and R. Bro, *PARAFAC2 - Part I. A direct fitting algorithm for the PARAFAC2 model*, J. Chemom, 13 (1999), pp. 275–294.

[49] D. Kim, S. Sra, and I. S. Dhillon, *Fast Newton-type methods for the least squares nonnegative matrix approximation problem*, in Proceedings of the SIAM Conference on Data Mining, 2007.

[50] Y.-D. Kim and S. Choi, *Nonnegative Tucker decomposition*, in Proc. of Conf. Computer Vision and Pattern Recognition (CVPR-2007), Minneapolis, Minnesota, June 2007.

[51] Y.-D. Kim, A. Cichocki, and S. Choi, *Nonnegative Tucker decomposition with alpha divergence*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP2008, Nevada, U.S.A., 2008.

[52] B. Klingenberg, J. Curry, and A. Dougherty, *Non-negative matrix factorization: Ill-posedness and a geometric algorithm*, Pattern Recognition, 42 (2009), pp. 918–928.

[53] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*. To appear in *SIAM Review*, Sept. 2009.

[54] W. Krijnen and J. ten Berge, *Contrastvrije oplossingen van het CANDECOMP/PARAFAC-model*, Kwantitatieve Methoden, 12 (1991), pp. 87–96.

[55] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra and its Applications, 18 (1977), pp. 95–138.

[56] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[57] ——, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing Systems 13, T. K. Leen, T. G. Dietterich, and V. Tresp, eds., MIT Press, 2001, pp. 556–562.

[58] J. S. Lee, D. D. Lee, S. Choi, and D. S. Lee, *Application of non-negative matrix factorization to dynamic positron emission tomography*, in Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA2001), San Diego, California, T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, eds., December 9-13 2001, pp. 629–632.

[59] D. J. Leggett, *Numerical analysis of multicomponent spectra*, Analytical Chemistry, 49 (1977), pp. 276–281.

[60] M. Lennon, G. Mercier, M. C. Mouchot, and L. Hubert-Moy, *Independent component analysis as a tool for the dimensionality reduction and the representation of hyperspectral images*, in Proceedings of the IEEE 2001 International Geoscience and Remote Sensing Symposium (IGARSS '01), vol. 6, IEEE, 9-13 July 2001, pp. 2893 – 2895.

[61] C.-J. Lin, *Projected gradient methods for nonnegative matrix factorization*, Neural Computation, 19 (2007), pp. 2756–2779.

[62] M. Lustig, D. Donoho, and J. M. Pauly, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine, 58 (2007), p. 11821195.

[63] M. Mørup, L. K. Hansen, and S. M. Arnfred, *Algorithms for sparse nonnegative Tucker decompositions*, Neural Computation, 20 (2008), pp. 2112–2131.

[64] H. H. Muhammed, P. Ammenberg, and E. Bengtsson, *Using feature-vector based analysis, based on principal component analysis and independent component analysis, for analysing hyperspectral images*, in Proceedings of the 11th International Conference on Image Analysis and Processing, Palermo, Italy, 26-28 September 2001, pp. 309 – 315.

[65] M. Novak and R. Mammone, *Use of non-negative matrix factorization for language model adaptation in a lecture transcription task*, in Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, Salt Lake City, UT, USA, 7-11 May 2001, pp. 541 – 544.

[66] E. Oja, *Neural networks, principal components, and subspaces*, International Journal of Neural Systems, 1 (1989), pp. 61–68.

[67] E. Oja, *The nonlinear PCA learning rule in independent component analysis*, Neurocomputing, 17 (1997), pp. 25–45.

[68] E. Oja and J. Karhunen, *On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix*, Journal of Mathematical Analysis and Applications, 106 (1985), pp. 69–84.

[69] B. A. Olshausen and D. J. Field, *Emergence of simple-cell receptive-field properties by learning a sparse code for natural images*, Nature, 381 (1996), pp. 607–609.

[70] P. Paatero, *Least squares formulation of robust non-negative factor analysis*, Chemometrics and Intelligent Laboratory Systems, 37 (1997), pp. 23–35.

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

[72] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.-R. Müller, *Unmixing hyperspectral data*, in Advances in Neural Information Processing Systems 12 (Proc. NIPS*99), MIT Press, 2000, pp. 942–948.

[73] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmean, and R. Pacual-Marqui, *Nonsmooth nonnegative matrix factorization (nsNMF)*, IEEE Transaction Pattern Analysis and Machine Intelligence, 28 (2006), pp. 403–415.

[74] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, *Text mining using non-negative matrix factorizations*, in Proc. of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, FL, 2004, pp. 452–456.

[75] A. Phan and A. Cichocki, *Fast and efficient algorithms for nonnegative Tucker decomposition*, in Proc. of The Fifth International Symposium on Neural Networks, Springer LNCS-5264, Beijing, China, 24–28, September 2008, pp. 772–782.

[76] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin, *Object characterization from spectral data using nonnegative factorization and information theory*, in In Proc. AMOS Technical Conf., Maui, HI, Sept. 2004.

[77] M. D. Plumbley, *Adaptive lateral inhibition for non-negative ICA*, in Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA2001), San Diego, California, T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, eds., December 9-13 2001, pp. 516–521.

[78] ——, *Conditions for nonnegative independent component analysis*, IEEE Signal Processing Letters, 9 (2002), pp. 177–180.

[79] ——, *Algorithms for nonnegative independent component analysis*, IEEE Transactions on Neural Networks, 14 (2003), pp. 534–543.

[80] ——, *Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras*, Neurocomputing, 67 (2005), pp. 161–197.

[81] M. D. Plumbley and E. Oja, *A "nonnegative PCA" algorithm for independent component analysis*, IEEE Transactions on Neural Networks, 15 (2004), pp. 66–76.

[82] A. Prieto, C. G. Puntonet, and B. Prieto, *A neural learning algorithm for blind separation of sources based on geometric properties*, Signal Processing, 64 (1998), pp. 315–331.

[83] C. G. Puntonet, A. Mansour, and C. Jutten, *Geometrical algorithm for blind separation of sources*, in Actes du XVème Colloque GRETSI, Juan-Les-Pins, France, 18-21 September 1995, pp. 273–276.

[84] C. G. Puntonet and A. Prieto, *Neural net approach for blind separation of sources based on geometric properties*, Neurocomputing, 18 (1998), pp. 141–164.

[85] M. N. Schmidt and H. Laurberg, *Nonnegative matrix factorization with Gaussian process priors*, Computational Intelligence and Neuroscience, 2008 (2008).

[86] M. N. Schmidt and M. Mørup, *Nonnegative matrix factor 2-D deconvolution for blind single channel source separation*, in Independent Component Analysis and Signal Separation, International Conference on, vol. 3889 of Lecture Notes in Computer Science (LNCS), Springer, Apr. 2006, pp. 700–707.

[87] P. Smaragdis, *Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs*, in Independent Component Analysis and Blind Signal Separation: Proceedings of the Fifth International Conference (ICA 2004), Granada, Spain, September 22–24 2004, pp. 494–499.

[88] ——, *Convolutive speech bases and their application to supervised speech separation*, IEEE Transactions on Audio, Speech, and Language Processing, 15 (2007), pp. 1–12.

[89] M. W. Spratling, *Pre-synaptic lateral inhibition provides a better architecture for self-organizing neural networks*, Network: Computation in Neural Systems, 10 (1999), pp. 285–301.

[90] S. Sra and I. S. Dhillon, *Nonnegative matrix approximation: Algorithms and applications*, Tech. Report TR-06-27, Dept. of Computer Sciences, University of Texas at Austin, Austin, TX 78712, USA, 21 June 2006.

[91] R. Tauler, E. Casassas, and A. Izquierdo-Ridorsa, *Self-modelling curve resolution in studies of spectrometric titrations of multi-equilibria systems by factor analysis*, Anal. Chim. Acta, 248 (1991), pp. 447–458.

[92] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, *Dimensionality reduction using non-negative matrix factorization for information retrieval*, in IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, USA, 7-10 October 2001, pp. 960 – 965 vol.2.

[93] J. M. van den Hof and J. H. van Schuppen, *Positive matrix factorization via extremal polyhedral cones*, Linear Algebra and its Applications, 293 (1999), pp. 171–186.

[94] T. Virtanen, *Sound source separation using sparse coding with temporal continuity objective*, in Proceedings of the International Computer Music Conference (ICMC 2003), H. C. Kong and B. T. G. Tan, eds., Singapore, 29 September – 4 October 2003, pp. 231–234.

[95] T. Virtanen, *Separation of sound sources by convolutive sparse coding*, in Proceedings of the ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA 2004), Jeju, Korea, 3 October 2004.

[96] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recognition Letters, 22 (2001), pp. 1255–1261.

[97] H. Wersing, J. Eggert, and E. Körner, *Sparse coding with invariance constraints*, in Proceedings of the International Conference on Artificial Neural Networks (ICANN 2003), Istanbul, 2003, pp. 385–392.

[98] R. J. Williams, *Feature discovery through error-correction learning*, ICS Report 8501, Institute for Cognitive Science, University of California, San Diego, May 1985.

[99] L. Xu, *Least mean square error reconstruction principle for self-organizing neural-nets*, Neural Networks, 6 (1993), pp. 627–648.

[100] W. Xu, X. Liu, and Y. Gong, *Document clustering based on non-negative matrix factorization*, in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR03), 2003, pp. 267–273.

[101] T. Yamaguchi, K. Hirokawa, and K. Itoh, *Independent component analysis by transforming a scatter diagram of mixtures of signals*, Optics Communications, 173 (2000), pp. 107–114.