# The Why and How of Nonnegative Matrix Factorization

Nicolas Gillis

Department of Mathematics and Operational Research

Faculté Polytechnique, Université de Mons

Rue de Houdain 9, 7000 Mons, Belgium

nicolas.gillis@umons.ac.be

**Abstract**

Nonnegative matrix factorization (NMF) has become a widely used tool for the analysis of high-dimensional data as it automatically extracts sparse and meaningful features from a set of nonnegative data vectors. We first illustrate this property of NMF on three applications, in image processing, text mining and hyperspectral imaging –this is the why. Then we address the problem of solving NMF, which is NP-hard in general. We review some standard NMF algorithms, and also present a recent subclass of NMF problems, referred to as near-separable NMF, that can be solved efficiently (that is, in polynomial time), even in the presence of noise –this is the how. Finally, we briefly describe some problems in mathematics and computer science closely related to NMF via the nonnegative rank.

**Keywords.** Nonnegative matrix factorization, applications, algorithms.

## 1   Introduction

Linear dimensionality reduction (LDR) techniques are a key tool in data analysis, and are widely used for example for compression, visualization, feature selection and noise filtering. Given a set of data points $x_j \in \mathbb{R}^p$ for $1 \leq j \leq n$ and a dimension $r < \min(p, n)$, LDR amounts to computing a set of $r$ basis elements $w_k \in \mathbb{R}^p$ for $1 \leq k \leq r$ such that the linear space spanned by the $w_k$'s approximates the data points as closely as possible, that is, such that we have for all $j$

$$x_j \approx \sum_{k=1}^{r} w_k h_j(k), \qquad \text{for some weights } h_j \in \mathbb{R}^r. \tag{1}$$

In other words, the $p$-dimensional data points are represented in a $r$-dimensional linear subspace spanned by the basis elements $w_k$'s and whose coordinates are given by the vectors $h_j$'s. LDR is equivalent to low-rank matrix approximation: in fact, constructing

- the matrix $X \in \mathbb{R}^{p \times n}$ such that each column is a data point, that is, $X(:, j) = x_j$ for $1 \leq j \leq n$,

- the matrix $W \in \mathbb{R}^{p \times r}$ such that each column is a basis element, that is, $W(:, k) = w_k$ for $1 \leq k \leq r$, and

- the matrix $H \in \mathbb{R}^{r \times n}$ such that each column of $H$ gives the coordinates of a data point $X(:, j)$ in the basis $W$, that is, $H(:, j) = h_j$ for $1 \leq j \leq n$,

the above LDR model (1) is equivalent to $X \approx WH$, that is, to approximate the data matrix $X$ with a low-rank matrix $WH$.

A first key aspect of LDR is the choice of the measure to assess the quality of the approximation. It should be chosen depending on the *noise model*. The most widely used measure is the Frobenius norm of the error, that is, $||X - WH||_F^2 = \sum_{i,j}(X - WH)_{ij}^2$. The reason for the popularity of the Frobenius norm is two-fold. First, it implicitly assumes the noise $N$ present in the matrix $X = WH + N$ to be Gaussian, which is reasonable in many practical situations (see also the introduction of Section 3). Second, an optimal approximation can be computed efficiently through the truncated singular value decomposition (SVD); see [57] and the references therein. Note that the SVD is equivalent to principal component analysis (PCA) after mean centering of the data points (that is, after shifting all data points so that their mean is on the origin).

A second key aspect of LDR is the assumption on the structure of the factors $W$ and $H$. The truncated SVD and PCA do not make any assumption on $W$ and $H$. For example, assuming independence of the columns of $W$ leads to independent component analysis (ICA) [29], or assuming sparsity of $W$ (and/or $H$) leads to sparse low-rank matrix decompositions, such as sparse PCA [32]. Nonnegative matrix factorization (NMF) is an LDR where both the basis elements $w_k$'s and the weights $h_j$'s are assumed to be *component-wise nonnegative*. Hence NMF aims at decomposing a given nonnegative data matrix $X$ as $X \approx WH$ where $W \geq 0$ and $H \geq 0$ (meaning that $W$ and $H$ are component-wise nonnegative). NMF was first introduced in 1994 by Paatero and Tapper [97] and gathered more and more interest after an article by Lee and Seung [79] in 1999.

In this paper, we explain *why* NMF has been so popular in different data mining applications, and *how* one can compute NMF's. The aim of this paper is not to give a comprehensive overview of all NMF applications and algorithms –and we apologize for not mentioning many relevant contributions– but rather to serve as an introduction to NMF, describing three applications and several standard algorithms.

# 2 The Why – NMF Generates Sparse and Meaningful Features

The reason why NMF has become so popular is because of its ability to automatically extract sparse and easily interpretable factors. In this section, we illustrate this property of NMF through three applications, in image processing, text mining and hyperspectral imaging. Other applications include air emission control [97], computational biology [34], blind source separation [22], single-channel source separation [82], clustering [35], music analysis [42], collaborative filtering [92], and community detection [106].

## 2.1 Image Processing – Facial Feature Extraction

Let each column of the data matrix $X \in \mathbb{R}_+^{p \times n}$ be a vectorized gray-level image of a face, with the $(i,j)$th entry of matrix $X$ being the intensity of the $i$th pixel in the $j$th face. NMF generates two factors $(W, H)$ so that each image $X(:,j)$ is approximated using a linear combination of the columns of $W$; see Equation (1), and Figure 1 for an illustration. Since $W$ is nonnegative, the columns of $W$ can be interpreted as images (that is, vectors of pixel intensities) which we refer to as the basis images. As the weights in the linear combinations are nonnegative ($H \geq 0$), these basis images can only be summed up to reconstruct each original image. Moreover, the large number of images in the data set must be reconstructed approximately with only a few basis images (in fact, $r$ is in general much smaller than $n$), hence the latter should be localized features (hence sparse) found simultaneously in several
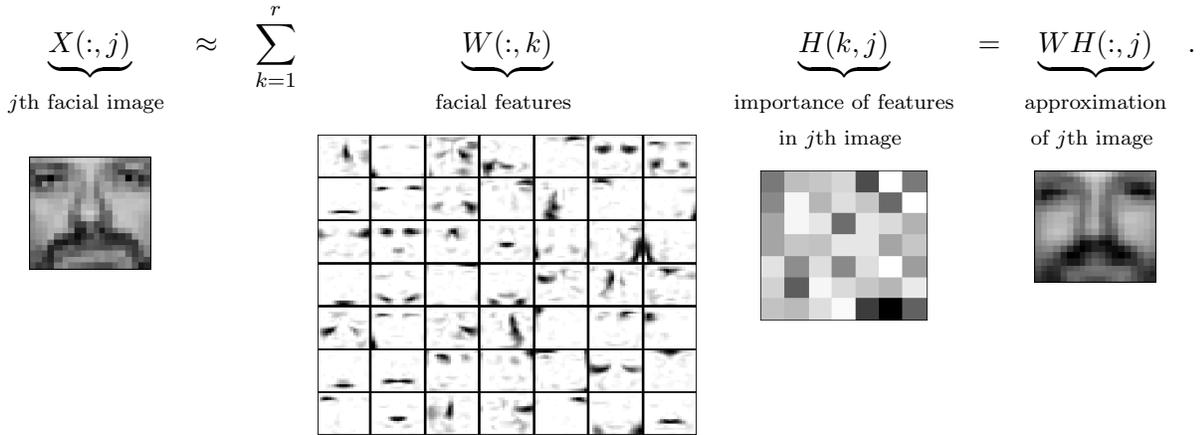
$$\underbrace{X(:,j)}_{j\text{th facial image}} \quad \approx \quad \sum_{k=1}^{r} \quad \underbrace{W(:,k)}_{\text{facial features}} \qquad \underbrace{H(k,j)}_{\substack{\text{importance of features} \\ \text{in } j\text{th image}}} \quad = \quad \underbrace{WH(:,j)}_{\substack{\text{approximation} \\ \text{of } j\text{th image}}} \quad .$$



Figure 1: Decomposition of the CBCL face database, MIT Center For Biological and Computation Learning (2429 gray-level 19-by-19 pixels images) using $r = 49$ as in [79].

images. In the case of facial images, the basis images are features such as eyes, noses, mustaches, and lips (see Figure 1) while the columns of $H$ indicate which feature is present in which image (see also [79, 61]).

A potential application of NMF is in face recognition. It has for example been observed that NMF is more robust to occlusion than PCA (which generates dense factors): in fact, if a new occluded face (e.g., with sun glasses) has to be mapped into the NMF basis, the non-occluded parts (e.g., the mustache or the lips) can still be well approximated [61].

## 2.2   Text Mining – Topic Recovery and Document Classification

Let each column of the nonnegative data matrix $X$ correspond to a document and each row to a word. The $(i,j)$th entry of the matrix $X$ could for example be equal to the number of times the $i$th word appears in the $j$th document in which case each column of $X$ is the vector of word counts of a document; in practice, more sophisticated constructions are used, e.g., the term frequency - inverse document frequency (tf-idf). This is the so-called bag-of-words model: each document is associated with a set of words with different weights, while the ordering of the words in the documents is not taken into account (see, e.g., the survey [10] for a discussion). Note that such a matrix $X$ is in general rather sparse as most documents only use a small subset of the dictionary. Given such a matrix $X$ and a factorization rank $r$, NMF generates two factors $(W, H)$ such that, for all $1 \leq j \leq n$, we have

$$\underbrace{X(:,j)}_{j\text{th document}} \quad \approx \quad \sum_{k=1}^{r} \quad \underbrace{W(:,k)}_{k\text{th topic}} \qquad \underbrace{H(k,j)}_{\substack{\text{importance of } k\text{th topic} \\ \text{in } j\text{th document}}} \quad , \qquad \text{with } W \geq 0 \text{ and } H \geq 0.$$

This decomposition can be interpreted as follows (see, also, e.g., [79, 101, 3]):

- Because $W$ is nonnegative, each column of $W$ can be interpreted as a document, that is, as a bag of words.

- Because the weights in the linear combinations are nonnegative ($H \geq 0$), one can only take the union of the sets of words defined by the columns of $W$ to reconstruct all the original documents.

- Moreover, because the number of documents in the data set is much larger than the number of basis elements (that is, the number of columns of $W$), the latter should be set of words found simultaneously in several documents. Hence the basis elements can be interpreted as *topics*, that is, set of words found simultaneously in different documents, while the weights in the linear combinations (that is, the matrix $H$) assign the documents to the different topics, that is, identify which document discusses which topic.

Therefore, given a set of documents, *NMF identifies topics and simultaneously classifies the documents among these different topics.* Note that NMF is closely related to existing topic models, in particular probabilistic latent semantic analysis and indexing (PLSA and PLSI) [45, 37].

## 2.3   Hyperspectral Unmixing – Identify Endmembers and Classify Pixels

Let the columns of the nonnegative data matrix $X$ be the spectral signatures of the pixels in a scene being imaged. The spectral signature of a pixel is the fraction of incident light being reflected by that pixel at different wavelengths, and is therefore nonnegative. For a hyperspectral image, there are usually between 100 and 200 wavelength-indexed bands, observed in much broader spectrum than the visible light. This allows for more accurate analysis of the scene under study.

Given a hyperspectral image (see Figure 2 for an illustration), the goal of blind hyperspectral unmixing (blind HU) is two-fold:

1. Identify the constitutive materials present in the image; for example, it could be grass, roads, or metallic surfaces. These are referred to as the *endmembers*.

2. Classify the pixels, that is, identify which pixel contains which endmember and in which proportion. (In fact, pixels are in general mixture of several endmembers, due for example to low spatial resolution or mixed materials.)

The simplest and most popular model used to address this problem is the *linear mixing model*. It assumes that the spectral signature of a pixel results from the linear combination of the spectral signature of the endmembers it contains. The weights in the linear combination correspond to the abundances of these endmembers in that pixel. For example, if a pixel contains 30% of grass and 70% of road surface, then, under the linear mixing model, its spectral signature will be 0.3 times the spectral signature of the grass plus 0.7 times the spectral signature of the road surface. This is exactly the NMF model: the spectral signatures of the endmembers are the basis elements, that is, the columns of $W$, while the abundances of the endmembers in each pixel are the weights, that is, the columns of $H$. Note that the factorization rank $r$ corresponds to the number of endmembers in the hyperspectral image. Figure 2 illustrates such a decomposition.

Therefore, given a hyperspectral image, *NMF is able to compute the spectral signatures of the endmembers and simultaneously the abundance of each endmember in each pixel.* We refer the reader to [8, 90] for recent surveys on blind HU techniques.
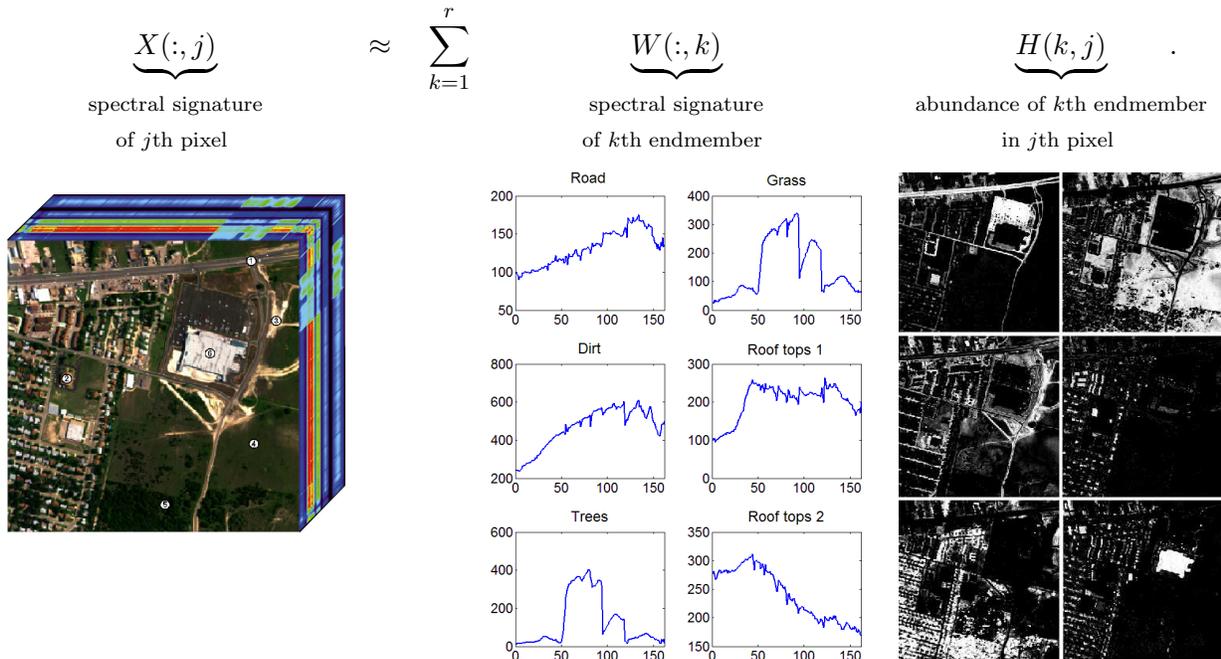
$$\underbrace{X(:,j)}_{\substack{\text{spectral signature} \\ \text{of } j\text{th pixel}}} \quad \approx \quad \sum_{k=1}^{r} \quad \underbrace{W(:,k)}_{\substack{\text{spectral signature} \\ \text{of } k\text{th endmember}}} \quad \underbrace{H(k,j)}_{\substack{\text{abundance of } k\text{th endmember} \\ \text{in } j\text{th pixel}}} \quad .$$



Figure 2: Decomposition of the Urban hyperspectral image from `http://www.agc.army.mil/`, constituted mainly of six endmembers ($r = 6$). Each column of the matrix $W$ is the spectral signature of an endmember, while each row of the matrix $H$ is the abundance map of the corresponding endmember, that is, it contains the abundance of all pixels for that endmember. (Note that to obtain this decomposition, we used a sparse prior on the matrix $H$; see Section 3.)

## 3  The How – Some Algorithms

We have seen in the previous section that NMF is a useful LDR technique for nonnegative data. The question is now: can we compute such factorizations? In this paper, we focus on the following optimization problem

$$\min_{W \in \mathbb{R}^{p \times r}, H \in \mathbb{R}^{r \times n}} ||X - WH||_F^2 \quad \text{such that} \quad W \geq 0 \text{ and } H \geq 0. \tag{2}$$

Hence we implicitly assume Gaussian noise on the data; see Introduction. Although this NMF model is arguably the most popular, it is not always reasonable to assume Gaussian noise for nonnegative data, especially for sparse matrices such as document data sets; see the discussion in [23]. In fact, many other objective functions are used in practice, e.g., the (generalized) Kullback-Leibler divergence for text mining [23], the Itakura-Saito distance for music analysis [42], the $\ell_1$ norm to improve robustness against outliers [71], and the earth mover's distance for computer vision tasks [100]. Other NMF models are motivated by statistical considerations; we refer the reader to the recent survey [102].

There are many issues when using NMF in practice. In particular,

- **NMF is NP-hard**. Unfortunately, as opposed to the unconstrained problem which can be solved efficiently using the SVD, NMF is NP-hard in general [105]. Hence, in practice, most

algorithms are applications of standard nonlinear optimization methods and may only be guaranteed to converge to stationary points; see Section 3.1. However, these heuristics have been proved to be successful in many applications. More recently, Arora et al. [4] described a subclass of nonnegative matrices for which NMF can be solved efficiently. These are the near-separable matrices which will be addressed in Section 3.2. Note that Arora et al. [4] also described an algorithmic approach for exact NMF[1] requiring $\mathcal{O}\big((pn)^{2^r r^2}\big)$ operations –later improved to $\mathcal{O}\big((pn)^{r^2}\big)$ by Moitra [94]– hence polynomial in the dimensions $p$ and $n$ for $r$ fixed. Although $r$ is usually small in practice, this approach cannot be used to solving real-world problems because of its high computational cost (in contrast, most heuristic NMF algorithms run in $\mathcal{O}(pnr)$ operations; see Section 3.1).

- **NMF is ill-posed**. Given an NMF $(W, H)$ of $X$, there usually exist equivalent NMF's $(W', H')$ with $W'H' = WH$. In particular, any matrix $Q$ satisfying $WQ \geq 0$ and $Q^{-1}H \geq 0$ generates such an equivalent factorization. The matrix $Q$ can always be chosen as the permutation of a diagonal matrix with positive diagonal elements (that is, as a monomial matrix) and this amounts to the scaling and permutation of the rank-one factors $W(:,k)H(k,:)$ for $1 \leq k \leq r$; this is not an issue in practice. The issue is when there exist non-monomial matrices $Q$ satisfying the above conditions. In that case, such equivalent factorizations generate different interpretations: for example, in text mining, they would lead to different topics and classifications; see the discussion in [48]. Here is a simple example

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

We refer the reader to [66] and the references therein for recent results on non-uniqueness of NMF.

In practice, this issue is tackled using other priors on the factors $W$ and $H$ and adding proper regularization terms in the objective function. The most popular prior is sparsity which can be tackled with projections [64] or with $\ell_1$-norm penalty [72, 48]. For example, in blind HU (Section 2.3), the abundance maps (that is, the rows of matrix $H$) are usually very sparse (most pixels contain only a few endmembers) and applying plain NMF (2) usually gives poor results for these data sets. Other priors for blind HU include piece-wise smoothness of the spectral signatures or spatial coherence (neighboring pixels are more likely to contain the same materials) which are usually tackled with TV-like regularizations (that is, $\ell_1$ norm of the difference between neighboring values to preserve the edges in the image); see, e.g., [68, 67], and the references therein. Note that the design and algorithmic implementation of refined NMF models for various applications is a very active area of research, e.g., graph regularized NMF [16], orthogonal NMF [24], tri-NMF [38, 85], semi and convex NMF [36], projective NMF [110], minimum volume NMF [93], and hierarchical NMF [86], to cite only a few.

- **Choice of $r$.** The choice of the factorization rank $r$, that is, the problem of order model selection, is usually rather tricky. Several popular approaches are: trial and error (that is, try different values of $r$ and pick the one performing best for the application at hand), estimation using the SVD (that is, look at the decay of the singular values of the input data matrix), and

---

[1]Exact NMF refers to the NMF problem where an exact factorization is sought: $X = WH$ with $W \geq 0$ and $H \geq 0$.

the use of experts insights (e.g., in blind HU, experts might have a good guess for the number of endmembers present in a scene); see also [7, 104, 70] and the references therein.

In this section, we focus on the first issue. In Section 3.1, we present several standard algorithms for the general problem (2). In Section 3.2, we describe the near-separable NMF problem and several recent algorithms.

## 3.1 Standard NMF Algorithms

Almost all NMF algorithms designed for (2) use a two-block coordinate descent scheme (exact or inexact; see below), that is, they optimize alternatively over one of the two factors, $W$ or $H$, while keeping the other fixed. The reason is that the subproblem in one factor is convex. More precisely, it is a nonnegative least squares problem (NNLS): for example, for $H$ fixed, we have to solve $\min_{W \geq 0} ||X - WH||_F^2$. Note that this problem has a particular structure as it can be decomposed into $p$ independent NNLS in $r$ variables since

$$||X - WH||_F^2 = \sum_{i=1}^p ||X_{i:} - W_{i:}H||_2^2 = \sum_{i=1}^p W_{i:} \left(HH^T\right) W_{i:}^T - 2W_{i:} \left(HX_{i:}^T\right) + ||X_{i:}||_2^2. \tag{3}$$

Many algorithms exist to solve the NNLS problem, and NMF algorithms based on two-block coordinate descent differ by which NNLS algorithm is used; see also, e.g., the discussion in [74]. It is interesting to notice that the problem is symmetric in $W$ and $H$ since $||X - WH||_F^2 = ||X^T - H^TW^T||_F^2$. Therefore, we can focus on the update of only one factor and, in fact, most NMF algorithms use the same update for $W$ and $H$, and therefore adhere to the framework described in Algorithm CD.

---

**Algorithm CD** Two-Block Coordinate Descent – Framework of Most NMF Algorithms

---

**Input:** Input nonnegative matrix $X \in \mathbb{R}_+^{p \times n}$ and factorization rank $r$.
**Output:** $(W, H) \geq 0$: A rank-$r$ NMF of $X \approx WH$.

1: Generate some initial matrices $W^{(0)} \geq 0$ and $H^{(0)} \geq 0$; see Section 3.1.8.
2: **for** $t = 1, 2, \ldots^\dagger$ **do**
3:      $W^{(t)} = \text{update}\left(X, H^{(t-1)}, W^{(t-1)}\right)$.
4:      $H^{(t)^T} = \text{update}\left(X^T, W^{(t)^T}, H^{(t-1)^T}\right)$.
5: **end for**
  $^\dagger$See Section 3.1.7 for stopping criteria.

---

The update in steps 3 and 4 of Algorithm CD usually guarantees the objective function to decrease. In this section, we describe the most widely used updates, that is, we describe several standard and widely used NMF algorithms, and compare them in Section 3.1.6. But first we address an important tool to designing NMF algorithms: the optimality conditions. To simplify notations, we will drop the iteration index $t$.

### 3.1.1 First-Order Optimality Conditions

Given $X$, let us denote $F(W, H) = \frac{1}{2}||X - WH||_F^2$. The first-order optimality conditions for (2) are

$$W \geq 0, \quad \nabla_W F = WHH^T - XH^T \geq 0, \quad W \circ \nabla_W F = 0, \tag{4}$$
$$H \geq 0, \quad \nabla_H F = W^TWH - W^TX \geq 0, \quad H \circ \nabla_H F = 0,$$

where $\circ$ is the component-wise product of two matrices. Any $(W, H)$ satisfying these conditions is a stationary point of (2).

It is interesting to observe that these conditions give a more formal explanation of why NMF naturally generates sparse solutions [51]: in fact, any stationary point of (2) is expected to have zero entries because of the conditions $W \circ \nabla_W F = 0$ and $H \circ \nabla_H F = 0$, that is, the conditions that for all $i, k$ either $W_{ik}$ is equal to zero or the partial derivative of $F$ with respect to $W_{ik}$ is, and similarly for $H$.

### 3.1.2  Multiplicative Updates

Given $X$, $W$ and $H$, the multiplicative updates (MU) modify $W$ as follows

$$W \leftarrow W \circ \frac{\left[XH^T\right]}{\left[WHH^T\right]} \tag{5}$$

where $\frac{[\,]}{[\,]}$ denotes the component-wise division between two matrices. The MU were first developed in [33] for solving NNLS problems, and later rediscovered and used for NMF in [80]. The MU are based on the majorization-minimization framework. In fact, (5) is the global minimizer of a quadratic function majorizing $F$, that is, a function that is larger than $F$ everywhere and is equal to $F$ at the current iterate [33, 80]. Hence minimizing that function guarantees $F$ to decrease and therefore leads to an algorithm for which $F$ monotonically decreases. The MU can also be interpreted as a rescaled gradient method: in fact,

$$W \circ \frac{\left[XH^T\right]}{\left[WHH^T\right]} = W - \frac{[W]}{\left[WHH^T\right]} \circ \nabla_W F. \tag{6}$$

Another more intuitive interpretation is as follows: we have that

$$\frac{\left[XH^T\right]_{ik}}{\left[WHH^T\right]_{ik}} \geq 1 \quad \Longleftrightarrow \quad (\nabla_W F)_{ik} \leq 0.$$

Therefore, in order to satisfy (4), for each entry of $W$, the MU either (i) increase it if its partial derivative is negative, (ii) decrease it if its partial derivative is positive, or (iii) leave it unchanged if its partial derivative is equal to zero.

If an entry of $W$ is equal to zero, the MU cannot modify it hence it may occur that an entry of $W$ is equal to zero while its partial derivative is negative which would not satisfy (4). Therefore, the MU are not guaranteed to converge to a stationary point[2]. There are several ways to fix this issue, e.g., rewriting the MU as a rescaled gradient descent method –see Equation (6): only entries in the same row interact– and modifying the step length [87], or using a small positive lower bound for the entries of $W$ and $H$ [52, 103]; see also [5]. A simpler and nice way to guarantee convergence of the MU to a stationary point is proposed in [23]: use the original updates (5) while reinitializing zero entries of $W$ to a small positive constant when their partial derivatives become negative.

The MU became extremely popular mainly because (i) they are simple to implement[3], (ii) they scale well and are applicable to sparse matrices[4], and (iii) they were proposed in the paper of Lee

---

[2]If the initial matrices are chosen positive, some entries can first converge to zero while their partial derivative eventually becomes negative or zero (when strict complementarity is not met) which is numerically unstable; see [52] for some numerical experiments.

[3]For example, in Matlab: `W = W.*(X*H')./(W*(H*H'))`.

[4]When computing the denominator $WHH^T$ in the MU, it is crucial to compute $HH^T$ first in order to have the lowest computational cost, and make the MU scalable for sparse matrices; see, e.g., footnote 3.

and Seung [79] which launched the research on NMF. However, the MU converge relatively slowly; see, e.g., [62] for a theoretical analysis, and Section 3.1.6 for some numerical experiments. Note that the original MU only update $W$ once before updating $H$. They can be significantly accelerated using a more effective alternation strategy [52]: the idea is to update $W$ several times before updating $H$ because the products $HH^T$ and $XH^T$ do not need to be recomputed.

### 3.1.3   Alternating Least Squares

The alternating least squares method (ALS) first computes the optimal solution of the unconstrained least squares problem $\min_W ||X - WH||_F$ and then project the solution onto the nonnegative orthant:

$$W \leftarrow \max \left( \operatorname{argmin}_{Z \in \mathbb{R}^{p \times r}} ||X - ZH||_F, 0 \right),$$

where the max is taken component-wise. The method has the advantage to be relatively cheap, and easy to implement[5]. ALS usually does not converge: the objective function of (2) might oscillate under the ALS updates (especially for dense input matrices $X$; see Section 3.1.6). It is interesting to notice that, because of the projection, the solution generated by ALS is not scaled properly. In fact, the error can be reduced (sometimes drastically) by multiplying the current solution $WH$ by the constant

$$\alpha^* = \operatorname{argmin}_{\alpha \geq 0} ||X - \alpha WH||_F = \frac{\langle X, WH \rangle}{\langle WH, WH \rangle} = \frac{\langle XH^T, W \rangle}{\langle W^T W, HH^T \rangle}. \tag{7}$$

Although it is in general not recommended to use ALS because of the convergence issues, ALS can be rather powerful for initialization purposes (that is, perform a few steps of ALS and then switch to another NMF algorithm), especially for sparse matrices [28].

### 3.1.4   Alternating Nonnegative Least Squares

Alternating nonnegative least squares (ANLS) is a class of methods where the subproblems in $W$ and $H$ are solved exactly, that is, the update for $W$ is given by

$$W \leftarrow \operatorname{argmin}_{W \geq 0} ||X - WH||_F.$$

Many methods can be used to solve the NNLS $\operatorname{argmin}_{W \geq 0} ||X - WH||_F$, and dedicated active-set methods have shown to perform very well in practice[6]; see [72, 73, 75]. Other methods are based for example on projected gradients [88], Quasi-Newton [26], or fast gradient methods [60]. ANLS is guaranteed to converge to a stationary point [59]. Since each iteration of ANLS computes an optimal solution of the NNLS subproblem, each iteration of ANLS decreases the error the most among NMF algorithms following the framework described in Algorithm CD. However, each iteration is computationally more expensive, and more difficult to implement.

Note that, because usually the initial guess $WH$ is a poor approximation of $X$, it does not make much sense to solve the NNLS subproblems exactly at the first steps of Algorithm CD, and therefore it might be profitable to use ANLS rather in a refinement step of a cheaper NMF algorithm (such as the MU or ALS).

---

[5]For example, in Matlab: `W = max(0,(X*H')/(H*H'))`.

[6]In particular, the Matlab function `lsqnonneg` implements an active-set method from [78].

### 3.1.5 Hierarchical Alternating Least Squares

Hierarchical alternating least squares (HALS) solves the NNLS subproblem using an exact coordinate descent method, updating one column of $W$ at a time. The optimal solutions of the corresponding subproblems can be written in closed form. In fact, the entries of a column of $W$ do not interact –see Equation (3)– hence the corresponding problem can be decoupled into $p$ quadratic problems with a single nonnegative variable. HALS updates $W$ as follows. For $\ell = 1, 2, \ldots, r$:

$$W(:,\ell) \leftarrow \text{argmin}_{W(:,\ell) \geq 0} \left\| X - \sum_{k \neq \ell} W(:,k) H(k,:) - W(:,\ell) H(\ell,:) \right\|_F$$

$$\leftarrow \max \left( 0, \frac{X H(\ell,:)^T - \sum_{k \neq \ell} W(:,k) \left( H(k,:) H(\ell,:)^T \right)}{\|H(\ell,:)\|_2^2} \right).$$

HALS has been rediscovered several times, originally in [27] (see also [25]), then as the rank-one residue iteration (RRI) in [63], as FastNMF in [84], and also in [89]. Actually, HALS was first described in Rasmus Bro's thesis [14, pp.161-170] (although it was not investigated thoroughly):

> ...to solve for a column vector $w$ of $W$ it is only necessary to solve the unconstrained problem and subsequently set negative values to zero. Though the algorithm for imposing non-negativity is thus simple and may be advantageous in some situations, it is not pursued here. Since it optimizes a smaller subset of parameters than the other approaches it may be unstable in difficult situations.

HALS was observed to converge much faster than the MU (see [47, p.131] for a theoretical explanation, and Section 3.1.6 for a comparison) while having almost the same computational cost; see [52] for a detailed account of the flops needed per iteration. Moreover, HALS is, under some mild assumptions, guaranteed to converge to a stationary point; see the discussion in [52]. Note that one should be particularly careful when initializing HALS otherwise the algorithm could set some columns of $W$ to zero initially (e.g., if $WH$ is badly scaled with $WH \gg X$) hence it is recommended to initially scale $(W, H)$ according to (7); see the discussion in [47, p.72].

In the original HALS, each column of $W$ is updated only once before updating $H$. However, as for the MU, it can be sped up by updating $W$ several times before updating $H$ [52], or selecting the entries of $W$ to update following a Gauss-Southwell-type rule [65]. HALS can also be generalized to other cost functions using Taylor expansion [83].

### 3.1.6 Comparison

Figure 3 displays the evolution of the objective function of (2) for the algorithms described in the previous section: on the left, the dense CBCL data set (see also Figure 1), and, on the right, the sparse Classic document data set. As anticipated in the description of the different algorithms in the previous sections, we observe that:

- The MU converge rather slowly.

- ALS oscillates for the dense matrix (CBCL data set) and performs quite poorly while, for the sparse matrix (Classic data set), it converges initially very fast but then stabilizes and cannot compute a solution with small objective function value.
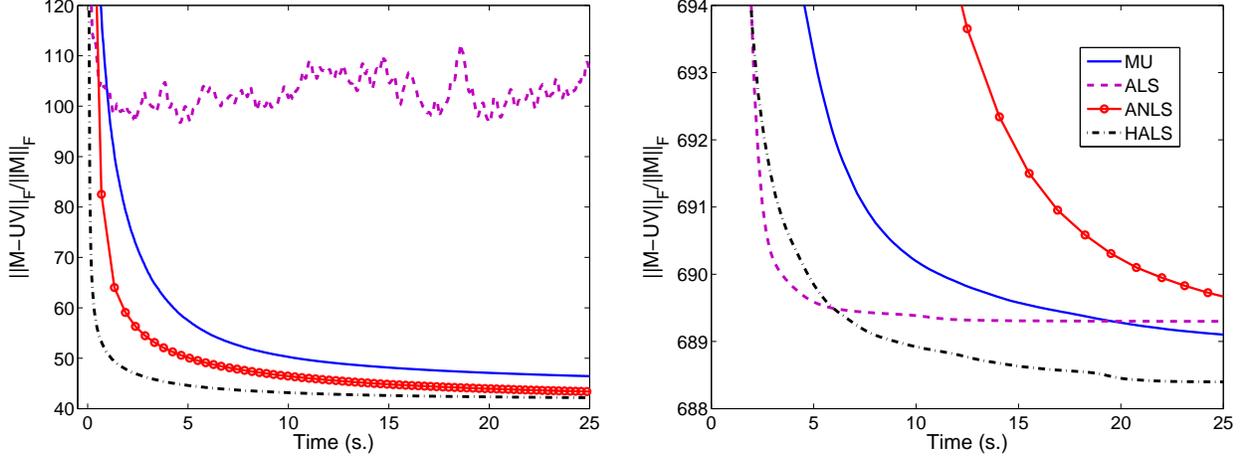
Figure 3: Comparison of MU, ALS, ANLS and HALS. On the left: CBCL facial images with $r = 49$; same data set as in Figure 1. On the right: Classic document data set with $m = 7094$, $n = 41681$ and $r = 20$; see, e.g., [113]. The figure displays the average error using the same ten initial matrices $W$ and $H$ for all algorithms, randomly generated with the `rand` function of Matlab. All tests were performed using Matlab on a laptop Intel CORE i5-3210M CPU @2.5GHz 2.5GHz 6Go RAM. Note that, for ALS, we display the error after scaling; see Equation (7). For MU and HALS, we used the implementation from `https://sites.google.com/site/nicolasgillis/`, for ANLS from `http://www.cc.gatech.edu/~hpark/nmfsoftware.php`, and ALS was implemented following footnote 5.

- ANLS performs rather well for the dense matrix and is the second best after HALS. However, it performs rather poorly for the sparse matrix.

- HALS performs the best as it generates the best solutions within the allotted time.

For other comparisons of NMF algorithms and more numerical experiments, we refer the reader to the book [28], the theses [63, 47], the survey [6], and the references therein.

Further research on NMF includes the design of more efficient algorithms, in particular for regularized problems; see, e.g., [98] for a recent example of imposing sparsity in a more robust and stable way. We conclude this section with some comments about stopping criteria and initializations of NMF algorithms.

### 3.1.7 Stopping Criterion

There are several approaches for the stopping criterion of NMF algorithms, as in usual non-linear optimization schemes, e.g., based on the evolution of the objective function, on the optimality conditions (4), or on the difference between consecutive iterates. These criteria are typically combined with either a maximum number of iterations or a time limit to ensure termination; see, e.g., the discussion in [47]. In this section, we would like to point out an issue which is sometimes overlooked in the literature when using the optimality conditions to assess the convergence of NMF algorithms. A criterion based on the optimality conditions is for example $C(W, H) = C_W(W) + C_H(H)$ where

$$C_W(W) = \underbrace{|| \min(W, 0)||_F}_{W \geq 0} + \underbrace{|| \min(\nabla_W F, 0)||_F}_{\nabla_W F \geq 0} + \underbrace{||W \circ \nabla_W F||_F}_{W \circ \nabla_W F = 0}, \tag{8}$$

11

and similarly $C_H(H)$ for $H$, so that $C(W, H) = 0 \iff (W, H)$ is a stationary point of (2). There are several problems to using $C(W, H)$ (and other similar variants) as a stopping criterion and for comparing the convergence of different algorithms:

- It is sensitive to scaling. For $\alpha > 0$ and $\alpha \neq 1$, we will have in general that

$$C_W(W) + C_H(H) = C(W, H) \neq C(\alpha W, \alpha^{-1} H).$$

  since the first two terms in (8) are sensitive to scaling. For example, if one solves the subproblem in $W$ exactly and obtains $C_W(W) = 0$ (this will be the case for ANLS; see Section 3.1.4), then $\nabla_H F$ can be made arbitrarily small by multiplying $W$ by a small constant and dividing $H$ by the same constant (while, if $H \geq 0$, it will not influence the first term which is equal to zero). This issue can be handled with proper normalization, e.g., imposing $||W(:, k)||_2 = ||H(k, :)||_2$ for all $k$; see [63].

- The value of $C(W, H)$ after the update of $W$ can be very different from the value after an update of $H$ (in particular, if the scaling is bad or if $|m - n| \gg 0$). Therefore, one should be very careful when using this type of criterion to compare ANLS-type methods with other algorithms such as the MU or HALS as the evolution of $C(W, H)$ can be misleading (in fact, an algorithm that monotonically decreases the objective function, such as the MU or HALS, is not guaranteed to monotonically decrease $C(W, H)$.) A potential fix would be to scale the columns of $W$ and the rows of $H$ so that $C_W(W)$ after the update of $H$ and $C_H(H)$ after the update of $W$ have the same order of magnitude.

### 3.1.8 Initialization

A simple way to initialize $W$ and $H$ is to generate them randomly (e.g., generating all entries uniformly at random in the interval [0,1]). Several more sophisticated initialization strategies have been developed in order to have better initial estimates in the hope to (i) obtain a good factorization with fewer iterations, and (ii) converge to a better stationary point. However, most initialization strategies come with no theoretical guarantee (e.g., a bound on the distance of the initial point to optimality) which can be explained in part by the complexity of the problem (in fact, NMF is NP-hard in general –see the introduction of this section). This could be an interesting direction for further research. We list some initialization strategies here, they are based on

- *Clustering techniques.* Use the centroids computed with some clustering method, e.g., with $k$-means or spherical $k$-means, to initialize the columns of $W$, and initialize $H$ as a proper scaling of the cluster indicator matrix (that is, $H_{kj} \neq 0 \iff X(:, j)$ belongs to the $k$th cluster) [107, 109]; see also [18] and the references therein for some recent results.

- *The SVD.* Let $\sum_{k=1}^{r} u_k v_k^T$ be the best rank-$r$ approximation of $X$ (which can be computed, e.g., using the SVD; see Introduction). Each rank-one factor $u_k v_k^T$ might contain positive and negative entries (except for the first one, by the Perron-Frobenius theorem[7]). However, denoting $[x]_+ = \max(x, 0)$, we have

$$u_k v_k^T = [u_k]_+ [v_k^T]_+ + [-u_k]_+ [-v_k^T]_+ - [-u_k]_+ [v_k^T]_+ - [u_k]_+ [-v_k^T]_+,$$

---

[7]Actually, the first factor could contain negative entries if the input matrix is reducible and its first two singular values are equal to one another; see, e.g., [47, p.16].

and the first two rank-one factors in this decomposition are nonnegative. Boutsidis et al. [11] proposed to replace each rank-one factor in $\sum_{k=1}^{r} u_k v_k^T$ with either $[u_k]_+[v_k^T]_+$ or $[-u_k]_+[-v_k^T]_+$, selecting the one with larger norm and scaling it properly.

- *Column subset selection.* It is possible to initialize the columns of $W$ using data points, that is, initialize $W = X(:,\mathcal{K})$ for some set $\mathcal{K}$ with cardinality $r$; see [21, 112] and Section 3.2.

In practice, one may use several initializations, and keep the best solution obtained; see, e.g., the discussion in [28].

## 3.2   Near-Separable NMF

A matrix $X$ is $r$-separable if there exists an index set $\mathcal{K}$ of cardinality $r$ such that

$$X = X(:,\mathcal{K})H \qquad \text{for some } H \geq 0.$$

In other words, there exists a subset of $r$ columns of $X$ which generates a convex cone containing all columns of $X$. Hence, given a separable matrix, the goal of separable NMF is to identify the subset of columns $\mathcal{K}$ that allows to reconstruct all columns of $X$ (in fact, given $X(:,\mathcal{K})$, $H$ can be computed by solving a convex optimization program; see Section 3.1). The separability assumption makes sense in several applications: for example,

- In text mining (see Section 2.2), separability of the word-by-document matrix requires that for each topic, there exists a document only on that topic. Note that we can also assume separability of the transpose of $X$ (that is, of the document-by-word matrix), i.e., for each topic there exists one word used only by that topic (referred to as an 'anchor' word). In fact, the latter is considered a more reasonable assumption in practice; see [77, 3, 39] and also the thesis [46] for more details.

- In hyperspectral unmixing (see Section 2.3), separability of the wavelength-by-pixel matrix requires that for each endmember there exists a pixel containing only that endmember. This is the so-called pure-pixel assumption, and makes sense for relatively high spatial resolution hyperspectral images; see [8, 90] and the references therein.

Separability has also been used successfully in blind source separation [95, 22], video summarization and image classification [40], and foreground-background separation in computer vision [76]. Note that for facial feature extraction described in Section 2.1, separability does not make sense since we cannot expect features to be present in the data set.

It is important to points out that separable NMF is closely related to several problems, including

- Column subset selection which is a long-standing problem in numerical linear algebra (see [12] and the references therein).

- Pure-pixel search in hyperspectral unmixing which has been addressed long before NMF was introduced; see [90] for a historical note.

- The problem of identifying a few important data points in a data set (see [40] and the references therein).

- Convex NMF [36], and the CUR decomposition [91].

Therefore, it is difficult to pinpoint the roots of separable NMF and a comprehensive overview of all methods related to separable NMF is out of the scope of this paper. However, to the best of our knowledge, it is only very recently that provably efficient algorithms for separable NMF have been proposed. This new direction of research was launched by a paper by Arora et al. [4] which shows that NMF of separable matrices can be computed efficiently (that is, in polynomial time), even in the presence of noise (the error can be bounded in terms of the noise level; see below). We focus in this section on these provably efficient algorithms for separable NMF.

In the noiseless case, separable NMF reduces to identifying the extreme rays of the cone spanned by the columns of $X$. If the columns of the input matrix $X$ are normalized so that their entries sum to one, that is, $X(:,j) \leftarrow ||X(:,j)||_1^{-1} X(:,j)$ for all $j$ (and discarding the zero columns of $X$), then the problem reduces to identifying the vertices of the convex hull of the columns of $X$. In fact, since the entries of each column of $X$ sum to one and $X = X(:,\mathcal{K})H$, the entries of each column of $H$ must also sum to one: as $X$ and $H$ are nonnegative, we have for all $j$

$$1 = ||X(:,j)||_1 = ||X(:,\mathcal{K})H(:,j)||_1$$
$$= \sum_k ||X(:,\mathcal{K}(k))||_1 H(k,j) = \sum_k H(k,j) = ||H(:,j)||_1.$$

Therefore, the columns of $X$ are convex combinations (that is, linear combinations with nonnegative weights summing to one) of the columns of $X(:,\mathcal{K})$.

In the presence of noise, the problem is referred to as near-separable NMF, and can be formulated as follows

(Near-Separable NMF) *Given a noisy $r$-separable matrix $\tilde{X} = X + N$ with $X = W[I_r, H']\Pi$ where $W$ and $H'$ are nonnegative matrices, $\Pi$ is a permutation matrix and $N$ is the noise, find a set $\mathcal{K}$ of $r$ indices such that $\tilde{X}(:,\mathcal{K}) \approx W$.*

In the following, we describe some algorithms for near-separable NMF; they are classified in two categories: algorithms based on self-dictionary and sparse regression (Section 3.2.1) and geometric algorithms (Section 3.2.2).

### 3.2.1 Self-Dictionary and Sparse Regression Framework

In the noiseless case, separable NMF can be formulated as follows

$$\min_{Y \in \mathbb{R}^{n \times n}} ||Y||_{\text{row},0} \quad \text{such that} \quad X = XY \text{ and } Y \geq 0, \tag{9}$$

where $||Y||_{\text{row},0}$ is the number of non-zero rows of $Y$. In fact, if all the entries of a row of $Y$ are equal to zero, then the corresponding column of $X$ is not needed to reconstruct the other columns of $X$. Therefore, minimizing the number of rows of $Y$ different from zero is equivalent to minimizing the number of columns of $X$ used to reconstruct all the other columns of $X$, which solves the separable NMF problem. In particular, given an optimal solution $Y^*$ of (9) and denoting $\mathcal{K} = \{i | Y^*(i,:) \neq 0\}$, we have $X = WY^*(\mathcal{K},:)$ where $W = X(:,\mathcal{K})$.

In the presence of noise, the constraints $X = XY$ are usually reformulated as $||X - XY|| \leq \delta$ for some $\delta > 0$ or put as a penalty $\lambda ||X - XY||$ in the objective function for some penalty parameter $\lambda > 0$. In [40, 41], $||Y||_{\text{row},0}$ is replaced using $\ell_1$-norm type relaxation:

$$||Y||_{q,1} = \sum_j ||Y(i,:)||_q,$$

14

where $q > 1$ so that $||Y||_{q,1}$ is convex and (9) becomes a convex optimization problem. Note that this idea is closely related to compressive sensing where $\ell_1$-norm relaxation is used to find the sparsest solution to an underdetermined linear system. This relaxation is exact given that the matrix involved in the linear system satisfies some incoherence properties. In separable NMF, the columns and rows of matrix $X$ are usually highly correlated hence it is not clear how to extend the results from the compressive sensing literature to this separable NMF model; see, e.g., the discussion in [90].

A potential problem in using convex relaxations of (9) is that it cannot distinguish duplicates of the columns of $W$. In fact, if a column of $W$ is present twice in the data matrix $X$, the corresponding rows of $Y$ can both be non-zero hence both columns of $W$ can potentially be extracted (this is because of the convexity and the symmetry of the objective function) –in [40], $k$-means is used as a pre-processing in order to remove duplicates. Moreover, although this model was successfully used to solve real-world problems, no robustness results were developed so far so it is not clear how this model behaves in the presence of noise (only asymptotic results were proved, that is, when the noise level goes to zero and when no duplicates are present [40]).

A rather different approach to enforce row sparsity was suggested in [9], and later improved in [54]. Row sparsity of $Y$ is enforced by (i) minimizing a weighted sum of the diagonal entries of $Y$ hence enforcing $\mathrm{diag}(Y)$ to be sparse (in fact, this is nothing but a weighted $\ell_1$ norm since $Y$ is nonnegative), and (ii) imposing all entries in a row of $Y$ to be smaller than the corresponding diagonal entry of $Y$ (we assume here that the columns of $X$ are normalized). The second condition implies that if $\mathrm{diag}(Y)$ is sparse then $Y$ is row sparse. The corresponding near-separable NMF model is:

$$\min_{Y \in \mathbb{R}^{n \times n}} p^T \mathrm{diag}(Y) \quad \text{such that} \quad ||X - XY||_1 \le \delta \quad \text{and} \quad 0 \le Y_{ij} \le Y_{ii} \le 1, \tag{10}$$

for some positive vector $p \in \mathbb{R}^n$ with distinct entries (this breaks the symmetry so that the model can distinguish duplicates). This model has been shown to be robust: defining the parameter[8] $\alpha$ as

$$\alpha(W) = \min_{1 \le j \le r} \min_{x \in \mathbb{R}_+^{r-1}} ||W(:,j) - W(:,\mathcal{J}_j)x||_1, \quad \text{where } \mathcal{J}_j = \{1,2,\ldots,r\}\backslash\{j\},$$

and for a near-separable matrix $\tilde{X} = W[I_r, H']\Pi + N$ (see above) with $\epsilon = \max_j ||N(:,j)||_1 \le \mathcal{O}\left(\frac{\alpha^2(W)}{r}\right)$, the model (10) can be used to identify the columns of $W$ with $\ell_1$ error proportional to $\mathcal{O}\left(\frac{r\epsilon}{\alpha(W)}\right)$, that is, the identified index set $\mathcal{K}$ satisfies $\max_j \min_{k \in \mathcal{K}} ||\tilde{X}(:,k) - W(:,j)||_1 \le \mathcal{O}\left(\frac{r\epsilon}{\alpha(W)}\right)$; see [54, Th.7] for more details.

Finally, a drawback of the approaches based on self-dictionary and sparse regression is that they are computationally expensive as they require to tackle optimization problems with $n^2$ variables.

### 3.2.2 Geometric Algorithms

Another class of near-separable algorithms are based on geometric insights and in particular on the fact that the columns of $W$ are the vertices of the convex hull of the normalized columns of $X$. The first geometric algorithms can be found in the remote sensing literature (they are referred to as endmember extraction algorithms or pure-pixel identification algorithms), see [90] for a historical note; and [8] for a comprehensive survey. Because of the large body of literature, we do not aim at surveying

---

[8]The larger the parameter $\alpha$ is, the less sensitive the data to noise. For example, it can be easily checked that $\epsilon = \max_j ||N(:,j)||_1 < \frac{\alpha}{2}$ is a necessary condition to being able to distinguish the columns of $W$ [49].

all algorithms but rather focus on a single algorithm which is particularly simple while being rather effective in practice: the successive projection algorithm (SPA). Moreover, the ideas behind SPA are at the heart of many geometric-based near-separable NMF algorithms (see below).

SPA looks for the vertices of the convex hull of the columns of the input data matrix $X$ and works as follows: at each step, it selects the column of $X$ with maximum $\ell_2$ norm and then updates $X$ by projecting each column onto the orthogonal complement of the extracted column; see Algorithm SPA. SPA is extremely fast as it can be implemented in $2pnr + \mathcal{O}(pr^2)$ operations, using the formula $||(I - uu^T)v||_2^2 = ||v||_2^2 - (u^Tv)^2$, for any $u, v \in \mathbb{R}^m$ with $||u||_2 = 1$ [55]. Moreover, if $r$ is unknown, it can be estimated using the norm of the residual $R$.

---

**Algorithm SPA** Successive Projection Algorithm [2]

---

**Input:** Near-separable matrix $\tilde{X} = W[I_r, H']\Pi + N$ where $W$ is full rank, $H' \geq 0$, the entries of each column of $H'$ sum to at most one, $\Pi$ is a permutation and $N$ is the noise, and the number $r$ of columns of $W$.

**Output:** Set of $r$ indices $\mathcal{K}$ such that $\tilde{X}(:, \mathcal{K}) \approx W$ (up to permutation).

1: Let $R = \tilde{X}$, $\mathcal{K} = \{\}$.
2: **for** $k = 1 : r$ **do**
3:     $p = \text{argmax}_j ||R_{:j}||_2$.
4:     $R = \left(I - \frac{R_{:p}R_{:p}^T}{||R_{:p}||_2^2}\right) R$.
5:     $\mathcal{K} = \mathcal{K} \cup \{p\}$.
6: **end for**

---

Let us prove the correctness of SPA in the noiseless case using induction, and assuming $W$ is full rank (this is a necessary and sufficient condition) and assuming the entries of each column of $H'$ sum to at most one (this can be achieved through normalization; see above). At the first step, SPA identifies a column of $W$ because the $\ell_2$ norm can only be maximized at a vertex of the convex hull of a set of points; see Figure 4 for an illustration. In fact, for all $1 \leq j \leq n$,

$$||X(:, j)||_2 = ||WH(:, j)||_2 \leq \sum_{k=1}^{r} H(k, j)||W(:, k)||_2 \leq \max_{1 \leq k \leq r} ||W(:, k)||_2.$$

The first inequality follows from the triangle inequality, and the second since $H(k, j) \geq 0$ and $\sum_k H(k, j) \leq 1$. Moreover, by strong convexity of the $\ell_2$ norm and the full rank assumption on $W$, the first inequality is strict unless $H(:, k)$ is a column of the identity matrix, that is, unless $X(:, j) = W(:, k)$ for some $k$. For the induction step, assume without loss of generality that SPA has extracted the first $\ell$ columns of $W$, and let $W_\ell = W(:, 1{:}\ell)$ and $P_{W_\ell}^{\perp}$ be the projection onto the orthogonal complement of the columns of $W_\ell$ so that $P_{W_\ell}^{\perp} W_\ell = 0$. We have, for all $1 \leq j \leq n$,

$$||P_{W_\ell}^{\perp}X(:, j)||_2 = ||P_{W_\ell}^{\perp}WH(:, j)||_2 \leq \sum_{k=1}^{r} H(k, j)||P_{W_\ell}^{\perp}W(:, k)||_2 \leq \max_{\ell+1 \leq k \leq r} ||P_{W_\ell}^{\perp}W(:, k)||_2,$$

where $P_{W_\ell}^{\perp}W(:, k) \neq 0$ for $\ell + 1 \leq k \leq r$ since $W$ is full rank. Hence, using the same reasoning as above, SPA will identify a column of $W$ not extracted yet, which concludes the proof.

Moreover, SPA is robust to noise: given a near-separable matrix $\tilde{X} = W[I_r, H']\Pi + N$ with $W$ full rank, $H'$ nonnegative with $||H'(:, j)||_1 \leq 1 \; \forall j$, and $\epsilon = \max_j ||N(:, j)||_2 \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{\sqrt{r}\kappa^2(W)}\right)$, SPA
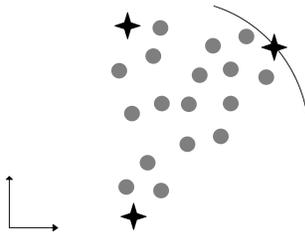
16

Figure 4: Illustration of SPA.

identifies the columns of $W$ up to $\ell_2$ error proportional to $\mathcal{O}\left(\epsilon\,\kappa^2(W)\right)$, where $\kappa(W) = \frac{\sigma_{\max}(W)}{\sigma_{\min}(W)}$ [55, Th.3]. These bounds can be improved using post-processing (see below) which reduces the error to $\mathcal{O}\left(\epsilon\,\kappa(W)\right)$ [3], or preconditioning which significantly increases the upper bound on the noise level, to $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{r\sqrt{r}}\right)$, and reduces the error to $\mathcal{O}\left(\epsilon\,\kappa(W)\right)$ [56].

It is interesting to note that SPA has been developed and used for rather different purposes in various fields:

- *Numerical linear algebra.* SPA is closely related to the modified Gram-Schmidt algorithm with column pivoting, used for example to solve linear least squares problems [15].

- *Chemistry* (and in particular spectroscopy). SPA is used for variable selection in spectroscopic multicomponent analysis; in fact, the name SPA comes from [2].

- *Hyperspectral imaging.* SPA is closely related to several endmember extraction algorithms; in particular N-FINDR [108] and its variants, the automatic target generation process (ATGP) [99], and the successive volume maximization algorithm (SVMAX) [21]; see the discussion in [90] for more details. The motivation behind all these approaches is to identify an index set $\mathcal{K}$ that maximizes the volume of the convex hull of the columns of $X(:,\mathcal{K})$. Note that most endmember extraction algorithms use an LDR (such as the SVD) as a pre-processing step for noise filtering, and SPA can be combined with an LDR to improve performance.

- *Text mining.* Arora et al. [3] proposed FastAnchorWords whose differences with SPA are that (i) the projection is made onto the affine hull of the columns extracted so far (instead of the linear span), and (ii) the index set extracted is refined using the following post-processing step: let $\mathcal{K}$ be the extracted index set by SPA, for each $k \in \mathcal{K}$:

  - Compute the projection $R$ of $X$ into the orthogonal complement of $X(:,\mathcal{K}\backslash\{k\})$.
  - Replace $k$ with the index corresponding to the column of $R$ with maximum $\ell_2$ norm.

- *Theoretical computer science.* SPA was proved to be a good heuristic to identify a subset of columns of a given matrix whose convex hull has maximum volume [19, 20] (in the sense that no polynomial-time algorithm can achieve better performance up to some logarithmic factors).

- *Sparse regression with self-dictionary.* SPA is closely related to orthogonal matching pursuit and can be interpreted as a greedy method to solve the sparse regression problem with self-dictionary (9); see [44] and the references therein.

Moreover, there exist many geometric algorithms which are variants of SPA, e.g., vertex component analysis (VCA) using linear functions instead of the $\ell_2$-norm [96], $\ell_p$-norm based pure pixel algorithm (TRI-P) using $p$-norms [1], FastSepNMF using strongly convex functions [55], the successive nonnegative projection algorithm (SNPA) [50] and the fast conical hull algorithm (XRAY) [77] using nonnegativity constraints for the projection step.

Further research on near-separable NMF includes the design of faster and/or provably more robust algorithms. In particular, there does not seem to exist an algorithm guaranteed to be robust for any matrix $W$ such that $\alpha(W) > 0$ and running in $\mathcal{O}(n)$ operations.

# 4 Connections with Problems in Mathematics and Computer Science

In this section, we briefly mention several connections of NMF with problems outside data mining and machine learning. The minimum $r$ such that an exact NMF of a nonnegative matrix $X$ exists is the nonnegative rank of $X$, denoted $\mathrm{rank}_+(X)$. More precisely, given $X \in \mathbb{R}_+^{p \times n}$, $\mathrm{rank}_+(X)$ is the minimum $r$ such that there exist $W \in \mathbb{R}_+^{p \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ with $X = WH$. The nonnegative rank has tight connections with several problems in mathematics and computer science:

- *Graph Theory.* Let $G(X) = (V_1 \cup V_2, E)$ be the bipartite graph induced by $X$ (that is, $(i, j) \in E \iff X_{ij} \neq 0$). The minimum biclique cover $\mathrm{bc}(G(X))$ of $G(X)$ is the minimum number of complete bipartite subgraphs needed to cover $G(X)$. It can be checked easily that for any $(W, H) \geq 0$ such that $X = WH = \sum_{k=1}^r W_{:k} H_{k:}$, we have

$$G(X) = \cup_{k=1}^r G(W_{:k} H_{k:}),$$

  where $G(W_{:k} H_{k:})$ are complete bipartite subgraphs hence $\mathrm{bc}(G(W_{:k} H_{k:})) = 1 \; \forall k$. Therefore,

$$\mathrm{bc}(G(X)) \leq \mathrm{rank}_+(X).$$

  This lower bound on the nonnegative rank is referred to as the rectangle covering bound [43].

- *Extended Formulations.* Given a polytope $P$, an extended formulation (or lift, or extension) is a higher dimensional polyhedron $Q$ and a linear projection $\pi$ such that $\pi(Q) = P$. When the polytope $P$ has exponentially many facets, finding extended formulations of polynomial size is of great importance since it allows to solve linear programs (LP) over $P$ in polynomial time. It turns out that the minimum number of facets of an extended formulation $Q$ of a polytope $P$ is equal to the nonnegative rank of its slack matrix [111], defined as $X(i, j) = a_i^T v_j - b_i$ where $v_j$ is the $j$th vertex of $P$ and $\{x \in \mathbb{R}^n \mid a_i^T x - b_i \geq 0\}$ its $i$th facet with $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, that is, $X$ is a facet-by-vertex matrix and $X(i, j)$ is the slack of the $j$th vertex with respect to $i$th facet; see the surveys [30, 69] and the references therein. These ideas can be generalized to approximate extended formulations, directly related to approximate factorizations (hence NMF) [13, 58].

- *Probability.* Let $X^{(k)} \in \{1, \dots, p\}$ and $Y^{(k)} \in \{1, \dots, n\}$ be two independent variables for each $1 \leq k \leq r$, and $P^{(k)}$ be the joint distribution with

$$P_{ij}^{(k)} = \mathbb{P}\left(X^{(k)} = i, Y^{(k)} = j\right) = \mathbb{P}\left(X^{(k)} = i\right) \mathbb{P}\left(Y^{(k)} = j\right).$$

Each distribution $P^{(k)}$ corresponds to a nonnegative rank-one matrix. Let us define the mixture $P$ of these $k$ independent distributions as follows:

– Choose the distribution $P^{(k)}$ with probability $\alpha_k$, where $\sum_{k=1}^{r} \alpha_k = 1$.
– Draw $X$ and $Y$ from the distribution $P^{(k)}$.

We have that $P = \sum_{k=1}^{r} \alpha_k P^{(k)}$ is the sum of $r$ rank-one nonnegative matrices. In practice, only $P$ is observed and computing its nonnegative rank and a corresponding factorization amounts to explaining the distribution $P$ with as few independent variables as possible; see [17] and the references therein.

- *Communication Complexity.* In its simplest variant, communication complexity addresses the following problem: Alice and Bob have to compute the following function

$$f : \{0,1\}^m \times \{0,1\}^n \mapsto \{0,1\} : (x,y) \mapsto f(x,y).$$

Alice only knows $x$ and Bob $y$, and the aim is to minimize the number of bits exchanged between Alice and Bob to compute $f$ exactly. Nondeterministic communication complexity (NCC) is a variant where Bob and Alice first receive a message before starting their communication; see [81, Ch.3] and the references therein for more details. The communication matrix $X \in \{0,1\}^{2^n \times 2^m}$ is equal to the function $f$ for all possible combinations of inputs. Yannakakis [111] showed that the NCC for computing $f$ is upper bounded by the logarithm of the nonnegative rank of the communication matrix (this result is closely related to the rectangle covering bound described above: in fact, $\lceil \log(\mathrm{bc}(G(X))) \rceil$ equals to the NCC of $f$).

- *Computational Geometry.* Computing the nonnegative rank is closely related to the problem of finding a polytope with minimum number of vertices nested between two given polytopes [53]. This is a well-known problem is computational geometry, referred to as the nested polytopes problem; see [31] and the references therein.

## 5   Conclusion

NMF is an easily interpretable linear dimensionality reduction technique for nonnegative data. It is a rather versatile technique with many applications, and brings together a broad range of researchers. In the context of 'Big Data' science, which becomes an increasingly important topic, we believe NMF has a bright future; see Figure 5 for an illustration of the number of publications related to NMF since the publication of the Lee and Seung paper [79].
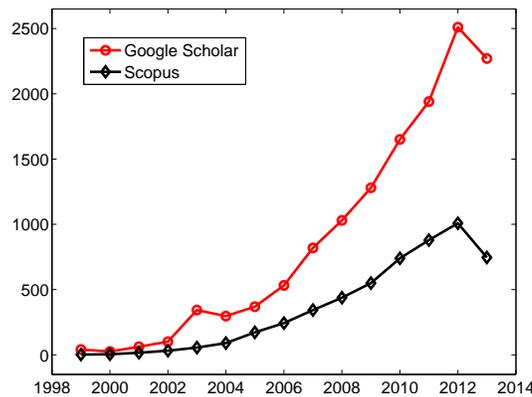
## Acknowledgment

Figure 5: Number of search results for papers containing either 'nonnegative matrix factorization' or 'nonnegative matrix factorization' on Google Scholar and Scopus (as of December 12, 2013).

# References

[1] Ambikapathi, A., Chan, T.H., Chi, C.Y., Keizer, K.: Hyperspectral data geometry based estimation of number of endmembers using p-norm based pure pixel identification. IEEE Trans. on Geoscience and Remote Sensing **51**(5), 2753–2769 (2013)

[2] Araújo, U., Saldanha, B., Galvão, R., Yoneyama, T., Chame, H., Visani, V.: The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. Chemometrics and Intelligent Laboratory Systems **57**(2), 65–73 (2001)

[3] Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., Zhu, M.: A practical algorithm for topic modeling with provable guarantees. In: Int. Conf. on Machine Learning (ICML '13), vol. 28, pp. 280–288 (2013)

[4] Arora, S., Ge, R., Kannan, R., Moitra, A.: Computing a nonnegative matrix factorization – provably. In: Proc. of the 44th Symp. on Theory of Computing (STOC '12), pp. 145–162 (2012)

[5] Badeau, R., Bertin, N., Vincent, E.: Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization. IEEE Trans. on Neural Networks **21**(12), 1869–1881 (2010)

[6] Berry, M., Browne, M., Langville, A., Pauca, V., Plemmons, R.: Algorithms and Applications for Approximate Nonnegative Matrix Factorization. Computational Statistics & Data Analysis **52**, 155–173 (2007)

[7] Bioucas-Dias, J., Nascimento, J.: Estimation of signal subspace on hyperspectral data. In: Remote Sensing, p. 59820L. International Society for Optics and Photonics (2005)

[8] Bioucas-Dias, J., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J.: Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing **5**(2), 354–379 (2012)

[9] Bittorf, V., Recht, B., Ré, E., Tropp, J.: Factoring nonnegative matrices with linear programs. In: Advances in Neural Information Processing Systems (NIPS '12), pp. 1223–1231 (2012)

[10] Blei, D.: Probabilistic topic models. Communications of the ACM **55**(4), 77–84 (2012)

[11] Boutsidis, C., Gallopoulos, E.: SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognition **41**, 1350–1362 (2008)

[12] Boutsidis, C., Mahoney, M., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: Proc. of the 20th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA '09), pp. 968–977 (2009)

[13] Braun, G., Fiorini, S., Pokutta, S., Steurer, D.: Approximation limits of linear programs (beyond hierarchies). In: Proc. of the 53rd Annual IEEE Symp. on Foundations of Computer Science (FOCS' 12), pp. 480–489 (2012)

[14] Bro, R.: Multi-way analysis in the food industry: Models, algorithms, and applications. Ph.D. thesis, University of Copenhagen (1998). `http://curis.ku.dk/ws/files/13035961/Rasmus_Bro.pdf`

[15] Businger, P., Golub, G.: Linear least squares solutions by householder transformations. Numerische Mathematik **7**, 269–276 (1965)

[16] Cai, D., He, X., Han, J., Huang, T.: Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. on Pattern Analysis and Machine Intelligence **33**(8), 1548–1560 (2011)

[17] Carlini, E., Rapallo, F.: Probability matrices, non-negative rank, and parameterization of mixture models. Linear Algebra and its Applications **433**, 424–432 (2010)

[18] Casalino, G., Del Buono, N., Mencar, C.: Subtractive clustering for seeding non-negative matrix factorizations. Information Sciences (257), 369–387 (2013)

[19] Çivril, A., Magdon-Ismail, M.: On selecting a maximum volume sub-matrix of a matrix and related problems. Theoretical Computer Science **410**(47-49), 4801–4811 (2009)

[20] Çivril, A., Magdon-Ismail, M.: Exponential inapproximability of selecting a maximum volume sub-matrix. Algorithmica **65**(1), 159–176 (2013)

[21] Chan, T.H., Ma, W.K., Ambikapathi, A., Chi, C.Y.: A simplex volume maximization framework for hyperspectral endmember extraction. IEEE Trans. on Geoscience and Remote Sensing **49**(11), 4177–4193 (2011)

[22] Chan, T.H., Ma, W.K., Chi, C.Y., Wang, Y.: A convex analysis framework for blind separation of non-negative sources. IEEE Trans. on Signal Processing **56**(10), 5120–5134 (2008)

[23] Chi, E., Kolda, T.: On tensors, sparsity, and nonnegative factorizations. SIAM J. on Matrix Analysis and Applications **33**(4), 1272–1299 (2012)

[24] Choi, S.: Algorithms for orthogonal nonnegative matrix factorization. In: Proc. of the Int. Joint Conf. on Neural Networks, pp. 1828–1832 (2008)

[25] Cichocki, A., Phan, A.: Fast local algorithms for large scale Nonnegative Matrix and Tensor Factorizations. IEICE Trans. on Fundamentals of Electronics **Vol. E92-A No.3**, 708–721 (2009)

[26] Cichocki, A., Zdunek, R., Amari, S.I.: Non-negative Matrix Factorization with Quasi-Newton Optimization. In: Lecture Notes in Artificial Intelligence, Springer, vol. 4029, pp. 870–879 (2006)

[27] Cichocki, A., Zdunek, R., Amari, S.I.: Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization. In: Lecture Notes in Computer Science, Vol. 4666, Springer, pp. 169–176 (2007)

[28] Cichocki, A., Zdunek, R., Phan, A., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley-Blackwell (2009)

[29] Comon, P.: Independent component analysis, A new concept? Signal Processing **36**, 287–314 (1994)

[30] Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR: A Quarterly Journal of Operations Research **10**(1), 1–48 (2010)

[31] Das, G., Joseph, D.: The Complexity of Minimum Convex Nested Polyhedra. In: Proc. of the 2nd Canadian Conf. on Computational Geometry, pp. 296–301 (1990)

[32] d'Aspremont, A., El Ghaoui, L., Jordan, M., Lanckriet, G.: A Direct Formulation for Sparse PCA Using Semidefinite Programming. SIAM Review **49**(3), 434–448 (2007)

[33] Daube-Witherspoon, M., Muehllehner, G.: An iterative image space reconstruction algorithm suitable for volume ECT. IEEE Trans. on Medical Imaging **5**, 61–66 (1986)

[34] Devarajan, K.: Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. PLoS Computational Biology **4**(7), e1000029 (2008)

[35] Ding, C., He, X., Simon, H.: On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In: SIAM Int. Conf. Data Mining (SDM'05), pp. 606–610 (2005)

[36] Ding, C., Li, T., Jordan, M.: Convex and semi-nonnegative matrix factorizations. IEEE Trans. on Pattern Analysis and Machine Intelligence **32**(1), 45–55 (2010)

[37] Ding, C., Li, T., Peng, W.: On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. Computational Statistics & Data Analysis **52**(8), 3913–3927 (2008)

[38] Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 126–135 (2006)

[39] Ding, W., Rohban, M., Ishwar, P., Saligrama, V.: Topic discovery through data dependent and random projections. In: Int. Conf. on Machine Learning (ICML '13), vol. 28, pp. 471–479 (2013)

[40] Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: Sparse modeling for finding representative objects. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '12) (2012)

[41] Esser, E., Moller, M., Osher, S., Sapiro, G., Xin, J.: A convex model for nonnegative matrix factorization and dimensionality reduction on physical space. IEEE Trans. on Image Processing **21**(7), 3239–3252 (2012)

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

[43] Fiorini, S., Kaibel, V., Pashkovich, K., Theis, D.: Combinatorial bounds on nonnegative rank and extended formulations. Discrete Mathematics **313**(1), 67–83 (2013)

[44] Fu, X., Ma, W.K., Chan, T.H., Bioucas-Dias, J., Iordache, M.D.: Greedy algorithms for pure pixels identification in hyperspectral unmixing: A multiple-measurement vector viewpoint. In: Proc. of 21st European Signal Processing Conf. (EUSIPCO '13) (2013)

[45] Gaussier, E., Goutte, C.: Relation between PLSA and NMF and implications. In: Proc. of the 28th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 601–602 (2005)

[46] Ge, R.: Provable algorithms for machine learning problems. Ph.D. thesis, Princeton University (2013). `http://dataspace.princeton.edu/jspui/bitstream/88435/dsp019k41zd62n/1/Ge_princeton_0181D_10819.pdf`

[47] Gillis, N.: Nonnegative matrix factorization: Complexity, algorithms and applications. Ph.D. thesis, Université catholique de Louvain (2011). `https://sites.google.com/site/nicolasgillis/`

[48] Gillis, N.: Sparse and unique nonnegative matrix factorization through data preprocessing. Journal of Machine Learning Research **13**(Nov), 3349–3386 (2012)

[49] Gillis, N.: Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. SIAM J. on Matrix Analysis and Applications **34**(3), 1189–1212 (2013)

[50] Gillis, N.: Successive nonnegative projection algorithm for robust nonnegative blind source separation (2013). arXiv:1310.7529

[51] Gillis, N., Glineur, F.: Using underapproximations for sparse nonnegative matrix factorization. Pattern Recognition **43**(4), 1676–1687 (2010)

[52] Gillis, N., Glineur, F.: Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Computation **24**(4), 1085–1105 (2012)

[53] Gillis, N., Glineur, F.: On the geometric interpretation of the nonnegative rank. Linear Algebra and its Applications **437**(11), 2685–2712 (2012)

[54] Gillis, N., Luce, R.: Robust near-separable nonnegative matrix factorization using linear optimization. Journal of Machine Learning Research (2014). to appear

[55] Gillis, N., Vavasis, S.: Fast and robust recursive algorithms for separable nonnegative matrix factorization. IEEE Trans. Pattern Anal. Mach. Intell. (2013). doi:10.1109/TPAMI.2013.226

[56] Gillis, N., Vavasis, S.: Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization (2013). arXiv:1310.2273

[57] Golub, G., Van Loan, C.: Matrix Computation, 3rd Edition. The Johns Hopkins University Press Baltimore (1996)

[58] Gouveia, J., Parrilo, P., Thomas, R.: Approximate cone factorizations and lifts of polytopes (2013). arXiv:1308.2162

[59] Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Operations Research Letters **26**, 127–136 (2000)

[60] Guan, N., Tao, D., Luo, Z., Yuan, B.: NeNMF: an optimal gradient method for nonnegative matrix factorization. IEEE Trans. on Signal Processing **60**(6), 2882–2898 (2012)

[61] Guillamet, D., Vitrià, J.: Non-negative matrix factorization for face recognition. In: Lecture Notes in Artificial Intelligence, pp. 336–344. Springer (2002)

[62] Han, J., Han, L., Neumann, M., Prasad, U.: On the rate of convergence of the image space reconstruction algorithm. Operators and Matrices **3**(1), 41–58 (2009)

[63] Ho, N.D.: Nonnegative matrix factorization - algorithms and applications. Ph.D. thesis, Université catholique de Louvain (2008)

[64] Hoyer, P.: Nonnegative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5**, 1457–1469 (2004)

[65] Hsieh, C.J., Dhillon, I.: Fast coordinate descent methods with variable selection for non-negative matrix factorization. In: Proc. of the 17th ACM SIGKDD int. conf. on Knowledge discovery and data mining, pp. 1064–1072 (2011)

[66] Huang, K., Sidiropoulos, N., Swami, A.: Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. IEEE Trans. on Signal Processing **62**(1), 211–224 (2014)

[67] Iordache, M.D., Bioucas-Dias, J., Plaza, A.: Total variation spatial regularization for sparse hyperspectral unmixing. IEEE Trans. on Geoscience and Remote Sensing **50**(11), 4484–4502 (2012)

[68] Jia, S., Qian, Y.: Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE Trans. on Geoscience and Remote Sensing **47**(1), 161–173 (2009)

[69] Kaibel, V.: Extended Formulations in Combinatorial Optimization. Optima **85**, 2–7 (2011)

[70] Kanagal, B., Sindhwani, V.: Rank selection in low-rank matrix approximations. In: Advances in Neural Information Processing Systems (NIPS '10) (2010)

[71] Ke, Q., Kanade, T.: Robust $L_1$ norm factorization in the presence of outliers and missing data by alternative convex programming. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '05), pp. 739–746 (2005)

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

[73] Kim, H., Park, H.: Non-negative Matrix Factorization Based on Alternating Non-negativity Constrained Least Squares and Active Set Method. SIAM J. on Matrix Analysis and Applications **30**(2), 713–730 (2008)

[74] Kim, J., He, Y., Park, H.: Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. Journal of Global Optimization (2013). doi:10.1007/s10898-013-0035-4

[75] Kim, J., Park, H.: Fast nonnegative matrix factorization: An active-set-like method and comparisons. SIAM J. on Scientific Computing **33**(6), 3261–3281 (2011)

[76] Kumar, A., Sindhwani, V.: Near-separable non-negative matrix factorization with $\ell_1$- and Bregman loss functions (2013). arXiv:1312.7167

[77] Kumar, A., Sindhwani, V., Kambadur, P.: Fast conical hull algorithms for near-separable non-negative matrix factorization. In: Int. Conf. on Machine Learning (ICML '13), vol. 28, pp. 231–239 (2013)

[78] Lawson, C., Hanson, R.: Solving Least Squares Problems. Prentice-Hall (1974)

[79] Lee, D., Seung, H.: Learning the Parts of Objects by Nonnegative Matrix Factorization. Nature **401**, 788–791 (1999)

[80] Lee, D., Seung, H.: Algorithms for Non-negative Matrix Factorization. In Advances in Neural Information Processing (NIPS '01) **13** (2001)

[81] Lee, T., Shraibman, A.: Lower bounds in communication complexity. Now Publishers Inc. (2009)

[82] Lefèvre, A.: Dictionary learning methods for single-channel source separations. Ph.D. thesis, Ecole Normale Supérieure de Cachan (2012)

[83] Li, L., Lebanon, G., Park, H.: Fast Bregman divergence NMF using Taylor expansion and coordinate descent. In: Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 307–315 (2012)

[84] Li, L., Zhang, Y.J.: FastNMF: highly efficient monotonic fixed-point nonnegative matrix factorization algorithm with good applicability. J. Electron. Imaging **Vol. 18**(033004) (2009)

[85] Li, T., Zhang, Y., Sindhwani, V.: A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In: Association of Computational Lingustics, pp. 244–252 (2009)

[86] Li, Y., Sima, D., Van Cauter, S., Croitor Sava, A., Himmelreich, U., Pi, Y., Van Huffel, S.: Hierarchical non-negative matrix factorization (hNMF): a tissue pattern differentiation method for glioblastoma multiforme diagnosis using MRSI. NMR in Biomedicine **26**(3), 307–319 (2013)

[87] Lin, C.J.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Trans. on Neural Networks **18**(6), 1589–1596 (2007)

[88] Lin, C.J.: Projected Gradient Methods for Nonnegative Matrix Factorization. Neural Computation **19**, 2756–2779 (2007)

[89] Liu, J., Liu, J., Wonka, P., Ye, J.: Sparse non-negative tensor factorization using columnwise coordinate descent. Pattern Recognition **45**(1), 649–656 (2012)

[90] Ma, W.K., Bioucas-Dias, J., Chan, T.H., Gillis, N., Gader, P., Plaza, A., Ambikapathi, A., Chi, C.Y.: A Signal Processing Perspective on Hyperspectral Unmixing. IEEE Signal Processing Magazine **31**(1), 67–81 (2014)

[91] Mahoney, M., Drineas, P.: CUR matrix decompositions for improved data analysis. Proc. of the National Academy of Sciences **106**(3), 697–702 (2009)

[92] Melville, P., Sindhwani, V.: Recommender systems. Encyclopedia of machine learning **1**, 829–838 (2010)

[93] Miao, L., Qi, H.: Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. IEEE Trans. on Geoscience and Remote Sensing **45**(3), 765–777 (2007)

[94] Moitra, A.: An almost optimal algorithm for computing nonnegative rank. In: Proc. of the 24th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA '13), pp. 1454–1464 (2013)

[95] Naanaa, W., Nuzillard, J.M.: Blind source separation of positive and partially correlated data. Signal Processing **85**(9), 1711–1722 (2005)

[96] Nascimento, J., Bioucas-Dias, J.: Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Trans. on Geoscience and Remote Sensing **43**(4), 898–910 (2005)

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

[98] Rapin, J., Bobin, J., Larue, A., Starck, J.L.: Sparse and non-negative BSS for noisy data. IEEE Trans. on Signal Processing **61**(22), 5620–5632 (2013)

[99] Ren, H., Chang, C.I.: Automatic spectral target recognition in hyperspectral imagery. IEEE Trans. on Aerospace and Electronic Systems **39**(4), 1232–1249 (2003)

[100] Sandler, R., Lindenbaum, M.: Nonnegative matrix factorization with earth mover's distance metric. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '09), pp. 1873–1880 (2009)

[101] Shahnaz, F., Berry, M., A., Pauca, V., Plemmons, R.: Document clustering using nonnegative matrix factorization. Information Processing and Management **42**, 373–386 (2006)

[102] Smaragdis, P., Févotte, C., Mysore, G., Mohammadiha, N., Hoffman, M.: A Unied View of Static and Dynamic Source Separation Using Non-Negative Factorizations. IEEE Signal Processing Magazine (2014)

[103] Takahashi, N., Hibi, R.: Global convergence of modified multiplicative updates for nonnegative matrix factorization. Computational Optimization and Applications (2013). doi:10.1007/s10589-013-9593-0

[104] Tan, V., Févotte, C.: Automatic relevance determination in nonnegative matrix factorization. In: Signal Processing with Adaptive Sparse Structured Representations (SPARS '09) (2009)

[105] Vavasis, S.: On the complexity of nonnegative matrix factorization. SIAM J. on Optimization **20**(3), 1364–1377 (2009)

[106] Wang, F., Li, T., Wang, X., Zhu, S., Ding, C.: Community discovery using nonnegative matrix factorization. Data Min. Knowl. Disc. **22**(3), 493–521 (2011)

[107] Wild, S., Curry, J., Dougherty, A.: Improving non-negative matrix factorizations through structured initialization. Pattern Recognition **37**(11), 2217–2232 (2004)

[108] Winter, M.: N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In: Proc. SPIE Conf. on Imaging Spectrometry V, pp. 266–275 (1999)

[109] Xue, Y., Tong, C., Chen, Y., Chen, W.S.: Clustering-based initialization for non-negative matrix factorization. Applied Mathematics and Computation **205**(2), 525–536 (2008)

[110] Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Trans. on Neural Networks **21**(5), 734–749 (2010)

[111] Yannakakis, M.: Expressing Combinatorial Optimization Problems by Linear Programs. Journal of Computer and System Sciences **43**(3), 441–466 (1991)

[112] Zdunek, R.: Initialization of nonnegative matrix factorization with vertices of convex polytope. In: Artificial Intelligence and Soft Computing, *Lecture Notes in Computer Science*, vol. 7267, pp. 448–455. Springer Berlin Heidelberg (2012)

[113] Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. Knowledge and Information Systems **8**(3), 374–384 (2005)