Université de Mons
Faculté Polytechnique
Mathématique et Recherche Opérationnelle

Low-Rank Matrix Factorizations with Volume-based Constraints and Regularizations

Olivier Vu Thanh

A thesis presented in partial fulfillment of the requirements for the degree of
Docteur en Sciences de l'Ingénieur et Technologies

Dissertation committee:

Prof. Nicolas Gillis	Université de Mons	Supervisor
Prof. Fabian Lecron	Université de Mons	Co-supervisor
Prof. Arnaud Vandaele	Université de Mons	Chair
Prof. Kejun Huang	University of Florida	
Prof. Clémence Prévost	University of Lille	
Prof. Matthieu Puigt	Université du Littoral Côte d'Opale	

Abstract

Low-rank matrix factorizations (LRMFs) are a class of linear models widely used in various fields such as machine learning, signal processing, and data analysis. These models approximate a matrix as the product of two smaller matrices, where the left matrix captures latent features—the most important components of the data—while the right matrix linearly decomposes the data based on these features. There are many ways to define what makes a component "important." Standard LRMFs, such as the truncated singular value decomposition, focus on minimizing the distance between the original matrix and its low-rank approximation. In this thesis, the notion of "importance" is closely linked to interpretability and uniqueness, which are key to obtaining reliable and meaningful results.

This thesis thus focuses on volume-based constraints and regularizations designed to enhance interpretability and uniqueness. We first introduce two new volume-constrained LRMFs designed to enhance these properties. The first assumes that data points are naturally bounded (e.g., movie ratings between 1 and 5 stars) and can be explained by convex combinations of features within the same bounds, allowing them to be interpreted in the same way as the data. The second model is more general, constraining the factors to belong to convex polytopes. Then, two variants of volume-regularized LRMFs are proposed. The first minimizes the volume of the latent features, encouraging them to cluster closely together, while the second maximizes the volume of the decompositions, promoting sparse representations. Across all these models, uniqueness is achieved under the core principle that the factors must be "sufficiently scattered" within their respective feasible sets.

Motivated by applications such as blind source separation (e.g., hyperspectral unmixing) and missing data imputation (e.g., in recommender systems), this thesis also proposes efficient algorithms that make these models scalable and practical for real-world applications.

Résumé

Les factorisations matricielles de faible rang (LRMFs) sont des modèles linéaires largement utilisés dans des domaines tels que l'apprentissage automatique, le traitement du signal et l'analyse de données. Ces modèles approchent une matrice en la décomposant en produit de deux matrices plus petites : la première capture les caractéristiques latentes, c'est-à-dire les composantes les plus importantes des données, tandis que la seconde décompose linéairement les données à partir de ces caractéristiques. Il existe cependant de nombreuses manières de définir ce qui rend une composante "importante". Les LRMFs classiques, comme la décomposition en valeurs singulières tronquée, se concentrent sur la minimisation de la distance entre la matrice originale et son approximation de faible rang. Dans cette thèse, l'importance d'une composante est étroitement déterminée par l'interprétabilité et l'unicité, des notions clés pour obtenir des résultats fiables et pertinents.

Cette thèse explore donc des contraintes et régularisations volumiques visant à renforcer l'interprétabilité et l'unicité. Dans un premier temps, nous introduisons deux nouvelles variantes de LRMFs à contraintes volumiques. La première suppose que les points du jeu de données sont naturellement bornés (ex: des films notés entre 1 et 5 étoiles) et peuvent être expliqués par des combinaisons convexes de caractéristiques bornées de la même manière, permettant ainsi de les interpréter comme les données. Le second modèle est plus général et contraint les facteurs à appartenir à des polytopes convexes. Par ailleurs, nous proposons deux variantes de LRMF avec régularisation volumique : la première minimise le volume des caractéristiques latentes, favorisant ainsi un rapprochement entre elles, tandis que la seconde maximise le volume des décompositions, encourageant des représentations parcimonieuses. Dans l'ensemble de ces modèles, l'unicité est assurée par le principe clé selon lequel les facteurs doivent être "suffisamment dispersés" dans leur ensemble de solutions possibles.

Motivée par des applications telles que la séparation de sources aveugles (par exemple, le démélange hyperspectral) et l'imputation de données manquantes (par exemple, dans les systèmes de recommandation), cette thèse propose également des algorithmes efficaces permettant à ces modèles d'être adaptés à des applications réelles.

Acknowledgements

I would like first to thank my PhD supervisor Nicolas Gillis, from whom I learned a lot, humanly and scientifically.

I am thankful to the colleagues I met, chronologically: Andersen, Hien for welcoming me, Fabian my co-supervisor, Arnaud for our discussions, Pierre, Nadisic мой товарищ, Christos my hearty laughing neighbor, Atharva the flawless, Seraghiti the early bird coffee lover[1], Subhayan the connoisseur[2], Barbarino the maidenless tarnished who should put these foolish ambitions to rest, Timothy the original Belgian, the goofy Florian and Amjad for his kindness.

Special thanks to Jule and Junior, the guardians of the Houdain park.

I also would like to thank the teachers from Grenoble INP who indirectly gave me the will to pursue a PhD.

I thank the jury members for agreeing to evaluate this thesis and for their useful comments.

Finally, thanks to Hélène for sailing with me.

I dedicate this thesis to Wallace.

[1] addict
[2] in debauchery

I acknowledge the support by the European Research Council (ERC Starting Grant, COLORAMAP, no 679515, and ERC consolidator Grant, eLinoR, no 101085607), by the Fonds de la Recherche Scientifique (F.R.S.) - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) under EOS Project no O005318F-RG47, by the Francqui Foundation, by the F.R.S.-FNRS under the Research Project T.0097.2 and under a FRIA PhD grant.

Contents

Abstract **3**

Contents **11**

Notation **14**

List of Figures **17**

List of Tables **19**

1 Introduction **21**
Motivations 21
Applications 22
Thesis outline and related publications 24
Open-source codes 27

2 Preliminaries **29**
2.1 Nonnegative Matrix Factorization (NMF) 29
2.2 Simplex-structured matrix factorization (SSMF) 30
2.3 Identifiability 32
2.3.1 Identifiability of NMF 32
2.3.2 Identifiability of SSMF 33
2.4 Brief summary of the thesis content 36

3 Bounded Simplex-Structured Matrix Factorization **37**
3.1 Motivation of BSSMF 38
3.2 Inertial block-coordinate descent algorithm for BSSMF 40
3.2.1 Proposed algorithm 41
3.2.2 Accelerating BSSMF algorithms via data centering 44
3.2.3 Convergence speed and effect of acceleration strategies on real data 46
3.3 Identifiability of BSSMF 48
3.4 Numerical experiments 54
3.4.1 Interpretability 54

3.4.2 Identifiability . . . 57
3.4.3 Robustness to overfitting . . . 59
3.5 Conclusion . . . 61

4 Identifiability of Polytopic Matrix Factorization 63
4.1 Polytopic Matrix Factorization . . . 64
4.2 Definitions and Properties . . . 64
4.3 Identifiability . . . 68
4.4 Examples of PMF . . . 69
4.4.1 Nonnegative Matrix Factorization (NMF) . . . 69
4.4.2 Factor-Bounded Matrix Factorization . . . 70
4.4.3 Bounded Simplex-Structured Matrix Factorization . . . 70
4.5 Conclusion . . . 71

5 Randomized Successive Projection Algorithm for Separable NMF 73
5.1 Successive Projection Algorithm . . . 74
5.2 Randomized SPA . . . 74
5.3 Numerical experiments . . . 76
5.4 Conclusion . . . 77

6 Minimum-Volume Nonnegative Matrix Factorization 81
6.1 Existing variants of MinVol NMF . . . 81
6.2 Identifiability of MinVol NMF . . . 83
6.3 Solving MinVol NMF with TITAN . . . 84
6.3.1 TITANized MinVol NMF . . . 84
6.3.2 Numerical Experiments . . . 89
6.4 Minimum-volume Nonnegative Matrix Completion . . . 92
6.4.1 Motivation . . . 93
6.4.2 Algorithms . . . 94
6.4.3 Experiments . . . 95
6.5 Identifiability of MinVol NMF with ℓ_1 penalty . . . 99
6.6 Conclusion . . . 99

7 Maximum-Volume Nonnegative Matrix Factorization 101
7.1 Motivation . . . 101
7.2 MaxVol NMF . . . 105
7.2.1 Identifiability of MaxVol NMF . . . 105
7.2.2 Behavior of MaxVol NMF . . . 106
7.3 Solving MaxVol NMF . . . 108
7.3.1 Adaptive accelerated gradient descent . . . 110
7.3.2 Alternating direction method of multipliers (ADMM) for the MaxvolMF problem . . . 112
7.3.3 Comparison of the two algorithms . . . 116
7.4 Normalized MaxVol NMF . . . 119
7.5 Solving Normalized MaxVol NMF . . . 121
7.6 Performance of Normalized MaxVol NMF on Hyperspectral Unmixing 122

7.7 Conclusion 124

8 Highlight of the contributions and discussions 129
Summary 129
Further research 131

Notation

$\mathbb{R}$	set of real numbers
$\mathbb{R}_*$	set of real nonzero numbers
$\mathbb{R}_+$	set of real nonnegative numbers
$\mathbb{R}^m$	set of real column vectors of dimension m
$\mathbb{R}^{m\times n}$	set of real matrices of dimension $m \times n$
x_i or $x(i)$	i-th entry of the vector x
$x(K)$	subvector x with indices in the set K
$A(i,:)$	i-th row of the matrix A
$A(:,j)$	j-th column of the matrix A
$A(i,j)$	entry of the matrix A indexed by (i,j)
$A(:,J)$	submatrix of A with column indices in the set J
$A^\top$	transpose of the matrix A
$A^{-\top}$	inverse of the transpose of the square matrix A
$A \circ B$	Hadamard product, that is $(A \circ B)(i,j) = A(i,j)B(i,j)$
e	vector of all ones of appropriate dimension
e_i	i-th canonical vector of appropriate dimension
J	matrix of all ones of appropriate dimension
$E_{i,j}$	matrix whose (i,j)-th element is equal to one and zero elsewhere, that is $e_i e_j^\top$ of appropriate dimension
$x \geq 0$	the vector x is entry-wise nonnegative
$A \geq 0$	the matrix A is entry-wise nonnegative
Δ^r	probability simplex, $\{x \in \mathbb{R}^r \mid x \geq 0, e^\top x = 1\}$
$\Delta^{r\times n}$	set of matrices whose columns lies in Δ^r
$\mathbb{S}^n$	set of symmetric $n \times n$ matrices, $\{X \in \mathbb{R}^{n\times n} \mid X = X^\top\}$
$\mathbb{S}^n_+$	set of symmetric positive semidefinite matrices, $\{X \in \mathbb{S}^n \mid X \succeq 0\}$
$\mathbb{S}^n_{++}$	set of symmetric positive definite matrices, $\{X \in \mathbb{S}^n \mid X \succ 0\}$
$\mathrm{cone}(A)$	conical hull of the columns of A, $\{y \mid y = Az, z \geq 0\}$
$\mathrm{conv}(A)$	convex hull of the columns of matrix $A \in \mathbb{R}^{m\times n}$, $\{y \mid y = Az, z \in \Delta^n\}$
$\mathrm{ext}(\mathcal{X})$	set of extreme points of the set $\mathcal{X}$
$\mathrm{bd}(\mathcal{X})$	boundary of the set $\mathcal{X}$
$\mathcal{X}^{*,g}$	polar of the set $\mathcal{X} \subset \mathbb{R}^r$ with respect to g, that is, $\{x \in \mathbb{R}^r \mid \langle x, y - g\rangle \geq 0, \text{ for all } y \in \mathcal{X}\}$
$\kappa(A)$	condition number of the matrix A
$\lvert S\rvert$	Cardinality of the set S, that is the number of elements in S
$\lVert x\rVert_0$	ℓ_0-"norm" of vector x, $\lvert\{i \mid x_i \neq 0\}\rvert$
$\lVert x\rVert_1$	ℓ_1-norm of vector $x \in \mathbb{R}^r$, $\sum_{i=1}^r \lvert x_i\rvert$
$\lVert x\rVert_2$	ℓ_2-norm of vector $x \in \mathbb{R}^r$, $\sqrt{\sum_{i=1}^r \lvert x_i\rvert^2}$
$\lVert A\rVert_F$	Frobenius norm of matrix $A \in \mathbb{R}^{m\times r}$, $\sqrt{\sum_{i=1}^m \sum_{j=1}^r A(i,j)^2}$
$\lVert A\rVert$	Spectral norm of matrix A, that is, its largest singular value

Acronyms

BSSMF	bounded simplex-structured matrix factorization	p. 37
CLRMF	constrained low-rank matrix factorization	p. 22
HU	hyperspectral unmixing	p. 22
MinVol	minimum-volume	p. 35
MinVol NMF	minimum volume matrix factorization	p. 81
MaxVol NMF	maximum volume matrix factorization	p. 101
MVIE	maximum-volume inscribed ellipsoid	p. 65
NMF	nonnegative matrix factorization	p. 29
ONMF	orthogonal matrix factorization	p. 119
PMF	polytopic matrix factorization	p. 63
RandSPA	randomized successive matrix factorization	p. 74
SPA	successive projection algorithm	p. 73
SSC	sufficiently scattered conditions	p. 33
SSMF	simplex-structured matrix factorization	p. 30
TITAN	inerTial block majorIzation minimization framework for non-smooth non-convex opTimizAtioN	p. 41
VCA	vertex component analysis	p. 73

List of Figures

2.1 Geometric interpretation of Exact NMF 30
2.2 Geometric interpretation of Exact SSMF 31
2.3 Illustration of the SSC in three dimensions 34

3.1 Influence of centering the data on the cost function topology regarding H via a small example 45
3.2 Evolution of the training error for ml-1m and MNIST 47
3.3 Geometric interpretation of BSSMF for Example 3.1 52
3.4 Reshaped columns of the basis matrix W for $r = 10$ for MNIST 55
3.5 Decomposition of an eight by BSSMF with $r = 10$. 56
3.6 Ratio on satisfying a necessary condition for SSC1 58
3.7 Boxplots of the average MRSA and subspace angle between the true W and the estimated W 60

4.1 Small example showing how [PMF.SSC1] can be satisfied without [PMF.SSC2] being satisfied 66
4.2 Visualization of how can $\binom{W}{1-W}^\top$ satisfy [SSC1] while $W^\top$ does not satisfy [PMF.SSC1] 72

5.1 Abundance maps in false color from the unmixing of hyperspectral images. 78
5.2 Average best reconstruction error on Samson 79

6.1 Evolution w.r.t. time of the error for different datasets 90
6.2 Approximation of the ℓ_0 and ℓ_1 norm via $f_\delta(x)$ 94
6.3 Average RMSE according to the rank r and to the percentage of missing values 97
6.4 Average angle according to the rank r and to the percentage of missing values 98
6.5 Average RMSE according to the noise level and to the percentage of missing values 98

7.1 Abundance maps and endmembers for MinVol on Samson 103
7.2 Abundance maps and endmembers for MinVol on Moffett 104

7.3 Abundance maps of MaxVol NMF on Samson, depending on λ. 109
7.4 ADMM on synthetic dataset with $\epsilon = 10^{-3}$ 117
7.5 Comparison of algorithms for MaxVol NMF on various datasets 118
7.6 Range of the logdet depending on r and δ 120
7.7 Abundance maps and endmembers by Normalized MaxVol NMF on Moffett 122
7.8 Abundance maps and endmembers by Normalized MaxVol NMF on Samson 123
7.9 Abundance maps and endmembers by Normalized MaxVol NMF on Samson with $r = 6$ 124
7.10 Abundance maps grouped by endmember varieties and endmembers by Normalized MaxVol NMF with $r = 6$ on Samson 125
7.11 Abundance maps and endmembers by Normalized MaxVol NMF on Urban depending on r 126
7.12 Abundance maps and endmembers by Normalized MaxVol NMF with $r = 4$ on Jasper 127
7.13 Abundance maps and endmembers by Normalized MaxVol NMF with $r = 5$ on Jasper 127

List of Tables

3.1 RMSE on the test set for ml-1m 46
3.2 Computation time of Algorithm 3.1 in the experiment settings of Figure 3.2a depending on the number of used threads. 48
3.3 Computation time of Algorithm 3.1 in the experiment settings of Figure 3.2b depending on the number of used threads. 48
3.4 RMSE on the test set according to r, averaged $\pm$ standard deviation on 10 runs on ml-1m 61
3.5 RMSE on the test set according to r, averaged $\pm$ standard deviation on 10 runs on ml-100k 62

5.1 Summary of the datasets, for which $X \in \mathbb{R}^{m \times n}$ 76
5.2 Relative reconstruction error $\|X - WH\|_F / \|X\|_F$ in percent. 77

6.1 Datasets used in our experiments and their respective dimensions . . . 91
6.2 TITANized MinVol's lead time 91
6.3 Ranking depending on the algorithm and the kind of data set 91
6.4 Updates for W according to the model 95
6.5 Updates for H according to the model 96

7.1 Norms of the spectral signature of the water retrieved by MinVol for the Samson dataset 102
7.2 Average time per run on synthetic datasets 117

Chapter 1

Introduction

Motivations

The objective of machine learning is mainly to predict, classify or analyze data. This is usually done by using an algorithm that recognizes common and useful features in the data, according to a model. Compared to data-driven approaches, model-based approaches require more understanding of the data, but less amount of data during the learning. Particularly, linear models are interesting for their simplicity and interpretability. Consider some data stored in a matrix $X \in \mathbb{R}^{m \times n}$ where m represents the dimension of a sample and n the number the samples. A general linear model assumes that X can be written as $X = WH + N$, where $W \in \mathbb{R}^{m \times r}$ can be interpreted as a basis matrix with each column of W representing a feature, $H \in \mathbb{R}^{r \times n}$ can be interpreted as a decomposition of X into the W basis, and $N \in \mathbb{R}^{m \times n}$ is noise and model misfit. Take the j-th sample $X(:, j)$, it can be approximated by

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

In other words, each sample can be approximated by a weighted sum of features. The features are stored in W and the weights are stored in H. This simple, yet powerful, data representation technique is applied in many domains, e.g., facial feature extraction [62], document clustering [35], blind source separation [72, 83], data fusion [84], demosaicing [1], community detection [88], gene expression analysis [110], in situ calibration of sensors [104], and recommender systems [87]. When $r < \text{rank}(X)$, we refer to such models as low-rank matrix approximations. Low-rank matrix approximations/factorizations are linear dimension reduction techniques, that have recently emerged as very efficient models for unsupervised learning; see, e.g., [94, 95] and the references therein. The most notable example is principal component analysis (PCA), which can be solved efficiently via the Singular Value Decomposition (SVD). In the last 20 years, many new more sophisticated models have been proposed, such as sparse PCA that requires one of the factors to be sparse to improve interpretability [25], robust PCA to handle gross corruption and outliers [17, 18], and low-rank

matrix completion, also known as PCA with missing data, to handle missing entries in the input matrix [57]. The low-rank assumption supposes that there is redundancy in the data that can be explained linearly. Typically, the factors W and H are learned by minimizing an objective function. Different objective functions will promote different behaviors. The main objective function used in this thesis is the Frobenius norm, that is, $\|X - WH\|_F^2 = \sum_{(i,j)}(X(i,j) - W(i,:)H(:,j))^2$. For the Frobenius norm, the best rank r matrix approximation is given by the truncated SVD. This result is also known as the Eckart-Young theorem [29]. Depending on the application, the data and the goal at hand (e.g., clustering, denoising, feature extraction), additional structures/constraints on the factors W and/or H, such as sparsity, nonnegativity and statistical independence to name a few, are more or less relevant in order to favor specific structures. We then talk of a Constrained Low-Rank Matrix Factorization (CLRMF) . In this thesis, we particularly focus on CLRMFs that encourage uniqueness, that is, a unique retrieval of W and H. Uniqueness is also called identifiability. More details on identifiability are given in Section 2.3. Identifiability is useful in applications where the true underlying features and decomposition are desired, like in hyperspectral unmixing for instance where we aim at recovering the true materials present in the image along with their abundances in each pixel; see below for more details.

Applications

CLRMF is a very generic model and can be used in many applications. It can be used for, but it is not limited to, data imputation, noise reduction, data visualization and cluster analysis. Here, we mention two applications that will be often used in this thesis.

- **Hyperspectral Unmixing (HU)** Light can be made of several electromagnetic waves that include radio waves, microwaves, infrared, visible light, ultraviolet, X-rays, and gamma rays. When light hits a material, this material absorbs an amount of the light, depending on the wavelengths of the electromagnetic spectrum. Some of the light is also reflected. When a white light hits a banana, we see the banana as being yellow because it absorbed the colors in the visible light spectrum except at the wavelengths corresponding to yellow. Even if we cannot see it, this phenomenon also happens outside of the visible light spectrum, providing very rich information. Each material has a unique spectral signature, which refers to how much light the material reflects at different wavelengths of the electromagnetic spectrum. In other words, a spectral signature is a pattern that shows how the reflectance of a material changes across various wavelengths, making it possible to identify different materials based on their reflectance behavior. A hyperspectral data cube of size $a \times b \times m$ contains the measured spectral reflectance in m bandwidths of a $a \times b$ sized pixelated area. The spatial information can be vectorized by horizontally concatenating each pixel. Thus, a matrix $X \in \mathbb{R}_+^{m \times n}$ is obtained where $n = a \times b$ is the number of

pixels. Due to physical constraints, satellites measuring the reflectance with a high spectral resolution have to compromise with the spatial resolution. Hence, a pixel can correspond to an area of several square meters. It is then possible that several materials are present in one pixel. HU consists in identifying the spectral signature of the materials present in the area, also called endmembers, as well as their abundance in each pixel. If we assume that the mixing process in a pixel is linear, HU can be performed with CLRMF. Properly doing so, the k-th column of W should contain the spectral signature of the k-th endmember, and $H(k, j)$ should contain the abundance of the k-th endmember in the j-th pixel. As previously said, identifiability is then a key feature in HU. A practitioner wants to retrieve the true materials present in the area, as well as their true abundances. HU is discussed in Chapters 5 to 7.

- **Matrix Completion for Recommender Systems** In some applications, either due to data corruption or simply due to missing measurements, it is possible that the data matrix X is incomplete. This is the case in recommender systems for instance. Consider a movie-user rating data matrix $X \in \mathbb{R}_+^{m\times n}$, where the entry $X(i, j)$ is the rating that the j-th user gave to the i-th movie. Obviously, X has some missing entries because all the users have not watched and rated all the movies. Let us assume that we have a way to estimate the missing values. It is then possible to recommend a movie to a user if, according to the estimation, this user should give a good rating to this movie. One of the most standard way to impute the missing entries is to assume that the hypothetical full matrix can be approximated by a low-rank matrix. Let us assume that we fix[1] the rank of the estimation to r. Call $M \in \{0, 1\}^{m\times n}$ the binary matrix[2] of observed entries, where $M(i, j) = 1$ if $X(i, j)$ is known, and $M(i, j) = 0$ otherwise. Let us minimize the fitting error

$$\|M \circ (X - WH)\|_F^2 = \sum_{(i,j),M(i,j)=1} (X(i, j) - W(i, :)H(:, j))^2$$

with respect to $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$, where $\circ$ is the Hadamard product. If $X(i, j)$ is unknown, it can be estimated just by computing $W(i, :)H(:, j)$. Typically, other constraints are imposed on the factors W and H in order to avoid over-fitting and improve the imputation. Matrix completion is discussed in Chapters 3 and 6.

[1] Some CLRMFs for missing data completion do not need to fix the rank [14]. We just make this assumption here for the sake of simplicity.

[2] This is a particular case of $M \in [0, 1]^{m\times n}$ being a weight matrix, where the weight $M(i, j)$ between 0 and 1 indicates how much the (i, j)-th entry can be trusted. $M(i, j) = 0$ means that you do not trust the value $X(i, j)$ and $M(i, j) = 1$ means that you trust the value $X(i, j)$.

Thesis outline and related publications

The aim of this thesis is fourfold:

1. create new interpretable and identifiable matrix factorization models,
2. improve existing matrix factorization models for some specific applications,
3. develop fast algorithms for these models, and
4. apply these models and algorithms on data sets and compare to the state of the art.

This thesis is structured as follows:

Chapter 2. Preliminaries

In this chapter, we introduce some background on Nonnegative Matrix Factorization, Simplex-Structured Matrix Factorization and identifiability, often needed through the thesis.

Chapter 3. Bounded Simplex-Structured Matrix Factorization

In this chapter, we propose a new low-rank matrix factorization model dubbed bounded simplex-structured matrix factorization (BSSMF). Given an input matrix X and a factorization rank r, BSSMF looks for a matrix W with r columns and a matrix H with r rows such that $X \approx WH$ where the entries in each column of W are bounded, that is, they belong to given intervals, and the columns of H belong to the probability simplex, that is, H is column stochastic. BSSMF generalizes nonnegative matrix factorization (NMF), and simplex-structured matrix factorization (SSMF). BSSMF is particularly well suited when the entries of the input matrix X belong to a given interval; for example when the rows of X represent images, or X is a rating matrix such as in the Netflix and MovieLens datasets where the entries of X belong to the interval $[1, 5]$. The simplex-structured matrix H not only leads to an easily understandable decomposition providing a soft clustering of the columns of X, but implies that the entries of each column of WH belong to the same intervals as the columns of W. In this chapter, we first propose a fast algorithm for BSSMF, even in the presence of missing data in X. Then we provide identifiability conditions for BSSMF, that is, we provide conditions under which BSSMF admits a unique decomposition, up to trivial ambiguities. Finally, we illustrate the effectiveness of BSSMF on two applications: extraction of features in a set of images, and the matrix completion problem for recommender systems.

The content of this chapter is mainly extracted from
[101] Vu Thanh, O., Gillis, N. & Lecron, F. *Bounded Simplex-Structured Matrix Factorization* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2022), 9062–9066
[102] Vu Thanh, O., Gillis, N. & Lecron, F. Bounded Simplex-Structured Matrix Factorization: Algorithms, Identifiability and Applications. *IEEE Transactions on Signal Processing* **71,** 2434–2447 (2023).

Chapter 4. Identifiability of Polytopic Matrix Factorization

Polytopic matrix factorization (PMF) decomposes a given matrix as the product of two factors where the rows of the first factor belong to a given convex polytope and the columns of the second factor belong to another given convex polytope. In this chapter we show that if the polytopes have certain invariant properties, and that if the rows of the first factor and the columns of the second factor are sufficiently scattered within their corresponding polytope, then this PMF is identifiable, that is, the factors are unique up to a signed permutation. The PMF framework is quite general, as it recovers other known structured matrix factorization models, and is highly customizable depending on the application. Hence, our result provides sufficient conditions that guarantee the identifiability of a large class of structured matrix factorization models.

The content of this chapter is mainly extracted from
[99] Vu Thanh, O. & Gillis, N. *Identifiability of Polytopic Matrix Factorization* in *2023 31st European Signal Processing Conference (EUSIPCO)* (2023), 1290–1294.

Chapter 5. Randomized Successive Projection Algorithm for Separable NMF

The successive projection algorithm (SPA) is a widely used algorithm for nonnegative matrix factorization (NMF) under the separability assumption. Separability assumes that the cone of W should be equal to the cone of the data X. In hyperspectral unmixing, that is, the extraction of materials in a hyperspectral image, separability is equivalent to the pure-pixel assumption and states that for each material present in the image there exists at least one pixel composed of only this material. SPA is fast and provably robust to noise, but is not robust to outliers. Also, it is deterministic, so for a given setting it always produces the same solution. Yet, it has been shown empirically that the non-deterministic algorithm vertex component analysis (VCA), when run sufficiently many times, often produces at least one solution that is better than the solution of SPA. In this chapter, we combine the best of both worlds and introduce a randomized version of SPA dubbed RandSPA, that produces potentially different results at each run. It can be run several times to keep the best solution, and it is still provably robust to noise. Experiments on the unmixing of hyperspectral

images show that the best solution among several runs of RandSPA is generally better that the solution of vanilla SPA.

The content of this chapter is mainly extracted from
[103] Vu Thanh, O., Nadisic, N. & Gillis, N. *Randomized Successive Projection Algorithm* in *GRETSI'22, XXVIIIème Colloque Francophone de Traitement du Signal et des Images* (2022).

Chapter 6. Minimum-Volume Nonnegative Matrix Factorization

Nonnegative matrix factorization with the minimum volume criterion (MinVol NMF) guarantees that, under some mild and realistic conditions, the factorization has an essentially unique solution. This result has been successfully leveraged in many applications, including topic modeling, hyperspectral image unmixing, and audio source separation. In this chapter, we propose a fast algorithm to solve MinVol NMF which is based on a recently introduced block majorization-minimization framework with extrapolation steps. We illustrate the effectiveness of our new algorithm compared to the state of the art on several real hyperspectral images and document datasets. We also focus on the use of the minimum volume criterion on the task of nonnegative data imputation, which, up to our knowledge, has never been explored before. The particular choice of the MinVol regularization is justified by its interesting identifiability property and by its link with the nuclear norm. We show experimentally that MinVol NMF is a relevant model for nonnegative data recovery, especially when the recovery of a unique embedding is desired. Additionally, we introduce a new version of MinVol NMF that outperforms vanilla MinVol for data recovery.

The content of this chapter is mainly extracted from
[98] Vu Thanh, O., Ang, A., Gillis, N. & Hien, L. T. K. *Inertial majorization-minimization algorithm for minimum-volume NMF* in *European Signal Processing Conference (EUSIPCO)* (2021), 1065–1069
[100] Vu Thanh, O. & Gillis, N. *Minimum-Volume Nonnegative Matrix Completion* in *European Signal Processing Conference (EUSIPCO)* (2024).

Chapter 7. Maximum-Volume Nonnegative Matrix Factorization

Nonnegative matrix factorization with a maximum volume criterion (MaxVol NMF) is an identifiable regularized low-rank model that has not been studied as much as its counterpart minimum-volume NMF (MinVol NMF). Given a matrix dataset X, MaxVol NMF consists in finding two nonnegative low-rank factors, W and H, such that their product approximates X while the volume spanned by the origin and the

rows of H is as large as possible. This MaxVol criterion, combined with nonnegativity, will incite H to be sparser. In MinVol NMF, the volume criterion is on W and should be minimized. In the exact case, that is, $X = WH$, we show that MinVol NMF is equivalent to MaxVol NMF. Moreover, we show that MaxVol NMF behaves rather differently than MinVol NMF in the presence of noise, especially when the penalty on the volume criterion is increased. We also show how MaxVol NMF creates a continuum between NMF and orthogonal NMF with even clusters. We propose several algorithms to solve MaxVol NMF, which we apply on real datasets. Finally, we introduce the "normalized" variant of MaxVol NMF which exhibits better results than MinVol NMF and MaxVol NMF on Hyperspectral Unmixing (HU).

Open-source codes

All the algorithms developed in this thesis are available online along with the data and (most of) the test scripts necessary to reproduce our experiments: `https://gitlab.com/vuthanho`

Chapter 2

Preliminaries

Radiohead - Karma Police

2.1 Nonnegative Matrix Factorization (NMF)

We say that a matrix is nonnegative if all its elements are larger or equal to zero. In the remaining of this thesis, $X \geq 0$ means that $X(i,j) \geq 0$ for all (i,j). Nonnegative matrix factorization (NMF) , popularized by Lee and Seung [62], is a linear dimensionality reduction technique that has become a standard tool to extract latent structures in nonnegative data. Given an input matrix $X \in \mathbb{R}^{m \times n}$ and a factorization rank $r < \min(m,n)$, NMF consists in finding two factors $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that $X \approx WH$. Columns of X are called data points, and if H is column-stochastic then the columns of W can be seen as the vertices of a convex hull containing the data points; see Section 2.2. Applications of NMF include feature extraction in images, topic modeling, audio source separation, chemometrics, or blind hyperspectral unmixing (HU), see for example [20, 33, 34, 41] and the references therein.

Let us define the Exact NMF and NMF problems.

Definition 2.1 (Exact NMF) *Given a nonnegative matrix $X \in \mathbb{R}_+^{m \times n}$, an exact NMF of size r consists in finding two matrices $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that $X = WH$. The smallest r such that you can find an exact NMF of X is called the nonnegative rank of X and is noted* $\operatorname{rank}_+(X)$.

Definition 2.2 (NMF) *Given a matrix $X \in \mathbb{R}^{m \times n}$, finding its NMF of size r consists in solving*

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \|X - WH\|_F^2 \\ \text{subject to} \quad & W \in \mathbb{R}_+^{m \times r}, \\ & H \in \mathbb{R}_+^{r \times n}. \end{aligned} \tag{2.1}$$

Note that we fixed our definition with the Frobenius norm because it is the only cost function used as a reconstruction error in this thesis. Nonetheless, other cost functions could be used, such as the beta-divergences [31].

When a data matrix is nonnegative, it makes sense to use NMF in order to decompose it with features that are also nonnegative, and in an additive way. The main advantage of NMF is that the nonnegativity constraints on the factors W and H lead to an easily interpretable part-based decomposition [62].

Geometric interpretation of NMF In the exact case, an NMF of size r is equivalent to finding a cone with r rays in the nonnegative orthant that contains all the data points $X(:, j)$. The columns of the matrix W of the corresponding NMF are the rays that generated this polyhedral cone. Consider an NMF $X = WH$. For every j, $X(:, j) = WH(:, j)$. Since $H(:, j) \geq 0$ for all j, we have $\text{cone}(X) \subseteq \text{cone}(W)$ by definition of a cone[1]; see Figure 2.1 for a 3D example.

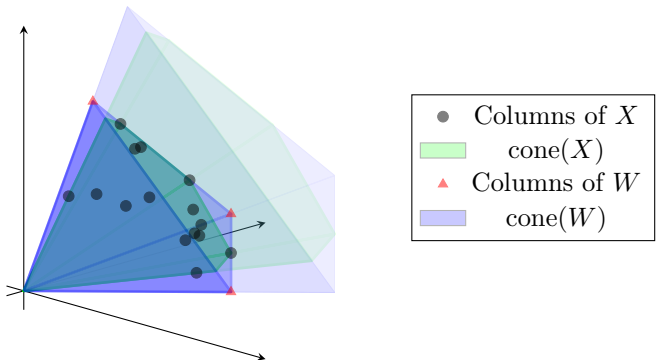


Figure 2.1: Geometric interpretation of Exact NMF with $r = 3$

2.2 Simplex-structured matrix factorization (SSMF)

A key set that will be used through the thesis is the probability simplex, that will allow us to define the SSMF problem.

Definition 2.3 (Probability simplex) *We denote Δ^r the probability simplex, that is, the set*

$$\{x \in \mathbb{R}^r \mid x \geq 0, e^\top x = 1\},$$

where e is the vector of all ones of appropriate dimension.

We then also denote $\Delta^{r \times n}$ the set of matrices of size $r \times n$ such that all their columns lie in Δ^r, that is,

$$\{X \in \mathbb{R}^{r \times n} \mid X(:, j) \in \Delta^r \text{ for all } j\}.$$

Definition 2.4 (Exact SSMF) *Given a matrix $X \in \mathbb{R}^{m \times n}$, an exact SSMF of size r consists in finding two matrices $W \in \mathbb{R}^{m \times r}$ and $H \in \Delta^{r \times n}$ such that $X = WH$.*

[1] Equality is equivalent under the so-called *separability assumption*; see Chapter 5

Definition 2.5 (SSMF) *Given a matrix $X \in \mathbb{R}^{m \times n}$, finding its SSMF of size r consists in solving*

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \|X - WH\|_F^2 \\ \text{subject to} \quad & W \in \mathbb{R}^{m \times r}, \\ & H \in \Delta^{r \times n}. \end{aligned} \tag{2.2}$$

With SSMF, each data point $X(:, j)$ has to be explained through a convex combination of some features. Due to that, SSMF is quite useful for providing a soft clustering decomposition of the data. In recommender systems for instance, SSMF could provide this kind of interpretation: "This user is behaving 80% like this typical user and 20% like this other typical user".

Geometric interpretation of SSMF In the exact case, an SSMF of size r is equivalent to finding a convex hull with r vertices that contains all the data points $X(:, j)$. The columns of the matrix W of the corresponding SSMF are then the vertices of this convex hull. Consider an SSMF $X = WH$. For every j, $X(:, j) = WH(:, j)$. Since $H(:, j) \in \Delta^r$ for all j, we have $\text{conv}(X) \subseteq \text{conv}(W)$ by definition of $\text{conv}(W)$; see Figure 2.2 for a 3D example.

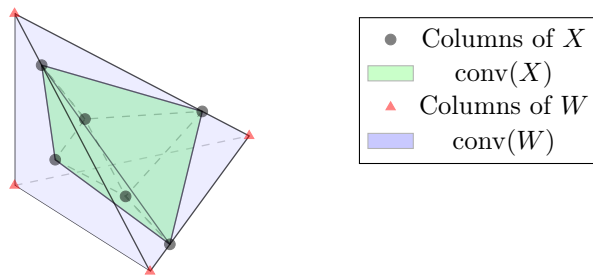


Figure 2.2: Geometric interpretation of Exact SSMF with $r = 4$ and $n = 6$

2.3 Identifiability

Let us first define a factorization model.

Definition 2.6 (Factorization model) *Given a matrix $X \in \mathbb{R}^{m \times n}$, and an integer $r \leq \min(m, n)$, a factorization model is an optimization model of the form*

$$\begin{aligned} \min_{W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n}} & g(W, H) \\ \text{such that } & X = WH, \\ & W \in \Omega_W \text{ and } H \in \Omega_H, \end{aligned} \tag{2.3}$$

where $g(W, H)$ is some criterion, and Ω_W and Ω_H are the feasible sets for W and H, respectively.

Let us define the identifiability of a factorization model, and essential uniqueness of a pair (W, H).

Definition 2.7 (Identifiability / Essential uniqueness) *Let $X \in \mathbb{R}^{m \times n}$, and $r \leq \min(m, n)$ be an integer. Let (W, H) be a solution to a given factorization model* (2.3)*. The pair (W, H) is essentially unique for the factorization model* (2.3) *of matrix X if and only if any other pair $(W', H') \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ that solves the factorization model* (2.3) *satisfies, for all k,*

$$W'(:, k) = \alpha_k W(:, \pi(k))$$

and

$$H'(k, :) = \alpha_k^{-1} H(\pi(k), :),$$

where π is a permutation of $\{1, 2, \ldots, r\}$, and $\alpha_k \neq 0$ for all k. In other terms, (W', H') can only be obtained as a permutation and scaling of (W, H). In that case, the factorization model is said to be identifiable for the matrix X.

A key question in theory and practice is to determine conditions on X, g, Ω_W and Ω_H that lead to identifiable factorization models; see, e.g., [34, 58] for discussions. This will be a major topic of this thesis.

2.3.1 Identifiability of NMF

NMF is not essentially unique in general. However, as opposed to SSMF (see Section 2.3.2), NMF decompositions can be identifiable without the use of additional requirements. The first identifiability result was proposed in [27]. Their conditions, based on separability, are quite strong. In the context of nonnegative source separation, [78] proposed some necessary conditions for the uniqueness of the solution. One of the most relaxed sufficient condition for identifiability is based on the Sufficiently scattered condition (SSC).

Theorem 2.1 *[54, Theorem 4] If $W^\top \in \mathbb{R}^{r \times m}$ and $H \in \mathbb{R}^{r \times n}$ are sufficiently scattered, then the Exact NMF (W, H) of $X = WH$ of size $r = \operatorname{rank}(X)$ is essentially unique.*

The SSC is defined as follows.

Definition 2.8 (Sufficiently scattered condition) *The matrix $H \in \mathbb{R}_+^{r\times n}$ is sufficiently scattered if the following two conditions are satisfied:*

[SSC1] $\mathcal{C} = \{x \in \mathbb{R}_+^r \mid e^\top x \geq \sqrt{r-1}\|x\|_2\} \subseteq \text{cone}(H)$.

[SSC2] There does not exist any orthogonal matrix Q such that $\text{cone}(H) \subseteq \text{cone}(Q)$, *except for permutation matrices.*

Lemma 2.1 *The dual cone of $\mathcal{C}$ is given by $\mathcal{C}^* = \left\{y \in \mathbb{R}^r,\ e^\top y \geq \|y\|_2\right\}$.*

The proof for this lemma is provided in [42, Section 4.2.3.2].

SSC1 requires the columns of H to contain the cone $\mathcal{C}$, which is tangent to every facet of the nonnegative orthant; see Figure 2.3. Hence, satisfying SSC1 requires some degree of sparsity as H needs to contain at least $r-1$ zeros per row [41, Th. 4.28]. SSC2 is a mild regularity condition which is typically satisfied when SSC1 is satisfied. For more discussions on the SSC, we refer the interested reader to [34] and [41, Chapter 4.2.3], and the references therein.

In practice, it is not likely for both $W^\top$ and H to satisfy the SSC. Typically, H will satisfy the SSC, as it is typically sparse. However, in many applications, $W^\top$ will not satisfy the SSC; in particular in applications where W is not sparse, e.g., in hyperspectral unmixing, recommender systems, or imaging. This is why regularized NMF models have been introduced, including sparse and volume regularized NMF. We refer the interested reader to Chapter 6, Chapter 7 and [41, Chapter 4] for more details.

2.3.2 Identifiability of SSMF

Without further requirements, SSMF is never identifiable; which follows from a result for semi-NMF which is a factorization model that requires only one factor, H, to be nonnegative [44]. Let $X = WH$ be an SSMF of X. We can obtain other SSMF of X using the following transformation: for any $\alpha \geq 0$, let

$$W(\alpha) := W\left((1+\alpha)I - \frac{\alpha}{r}J\right),$$

and

$$\begin{aligned} H(\alpha) &:= \left(\frac{1}{1+\alpha}H + \frac{\alpha}{(1+\alpha)r}J\right) \\ &= \left(\frac{1}{1+\alpha}I + \frac{\alpha}{(1+\alpha)r}J\right)H, \end{aligned}$$

where I is the identity matrix of appropriate dimension, J is the matrix of all ones of appropriate dimension. The second equality follows from the fact that $e^\top H = e^\top$. The matrix $H(\alpha)$ is column stochastic since H and $\frac{J}{r}$ are. One can check that

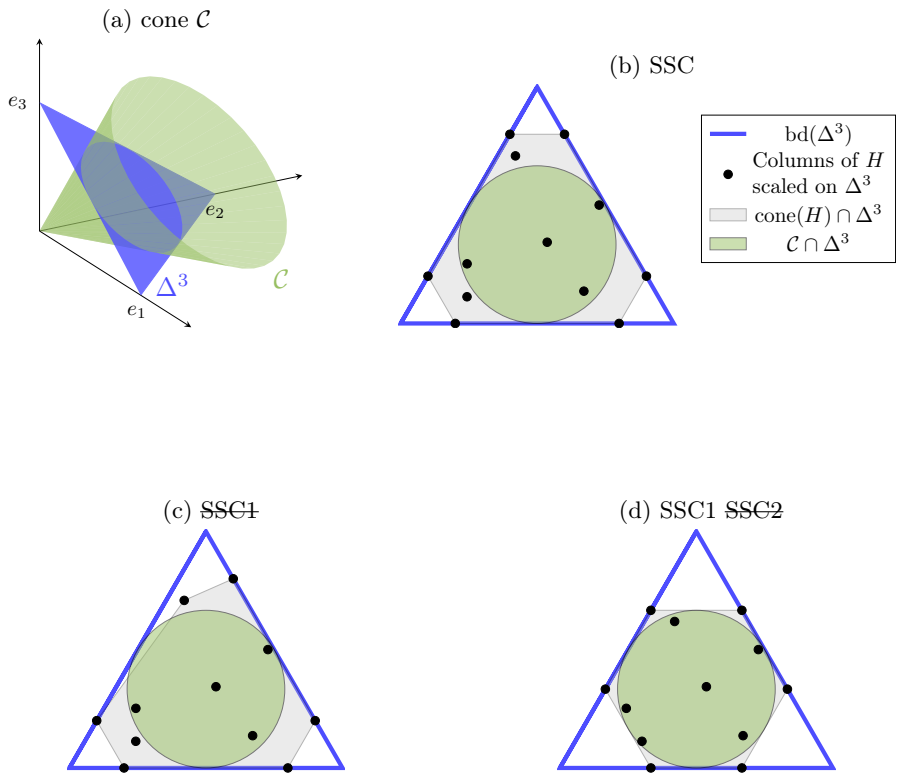


Figure 2.3: Illustration of the SSC in three dimensions. On (a): the sets Δ^3 and $\mathcal{C}$, they intersect at (0,0.5,0.5), (0.5,0,0.5), and (0.5,0.5,0). On (b), (c) and (d): examples of a matrix $H \in \mathbb{R}^{3\times n}$ respectively satisfying the SSC, not satisfying SSC1 and satisfying SSC1 but not SSC2.

$(W(\alpha), H(\alpha))$ is not a permutation and scaling of (W, H) for $\alpha > 0$, while $WH = W(\alpha)H(\alpha)$ since[2]

$$\begin{aligned} A(\alpha) :=& \left((1+\alpha)I - \frac{\alpha}{r}J\right)^{-1} \\ =& \frac{1}{1+\alpha}I + \frac{\alpha}{(1+\alpha)r}J. \end{aligned}$$

Geometrically, to obtain $W(\alpha)$, the columns of W are moved towards the exterior of $\operatorname{conv}(W)$ and hence the convex hull of the column of $W(\alpha)$ contains the convex hull of the columns of W and hence contains $\operatorname{conv}(X)$. This follows from the fact that $W = W(\alpha)A(\alpha)$, where A is column stochastic.

To obtain identifiability of SSMF, one needs either to impose additional constraints on W and/or H such as sparsity [2], or look for a solution minimizing a certain function g. In particular, MinVol SSMF , that is the solution (W, H) that minimizes the volume of the convex hull of W and the origin within its column space

$$\begin{array}{ll} \underset{W\in\mathbb{R}^{m\times r}, H\in\mathbb{R}^{r\times n}}{\text{minimize}} & \det(W^\top W) \\ \text{subject to} & X = WH, \\ & H \in \Delta^{r\times n}, \end{array} \tag{2.4}$$

is essentially unique given that H satisfies the so-called sufficiently scattered condition (SSC). Note that the quantity $\det(W^\top W)$ is only relative to the aforementioned volume. The true volume is $\frac{1}{r!}\sqrt{\det(W^\top W)}$.

For SSMF, we have the following identifiability result.

Theorem 2.2 *[37, 68] If $H \in \mathbb{R}^{r\times n}$ is sufficiently scattered, then the Exact MinVol SSMF (W, H), in the sense of (2.4), of $X = WH$ of size $r = \operatorname{rank} X$ is essentially unique.*

Note that this result has been generalized to the case where the columns of H belong to a given polytope instead of the probability simplex; see [91].

In practice, because of noise and model misfit, SSMF optimization models need to balance the data fitting term which measures the discrepancy between X and WH, and the volume regularization for $\operatorname{conv}(W)$. Typically, a problem with objective function of the form

$$\|X - WH\|_F^2 + \lambda \det(W^\top W),$$

is solved. This requires the tuning of the parameter λ, which is a nontrivial process [3, 82, 113].

[2]This is an invertible M-matrix, with positive diagonal elements and negative off-diagonal elements, whose inverse is nonnegative [8].

2.4 Brief summary of the thesis content

In the following chapters, we will study various CLRMFs; they are organized as follows:

- In Chapter 3, we introduce a new model, dubbed BSSMF, for Bounded Simplex-Structured Matrix Factorization. BSSMF imposes that the columns of W belong to a chosen hyperrectangle while the columns of H have to belong to the probability simplex. The resulting factorization WH naturally belongs to the same chosen hyperrectangle. This behavior is particularly meaningful for naturally bounded data. BSSMF is also identifiable under milder conditions than NMF.

- In Chapter 4, we introduce a new model, dubbed PMF, for Polytopic Matrix Factorization. PMF imposes that the rows of W belong to a chosen polytope and the columns of H belong to another chosen polytope. PMF is identifiable with the same core idea than Theorem 2.1, that is, the rows of W and the columns of H should be "sufficiently scattered" within their respective feasible set. PMF is a very generic model, meaning that we provide identifiability for a wild class of matrix factorization models.

- In Chapter 5, we introduce RandSPA, a greedy algorithm to solve NMF under the separability assumption. RandSPA creates a continuum between SPA and VCA. Thus, it combines the robustness of SPA and the randomness of VCA, allowing to outperform them when taking the best run among several.

- In Chapter 6, we study MinVol NMF, a model similar to MinVol SSMF (2.4). Thus, it inherits from its identifiability. As a reminder, the MinVol criterion penalizes the volume of the convex hull generated by the columns of W and the origin. Compared to MinVol SSMF, W has to be nonnegative. Also, there are variants of MinVol NMF where the simplex-structure can be either on rows of H or the columns of W. We develop a fast algorithm for MinVol NMF based on an inertial block majorization-minimization framework for non-smooth non-convex optimization, that was also used for BSSMF. Then, we show that the MinVol criterion shows promising results for matrix completion.

- In Chapter 7, we introduce a new model, dubbed MaxVol NMF. The core idea revolves around volume penalization, like with MinVol NMF. The difference is that the volume of H is maximized, instead of the volume of W being minimized. In the exact case, MinVol NMF and MaxVol NMF are equivalent. Thus, MaxVol NMF is as identifiable as MinVol NMF. However, in the inexact case, MaxVol NMF offers more control on the sparsity of the decomposition factor H. This behavior is interesting in the context of HU. In fact, MaxVol NMF shows better results than MinVol NMF on real hyperspectral datasets.

Chapter 3

Bounded Simplex-Structured Matrix Factorization

Recently, simplex-structured matrix factorization (SSMF), discussed in Section 2.2, was introduced as a generalization of NMF [108]; see also [2] and the references therein. SSMF does not impose any constraint on W, while it requires H to be column stochastic, that is, $H(:,j) \in \Delta^r$ for all j. As a reminder, $\Delta^r = \{x \in \mathbb{R}^r \mid x \geq 0, e^\top x = 1\}$ is the probability simplex and e is the vector of all ones of appropriate dimension. SSMF is closely related to various machine learning problems, such as latent Dirichlet allocation, clustering, and the mixed membership stochastic block model; see [6] and the references therein. Let us recall why SSMF is a generalizarion of NMF by considering the exact NMF model, $X = WH$. Let us normalize the input matrix such that the entries in each column sum to one (w.l.o.g. we assume X, and W, do not have a zero column), that is, such that $X^\top e = e$, and let us impose w.l.o.g. that the entries in each column of W also sum to one (by the scaling degree of freedom in the factorization WH), that is, $W^\top e = e$. Then, we have

$$X^\top e = e = (WH)^\top e = H^\top W^\top e = H^\top e, \tag{3.1}$$

so that H has to be column stochastic, since $H \geq 0$ and $H^\top e = e$ is equivalent to $H(:,j) \in \Delta^r$ for all j.

In this chapter, we introduce bounded simplex-structured matrix factorization (BSSMF) . BSSMF imposes the columns of W to belong to a hyperrectangle, namely $W(i,j) \in [a_i, b_i]$ for all i, j for some parameters $a_i \leq b_i$ for all i. For simplicity, given $a \leq b \in \mathbb{R}^m$, we denote the hyperrectangle

$$[a,b] = \{y \in \mathbb{R}^m \mid a_i \leq y_i \leq b_i \text{ for all } i\},$$

and refer to it as an interval. The hyperrectangle constraint on W is denoted as $W(:,j) \in [a,b]$ for all j. Let us formally define BSSMF.

Definition 3.1 (BSSMF) *Let $X \in \mathbb{R}^{m \times n}$, let $r \leq \min(m,n)$ be an integer, and let $a, b \in \mathbb{R}^m$ with $a \leq b$. The pair $(W,H) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ is a BSSMF of X of size r*

for the interval $[a, b]$ *if*

$$X = WH, \quad W(:, k) \in [a, b] \text{ for all } k, \quad H(:, j) \in \Delta^r \text{ for all } j.$$

Since the columns of H define convex combinations, the convex hull of the columns of $X = WH$ is contained in the convex hull of the columns of W, which is itself contained in the hyperrectangle $[a, b]$. This implies that the hyperrectangle $[a, b]$ must contain the columns of the data matrix, $X = WH$. BSSMF reduces to SSMF when $a_i = -\infty$ and $b_i = +\infty$ for all i. When $X \geq 0$, BSSMF reduces to NMF when $a_i = 0$ and $b_i = +\infty$ for all i, after a proper normalization of X; see the discussion around Equation (3.1).

Outline and contribution of the chapter The chapter is organized as follows. In Section 3.1, we explain the motivation of introducing BSSMF. In Section 3.2, we propose an efficient algorithm for BSSMF. In Section 3.3, we provide an identifiability result for BSSMF, which follows from an identifiability result for NMF. In Section 3.4, we illustrate the effectiveness of BSSMF on two applications:

- Image feature extraction: the entries of X are pixel intensities. For example, for a gray level image, the entries of X belong to the interval $[0, 255]$.
- Recommender systems: the entries of X are ratings of users for some items (e.g., movies). These ratings belong to an interval, e.g., [1,5] for the Netflix and MovieLens datasets.

3.1 Motivation of BSSMF

The motivation to introduce BSSMF is mostly fourfold; this is described in the next four paragraphs.

Bounded low-rank approximation When the data naturally belong to intervals, imposing the approximation to belong to the same interval allows to provide better approximations, taking into account this prior information. Imposing that the entries in W belong to some interval and that H is column stochastic resolves this issue. BSSMF implies that the columns of the approximation WH belong to the same interval as the columns of W. In fact, for all j,

$$X(:, j) \approx WH(:, j) \in [a, b],$$

since $W(:, k) \in [a, b]^m$ for all k, and the entries of $H(:, j)$ are nonnegative and sum to one.

Another closely related model was proposed in [70] where the entries of the factors W and H are required to belong to bounded intervals. The authors showed that their model is suitable for clustering. Nonetheless, it is not clear how to choose the lower and upper bounds on the entries of W and H to obtain tight lower and upper bounds for their product WH. With BSSMF the choice for the lower and upper bounds

is easier, e.g., choosing a_i and b_i to be the smallest and largest entry in $X(i,:)$, respectively, that is, bounding W in the same way the data matrix is; see Section 3.3 for more details.

Interpretability BSSMF allows us to easily interpret both factors: the columns of W can be interpreted in the same way as the columns of X (e.g., as movie ratings, or pixel intensities), while the columns of H provide a soft clustering of the columns of X as they are column stochastic. BSSMF can be interpreted geometrically similarly as SSMF and NMF: the convex hull of the columns of W, $\text{conv}(W)$, must contain $\text{conv}(X)$, since $X(:,j) = WH(:,j)$ for all j where H is column stochastic, while it is contained in the hyperrectangle $[a,b]$:

$$\text{conv}(X) \quad \subseteq \quad \text{conv}(W) \quad \subseteq \quad [a,b].$$

Imposing bounds on the approximation, via the element-wise constraints $a \leq WH \leq b$ for some $a, b \in \mathbb{R}$, was proposed in [55] and applied successfully to recommender systems. However, this model does not allow to interpret the basis factor, W, in the same way as the data. Some elements in W will probably be out of the rating range because W is not directly constrained. Hence, the basis elements in W can only be interpreted as "eigen users", while with BSSMF, the basis elements can be interpreted as virtual meaningful users. It is also difficult to interpret the factor H as it could contain negative contributions. In fact, only imposing $a \leq WH \leq b$ typically leads to dense factors W and H (that is, factors that do not contain many zeros, as opposed to sparse factors), while in most applications interpretability usually comes with a certain sparsity degree in at least one of the factors.

A closely related model that tackles blind source separation is bounded component analysis (BCA) proposed in [24, 30], where the sources are assumed to belong to compact sets (hyperrectangle being a special case), while no constraints is imposed on the mixing matrix. Again, without any constraints on the mixing matrix, BCA will generate dense factors with negative linear combinations which are difficult to interpret. Let us note that their motivation is different from ours, as their objective is to extract mixed sources, while ours is to extract interpretable features and decompose data through them. In [75], the authors also proposed a blind source separation algorithm for bounded sources based on geometrical concepts. The mixtures are assumed to belong to a parallelogram. The proposed separation technique is relies on mapping this parallelogram to a rectangle. Again, their objective is to extract mixed sources. Nonetheless, working with a domain different from a hyperrectangle could be of interest for future work.

Identifiability Identifiability is key in practice as it allows to recover the ground truth that generated the data; see the discussion in Section 2.3, [34, 58] and the references therein. A drawback of SSMF is that it is never identifiable, see Section 2.3.2 for further details. On the counterpart NMF can be identifiable, which is discussed in Section 2.3.1. Nonetheless, the conditions are not mild. For NMF to be identifiable, it is necessary that the supports of the columns of W (that is, the set of

non-zero entries) are not contained in one another (this is called a Sperner family), and similarly for the supports of the rows of H; see, e.g., [59, 78]. This requires the presence of zeros in each column of W and row of H, which can be a strong condition in some applications. For example, in hyperspectral unmixing, W is typically not sparse because it recovers spectral signatures of constitutive materials which are typically positive. Although the conditions for NMF (and SSMF) to be identifiable can be weakened using additional constraints and regularization terms, it then requires hyperparameter tuning procedures. In [91], they propose a model where the columns of H belong to a polytope. Using a maximum volume criterion on the convex hull of H, their model is identifiable under the condition that the convex hull of H contains the ellipsoid of maximum volume inscribed in the constraining polytope. The use of the maximum volume criterion also requires hyperparameter tuning. In [24, 30], the sources are identifiable by optimizing some geometric criterion, respectively minimizing a perimeter, and maximizing the ratio between the volume of an ellipsoid and the volume of a hyperrectangle. These identifiability conditions are not relevant to our model. As we will see in Section 3.3, BSSMF is identifiable under relatively mild conditions, while it does not require parameter tuning, as opposed to most regularized structured matrix factorization models that are identifiable. Let us note that it is also possible to formulate identifiable nonlinear matrix approximation models like the bilinear model of [26], but this is out of the scope of this chapter.

Robustness to overfitting Another drawback of NMF and SSMF is that they are rather sensitive to the choice of r. When r is chosen too large, these two models are over-parameterized and will typically lead to overfitting. This is a well-known behaviour that can be addressed with additional regularization terms that need to be fine-tuned [86]. As we will see experimentally in Section 3.4.3 for matrix completion, without any parameter tuning, BSSMF is much more robust to overfitting than NMF and unconstrained matrix factorization. The reason is that the additional bound constraints on W and sum-to-one constraint on H prevents columns of W and of WH from going outside the feasible range, $[a, b]$. In turn, BSSMF will be less sensitive to noise and an overestimation of r. For example, an outlier that falls outside the feasible set $[a, b]$ will not pose problems to BSSMF, while it may significantly impact the NMF and SSMF solutions.

3.2 Inertial block-coordinate descent algorithm for BSSMF

In this chapter, we consider the following BSSMF problem

$$\begin{aligned} \min_{W,H} g(W,H) &:= \frac{1}{2}\|X - WH\|_F^2 \\ \text{such that } & W(:,k) \in [a,b] \text{ for all } k, \\ & H \geq 0, \text{ and } H^\top e = e, \end{aligned} \tag{3.2}$$

that uses the squared Frobenius norm to measure the error of the approximation.

3.2.1 Proposed algorithm

Most NMF algorithms rely on block coordinate descent methods, that is, they update a subset of the variables at a time, such as the popular multiplicative updates of Lee and Seung [61], the hierarchical alternating least squares algorithm [19, 40], and a fast gradient based algorithm [48]; see, e.g., [41, Chapter 8] and the references therein for more detail. More recently, an inerTial block majorIzation minimization framework for non-smooth non-convex opTimizAtioN (TITAN) was introduced in [51] and has been shown to be particularly powerful to solve matrix and tensor factorization problems [50, 74, 98].

To solve (3.2), we therefore apply TITAN which updates one block W or H at a time while fixing the value of the other block. In order to update W (resp. H), TITAN chooses a block surrogate function for W (resp. H), embeds an inertial term to this surrogate function and then minimizes the obtained inertial surrogate function. We have $\nabla_W g(W,H) = -(X - WH)H^\top$ which is Lipschitz continuous in W with the Lipschitz constant $\|HH^\top\|$, where $\|.\|$ is the spectral norm. Similarly, $\nabla_H g(W,H) = -W^\top(X - WH)$ is Lipschitz continuous in H with constant $\|W^\top W\|$. Hence, we choose the Lipschitz gradient surrogate for both W and H and choose the Nesterov-type acceleration as analyzed in [51, Section 4.2.1] and [51, Remark 4.1], see also [51, Section 6.1] and [98] for similar applications.

Recall that applying BSSMF to recommender systems is one of our motivations, meaning that our model should be able to handle missing entries in X. Let us consider the more general model

$$\begin{aligned} \min_{W,H} g(W,H) &:= \frac{1}{2}\|M \circ (X - WH)\|_F^2 \\ \text{such that } W(:,k) &\in [a,b] \text{ for all } k, \\ H &\geq 0, \text{ and } H^\top e = e, \end{aligned} \tag{3.3}$$

where $\circ$ corresponds to the Hadamard product, and M is a weight matrix which can model missing entries using $M(i,j) = 0$ when $X(i,j)$ is missing, and $M(i,j) = 1$ otherwise. It can also be used in other contexts; see, e.g., [39, 43, 89]. TITAN can also be used to solve (3.3), where the gradients are equal to $\nabla_W g(W,H) = -(M \circ (X - WH))H^\top$ and $\nabla_H g(W,H) = -W^\top(M \circ (X - WH))$. We acknowledge that the identifiability result that will be presented in Section 3.3 does not hold for the case where some data are missing, this is an interesting direction of future research. Algorithm 3.1 describes TITAN for solving the general problem (3.3), where $[.]_b^a$ is the column-wise projection on $[a,b]$ and $[.]_{\Delta^r}$ is the column wise projection on the simplex Δ^r. Let us clarify that our implementation of TITAN, although looking similar to alternating fast projection gradient methods (AFPGMs), differs from them. Concretely, with TITAN, the inertial sequence is evolving at every iteration and is not restarted when the algorithm alternates between updating W and updating H. Typically, AFPGMs would restart the inertial sequence when the algorithm alternates between the blocks, because their goal is to solve alternatively the sub-problems. TITAN considers the whole problem instead of considering several sub-problems. Hence, TITAN tries to accelerate the global convergence of the

sequences rather than trying to accelerate the convergence for the sub-problems. For more details, see Section 6.3.1 where an implementation of TITAN for MinVol NMF is shown to be faster than an alternating projection gradient method with Nesterov extrapolation.

Due to our derived algorithm being a particular instance of TITAN with Lipschitz gradient surrogates [51, Section 4.2], Algorithm 3.1 guarantees a subsequential convergence, that is, every limit point of the generated sequence is a stationary point of Problem (3.2). The Julia code for Algorithm 3.1 is available on gitlab[1] (a MATLAB code is also available on gitlab[2] but it does not handle missing data). We omit the implementation details here, but let us mention that when data are missing, our Julia implementation does not compute the Hadamard product with M explicitly but rather takes advantage of the sparsity of the data by using multithreading to improve the computational time. The projections $[.]_b^a$ and $[.]_{\Delta^r}$ are also computed using multithreading.

Initialization A simple choice to initialize the factors, W and H, in Algorithm 3.1 is to randomly initialize them: for all i, each entry of $W(i,:)$ is generated using the uniform distribution in the interval $[a_i, b_i]$, while H is generated using a uniform distribution in $[0,1]^{r\times n}$ whose columns are then projected on the simplex Δ^r.

Choice of Lipschitz constant When some data are missing, the Lipschitz constant of the gradients relatively to W and H could be smaller than $\|HH^\top\|$ and $\|W^\top W\|$, respectively. Relatively to H for instance, a smaller Lipschitz constant would be $\max_j \|W^\top \operatorname{Diag}(M(:,j))W\|$. Indeed,

$$\begin{aligned}
\|\nabla_H g(W,H_1) - \nabla_H g(W,H_2)\|_F &= \|W^\top(M\circ(W(H_1-H_2)))\|_F \\
\|W^\top(M\circ(W(H_1-H_2)))\|_F^2 &= \sum_j \|W^\top(M(:,j)\circ(W(H_1(:,j)-H_2(:,j))))\|_F^2 \\
&= \sum_j \|W^\top \operatorname{Diag}(M(:,j))W(H_1(:,j)-H_2(:,j))\|_F^2 \\
&\le \max_j \|W^\top \operatorname{Diag}(M(:,j))W\|^2 \|H_1-H_2\|_F^2.
\end{aligned}$$

Obviously, $\max_j \|W^\top \operatorname{Diag}(M(:,j))W\| \le \|W^\top W\|$ due to M being binary. Equality is achieved if there exists a j such that $M(:,j) = e$, that is, if at least one column is fully observed. Consequently, $\|W^\top W\|$ is clearly a Lipschitz constant of $\nabla_H g(W,H)$ relatively to H, but $\max_j \|W^\top \operatorname{Diag}(M(:,j))W\|$ is a tighter one. By symmetry of the problem, this also applies to $\nabla_W g(W,H)$ relatively to W, where $\|HH^\top\|$ is a Lipschitz constant, but $\max_i \|H \operatorname{Diag}(M(i,:))H^\top\|$ is a tighter one. Yet, we choose to keep $\|HH^\top\|$ and $\|W^\top W\|$ even when some data are missing since those values are faster to compute. This choice can be compensated by data centering; see Section 3.2.2. Note that when M is a weight matrix, $\nabla_W g(W,H) = -(M^{\circ 2}\circ(X-WH))H^\top$

[1] https://gitlab.com/vuthanho/bssmf.jl
[2] https://gitlab.com/vuthanho/bounded-simplex-structured-matrix-factorization

Algorithm 3.1: Proposed algorithm for BSSMF

input : Input data matrix $X \in \mathbb{R}^{m \times n}$, bounds $a \leq b \in \mathbb{R}^m$, initial factors $W \in \mathbb{R}^{m \times r}$ s.t. $W(:,k) \in [a,b]$ for all k and simplex structured $H \in \mathbb{R}_+^{r \times n}$, weights $M \in [0,1]^{m \times n}$

output: W and H

$\alpha_1 = 1$, $\alpha_2 = 1$, $W_{old} = W, H_{old} = H$, $L_W^{prev} = L_W = \|HH^\top\|$, $L_H^{prev} = L_H = \|W^\top W\|$

repeat

  while *stopping criteria not satisfied* **do**

    $\alpha_0 = \alpha_1, \alpha_1 = (1 + \sqrt{1 + 4\alpha_0^2})/2$

    $\beta_W = \min\left[\ (\alpha_0 - 1)/\alpha_1, 0.9999\sqrt{L_W^{prev}/L_W}\ \right]$

    $\overline{W} \leftarrow W + \beta_W(W - W_{old})$

    $W_{old} \leftarrow W$

    $W \leftarrow \left[\overline{W} + \frac{(M \circ (X - \overline{W}H))H^\top}{L_W}\right]_a^b$

    $L_W^{prev} = L_W$

  $L_H \leftarrow \|W^\top W\|$

  while *stopping criteria not satisfied* **do**

    $\alpha_0 = \alpha_2, \alpha_2 = (1 + \sqrt{1 + 4\alpha_0^2})/2$

    $\beta_H = \min\left[\ (\alpha_0 - 1)/\alpha_2, 0.9999\sqrt{L_H^{prev}/L_H}\ \right]$

    $\overline{H} \leftarrow H + \beta_H(H - H_{old})$

    $H_{old} \leftarrow H$

    $H \leftarrow \left[\overline{H} + \frac{W^\top(M \circ (X - W\overline{H}))}{L_H}\right]_{\Delta^r}$

    $L_H^{prev} \leftarrow L_H$

  $L_W = \|HH^\top\|$

until *some stopping criteria is satisfied*

and $\nabla_H g(W,H) = -W^\top(M^{\circ 2} \circ (X - WH))$, where $M^{\circ 2} = M \circ M$. Similarly to the case where M is binary, we then retrieve $\max_i \|H \operatorname{Diag}(M(i,:))^2 H^\top\|$ and $\max_j \|W^\top \operatorname{Diag}(M(:,j))^2 W\|$ as tighter Lipschitz constants than $\|HH^\top\|$ and $\|W^\top W\|$.

3.2.2 Accelerating BSSMF algorithms via data centering

Not only the BSSMF model is invariant to translations of the input data (this is explained in details in Section 3.3), but also the optimization, because of the simplex constraints. In particular, for any $\mu \in \mathbb{R}^m$, minimizing

$$f(W,H) := \frac{1}{2}\|X - WH\|_F^2 \tag{3.4}$$

or

$$f_\mu(W,H) := \frac{1}{2}\|X - \mu e^\top - (W - \mu e^\top)H\|_F^2 \tag{3.5}$$

is equivalent in BSSMF, since $\mu e^\top H = \mu e^\top$ as H is column stochastic. However, *outside the feasible set, f and f_μ do not have the same topology.* Computing the gradients, we have $\nabla_H f(W,H) = W^\top(WH - X)$ which is Lipschitz continuous in H with the Lipschitz constant $\|W^\top W\|$, and $\nabla_H f_\mu(W,H) = W_\mu^\top(W_\mu H - X_\mu)$ which is Lipschitz continuous in H with the Lipschitz constant $\|W_\mu^\top W_\mu\|$, where $W_\mu = W - \mu e^\top$ and $X_\mu = X - \mu e^\top$. Particularly, for BSSMF, since W can be interpreted in the same way as X, we expect $\text{mean}_{\text{row}}(X) = \frac{1}{n}Xe \approx \text{mean}_{\text{row}}(W) \in \mathbb{R}^m$, where $\text{mean}_{\text{row}}(.)$ is the empirical mean of each row of the input. Let us in fact choose $\mu = \text{mean}_{\text{row}}(X)$. From [52, Theorem 3], we have $\|X_\mu^\top X_\mu\| \leq \|X^\top X\|$. Consequently, we expect the Lipschitz constant $\|W_\mu^\top W_\mu\|$ to be smaller than $\|W^\top W\|$. A smaller Lipschitz constant means that, when updating H, the gradient steps are allowed to be larger without losing any convergence guarantee. Hence, with the right translation on our data X, the optimization problem on H is unchanged on the feasible set but Algorithm 3.1 can be accelerated.

Let us illustrate this behavior on a small example with $m = 2$, $n = 1$, $r = 2$. We choose

$$X = \begin{pmatrix} 0.4 & 0.3 \\ 0.7 & 0.2 \end{pmatrix} \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}.$$

We fix

$$W = \begin{pmatrix} 0.4 & 0.3 \\ 0.7 & 0.2 \end{pmatrix},$$

and try to solve, with respect to H, Eq. (3.4) and Eq. (3.5) with $\mu = \text{mean}_{\text{row}}(X)$. We perform 5 projected gradient steps and display the results on Figure 3.1. On the top, 5 projected gradient steps are performed to update H based on the original data X. On the bottom, 5 projected gradient steps are performed to update H based on the centered data X. The feasible sets (in dash) are exactly the same, and therefore the optimal solutions are also the same. However, we observe that the landscape of the cost function outside the feasible region is smoother when the data are centered. This allows the solver to converge faster towards the optimal solution, as the gradients point

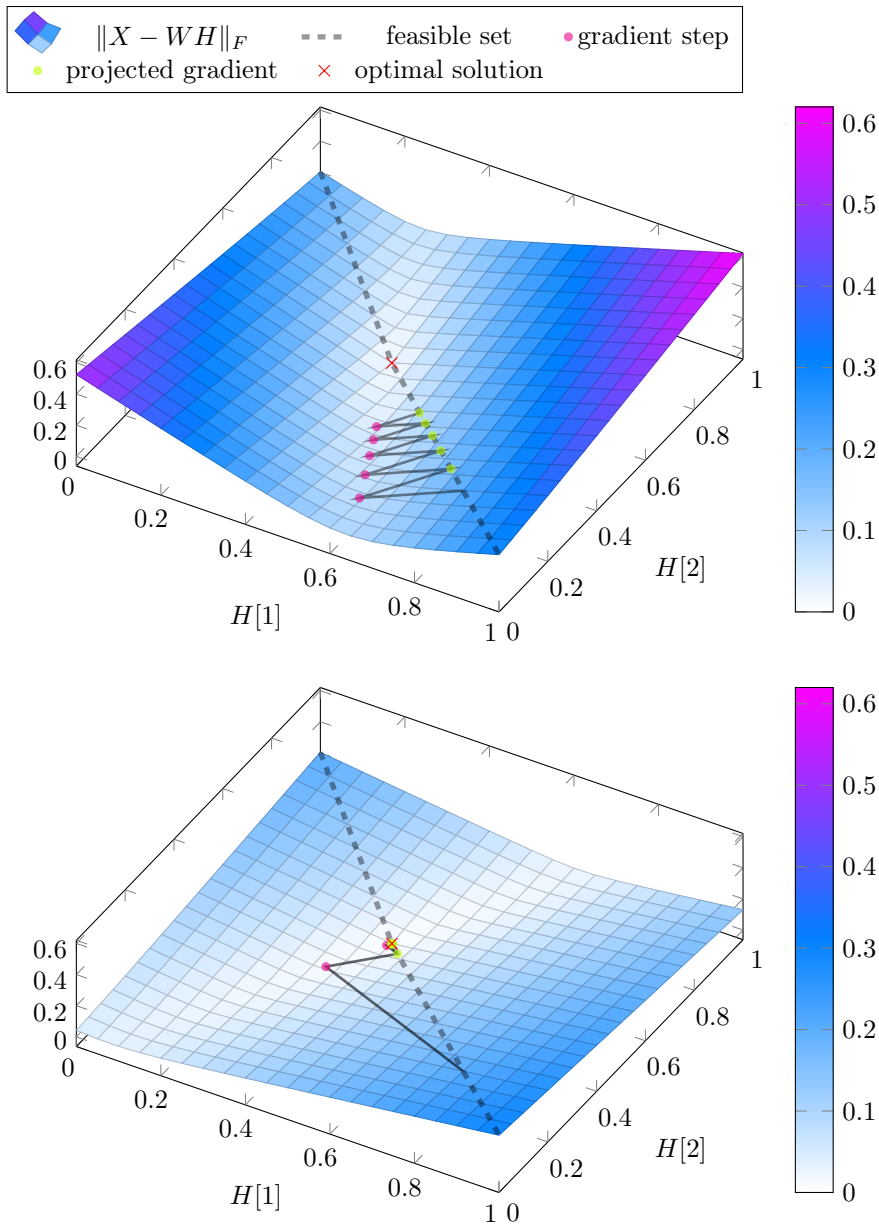


Figure 3.1: Influence of centering the data on the cost function topology regarding H via a small example ($m = 2, r = 2, n = 1$). Top: without centering. Bottom: with centering. Five projected gradient steps are shown, decomposed through one gradient descent step followed by its projection onto the feasible set.

better towards the optimal solution and the step sizes are larger. The improvement regarding the convergence speed by applying centering with real data will probably not be as drastic as in this small example. Still, minimizing a smoother function is always advantageous, and this will be shown empirically on real data in Section 3.2.3.

3.2.3 Convergence speed and effect of acceleration strategies on real data

In this subsection, the goal is twofold: (1) show the effect of the extrapolation in TITAN by comparing Algorithm 3.1 to a non-extrapolated block coordinate descent, and (2) show the acceleration effect of centering the data.

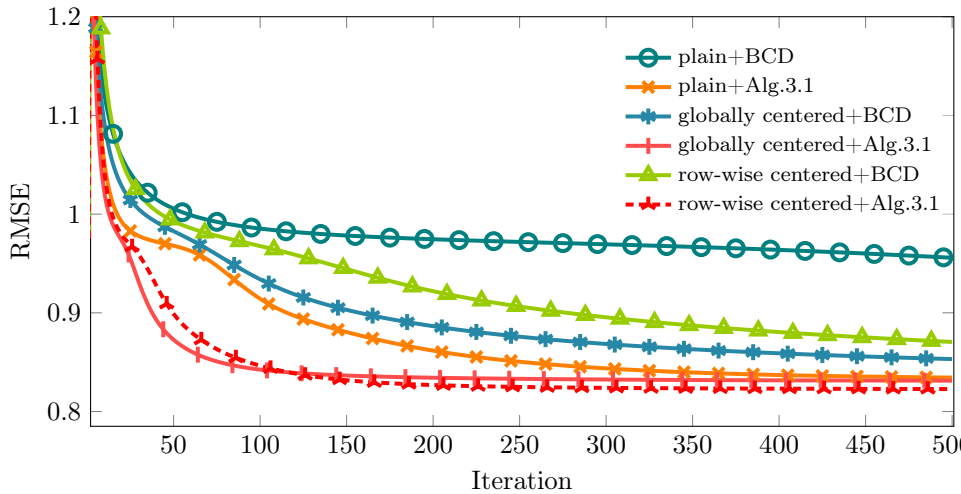
We will apply the BSSMF model on MNIST and ml-1m (these two datasets are properly introduced respectively in Section 3.4.1 and Section 3.4.3) in six different scenarios: 3 data related scenarios $\times$ 2 algorithmic related scenarios. The data scenarios are raw data, globally centered data, and row-wise centered data (respectively called 'plain', 'globally centered' and 'row-wise centered' in Figure 3.2). Globally centered data are such that $\mu = \frac{e^\top X e}{mn} e$ and row-wise centered data are such that $\mu = \frac{1}{n} X e$. Note that with a global centering, result from [52, Theorem 3] does not hold anymore. Yet, we propose to see here how this centering strategy behaves. For each data case, 2 algorithms are tested: (1) Algorithm 3.1, and (2) a standard block coordinate descent (BCD) which is Algorithm 3.1 where the β's are fixed to 0; this corresponds to the popular proximal alternating linearized minimization (PALM) algorithm [12]. When the algorithms are compared on the same data scenario, Algorithm 3.1 always converges faster and to a better solution than BCD. We also observe that when the data are centered, globally or row-wise, applying the same algorithm always lead to faster convergence than on the plain case. Let us first comment the results on ml-1m (Figure 3.2a). Applying BCD on the globally centered data is almost as fast as applying Algorithm 3.1 in the plain case, meaning that a good preprocessing is almost as important as a good acceleration strategy. With BCD only, global centering is faster than row-wise centering. With Algorithm 3.1, global centering converges faster but row-wise centering converges to slightly better solutions. The root-mean-square errors

	plain	globally centered	row-wise centered
BCD	0.93	0.89	0.91
Algorithm 3.1	0.87	0.87	0.87

Table 3.1: RMSE on the test set for ml-1m depending on the algorithm and the centering strategy

(RMSEs) on the test set are available in Table 3.1, highlighting the importance of a good acceleration strategy. This could be expected, since the centering only affects the convergence speed, but does not change the solutions. Let us now comment the results on MNIST shown on Figure 3.2b. Global centering does not really improve the convergence speed compared to the plain case, regardless of the used algorithm. Interestingly, row-wise centering provides a great improvement in convergence speed

(a) ml-1m



(b) MNIST

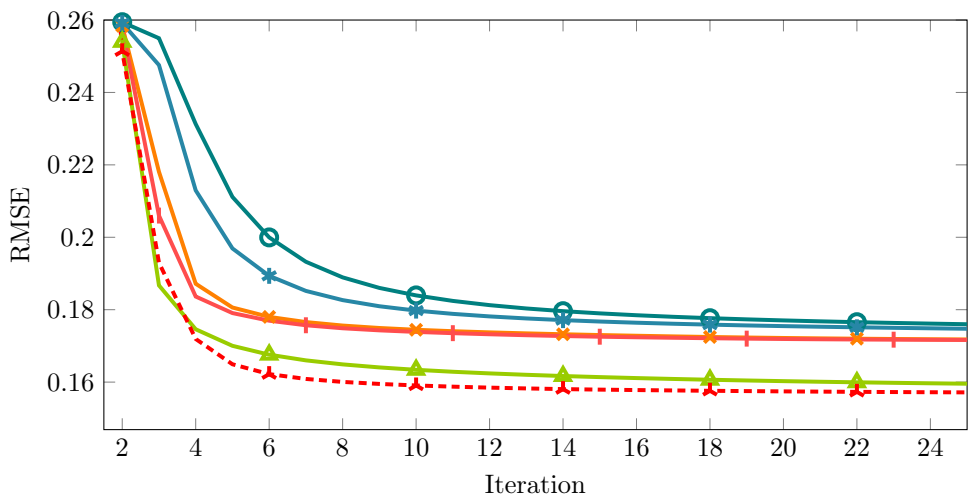


Figure 3.2: Evolution of the training error for ml-1m and MNIST, averaged on 10 runs. For ml-1m, $r = 5$, 1 inner iteration. For MNIST, $r = 50$, 10 inner iterations.

and better local minima, regardless of the used algorithm. In this case, a good centering strategy seems even more important than a good acceleration strategy. Globally, regardless of the dataset, applying Algorithm 3.1 on centered data is the best strategy as compared with using plain data. As a consequence, it will be our default choice for the experiments in Section 3.4.

As mentioned above, when entries are missing, Algorithm 3.1 can take advantage of the sparsity of the data and uses multithreading. We report in Table 3.2 the computation time of Algorithm 3.1 in the experiment settings of Figure 3.2a, given by the macro `@btime` from the package `BenchmarkTools.jl`. The computation time in the settings of the experiment in Figure 3.2b is reported in Table 3.3. When the dataset is full, like with MNIST, multithreading is only used for the projections $[.]_b^a$ and $[.]_{\Delta^r}$. Of course, multithreading is also employed for every matrix multiplication. However, the numbers of threads shown in Tables 3.2 and 3.3 do not affect the computation time of matrix multiplications, as BLAS selects its own number of threads, independent of the number assigned to Julia. Note that there is no distinction between Algorithm 3.1 and BCD in terms of computation time because the computation of the acceleration is negligible compared to the other computations.

# threads	1	2	4	6	8	10	12
time (s)	30.53	5.14	2.98	2.85	3.00	2.78	3.31

Table 3.2: Computation time of Algorithm 3.1 in the experiment settings of Figure 3.2a depending on the number of used threads.

# threads	1	2	4	6	8	10	12
time (s)	27.79	21.92	16.67	15.22	15.73	16.01	16.65

Table 3.3: Computation time of Algorithm 3.1 in the experiment settings of Figure 3.2b depending on the number of used threads.

3.3 Identifiability of BSSMF

A main motivation to introduce Bounded simplex-structured matrix factorization (BSSMF) is that it is identifiable under weaker conditions than NMF. We now state our main identifiability result for BSSMF, it is a consequence of the identifiability result of NMF and the following simple observation: $X = WH$ is a BSSMF for the interval $[a, b]$ implies that $be^\top - X = (be^\top - W)H$ and $X - ae^\top = (W - ae^\top)H$ are Exact NMF decompositions.

Theorem 3.1 *Let $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$ satisfy $W(:,k) \in [a,b]$ for all k for some $a \leq b$, $H \geq 0$, and $H^\top e = e$. If $\begin{pmatrix} W-ae^\top \\ be^\top - W \end{pmatrix}^\top \in \mathbb{R}^{r\times 2m}$ and $H \in \mathbb{R}^{r\times n}$ are sufficiently scattered, then the BSSMF (W,H) of $X = WH$ of size $r = \operatorname{rank}(X)$ for the interval $[a,b]$ is essentially unique.*

Proof 3.1 *Let (W, H) be a BSSMF of X for the interval $[a, b]$. As in the proof of Lemma 3.2, we have*

$$X - ae^\top = WH - ae^\top = (W - ae^\top)H,$$

since $e^\top = e^\top H$. This implies that $(W - ae^\top, H)$ is an Exact NMF of $X - ae^\top$, since $W - ae^\top$ and H are nonnegative. Similarly, we have

$$be^\top - X = be^\top - WH = (be^\top - W)H,$$

which implies that $(be^\top - W, H)$ is an Exact NMF of $be^\top - X$, since $be^\top - W \geq 0$. Therefore, we have the Exact NMF

$$\begin{pmatrix} X - ae^\top \\ be^\top - X \end{pmatrix} = \begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix} H.$$

By Theorem 2.1, this Exact NMF is unique if $\binom{W-ae^\top}{be^\top-W}^\top$ and H satisfy the SSC. This proves the result: in fact, the derivations above hold for any BSSMF of X. Hence, if (W, H) was not an essentially unique BSSMF of X, there would exist another Exact NMF of $\binom{W-ae^\top}{be^\top-W}^\top$, not obtained by permutation and scaling of $\left(\binom{W-ae^\top}{be^\top-W}, H\right)$, a contradiction.

Let us note that $W - ae^\top$ and H being SSC, or $be^\top - W$ and H being SSC, are also sufficient conditions for identifiability. These conditions are stronger, as $W - ae^\top$ being SSC or $be^\top - W$ being SSC implies that $\binom{W-ae^\top}{be^\top-W}^\top$ is SSC. However, $\binom{W-ae^\top}{be^\top-W}^\top$ does not imply that $W - ae^\top$ or $be^\top - W$ is SSC. The condition that $\binom{W-ae^\top}{be^\top-W}^\top$ is SSC is much weaker than requiring $W^\top$ to be SSC in NMF. In fact, in NMF, $W^\top$ being SSC requires that it contains some zero entries (at least $r - 1$ per row [41, Th. 4.28]; this can also be seen on Figure 2.3 in the case $r = 3$). Since the SSC is only defined for nonnegative matrices and $W^\top$ contains zeros, a has to be equal to the zero vector. In this case, $W^\top$ being SSC implies that $W^\top - ea^\top$ is SSC, and hence the corresponding BSSMF is identifiable. However, the reverse is not true. In fact, $\binom{W-ae^\top}{be^\top-W}^\top$ being SSC means that sufficiently many values in W are equal to its minimum and maximum bounds in a and b. For example, in recommender systems, with $W(i, j) \in [1, 5]$ for all (i, j), many entries of W are expected to be equal to 1 or to 5 (the minimum and maximum ratings), so that $\binom{W-ae^\top}{be^\top-W}^\top$ will contain many zero entries, and hence likely to satisfy the SSC [32]. On the other hand, W is positive, and hence it cannot be part of an essentially unique Exact NMF.

Let us illustrate the difference between NMF and BSSMF on a simple example.

Example 3.1 (Non-unique NMF vs. unique BSSMF) *Let $\omega \in [0, 1)$ and let*

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

For $\omega < 0.5$, A_ω satisfies the SSC, while it does not for $\omega \leq 0.5$; see [59, Example 3], [54, Example 2], [41, Example 4.16]. Let us take

$$H = 3A_{1/3} = \begin{pmatrix} 1 & 3 & 3 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 3 & 3 & 1 \end{pmatrix},$$

which satisfies the SSC, and

$$W^\top = 3A_{2/3} = \begin{pmatrix} 2 & 3 & 3 & 2 & 0 & 0 \\ 3 & 2 & 0 & 0 & 2 & 3 \\ 0 & 0 & 2 & 3 & 3 & 2 \end{pmatrix},$$

which does not satisfy the SSC, but has some degree of sparsity. The NMF of

$$X = WH = \begin{pmatrix} 11 & 9 & 6 & 2 & 3 & 9 \\ 9 & 11 & 9 & 3 & 2 & 6 \\ 3 & 9 & 11 & 9 & 6 & 2 \\ 2 & 6 & 9 & 11 & 9 & 3 \\ 6 & 2 & 3 & 9 & 11 & 9 \\ 9 & 3 & 2 & 6 & 9 & 11 \end{pmatrix}$$

is not essentially unique. For example,

$$X = \begin{pmatrix} 0 & 3 & 1 \\ 1 & 3 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 1 \\ 1 & 0 & 3 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} 0 & 2 & 3 & 3 & 2 & 0 \\ 3 & 3 & 2 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 3 & 3 \end{pmatrix}$$

is another decomposition which cannot be obtained as a scaling and permutation of (W, H).

However, the BSSMF of X is unique, taking $a_i = 0$ and $b_i = 3$ for all i. In fact, $(3 - W)^\top$ satisfies the SSC, as it is equal to $3A_{1/3}$, up to permutation of its columns:

$$\begin{aligned} 3 - W^\top &= \begin{pmatrix} 1 & 0 & 0 & 1 & 3 & 3 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 3 & 3 & 1 & 0 & 0 & 1 \end{pmatrix} \\ &= 3A_{1/3}(:, [4, 5, 6, 1, 2, 3]). \end{aligned}$$

Therefore, by Theorem 3.1, the BSSMF of X is unique.

Scaling ambiguity BSSMF is in fact more than essentially unique in the sense of Definition 2.7. In fact, the scaling ambiguity can be removed because of H being simplex structured, as shown in the following lemma.

Lemma 3.1 *Let $H \in \mathbb{R}^{r\times n}$ such that $e^\top H = e^\top$ and $\operatorname{rank}(H) = r$. Let $D \in \mathbb{R}^{r\times r}$ be a diagonal matrix, and let $H' = DH$ be a scaling of the rows of H, and such that $e^\top H' = e^\top$. Then D must be the identity matrix, that is, $D = I$.*

Proof 3.2 *Let us denote $H^\dagger \in \mathbb{R}^{n\times r}$ the right inverse of H, which exists and is unique since $\operatorname{rank}(H) = r$, so that $HH^\dagger = I$. We have*

$$\begin{aligned} & e^\top H' = e^\top DH = e^\top \\ \Rightarrow \quad & e^\top DHH^\dagger = e^\top H^\dagger = e^\top \\ & \qquad \textit{since } e^\top H^\dagger = e^\top HH^\dagger = e^\top \\ \Rightarrow \quad & e^\top D = e^\top \quad \Rightarrow \quad D = I. \end{aligned}$$

Note that this lemma does not require H, H' and D to be nonnegative.

Geometric interpretation of BSSMF Solving BSSMF is equivalent to finding a polytope with r vertices within the hyperrectangle defined by $[a, b]$ that reconstructs as well as possible the data points. The fact that BSSMF is constrained within a hyperrectangle makes BSSMF more constrained than NMF, and hence more likely to be essentially unique. This will be illustrated empirically in Section 3.4.2. Let us provide a toy example to better understand the distinction between NMF and BSSMF, namely let us use Example 3.1 with $W = \frac{3}{10}A_{2/3}$ and $H = \frac{2}{3}A_{1/2}$ so that $X = WH$ is column stochastic. Figure 3.3 represents the feasible regions of NMF and BSSMF for the hypercube $[a, b] = [0, \frac{3}{10}]^3$ in a two-dimensional space within the affine hull of W; see [41] for the details on how to construct such a representation. For this rank-3 factorization problem, solving NMF and BSSMF is equivalent to finding a triangle nested between the convex hull of the data points and the corresponding feasible region. BSSMF has a unique solution, that is, there is a unique triangle between the data points and the BSSMF feasible region. On the other hand, NMF is not identifiable: for example, any triangle within the gray area containing the data points is a solution.

In summary, for the BSSMF of $X = WH$ to be essentially unique, W must contain sufficiently many entries equal to the lower and upper bounds, while H must be sufficiently sparse.

Choice of a and b In practice, if a and b are unknown, it may be beneficial to choose them such that as many entries of X are equal to the lower and upper bounds, and hence BSSMF is more likely to be identifiable. Let us denote $\tilde{a}_i = \min_j X(i, j)$ and $\tilde{b}_i = \max_j X(i, j)$ for all i, and let $X = WH$ be a BSSMF for the hyperrectangle $[a, b]$. We have $\tilde{a} \geq a$ and $\tilde{b} \leq b$ since $H(:, j) \in \Delta^r$ for all j. Hence, without any prior information, it makes sense to use a BSSMF with interval $[\tilde{a}, \tilde{b}]$ which is contained in $[a, b]$. Note that such strategy will make BSSMF sensitive to outliers that are out of the ideal bounds a and b.

Remark 3.1 *Interestingly, as shown in Lemma 3.2 below, in the exact case, that is, when $X = WH$, we can assume w.l.o.g. that $[a_i, b_i] = [0, 1]$ for all i in BSSMF.*

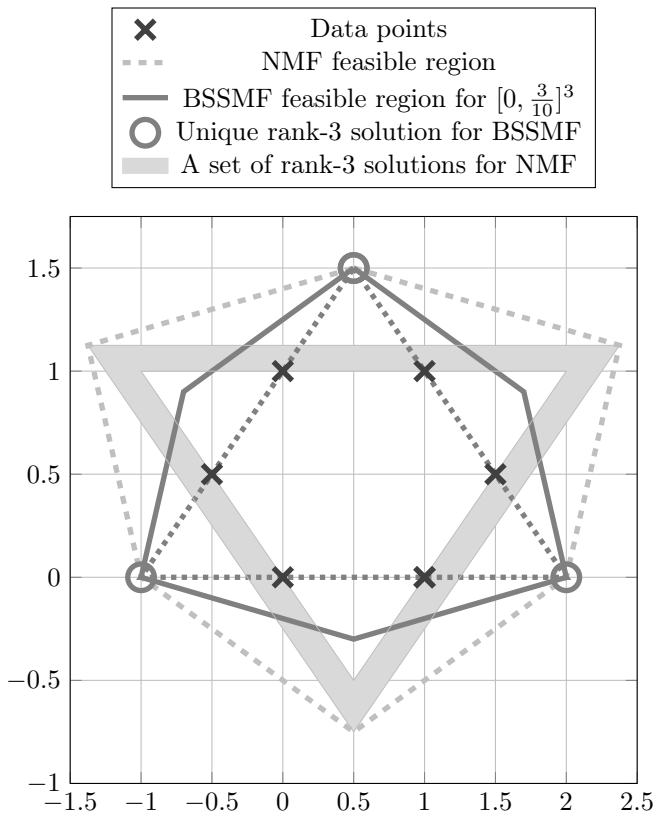


Figure 3.3: Geometric interpretation of BSSMF for Example 3.1. Any triangle in the gray filled area containing the data points is a rank-3 solution for NMF. On the contrary, there is a unique rank-3 solution for BSSMF since there is a unique triangle containing the data points in the BSSMF feasible set.

Lemma 3.2 *Let $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^m$ be such that $a_i < b_i$ for all i. The matrix $X = WH$ admits a BSSMF for the interval $[a, b]$ if and only if the matrix $\frac{[X - ae^\top]}{[(b-a)e^\top]}$ admits a BSSMF for the interval $[0, 1]^m$, where $\frac{[\cdot]}{[\cdot]}$ is the component-wise division of two matrices of the same size.*

Proof 3.3 *Let us show the direction $\Rightarrow$, the other is obtained exactly in the same way. Let the matrix $X = WH$ admit a BSSMF for the interval $[a, b]$. We have*

$$X - ae^\top = WH - ae^\top = (W - ae^\top)H,$$

since $e^\top H = e^\top$, as H is column stochastic. This shows that $X' = X - ae^\top$ admits a BSSMF for the interval $[0, b - a]$ since $W' = (W - ae^\top) \in [0, b - a]$. For simplicity, let us denote $c = b - a > 0$. We have $X' = W'H$, while

$$\frac{[X - ae^\top]}{[(b-a)e^\top]} = \frac{[X']}{[ce^\top]} = \frac{[W'H]}{[ce^\top]} = \frac{[W']}{[ce^\top]}H,$$

because H is column stochastic. In fact, for all i, j,

$$\begin{aligned}\frac{[W'H]_{i,j}}{[ce^\top]_{i,j}} &= \frac{\sum_k W'(k,i)H(k,j)]_{i,j}}{c_i} \\ &= \sum_k \frac{W'(k,i)}{c_i} H(k,j) \\ &= \left(\frac{[W']}{[ce^\top]}H\right)_{i,j}.\end{aligned}$$

Hence, $\frac{[X - ae^\top]}{[(b-a)e^\top]}$ admits a BSSMF for the interval $[0, 1]^m$ since H is column stochastic, and all columns of $\frac{[W']}{[ce^\top]} = \frac{[W - ae^\top]}{[(b-a)e^\top]}$ belong to $[0, 1]^m$.

Remark 3.2 (What if $a_i = b_i$ for some i?) *Lemma 3.2 does not cover the case $a_i = b_i$ for some i. In that case, we have $W(i,:) = a_i = b_i$ and therefore $X(i,:) = W(i,:)H = a_i e^\top = b_i e^\top$. This is not an interesting situation, and rows of X with identical entries can be removed. In fact, after the transformation $X - ae^\top$, these rows are identically zero.*

Lemma 3.2 highlights another interesting property of BSSMF: as opposed to NMF, it is invariant to translations of the entries of the input matrix, given that a and b are translated accordingly. For example, in recommender systems datasets such as Netflix and MovieLens, $X(i, j) \in \{1, 2, 3, 4, 5\}$ for all i, j. Changing the scale, say to $\{0, 1, 2, 3, 4\}$, does not change the interpretation of the data, but will typically impact the NMF solution significantly[3], while the BSSMF solution will be unchanged, if the interval is translated from $[1, 5]$ to $[0, 4]$ since H is invariant by translation on X. This property is in fact coming from SSMF.

[3]In fact, for NMF, it would make more sense to work on the datasets translated to $[0, 4]$, as it would potentially allow it to be identifiable: zeros in X imply zeros in W and H, which are therefore more likely to satisfy the SSC.

Tightness of Theorem 3.1 Unfortunately, the conditions in Theorem 3.1 are not necessary. This is due to SSC not being necessary for the uniqueness of NMF. Here is an example with

$$X = \begin{pmatrix} 0.25 & 0.25 & 0.75 & 0.75 \\ 0.2 & 0.6 & 0.6 & 0.2 \\ 0.75 & 0.75 & 0.25 & 0.25 \\ 0.8 & 0.4 & 0.4 & 0.8 \end{pmatrix}.$$

The unique BSSMF of X of size 3 with the bounds 0 and 1 is given by

$$X = \underbrace{\begin{pmatrix} 0 & 0.5 & 1 \\ 0.2 & 1 & 0.2 \\ 1 & 0.5 & 0 \\ 0.8 & 0 & 0.8 \end{pmatrix}}_{W} \underbrace{\begin{pmatrix} 0.75 & 0.5 & 0 & 0.25 \\ 0 & 0.5 & 0.5 & 0 \\ 0.25 & 0 & 0.5 & 0.75 \end{pmatrix}}_{H}.$$

However, H cannot be SSC since there are not at least $r - 1 = 2$ zeros per row. The matrix $[W^\top \; J - W^\top]$ is sparse enough, yet, it cannot be SSC since its cone does not contain $e - e_i$ for i in $1, \ldots, 3$. See [42, Chapter 4.2.5] for the details on how the aforementioned factorization $X = WH$ is a unique NMF, and hence, a unique BSSMF for our chosen bounds.

3.4 Numerical experiments

The goal of this section is to highlight the motivation points mentioned in Section 3.1 on real data sets. All experiments are run on a PC with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and 16GiB RAM. Let us recall that in order to retrieve NMF from Algorithm 3.1, the bounds need to be set to $(a, b) = (0, +\infty)$ and the projection step on the probability simplex in Algorithm 3.1 should be replaced by a projection on the nonnegative orthant. Hence, in our experiments, both NMF and BSSMF are solved with the same code implementation.

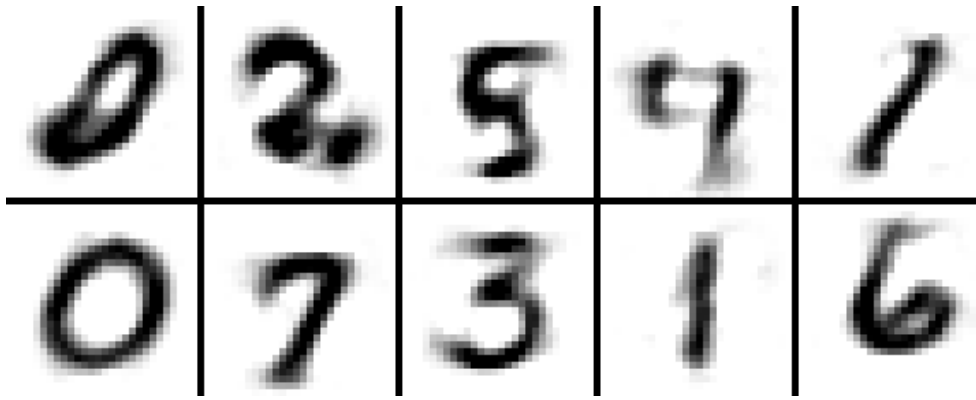
3.4.1 Interpretability

When applied on a pixel-by-image matrix, NMF allows to automatically extract common features among a set of images. For example, if each row of X is a vectorized facial image, the rows of W will correspond to facial features [62].

Let us compare NMF with BSSMF on the widely used MNIST handwritten digits dataset ($60,000$ images, 28×28 pixels) [60]. Each column of X is a vectorized handwritten digit. For BSSMF to make more sense, we preprocess X so that the intensities of the pixels in each digit belong to the interval $[0, 1]$ (first remove from $X(:, j)$ its minimum entry, then divide by the maximum entry minus the minimum entry).

Let us take a toy example with $n = 500$ randomly selected digits and $r = 10$, in order to visualize the natural interpretability of BSSMF. The choice of n is made solely for computational time considerations. For larger n, Figure 3.4b might change

(a) NMF



(b) BSSMF

Figure 3.4: Reshaped columns of the basis matrix W for $r = 10$ for MNIST with 500 digits.

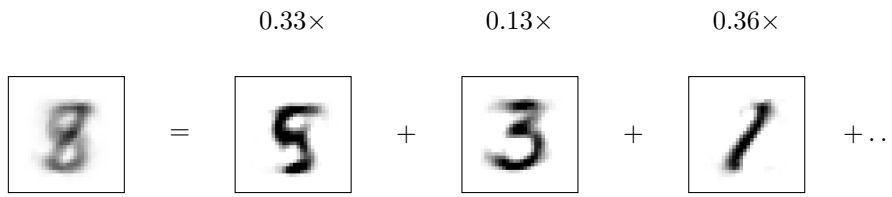


Figure 3.5: Decomposition of an eight by BSSMF with $r = 10$.

but we will not lose interpretability. Figure 3.4a shows the features learned by NMF which look like parts of digits. On the other hand, the features learned by BSSMF in Figure 3.4b look mostly like real digits, because of the bound constraint and the simplex structure. In fact, as it is well known [62] that NMF learns part-based representations, in this case, parts of digits. In other words, the columns of W in NMF identify subset of pixels that are activated simultaneously in as many images as possible. Now, by the scaling degree of freedom, assume w.l.o.g. that $W(:, j) \in [0, 1]^m$ for all j in NMF. Since the columns of W are parts of digits, each digit will have to use several of these parts, with an intensity close to one, so that H will be far from being column stochastic. BSSMF, with the simplex constraint on H and the bound constraints on W, therefore cannot learn such a part-based representation. This is the reason why BSSMF learns more global features that, added on top of each other, reconstruct the digits. As it is shown in the MNIST experiment, these features look like digits themselves. Interestingly, if we progressively increase the upper bound, we would see that BSSMF progressively learns parts of digits, like NMF (using a lower bound of zero, that is, BSSMF with $[0, u]^m$ with $u \geq 1$). This is an indirect way of balancing the sparsity between W and H. The larger the upper bound, the more relaxed is BSSMF and hence the sparser W will be (given that the lower bound is 0). In Figure 3.4b, we distinguish numbers (like 7, 3 and 6). From a clustering point of view, this is of much interest because a column of H which is near a ray of the probability simplex can directly be associated with the corresponding digit from W. In this toy example, due to r being small, an 8 cannot be seen. Nonetheless, an eight can be reconstructed as the weighted sum of the representations of a 5, a 3 and an italic 1; see Figure 3.5 for an example. Note that since BSSMF is more constrained than NMF, its reconstruction error might be larger than that of NMF. For our example ($r = 10$), BSSMF has relative error $\|X - WH\|_F / \|X\|_F$ of 61.56%, and NMF of 59.04%. This is not always a drawback. In some applications, due to the presence of noise, although the reconstruction error of BSSMF is larger than that of NMF, the accuracy of the estimated factors W and H could be better, because it uses the prior information and is less prone to overfitting and less sensitive to outliers that might be outside the bounds. See also the discussion in Section 3.4.3 where NMF has a lower RMSE than BSSMF on the training set, but a larger RMSE than BSSMF on the test set. Note that we also compute NMFs using Algorithm 3.1 where the projections are performed on the nonnegative orthant, instead of on the bounded set for W and on the probability simplex for H. The stopping criteria in Algorithm 3.1 of Algorithm 3.1 are a maximum number of iterations equal to 500, 20 and 20, respectively, for both

algorithms.

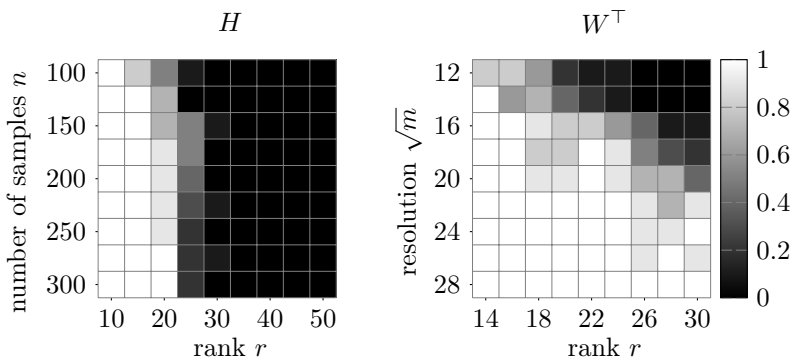
3.4.2 Identifiability

As it is NP-hard to check the SSC [54], we perform experiments on MNIST and synthetic data where only a necessary condition for SSC1 is verified, namely [41, Alg. 4.2].

MNIST dataset On MNIST, to see when H satisfies this condition, we first vary n from 100 to 300 for m fixed (=28×28). For $W^\top$, we fix n to 300, and downscale the resolution m from 28×28 to 12×12 with a linear interpolation (`imresize3` in MATLAB), and the rank r is varied from 12 to 30. Recall that both factors need to satisfy the SSC to correspond to an essentially unique factorization. In Figure 3.6a, we see that $W^\top$ of NMF often satisfies the necessary condition. This is due to NMF learning "parts" of objects [62], which are sparse by nature, and sparse matrices are more likely to satisfy the SSC (Definition 2.8). On the contrary, even for a relatively large n, H is too dense to satisfy the necessary condition. For $r \geq 30$, the factor H generated by NMF never satisfies the condition. Meanwhile, in Figure 3.6b we see that H of BSSMF always satisfies the condition when $n \geq 225$ for $r = 30$ and more generally, if n and m are large enough, both H and $\begin{pmatrix} W \\ J-W \end{pmatrix}^\top$ satisfy the necessary condition. This substantiates that BSSMF provides essentially unique factorizations more often than NMF does.

Synthetic datasets Let us now perform an experiment to show how BSSMF is more likely than NMF to recover factors closer to the true ones, even when the sufficient conditions for identifiability are not satisfied. As there is no groundtruth for NMF and BSSMF on MNIST, we generate synthetic data as follows. Our synthetic datasets are of size 100×100, and their factorization rank is 10. The matrix H is generated randomly with values uniformly distributed between zero and one, and we randomly set 30% of the values to zero. This allows us to ensure that H satisfies the SSC. The reason behind ensuring that H is SSC is that both NMF (Theorem 2.1) and BSSMF (Theorem 3.1) require that H satisfies the SSC[4]. As we want to emphasize on how likely it is to retrieve the true factors for NMF and BSSMF, we make sure that their common conditions for identifiability are satisfied. The matrix W is also generated randomly with values uniformly distributed between zero and one, and we then set a percentage of $p_{0,1}$ of the entries to zero and one, with the same probability to be equal to zero or one. Hence, $p_{0,1}$ percent of the values in W touches the lower and upper bounds in BSSMF. Finally, we let $X = WH$ to get our synthetic data. We solve NMF and BSSMF on X using Algorithm 3.1. To assess the quality of the solutions, we report the average of the mean removed spectral angle (MRSA) and the subspace angle (see Definition 6.2) between the columns of the true W and the estimated W (after an optimal permutation of the columns), as this is standard in

[4]In this experiment, because n and r are smaller, we could check that the SSC is satisfied (not a necessary condition), using Gurobi (`https://www.gurobi.com/`), a global optimization software.



(a) NMF

(b) BSSMF

Figure 3.6: Ratio, over 10 runs, of the factors generated by NMF in Figure 3.6a and by BSSMF in Figure 3.6b that satisfy the necessary condition for SSC1 (white squares indicate that all matrices meet the necessary condition, black squares that none do).

the NMF literature. Given any two vectors a and b, their MRSA is defined as

$$\text{MRSA}(a,b) = \frac{100}{\pi}\arccos\left(\frac{(a-\bar{a}e)^\top(b-\bar{b}e)}{\|a-\bar{a}e\|_2\|b-\bar{b}e\|_2}\right) \in [0,100]\,,$$

where $\bar{\cdot}$ is the average of the entries of a vector.

We vary the percentage $p_{0,1}$ of values touching the lower and uppper bounds in W (namely, 0 and 1) from 0% to 30% with a 5% increment. For each value of $p_{0,1}$, the test is performed 20 times. Let us note that among the generated true W's, between $p_{0,1} = 0\%$ and $p_{0,1} = 15\%$, $\binom{W}{J-W}^\top$ never satisfies the necessary conditions for SSC1. For $p_{0,1} = 20\%$, 3 out of the 20 generated $\binom{W}{J-W}^\top$ satisfies the necessary conditions for SSC1, 10 out of 20 for $p_{0,1} = 25\%$, and 17 out of 20 for $p_{0,1} = 30\%$. Let us also note that for all values of $p_{0,1}$ within the considered range, W never satisfies the necessary conditions for SSC1. The distribution of the average MRSAs and the subspace angle are respectively reported in Section 3.4.2 and Section 3.4.2. Clearly, the MRSA is always smaller for BSSMF compared to NMF, even when the necessary conditions for SSC1 are not satisfied for $\binom{W}{J-W}^\top$; this is because the feasible set of BSsMF is contained in that of NMF, and hence the generated factors are more likely to be closer to the ground truth. This also illustrates that the conditions of Theorem 3.1 for the identifiability of BSSMF are only sufficient, since BSSMF finds solutions with MRSA close to machine epsilon when these conditions are not fulfilled.

3.4.3 Robustness to overfitting

In this section we compare unconstrained matrix factorization (MF), NMF and BSSMF on the matrix completion problem; more precisely, on rating datasets for recommendation systems. Let X be an item-by-user matrix and suppose that user j has rated item i, that rating would be stored in $X_{i,j}$. The matrix X is then highly incomplete since a user has typically only rated a few of the items. In this context, NMF looks for nonnegative factors W and H such that $M \circ X \approx M \circ (WH)$, where $M_{i,j}$ is equal to 1 when user j rated item i and is equal to 0 otherwise. A missing rating $X_{i,j}$ is then estimated by computing $W(i,:)H(:,j)$. Features learned by NMF on rating datasets tend to be parts of typical users. Yet, the nonnegative constraint on the factors hardly makes the features interpretable by a practitioner. Suppose that the rating a user can give is an integer between 1 and 5 like in many rating systems, NMF can learn features whose values may fall under the minimum rating 1 or may exceed the maximum rating 5. Consequently, the features cannot directly be interpreted as typical users. On the contrary, with BSSMF, the extracted features will directly be interpretable if the lower and upper bounds are set to the minimum and maximum ratings. On top of that, BSSMF is expected to be less sensitive to overfitting than NMF since its feasible set is more constrained.

This last point will be highlighted in the following experiment on the ml-1m dataset[5], which contains 1 million ratings from 6040 users on 3952 movies. As in [67], we split the data in two sets : a training set and a test set. The test set contains 500

[5]https://grouplens.org/datasets/movielens/1m/

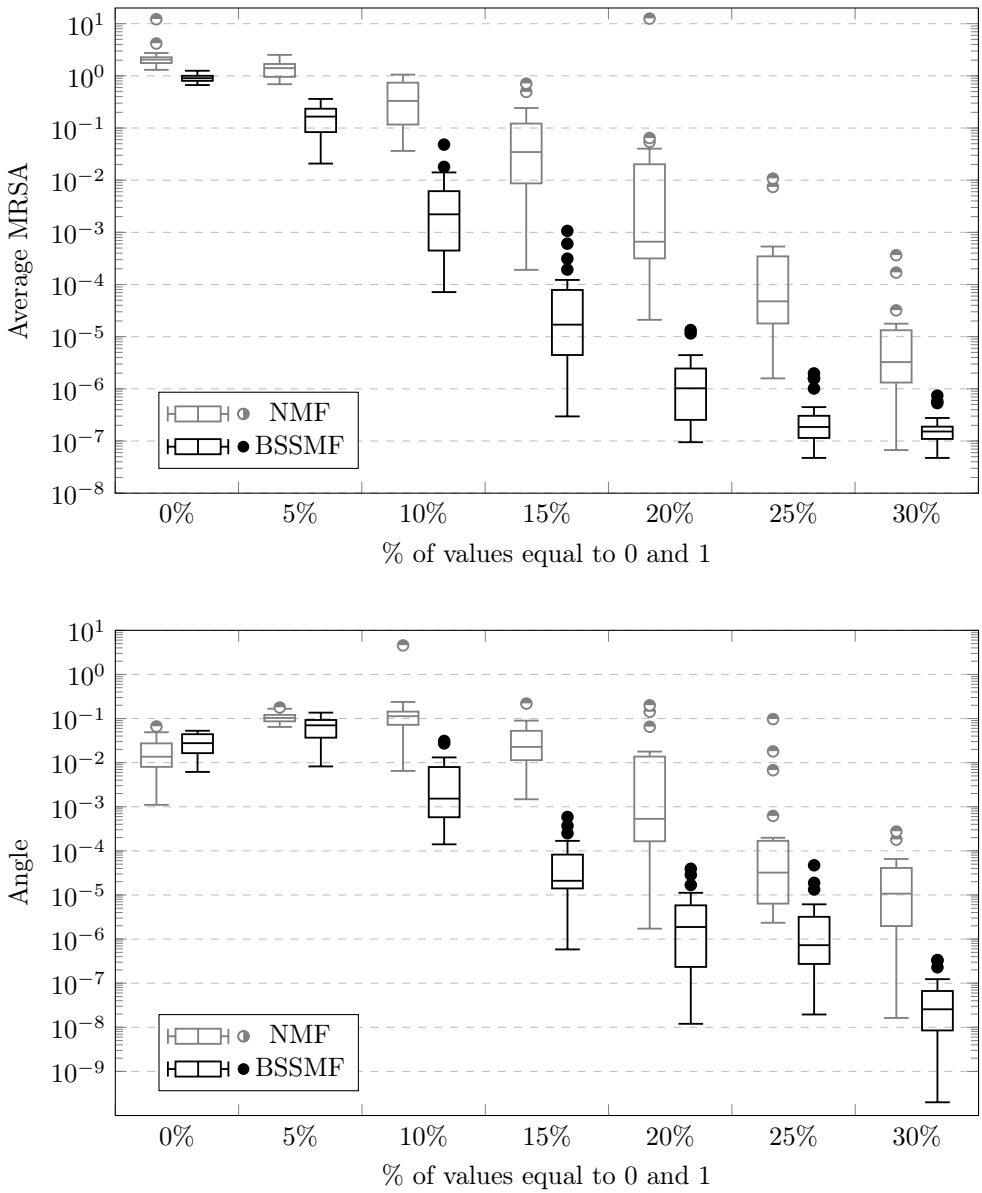


Figure 3.7: Boxplots of the average MRSA and subspace angle between the true W and the estimated W by NMF and BSSMF for the hypercube $[0,1]^{100}$ over 20 trials, depending on the percentage, $p_{0,1}$, of values equal to 0 and 1 in the true W.

r	BSSMF	NMF	MF
1	$0.97 \pm 2 \cdot 10^{-5}$	0.88 ± 0.002	$0.91 \pm 5 \cdot 10^{-6}$
5	0.87 ± 0.001	0.87 ± 0.003	0.87 ± 0.003
10	0.86 ± 0.002	0.87 ± 0.001	0.87 ± 0.002
20	0.87 ± 0.002	0.87 ± 0.002	0.88 ± 0.002
50	0.88 ± 0.002	0.90 ± 0.004	0.93 ± 0.004
100	0.89 ± 0.003	0.92 ± 0.003	0.99 ± 0.004

Table 3.4: RMSE on the test set according to r, averaged $\pm$ standard deviation on 10 runs on ml-1m

users. We also remove any movie that has been rated less than 5 times from both the training and test sets. For the test set, 80% of a user's ratings are considered as known. The remaining 20% are kept for evaluation. During the training, we learn W only on the training set. During the testing, the learned W is used to predict those 20% kept ratings of the test set by solving the H part only on the 80% known ratings. This simulates new users that were not taken into account during the training, but for whom we would still want to predict the ratings. The reported RMSEs are computed on the 20% kept ratings of the test set. In order to challenge the overfitting issue, we vary r in $\{1, 5, 10, 20, 50, 100\}$ for BSSMF, NMF and an unconstrained MF which are all computed using Algorithm 3.1, where the projections onto the feasible sets are adapted accordingly (projection onto the nonnegative orthant for NMF, no projection for unconstrained MF). The stopping criteria in Algorithm 3.1 of Algorithm 3.1 are a maximum number of iterations equal to 200, 1 and 1, respectively, for all algorithms. The experiment is conducted on 10 random initializations and the average RMSEs are reported is Table 3.4. As expected, BSSMF and NMF are more robust to overfitting than unconstrained MF. Additionnaly, BSSMF is also clearly more robust to overfitting than NMF. Its worse RMSE is 0.89 with $r = 100$ (and it is still equal to 0.89 with $r = 200$), while, for NMF, the RMSE is 0.92 when $r = 100$ (which is worse than a rank-one factorization giving a RMSE of 0.91).

The same experiment is conducted on the ml-100k dataset[6] which contains 100,000 ratings from 1,700 movies rated by 1,000 users. The test set contains 50 users. The results are reported in Table 3.5, and the observations are similar: BSSMF is significantly more robust to overfitting than NMF and unconstrained MF.

3.5 Conclusion

In this chapter, we proposed a new factorization model, namely bounded simplex structured matrix factorization (BSSMF). Fitting this model retrieves interpretable factors: the learned basis features can be interpreted in the same way as the original data while the activations are nonnegative and sum to one, leading to a straightforward soft clustering interpretation. Instead of learning parts of objects as NMF,

[6] https://grouplens.org/datasets/movielens/100k/

r	BSSMF	NMF	MF
1	$0.98 \pm 1 \cdot 10^{-4}$	$0.91 \pm 3 \cdot 10^{-5}$	$0.91 \pm 5 \cdot 10^{-5}$
5	0.89 ± 0.005	0.89 ± 0.01	0.89 ± 0.008
10	0.90 ± 0.008	0.90 ± 0.009	0.92 ± 0.01
20	0.91 ± 0.01	0.93 ± 0.01	0.97 ± 0.02
50	0.93 ± 0.01	0.97 ± 0.01	1.06 ± 0.03
100	0.94 ± 0.01	1.01 ± 0.007	1.13 ± 0.02

Table 3.5: RMSE on the test set according to r, averaged $\pm$ standard deviation on 10 runs on ml-100k

BSSMF learns objects that can be used to explain the data through convex combinations. We have proposed a dedicated fast algorithm for BSSMF, and showed that, under mild conditions, BSSMF is essentially unique. We also showed that the constraints in BSSMF make it robust to overfitting on rating datasets without adding any regularization term. Further work could include:

- the use of BSSMF for other applications,
- the design of more efficient algorithms for BSSMF, and
- the design of algorithms for other BSSMF models, e.g., with other data fitting terms such as the Kullback-Leibler divergence, as done recently in [66] for SSMF with nonnegativity constraint on W.

Chapter 4

Identifiability of Polytopic Matrix Factorization

Kino - Спокойная ночь

NMF is not essentially unique in general. However, it has been proven to be identifiable under the sufficiently scattered conditions (SSC). A geometric interpretation of these sufficient conditions is the following: while making sure that $X = WH$ and that W is nonnegative, it is not possible to decrease the "volume" of the cone of $W^\top$ without making the cone of H get out of the nonnegative orthant, and vice versa; see Section 2.3.1 for details.

In this chapter, we focus on the identifiability of polytopic matrix factorization (PMF) . With NMF, the feasible domain is the nonnegative orthant. With PMF, the feasible domains are convex polytopes: the columns of $W^\top$ and H belong to the polytopes $\mathcal{P}_W$ and $\mathcal{P}_H$, respectively. A variant of PMF has already been studied in [91, 92] where the authors proposed a structured matrix factorization where: (i) the matrix W is unconstrained, (ii) the columns of H belong to a convex polytope, and (iii) the goal is to find a factorization maximizing the volume of the convex hull of the columns of H. This model, proposed in [91, 92], is also referred to as PMF, although it would have been more appropriate to refer to it as maximum-volume PMF. In fact, their proposed model could be viewed as a polytopic variant of minimum-volume semi-NMF; see [35] and Section 2.3.2, while our proposed model would rather be a polytopic variant of NMF.

Outline and contribution of the chapter Inspired by the identifiability conditions in [92] and similarly to Theorem 2.1, our main contribution in this chapter is to show that if the convex hull of $W^\top$ and H are sufficiently scattered within their respective polytope, then the corresponding PMF is identifiable (Theorem 4.1).

In Section 4.1 we introduce PMF. Section 4.2 provides important definitions and properties. In Section 4.3 we prove our main result. Section 4.4 presents known structured matrix factorization that are special cases of PMF, and how our theoretical finding relates to previous results.

4.1 Polytopic Matrix Factorization

In this chapter, we consider convex polytopes, that is, bounded polyhedra. A convex polytope $\mathcal{P}$ can always be expressed in V-form, through a convex combination of its vertices:

$$\mathcal{P} = \text{conv}(V) = \{x \mid x = Vh, h \geq 0, \sum_i h_i = 1\}, \tag{4.1}$$

where the columns of V are the vertices, or the extremum points, of $\mathcal{P}$. We can now define PMF. Given a data matrix $X \in \mathbb{R}^{m \times n}$ and r, PMF computes W and H such that

$$\begin{aligned} X = WH \text{ s.t. } & W(i,:) \in \mathcal{P}_W \text{ for all } i \text{ in } 1, \ldots, m, \\ & H(:,j) \in \mathcal{P}_H \text{ for all } j \text{ in } 1, \ldots, n, \end{aligned} \tag{4.2}$$

where $W \in \mathbb{R}^{m \times r}$ is the basis matrix, $H \in \mathbb{R}^{r \times n}$ is the coefficient matrix, $\mathcal{P}_W$ and $\mathcal{P}_H$ are convex polytopes that respectively constrain the rows of W and the columns of H. This PMF is referred to as the quadruple $(W, H, \mathcal{P}_W, \mathcal{P}_H)$. This framework is quite general: it offers infinite varieties of structured matrix factorizations that promote different behaviors in the latent space, depending on the choice of $\mathcal{P}_W$ and $\mathcal{P}_H$. As we will show in Section 4.4, PMF recovers factorizations that have been studied in the literature.

4.2 Definitions and Properties

In this section, we provide important definitions and properties that are needed to achieve our main result on the identifiability of PMF (Theorem 4.1 in Section 4.3).

Identifiability. Let us clarify what is meant by identifiability. A PMF $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ is identifiable if for any other PMF $(W_*, H_*, \mathcal{P}_W, \mathcal{P}_H)$ of X, there exist a permutation matrix Π and a diagonal matrix D with diagonal values in $\{-1, 1\}$ such that $W_* = W\Pi^\top D^{-1}$ and $H_* = D\Pi H$. We will refer to a matrix of the form $D\Pi$ as a signed permutation. Essential uniqueness of PMF is stronger than the NMF one, as it only allows a sign ambiguity, while NMF allows a scaling ambiguity.

Maximum-volume ellipsoid and sufficient scatteredness. Our sufficient scatteredness conditions that guarantee identifiability heavily rely on the notion of ellipsoids. Given a center, $\bar{x} \in \mathbb{R}^r$, and a positive definite matrix E, an ellipsoid is defined as $\mathcal{E}(E, \bar{x}) := \left\{x \in \mathbb{R}^r | (x - \bar{x})^\top E (x - \bar{x}) \leq r\right\}$. Its volume is given by $\text{vol}(\mathcal{E}(E, \bar{x})) = \frac{r^{r/2}\Omega_r}{\sqrt{\det(E)}}$ where Ω_r is the volume of a ball of radius 1 in $\mathbb{R}^r$. The axis of the ellipsoid are given by the eigenvectors of E, and their length is inversely proportional to the square root of corresponding eigenvalues; see, e.g., [93]. Given an ellipsoid $\mathcal{E}(E, \bar{x})$ and an invertible matrix Q, it can be shown that $Q(\mathcal{E}(E, \bar{x})) = \{Qx | x \in \mathcal{E}\} = \mathcal{E}(Q^{-\top} E Q^{-1}, \bar{y})$ where $\bar{y} = Q\bar{x}$, and hence the volume of $Q\mathcal{E}$ equals the volume of $\mathcal{E}$ times $|\det(Q)|$. This will be useful in our identifiability proof.

The Maximum-Volume Inscribed Ellipsoid (MVIE) of a polytope $\mathcal{P}$, denoted $\mathcal{E}_\mathcal{P}$, is defined as the ellipsoid $\mathcal{E}_\mathcal{P} \subset \mathcal{P}$ with maximum volume $\mathrm{vol}(\mathcal{E}(E, \bar{x}))$, that is, for which $\det(E)$ is minimized. It can be computed by solving a convex semidefinite program; see, e.g., [13, Chap. 8.4.2]. A convex set is said to be sufficiently scattered relative to a polytope when it is contained in that polytope while containing the MVIE of this polytope [92].

Our identifiability result will be based on the following sufficient scatteredness condition:

Definition 4.1 (Sufficiently Scattered Factor [92]) *The matrix $H \in \mathbb{R}^{r\times n}$ is called a sufficiently scattered factor (SSF) corresponding to $\mathcal{P}$ if*

[PMF.SSC1] $\mathcal{P} \supseteq \mathrm{conv}(H) \supset \mathcal{E}_\mathcal{P}$, *and*

[PMF.SSC2] $\mathrm{conv}(H)^{*,g_\mathcal{P}} \cap \mathrm{bd}(\mathcal{E}_\mathcal{P}^{*,g_\mathcal{P}}) = \mathrm{ext}(\mathcal{P}^{*,g_\mathcal{P}})$,
where $\mathcal{E}_\mathcal{P}$ is the MVIE of $\mathcal{P}$ centered at $g_\mathcal{P}$.

The idea behind the condition [PMF.SSC1] is similar to [SSC1] in Theorem 2.1, as both conditions ensure that the considered factor is sufficiently scattered within its feasible set. The MVIE acts like the second order cone $\mathcal{C}$ in [SSC1] which is the largest cone contained in the nonnegative orthant. Here, [PMF.SSC1] ensures that the convex hull of a factor H is contained in the polytope $\mathcal{P}$ and contains the MVIE of $\mathcal{P}$. The second condition [PMF.SSC2] makes sure that the MVIE is not contained too tightly. Let us illustrate why [PMF.SSC2] is important with the PMF $(H^\top, H, \Delta^3, \Delta^3)$ using Example 3 from [59], see also [54, Example 2]:

$$H = \frac{1}{3}\begin{pmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 & 2 & 1 \end{pmatrix}. \tag{4.3}$$

As it can be seen on Fig. 4.1a, H satisfies [PMF.SSC1]. However, Fig. 4.1b exposes why H does not satisfy [PMF.SSC2], and it turns out that the PMF $(H^\top, H, \Delta^3, \Delta^3)$ is not identifiable:

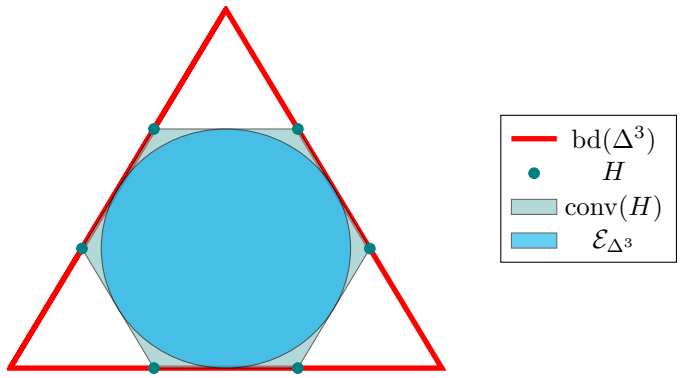
$$Q = \frac{1}{3}\begin{pmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{pmatrix}$$

provides another PMF, $(H^\top Q^\top, QH, \Delta^3, \Delta^3)$, while QH is not a signed permutation of the rows of H:

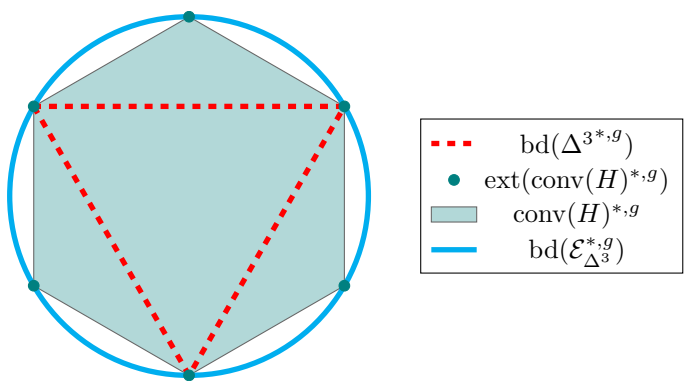
$$QH = \frac{1}{3}\begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Permutation-and/or-sign-only invariant sets. In addition to the sufficient scatteredness, the identifiability of PMF will rely on the following condition for the sets of vertices of $\mathcal{P}_W$ and $\mathcal{P}_H$.

Definition 4.2 *A set $\mathcal{X}$ is called a permutation-and/or-sign-only invariant (PSOI) set if, and only if, every linear transformation A such that $A(\mathcal{X}) = \mathcal{X}$ is a signed*



(a) Visualization of why H satisfies [PMF.SSC1].



(b) Visualization of why H does not satisfy [PMF.SSC2].

Figure 4.1: A small example, with H from Eq. (4.3) and $g = \begin{pmatrix}1/3 & 1/3 & 1/3\end{pmatrix}^\top$, showing how [PMF.SSC1] can be satisfied without [PMF.SSC2] being satisfied.

permutation, that is, $A = D\Pi$ *where* Π *is a permutation matrix and* D *is a diagonal matrix with diagonal entries in* $\{-1, 1\}$.

The set of vertices of full-dimensional polytopes will in most cases be PSOI sets.

Lemma 4.1 *Let the columns of* $V \in \mathbb{R}^{r\times n}$ *contain the vertices of the polytope* $\mathcal{V} \subset \mathbb{R}^r$ *and such that* $\operatorname{rank}(V) = r$ *(this holds for full-dimensional polytopes). Let* $A \in \mathbb{R}^{r\times r}$ *be such that* $AV = V(:, \Pi)$ *for some permutation* Π. *Then* A *is an orthogonal matrix, that is, a rotation of* $\mathbb{R}^r$.

Proof 4.1 *Since* A *permutes the columns of* V, *and the set of permutations is finite, there exists* n *such that* $A^nV = V$. *Since* V *has rank* r, *it admits a right inverse, so that* $A^n = I_r$, *where* I_r *is identity matrix of dimension* r. *This implies that the eigenvalues of* A *are roots of 1, and hence* A *is orthogonal, that is,* $A^\top A = I_r$.

In two dimensions, sets that are not PSOI are any regular polygon centered at the origin, except for the square (which is obtained by a rotation of 90 or 180 degrees in which case A is a signed permutation). For example, the vertices of the regular triangle given by the columns of

$$V = \begin{pmatrix} 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1 & -1/2 & -1/2 \end{pmatrix}$$

are preserved by a rotation of 120 degrees, corresponding to $A = \begin{pmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{pmatrix}$, and $AV = \begin{pmatrix} \sqrt{3}/2 & -\sqrt{3}/2 & 0 \\ -1/2 & 1/2 & 1 \end{pmatrix}$.

In Section 4.4, we will use two polytopes: Δ^r and $[a, b]^r$ for $b > a$. Let us show that their vertices are PSOI sets. For Δ^r, this is trivial since $\Delta^r = \operatorname{conv}(I_r)$, hence any A that satisfies $AI_r = I_r(:, \Pi)$ for some permutation Π must be a permutation (note there is no sign ambiguity possible here). For the hypercube $[a, b]^r$, let us first prove the following lemma.

Lemma 4.2 *Let* $a < b$ *be scalars, and* $d \in \mathbb{R}^r$ *with* $\|d\|_2 = 1$ *be such that* $d^\top x \in \{a, b\}$ *for all* $x \in \{a, b\}^r$. *Then* d *is a unit vector, up to multiplication by -1.*

Proof 4.2 *Let us prove the result by induction. For* $r = 1$, *the result is trivial, we must have* $d = 1$. *Assume the result holds for all* $r' < r$, *and let us denote* $d = [d_{r-1}, d_r]$ *with* $d_{r-1} \in \mathbb{R}^{r-1}$, *and similarly for* x. *We have for all* $x \in \{a, b\}^r$ *that*

$$d^\top x = d_{r-1}^\top x_{r-1} + d_r x_r \in \{a, b\}.$$

If $d_r \in \{-1, 0, 1\}$, *the result follows by induction since* $\|d\|_2 = 1$. *Hence, assume* $d_r \notin \{-1, 0, 1\}$. *We have*

$$d_{r-1}^\top x_{r-1} + d_r a \in \{a, b\} \quad \text{and} \quad d_{r-1}^\top x_{r-1} + d_r b \in \{a, b\}.$$

Let us denote $\alpha = d_{r-1}^\top x_{r-1}$, *we have*

$$\alpha \in \{a - d_r a, b - d_r a\} \quad \text{and} \quad \alpha \in \{a - d_r b, b - d_r b\}.$$

Since $a \neq b$, $a - d_r a \neq a - d_r b$ and $b - d_r a \neq b - d_r b$ as $d_r \neq 0$, $a - d_r a \neq b - d_r b$ as $d_r \neq 1$, and $b - d_r a \neq b - d_r b$ as $d_r \neq -1$. Hence, α cannot exist for $x_r \in \{a, b\}$, a contradiction.

Corollary 4.1 *The set of vertices of $[a, b]^r$ is a PSOI set.*

Proof 4.3 *The set of vertices of $[a, b]^r$ are all vectors in $\{a, b\}^r$. Let the columns of $V \in \mathbb{R}^{r \times 2^r}$ contain the vertices of $[a, b]^r$, and the linear transformation A satisfy $AV = V(:, \Pi)$ for some permutation Π. By Lemma 4.1, A is orthogonal hence its rows have unit ℓ_2 norm. This implies that every row of A must satisfy the condition of Lemma 4.2 and hence are unit vectors. Since rows of A are orthogonal, A must be a signed permutation.*

4.3 Identifiability

We can now state our main result: it fills a gap in the literature by combining the ideas of the identifiability of maximum-volume PMF in [92], and of NMF in [54].

Theorem 4.1 *Let $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ be a PMF of X of size $r = \operatorname{rank}(X)$. If $W^\top$ and H are SSFs, and $\operatorname{ext}(\mathcal{P}_W)$ and $\operatorname{ext}(\mathcal{P}_H)$ are PSOI sets, then the PMF $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ of $X = WH$ of size $r = \operatorname{rank}(X)$ is identifiable.*

Proof 4.4 *This proof follows that from [92, Th. 6] where only H is required to be sufficiently scattered while its volume is maximized. Let $Q \in \mathbb{R}^{r \times r}$ be an invertible matrix such that (WQ^{-1}, QH) is a PMF of X with*

$$\operatorname{conv}(Q^{-\top} W^\top) \subseteq \mathcal{P}_W \quad \text{and} \quad \operatorname{conv}(QH) \subseteq \mathcal{P}_H. \tag{4.4}$$

Since $W^\top$ and H are sufficiently scattered factors, their convex hull contains their corresponding MVIE:

$$\mathcal{E}_{\mathcal{P}_W} \subset \operatorname{conv}(W^\top) \quad \text{and} \quad \mathcal{E}_{\mathcal{P}_H} \subset \operatorname{conv}(H). \tag{4.5}$$

Then, Eq. (4.4) leads to

$$Q^{-\top}(\mathcal{E}_{\mathcal{P}_W}) \subseteq \mathcal{P}_W \quad \text{and} \quad Q(\mathcal{E}_{\mathcal{P}_H}) \subseteq \mathcal{P}_H. \tag{4.6}$$

The set $Q^{-\top}(\mathcal{E}_{\mathcal{P}_W})$ (resp. $Q(\mathcal{E}_{\mathcal{P}_H})$) is still an ellipsoid of volume $|\det(Q^{-1})| \operatorname{vol}(\mathcal{E}_{\mathcal{P}_W})$ (resp. $|\det(Q)| \operatorname{vol}(\mathcal{E}_{\mathcal{P}_H})$). By definition of the MVIE, we have

$$\begin{aligned} &|\det(Q^{-1})| \operatorname{vol}(\mathcal{E}_{\mathcal{P}_W}) \leq \operatorname{vol}(\mathcal{E}_{\mathcal{P}_W}) \text{ and } |\det(Q)| \operatorname{vol}(\mathcal{E}_{\mathcal{P}_H}) \leq \operatorname{vol}(\mathcal{E}_{\mathcal{P}_H}) \\ \Leftrightarrow &|\det(Q^{-1})| \leq 1 \text{ and } |\det(Q)| \leq 1 \Leftrightarrow |\det(Q)| = 1. \end{aligned}$$

This implies that $Q^{-\top}$ and Q respectively map $\mathcal{E}_{\mathcal{P}_W}$ and $\mathcal{E}_{\mathcal{P}_H}$ onto themselves :

$$Q^{-\top}(\mathcal{E}_{\mathcal{P}_W}) = \mathcal{E}_{\mathcal{P}_W} \quad \text{and} \quad Q(\mathcal{E}_{\mathcal{P}_H}) = \mathcal{E}_{\mathcal{P}_H}. \tag{4.7}$$

The remaining of the proof is exactly like in the remaining proof of [92, Th. 6] by focusing on either H or $W^\top$. Focus on H for example, and using [PMF.SSC2], the idea is to show that $Q(\operatorname{ext}(\mathcal{P}_H)) = \operatorname{ext}(\mathcal{P}_H)$. Then, because $\operatorname{ext}(\mathcal{P}_H)$ is a PSOI set, Q has to be a signed permutation.

The last part of the proof of Theorem 4.1 does not rely on both $W^\top$ and H satisfying [PMF.SSC2], and on both $\text{ext}(\mathcal{P}_W)$ and $\text{ext}(\mathcal{P}_H)$ being PSOI sets. Actually, Theorem 4.1 remains valid if only one the factors satisfies [PMF.SSC2] and if the vertices of its corresponding polytope form a PSOI set.

Corollary 4.2 *Let $W^\top$ and H satisfy [PMF.SSC1] and*
(i) $W^\top$ satisfy [PMF.SSC2] and $\text{ext}(\mathcal{P}_W)$ be a PSOI set,
or
(ii) H satisfy [PMF.SSC2] and $\text{ext}(\mathcal{P}_H)$ be a PSOI set,

then the PMF $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ of $X = WH$ of size $r = \text{rank}(X)$ is identifiable.

Proof 4.5 *The same proof as Theorem 4.1 applies. By symmetry, whether it is (i) or (ii) that is verified allows us to conclude that Q is a signed permutation.*

The PSOI set condition can be relaxed to sets that are "mutually" PSOI, that is, there cannot exist a matrix A which is not a signed permutation such that $A^{-\top}(\text{ext}(\mathcal{P}_W)) = \text{ext}(\mathcal{P}_W)$ and $A(\text{ext}(\mathcal{P}_H)) = \text{ext}(\mathcal{P}_H)$.

Corollary 4.3 *Let $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ be a PMF of X of size $r = \text{rank}(X)$. If $W^\top$ and H are SSFs, and $\text{ext}(\mathcal{P}_W)$ and $\text{ext}(\mathcal{P}_H)$ are mutually PSOI sets, then the PMF $(W, H, \mathcal{P}_W, \mathcal{P}_H)$ of $X = WH$ of size $r = \text{rank}(X)$ is identifiable.*

Proof 4.6 *The same proof as Theorem 4.1 applies up to Eq. (4.7). Then, $W^\top$ and H satisfying [PMF.SSC2] leads to $Q^{-\top}(\text{ext}(\mathcal{E}_{\mathcal{P}_W})) = \text{ext}(\mathcal{E}_{\mathcal{P}_W})$ and $Q(\text{ext}(\mathcal{E}_{\mathcal{P}_H})) = \text{ext}(\mathcal{E}_{\mathcal{P}_H})$. Then, because $\text{ext}(\mathcal{P}_W)$ and $\text{ext}(\mathcal{P}_H)$ are mutually PSOI sets, Q has to be a signed permutation.*

4.4 Examples of PMF

In this section, we show that some known constrained matrix factorizations are special instances of PMF, and explain how Theorem 4.1 relates to known identifiability results for these special cases.

4.4.1 Nonnegative Matrix Factorization (NMF)

An NMF, $X = WH$, requires W and H to be component-wise nonnegative. This is not a PMF since the nonnegative orthant is unbounded. However, if $W^\top$ and H do not contain a column full of zeros (which can be assumed w.l.o.g.), then there exist two diagonal matrices, D_l and D_r, such that $D_l W e = e$ and $e^\top H D_r = e^\top$. Hence, we can transform the NMF $X = WH$ into the PMF $(\tilde{W}, \tilde{H}, \Delta^r, \Delta^r)$ of $\tilde{X}$ with $\tilde{X} = D_l X D_r$, where $\tilde{W} = D_l W$ and $\tilde{H} = H D_r$.

Interestingly, the identifiability conditions for NMF in Theorem 2.1 and for PMF in Theorem 4.1 are equivalent, because $\tilde{H}$ satisfies the SSC in Definition 2.8 *if and only if* $\tilde{H}$ is an SSF according to Definition 4.1, while $\text{ext}(\Delta^r)$ is a PSOI set (see Section 4.2). This is due to the fact that $\mathcal{E}_{\mathcal{P}_W} = \mathcal{E}_{\mathcal{P}_H} = \mathcal{C} \cap \Delta^r$, since the MVIE of

Δ^r is an $(r-1)$-dimensional ball centered at $\frac{1}{r}e$ of radius $\frac{1}{\sqrt{r(r-1)}}$, within the affine subspace $\{x \in \mathbb{R}^r,\ e^\top x = 1\}$. Indeed, the diagonal matrices are just rescaling the rows of W and the columns of H such that they belong to Δ^r. Hence, $\mathcal{C} \cap \Delta^r \subseteq \text{conv}(\tilde{H})$ *if and only if* $\mathcal{C} \subseteq \text{cone}(H)$, and by symmetry this also holds for $\tilde{W}^\top$ and $W^\top$.

4.4.2 Factor-Bounded Matrix Factorization

Factor-bounded matrix factorization (FBMF) requires the elements of each factor to be bounded. Given $a < b \in \mathbb{R}$, we write $a \leq W \leq b$ if $a \leq W(i,k) \leq b$ for all (i,k).

Definition 4.3 (Factor-Bounded MF) *Let $X \in \mathbb{R}^{m \times n}$, r be an integer, $l_W < u_W \in \mathbb{R}$ and $l_H < u_H \in \mathbb{R}$. The pair $(W,H) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ is a FBMF of X of size r for the intervals $[l_W, u_W]$ and $[l_H, u_H]$ if*

$$X = WH \text{ such that } l_W \leq W \leq u_W, l_H \leq H \leq u_H. \tag{4.8}$$

This means that each row of W then belongs to the hypercube $[l_W, u_W]^r$ and each column of H belongs to the hypercube $[l_H, u_H]^r$. In [70], the authors propose a nonnegative FBMF (NFBMF), where $0 \leq l_W$ and $0 \leq l_H$ in Eq. (4.8). They showed that NFBMF is particularly well suited for clustering tasks. To the best of our knowledge, FBMF has never been proven to be identifiable. Since Eq. (4.8) is a PMF with the choice $\mathcal{P}_W = [l_W, u_W]^r$ and $\mathcal{P}_H = [l_H, u_H]^r$, Theorem 4.1 applies to FBMF. The MVIE $\mathcal{E}_{\mathcal{P}_W}$ is an r-dimensional ball centered at $\frac{u_W + 2l_W}{2}e$ of radius $\frac{u_W - l_W}{2}$, and similarly for $\mathcal{E}_{\mathcal{P}_H}$, while $\text{ext}(\mathcal{P}_W)$ and $\text{ext}(\mathcal{P}_H)$ are PSOI sets (Corollary 4.1).

4.4.3 Bounded Simplex-Structured Matrix Factorization

Bounded simplex-structured matrix factorization (BSSMF) was already presented in Chapter 3 as model useful to explain data that are convex combinations of vectors belonging to a hyperrectangle $[a,b]$, where $a \leq b \in \mathbb{R}^m$. The convex combinations are the columns of H and the vectors belonging to $[a,b]$ are the columns of W. For more details on BSSMF, refer to Chapter 3. BSSMF does not belong to the class of PMFs. The hyperrectangle constraint on the columns of W cannot in general be expressed as a polytopic constraint on the rows of W. However, when all entries of a, and of b, are equal to one another, the hyperrectangle constraint becomes a hypercube constraint that can be expressed by a polytopic row wise constraint. For example, when X corresponds to a set of vectorized images, the intensity of a pixel belongs to $[0,1]$. If there is no specific pixel position that should be bounded differently than the others, every row of W is bounded in the same way. In other words, the rows of W belong to the hypercube $[0,1]^r$. Another example is when X is a rating matrix whose entries are ordinal, e.g., the Netflix matrix with entries in $\{1,2,3,4,5\} \in [1,5]$. In these cases, BSSMF uses a hypercube $[a,b]^m$ and is equivalent to PMF since Equation (3.2) is equivalent to Eq. (4.2) with $\mathcal{P}_W = [a,b]^r$ and $\mathcal{P}_H = \Delta^r$. BSSMF was shown to be identifiable under conditions described in Theorem 3.1, different from the ones in Theorem 4.1. When BSSMF and PMF are equivalent, which identifiability theorem is the strongest? Since BSSMF is invariant by translation along e, we can assume

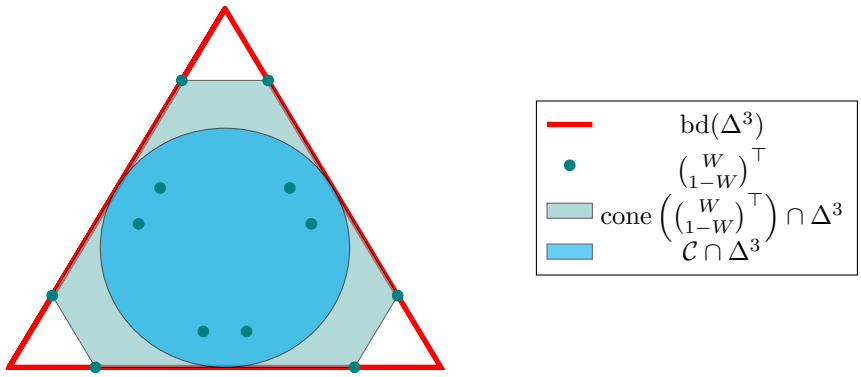
w.l.o.g. that $a = 0$ for the sake of simplicity. Also, we do not need to focus on the conditions for H. Indeed, when $H(:, j) \in \Delta^r$ for all j, [SSC1] is equivalent to [PMF.SSC1] because the MVIE of $\mathcal{P}_H = \Delta^r$ is equal to $\mathcal{C} \cap \Delta^r$. We then focus on the sufficient scatteredness of $W^\top$. The MVIE of $[0, b]^r$ is a ball $\mathcal{E}_{[0,b]^r}$ centered at $\frac{b}{2}e$ of radius $\frac{b}{2}$. This ball is tightly contained by $\mathcal{C}$, which means that for any convex set A that contains $\mathcal{E}_{[0,b]^r}$, $\mathcal{C} \subseteq \text{cone}(A)$. As a consequence, if $W^\top$ satisfies [PMF.SSC1], $W^\top$ satisfies [SSC1], which implies that $\begin{pmatrix} W \\ be^\top - W \end{pmatrix}^\top$ satisfies [SSC1]. However, it is possible that $\begin{pmatrix} W \\ be^\top - W \end{pmatrix}^\top$ satisfies [SSC1] while $W^\top$ does not satisfy [PMF.SSC1]. Here is an example with $\mathcal{P}_W = [0, 1]^3$:

$$W^\top = \begin{pmatrix} 0.8 & 0 & 0.2 & 0.2 & 0.8 & 1 \\ 0.2 & 0.8 & 0 & 1 & 0.2 & 0.8 \\ 0 & 0.2 & 0.8 & 0.8 & 1 & 0.2 \end{pmatrix}. \tag{4.9}$$

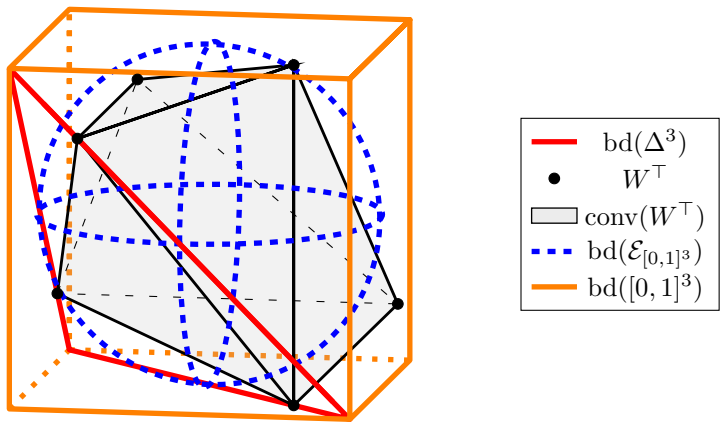
As it can be seen in Fig. 4.2a, the cone of $\begin{pmatrix} W \\ 1 - W \end{pmatrix}^\top$ contains $\mathcal{C}$ because W reaches enough times the minimum and maximum bounds 0 and 1. However, in Figure 4.2b the convex hull of $W^\top$ does not contain the MVIE of $[0, 1]^3$. Therefore, Theorem 4.1 is quite general but is not as strong as Theorem 3.1 for BSSMF.

4.5 Conclusion

We presented PMF, a structured matrix factorization model where the latent space of the factors is constrained by given polytopes. The choice of the polytopes should depend on the data and the application at hand. When the polytopes have certain invariant properties, we derived some sufficient conditions under which the identifiability of a PMF is guaranteed. Geometrically, these conditions are based on the scatteredness of the factors within the constraining polytopes.



(a) [SSC1] being satisfied.



(b) [PMF.SSC1] not being satisfied.

Figure 4.2: Visualization of $\binom{W}{1-W}^\top$ from Eq. (4.9) satisfying [SSC1] while $W^\top$ does not satisfy [PMF.SSC1]. The cone of $\binom{W}{1-W}^\top$ contains $\mathcal{C}$, while the convex hull of $W^\top$ does not contain the ball $\mathcal{E}_{[0,1]^3}$.

Chapter 5

Randomized Successive Projection Algorithm for Separable NMF

YĪN YĪN - One Inch Punch

In general, NMF is NP-hard [96] and not necessarily identifiable (Section 2.3.1), which are two main issues of NMF. However, under the *separability assumption*, it is solvable in polynomial time and is identifiable [5]. This assumption states that for every vertex (column of W), there exists at least one data point (column of X) equal to this vertex. In blind HU, which consists in identifying the materials present in a hyperspectral image as well as their distribution in the pixels of the image, this is known as the *pure-pixel assumption* and means that for each material, there is at least one pixel composed almost purely of this material. Many algorithms have been introduced that leverage this assumption, see for instance [42, Chapter 7] and the references therein. Recently, algorithms for separable NMF that are provably robust to noise have been introduced [5]. One of the most widely used is the successive projection algorithm (SPA) [4] .

SPA is robust to noise and generally works well in practice. However, it suffers from several drawbacks, notably it is sensitivity to outliers. SPA is deterministic, that is for a given problem it gives the same result at every run. It is also greedy, in the sense that it extract vertices sequentially, so an error at a given iteration cannot be compensated in the following iterations. In this chapter, we aim at addressing the sensitivity to outliers by designing a non-deterministic variant of SPA that could be run several times, in the hope that at least one run will not extract outliers.

Let us discuss an observation from [79]. The separable NMF algorithm called vertex component analysis (VCA) [80] includes a random projection, therefore it is non-deterministic and at each run it produces potentially a different result. VCA is simpler and its guarantees are weaker than those of SPA, and the experiments in [79] show that VCA performs worse than SPA on average, but they also show that the best result of VCA over many runs is in most cases better that the result of SPA in

terms of reconstruction error. This observation is our main motivation to design a non-deterministic variant of SPA, that we coin as randomized SPA (RandSPA).

Outline and contribution of the chapter In Section 5.1 we introduce the general form of recursive algorithm for separable NMF analyzed in [46] which generalizes SPA. In Section 5.2 we present the main contribution of this chapter, that is a randomized variant of SPA, called RandSPA . We show the theoretical results on the robustness to noise of SPA still hold for RandSPA, while the randomization allows to better handle outliers by allowing a diversity in the solutions produced. In Section 5.3 we illustrate the advantages of our method with experiments on both synthetic datasets and the unmixing of hyperspectral images.

5.1 Successive Projection Algorithm

In this section, we discuss the successive projection algorithm (SPA). It is based on the *separability assumption*, detailed below.

Assumption 5.1 (Separability) *The m-by-n matrix $X \in \mathbb{R}^{m\times n}$ is r-separable if there exist a nonnegative matrix H such that $X = X(:, \mathcal{J})H$, where $X(:, \mathcal{J})$ denotes the subset of columns of X indexed by $\mathcal{J}$ and $|\mathcal{J}| = r$.*

The pseudocode for a general recursive algorithm for separable NMF is given in Algorithm 5.1. Historically, the first variant of Algorithm 5.1 has been introduced by Araújo et al. [4] for spectroscopic component analysis with $f(x) = \|x\|_2^2 = x^\top x$, which is the so-called SPA. In the noiseless case, that is, under Assumption 5.1, SPA is guaranteed to retrieve $\mathcal{J}$ and more generally, the vertices of the set of points which are the columns of X [71]. This particular choice of f is proved to be the most robust to noise given the bounds in [46]. See Theorem 5.1 with $Q = I$ for the error bounds. The algorithm is iterative and is composed of the following two main steps:

- Selection step: the column that maximizes a given function f is selected (Algorithm 5.1).
- Projection step: all the columns are projected onto the orthogonal complement of the current selected columns (Algorithm 5.1).

These two steps are repeated r times, r being the target number of extracted columns. The drawback with the ℓ_2-norm is its sensitivity to outliers and the fact that it makes SPA deterministic. If some outliers are selected, running SPA again would still retrieve the exact same outliers.

5.2 Randomized SPA

In this section, we introduce the main contribution of this work, that is a randomized variant of SPA called RandSPA. Its key features are that it computes potentially

Algorithm 5.1: Recursive algorithm for separable NMF [46]. It coincides with SPA when $f(x) = \|x\|_2^2$.

Input: An r-separable matrix $X \in \mathbb{R}^{m\times n}$, a function f to maximize.
Output: Index set $\mathcal{J}$ of cardinality r such that $X \approx X(:,\mathcal{J})H$ for some $H \geq 0$.

Let $\mathcal{J} = \emptyset$, $P^\perp = I_m$, $V = [\,]$.
for $k = 1 : r$ **do**
  Let $j_k = \operatorname{argmax}_{1\leq j\leq n} f(P^\perp X(:,j))$. (Break ties arbitrarily, if necessary.)
  Let $\mathcal{J} = \mathcal{J} \cup \{j_k\}$.
  Update the projector $P^\perp$ onto the orthogonal complement of $X(:,\mathcal{J})$:

$$v_k = \frac{P^\perp X(:,j_k)}{\|P^\perp X(:,j_k)\|_2},$$
$$V = [V\, v_k],$$
$$P^\perp \leftarrow \left(I_m - VV^T\right).$$

different solutions at each run, thus allowing a multi-start strategy, and that the theoretical robustness results of SPA still hold.

RandSPA follows Algorithm 5.1 with $f(x) = x^\top QQ^\top x$, with $Q \in \mathbb{R}^{m\times\nu}$ being a randomly generated matrix with $\nu \geq r$. To control the conditioning of Q, we generate the columns of Q such that they are mutually orthogonal and such that

$$\|Q(:,1)\|_2 = 1 \geq \cdots \geq \|Q(:,\nu)\|_2 = 1/\sqrt{\kappa}$$

where κ is the desired conditioning of $QQ^\top$. For the columns between the first and the last one, we make the arbitrary choice to fix them also to $1/\sqrt{\kappa}$. If $Q^\top W$ has full column rank, which happens with probability one if $\nu \geq r$, RandSPA is robust to noise with the following bounds:

Theorem 5.1 *[45, Corollary 1] Let $\tilde{X} = X + N$, where X satisfies Assumption 5.1, W has full column rank, and N is noise with $\max_j \|N(:,j)\|_2 \leq \epsilon$; and let $Q \in \mathbb{R}^{m\times\nu}$ with $\nu \geq r$. If $Q^\top W$ has full column rank and*

$$\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{\sqrt{r}\kappa^3(Q^\top W)}\right),$$

then SPA applied on matrix $Q^\top\tilde{X}$ identifies a set of indices $\mathcal{J}$ corresponding to the columns of W up to the error

$$\max_{1\leq j\leq r}\min_{k\in\mathcal{J}} \left\|W(:,j) - \tilde{X}(:,k)\right\|_2 \leq \mathcal{O}\left(\epsilon\kappa(W)\kappa(Q^\top W)^3\right).$$

Theorem 5.1 is directly applicable to RandSPA since choosing $f(x) = x^\top QQ^\top x$ is equivalent to performing SPA on $Q^\top \tilde{X}$. The only subtlety is that with RandSPA, a random Q is drawn at each column extraction. The error bound for RandSPA is then the one with the highest drawn $\kappa(Q^\top W)$.

Let us note that choosing $\nu = 1$ or $\|Q(:,j)\| = 1/\sqrt{\kappa}$ with $\kappa \to \infty$ for all $j > 1$ retrieves VCA. Choosing $\nu = m$ and $\kappa(Q) = 1$ retrieves SPA. Hence, RandSPA creates a continuum between SPA, with more provable robustness, and VCA, with more solution diversity.

5.3 Numerical experiments

In this section, we study empirically the performance of the proposed algorithm RandSPA on the unmixing of hyperspectral images. The algorithms have been implemented in Julia [10]. The code for the algorithm is available as a Julia package in an online repository[1]. A different repository with the data and test scripts used in our experiments is also available[2]. Our tests are performed on 5 real hyperspectral datasets[3] described in Table 5.1.

Dataset	m	n	r
Jasper	198	$100 \times 100 = 10000$	4
Samson	156	$95 \times 95 = 9025$	3
Urban	162	$307 \times 307 = 94249$	5
Cuprite	188	$250 \times 191 = 47750$	12
San Diego	188	$400 \times 400 = 160000$	8

Table 5.1: Summary of the datasets, for which $X \in \mathbb{R}^{m \times n}$.

For all the tests, we choose $\nu = r + 1$ and a relatively well conditioned Q with $\kappa(Q) = 1.5$. We then compute $W = X(:, \mathcal{J})$ once with SPA and 30 times with RandSPA. Next, we compute H by solving the nonnegative least squares (NNLS) subproblem $\min_{H \geq 0} \|X - WH\|_F^2$ exactly with an active-set algorithm [56], and we compute the relative reconstruction error $\|X - WH\|_F / \|X\|_F$. For RandSPA, we show the best error and the median error among the 30 runs. Note that in our setting we choose the best solution as the one with the lower reconstruction error, but other methods could be used to choose the best solution among all the computed ones.

The results of the experiments for SPA and RandSPA are presented in Table 5.2. The median error of RandSPA is on the same order than that of SPA, except for Cuprite where it is higher. This is probably because r is greater than on other datasets. A good RandSPA run needs r good successive Q's, which is less probable when r gets greater. This highlights that RandSPA could be improved. One possible improvement would be to draw several matrices Q's at each iteration and select the

[1] `https://gitlab.com/vuthanho/randspa.jl`

[2] `https://gitlab.com/nnadisic/randspa`

[3] Originally downloaded from `http://lesun.weebly.com`, available at `https://gitlab.com/vuthanho/data`

best based on a criterion. The open question is then which criterion, and why. Going back to the median error of RandSPA, it is even slightly smaller than that of SPA for Samson and Urban. On the other hand, the error from the best run of RandSPA is always smaller than that of SPA. Particularly, the error is decreased respectively by 37%, 32% and 27% for Samson, Urban and San Diego. This improvement is quite noticeable.

Dataset	SPA	Med. RandSPA	Best RandSPA
Jasper	8.6869	8.7577	8.0206
Samson	6.4914	6.3114	3.9706
Urban	10.9367	9.6354	6.5402
Cuprite	2.6975	3.526	2.2824
San Diego	12.6845	12.8714	9.2032

Table 5.2: Relative reconstruction error $\|X - WH\|_F / \|X\|_F$ in percent.

The resulting false color images for Jasper, Samson, Urban and Cuprite are shown on Figure 5.1. They represent the repartition of the materials identified by SPA and RandSPA in the image. As we can see for Urban, SPA does not manage to separate well the grass and the trees (both the grass and trees are in green), while with RandSPA, it occurred that some random Q amplified some directions that separate better the grass (in blue) and the trees (in green). Similarly, in the abundance maps from the unmixing of Samson in Figure 5.1, RandSPA separates the soil (in red), the water (in blue) and the trees (in green) better than SPA where the soil (in blue) is extracted but the water is not clearly identified.

Let us discuss another experiment on the dataset Samson. We add some Gaussian noise such that $SNR = 20dB$, we fix $\kappa = 1$ and vary ν, and then show the average best error in 1,5,10 and 20 runs on Figure 5.2. As we can see, with a sufficient amount of runs that is 10 in this experiment, the relative error significantly improves for a ν near 10 in comparison to other choices of ν. In particular, it is also better than both $\nu = 1$ (VCA) and a high ν like 50 that should behave like SPA. Without added noise, VCA would perform better than every ν higher than 1 starting from 10 runs. However, when the data is noisy, this experiment highlights that VCA is not robust enough to noise and that the best run from a method between SPA and VCA is better than both SPA and VCA.

5.4 Conclusion

In this chapter, we introduced RandSPA, a variant of the separable NMF algorithm SPA that introduces randomness to allow a multi-start strategy. The robustness results of SPA still hold for RandSPA, provided a bound on the noise that depends on the parameters used. We showed empirically on the unmixing of hyperspectral images that, with sufficiently many runs, the best solution from RandSPA is generally better that the solution from SPA. We also showed that RandSPA creates a continuum

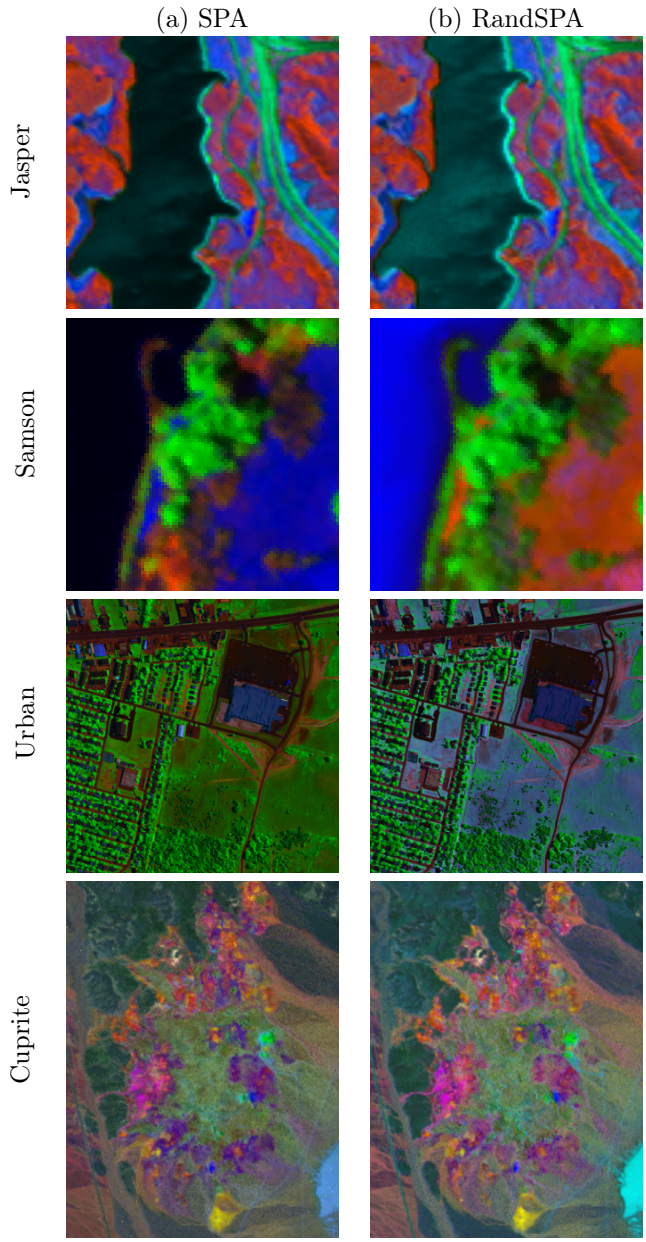


Figure 5.1: Abundance maps in false color from the unmixing of hyperspectral images.

between the two algorithms SPA and VCA, as we can recover these algorithms by running RandSPA with some given parameter values.

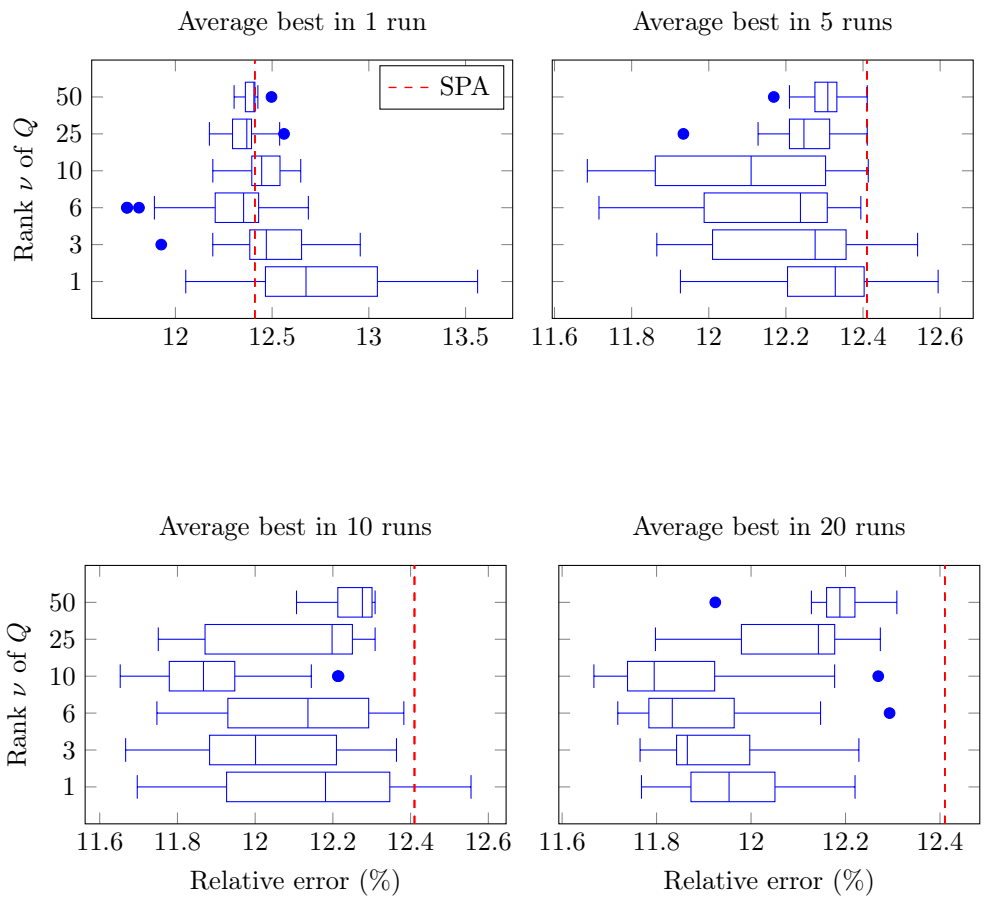


Figure 5.2: Average best reconstruction error on several runs, depending on ν, with $\kappa = 1$, on the hyperspectral image Samson with added noise such that $SNR = 20dB$.

Chapter 6

Minimum-Volume Nonnegative Matrix Factorization

The minimum-volume criterion was originally thought by [38] and the so called Craig's belief [23]. It also appeared later in the chemometrics community. Up to our knowledge, the first implementation of the minimum-volume criterion coupled with NMF was proposed in [77]. The idea is that in the absence of pure pixels, given that all the data points are not strong mixtures, finding endmembers whose cone or convex hull tightly contains the data points retrieves the true endmembers. If the main motivation was spectral unmixing, the minimum-volume criterion has also been shown to be useful in other applications, like blind audio source separation for instance [65, 106]. Regardless of the application, the minimum-volume criterion encourages interpretability of the features since they are close to the data points.

Outline and contribution of the chapter In this chapter, we present known declinations of MinVol NMF in Section 6.1, we quickly recall on the identifiability of MinVol NMF in Section 6.2, we propose a fast algorithm for MinVol NMF in Section 6.3 and finally, we show how the MinVol criterion is promising for matrix completion in Section 6.4.

6.1 Existing variants of MinVol NMF

Geometrically, the NMF $X = WH$ implies that $\text{cone}(X) \subseteq \text{cone}(W)$. Distinctively, with MinVol NMF, the convex hull of W should enclose the convex hull of X as tightly as possible[1], hence the expression "minimum-volume". In other words, MinVol NMF consists in finding a couple of factors $(W, H) \in \mathbb{R}_+^{m\times r} \times \mathbb{R}_+^{r\times n}$ such that $X = WH$ while minimizing the volume of the convex hull of the columns of W and the origin, which is given by $\frac{1}{r!}\sqrt{\det(W^\top W)}$. This improves the interpretability of the features (the columns of W) while prioritizing a unique decomposition of the data under

[1] This interpretation only holds with a column-wise simplex-structured H.

relatively mild assumptions, that are given in Theorem 2.2. Additionally, one of the factors should be constrained such that the scaling ambiguity between W and H coupled with the minimized volume does not make W tend to zero at optimality. Identifiable MinVol NMFs typically use simplex structuring constraints, namely $W \in \Delta^{m\times r}$ [65] or $H \in \Delta^{r\times n}$ [37] or $H^\top \in \Delta^{n\times r}$ [32], where $\Delta^{m\times r} = \{Y \in \mathbb{R}_+^{m\times r}, e^\top Y = e^\top\}$ and e is the all-one vector of appropriate dimension. See Section 6.2 for more details on the identifiability of MinVol NMF. The constraint $W \in \Delta^{m\times r}$ ensures that the columns of W lie within the probability simplex. The constraint $H^\top \in \Delta^{n\times r}$ can be seen as a budget assignment constraint: each feature should be used in the decomposition as much as the others. Both aforementioned constraints are without loss of generality relatively to NMF because of the scaling ambiguity between W and H, that is, any NMF (W, H) can be scaled so that $W \in \Delta^{m\times r}$ or $H^\top \in \Delta^{n\times r}$ is satisfied. The constraint $H \in \Delta^{r\times n}$ is stronger than the other two as it is not without loss of generality relatively to NMF. For example, for the matrix $X = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, there exists a rank-2 NMF of X which is simply $X = IX$. However, any exact NMF of X with the additional constraint that H has to be column wise stochastic is of rank at least 3. Despite the loss of generality, this constraint remains useful in practice as it provides a soft clustering interpretation of the decomposition. It has been instrumental in hyperspectral imaging where each column of H contains the abundances of the pure materials in a pixel which are nonnegative and sum to one [72]. The constraint $H^\top \in \Delta^{n\times r}$ can be responsible for an ill conditioned W that can lead to numerical issues [65]. Consequently, among the three mentioned variants of MinVol, we will only consider the following exact formulation in the remainder of this chapter:

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \det(W^\top W) \\ \text{subject to} \quad & X = WH, \\ & W \in \Delta^{m\times r},\ H \in \mathbb{R}_+^{r\times n}. \end{aligned} \tag{6.1}$$

Consequently, the inexact formulation is

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \frac{1}{2}\|X - WH\|_F^2 + \frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I) \\ \text{subject to} \quad & W \in \Delta^{m\times r},\ H \in \mathbb{R}_+^{r\times n}, \end{aligned} \tag{6.2}$$

where δ is a parameter that prevents the logdet from going to $-\infty$ when W is rank deficient, and $\lambda \geq 0$ balances the two terms. Note that the true volume spanned by the columns of W and the origin is equal to $\frac{1}{r!}\sqrt{\det(W^\top W)}$, but minimizing $\operatorname{logdet}(W^\top W)$ is equivalent in the exact case and makes the problem numerically easier to solve because the function $\operatorname{logdet}(\cdot)$ is concave and it is easier to design a "nice" majorizer for it [35].

6.2 Identifiability of MinVol NMF

The identifiability of MinVol NMF with $H \in \Delta^{r\times n}$ was indirectly mentioned through the identifiability of MinVol SSMF in Theorem 2.2. The additional nonnegative constraint on W does not change this identifiability result. MinVol NMF with $H^\top \in \Delta^{n\times r}$ and $W \in \Delta^{m\times r}$ are also identifiable with very similar proofs. Since we only use MinVol NMF with $W \in \Delta^{m\times r}$, let us remind the proof of its identifiability. The proof was given in [65] and adapted from [37].

Theorem 6.1 ([65]) *Let $X = WH$ be a MinVol NMF of X of size $r = \operatorname{rank}(X)$, in the sense of (6.1). If H satisfies SSC as in Definition 2.8, then MinVol NMF (W, H) of X is essentially unique.*

Proof 6.1 *Let $Q \in \mathbb{R}^{r\times r}$ be an invertible matrix such that (WQ^{-1}, QH) is another feasible solution of (6.1). Since $W^\top e = e$ and $Q^{-\top}W^\top e = e$ because (WQ^{-1}, QH) is feasible, we have*

$$Q^{-\top}W^\top e = e \quad \Leftrightarrow \quad Q^{-\top}e = e. \tag{6.3}$$

Multiplying on the left by $Q^\top$ leads to $e = Q^\top e$. Using again feasibility of (WQ^{-1}, QH),

$$\begin{aligned} QH \geq 0 \quad &\Leftrightarrow \quad H^\top Q^\top \geq 0 && (6.4)\\ &\Leftrightarrow \quad Q(i,:)^\top \in \operatorname{cone}^*(H) && (6.5)\\ &\Leftrightarrow \quad \operatorname{cone}(Q^\top) \subseteq \operatorname{cone}^*(H). && (6.6) \end{aligned}$$

Since H satisfies SSC1, $\mathcal{C} \subseteq \operatorname{cone}(H)$. By duality, $\operatorname{cone}^*(H) \subseteq \mathcal{C}^*$, where $\mathcal{C}^*$ is given in Lemma 2.1. With (6.6), this implies that $\operatorname{cone}(Q^\top) \subseteq \mathcal{C}^*$. More explicitly,

$$Q(i,:)e \geq \|Q(i,:)\|_2 \text{ for } i = 1, \ldots, r. \tag{6.7}$$

Therefore,

$$\begin{aligned} |\det(Q)| &\leq \prod_{i=1}^{r} \|Q(i,:)\|_2 \\ &\leq \prod_{i=1}^{r} Q(i,:)e \\ &\leq \left(\frac{\sum_{i=1}^{r} Q(i,:)e}{r}\right)^r = \left(\frac{e^\top Q^\top e}{r}\right)^r = 1, \end{aligned} \tag{6.8}$$

where the first inequality is coming from the Hadamard's inequality, the second from (6.7), and the last one from the arithmetic-geometric mean inequality and that $Q^\top e = e$.

Suppose now that (WQ^{-1}, QH) is also an optimal solution to (6.1). Then,

$$\begin{aligned} &\det(Q^{-\top}W^\top WQ^{-1}) = \det(W^\top W) && (6.9)\\ \Leftrightarrow \quad &|\det(Q)|^{-2}\det(W^\top W) = \det(W^\top W) && (6.10)\\ \Leftrightarrow \quad &|\det(Q)| = 1. && (6.11) \end{aligned}$$

With $|\det(Q)| = 1$*, all inequalities in* (6.8) *are equalities. Particularly, for all* i*,*

$$Q(i,:)e = \|Q(i,:)\|_2 = 1 \tag{6.12}$$

and $|\det(Q)| = \prod_{i=1}^{r} \|Q(i,:)\|_2$*, implying that* $Q^\top$ *is orthogonal. By duality of* (6.6) *and using that the cone of any orthogonal matrix is self dual, we have that* $\text{cone}(H) \subseteq \text{cone}(Q^\top)$*. Finally, since* H *satisfies SSC2,* $Q^\top$ *can only be a permutation matrix.*

6.3 Solving MinVol NMF with TITAN

As opposed to PCA/SVD, solving NMF is NP-hard in general [97]. Hence, most NMF algorithms rely on standard non-linear optimization schemes without global optimality guarantee. This also applies to MinVol NMF. In this section, we propose a fast method to solve MinVol NMF in Section 6.3.1. Our method is an application of a recent inertial block majorization-minimization framework called TITAN [51], that we already used in Chapter 3. Experimental results on real datasets show that the proposed method performs better than the state of the art; see Section 6.3.2.

6.3.1 TITANized MinVol NMF

As far as we know, all algorithms for MinVol NMF rely on two-block coordinate descent methods that update each block (W or H) by using some outer optimization algorithm to solve the subproblems formed by restricting the MinVol NMF problem to each block. For example, the state-of-the-art method from [63] uses Nesterov fast gradient method to update each factor matrix, one at a time.

Our proposed algorithm for (6.2) will be based on the TITAN framework from [51]. TITAN is an inertial block majorization minimization framework for nonsmooth non-convex optimization. It updates one block at a time while fixing the values of the other blocks, as previous MinVol NMF algorithms. In order to update a block, TITAN chooses a block surrogate function for the corresponding objective function (a.k.a. a majorizer), embeds an inertial term to this surrogate function and then minimizes the obtained inertial surrogate function. When a Lipschitz gradient surrogate is used, TITAN reduces to the Nesterov-type accelerated gradient descent step for each block of variables [51, Section 4.2]. The difference of TITAN compared to previous MinVol NMF algorithms is threefold:

1. The inertial force (also known as the extrapolation, or momentum) is used between block updates. This is a crucial aspect that will make our proposed algorithm faster: when we start the update of a block of variables (here, W or H), we can use the inertial force (using the previous iterate) although the other blocks have been updated in the meantime.

2. TITAN allows to update the surrogate after each update of W and H, which was not possible with the algorithm from [63] because it applied fast gradient from convex optimization on a fixed surrogate.

3. It has subsequential convergence guarantee, that is, every limit point of the generated sequence is a stationary point of Problem (6.2). Note that the state-of-the-art algorithm from [63] does not have convergence guarantees.

Remark. The block prox-linear (BPL) method from [109] can be used to solve (6.2) since the block functions in $W \mapsto \frac{1}{2}\|X - WH\|_F^2$ and in $H \mapsto \frac{1}{2}\|X - WH\|_F^2$ have Lipschitz continuous gradients. However, BPL applies extrapolation to the Lipschitz gradient surrogate of these block functions and requires to compute the proximal point of the regularizer $\frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I)$, which does not have a closed form. In contrast, TITAN applies extrapolation to the surrogate function of $W \mapsto f(W, H)$ with a surrogate function for the regularizer $\frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I)$ (see Section 6.3.1.1). This allows TITAN to have closed-form solutions for the subproblems, an acceleration effect, and convergence guarantee.

6.3.1.1 Surrogate functions

An important step of TITAN is to define a surrogate function for each block of variables. These surrogate functions are upper approximation of the objective function at the current iterate. Denote

$$f(W, H) = \frac{1}{2}\|X - WH\|_F^2 + \frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I)$$

and suppose we are cyclically updating (W, H). Let us denote $u_{W_k}(W)$ the surrogate function of $W \mapsto f(W, H_k)$ to update W_k, that is,

$$f(W, H_k) \leq u_{W_k}(W) \quad \text{for all} \quad W \in \mathcal{X}_W, \tag{6.13}$$

where $u_{W_k}(W_k) = f(W_k, H_k)$ and $\mathcal{X}_W$ is the feasible domain of W. Similarly, let us denote $u_{H_k}(H)$ the surrogate function of $H \mapsto f(W_{k+1}, H)$ to update H_k, that is

$$f(W_{k+1}, H) \leq u_{H_k}(H) \quad \text{for all} \quad H \in \mathcal{X}_H, \tag{6.14}$$

where $u_{H_k}(H_k) = f(W_{k+1}, H_k)$ and $\mathcal{X}_H$ is the feasible domain of H.

Surrogate function and update of W Denote $A = W^\top W + \delta I$, $B_k = W_k^\top W_k + \delta I$ and $P_k = (B_k)^{-1}$. Since logdet is concave, its first-order Taylor expansion around B_k leads to $\operatorname{logdet}(A) \leq \operatorname{logdet}(B_k) + \langle (B_k)^{-1}, A - B_k \rangle$. Hence,

$$f(W, H_k) \leq \widetilde{f}_{W_k}(W) := \frac{1}{2}\|X - WH_k\|_F^2 + \frac{\lambda}{2}\langle P_k, W^\top W \rangle + C_1, \tag{6.15}$$

where C_1 is a constant independent of W. Note that the gradient of $W \mapsto \widetilde{f}_{W_k}(W)$, being equal to

$$(WH_k - X)H_k^\top + \lambda W P_k,$$

is L_W^k-Lipschitz continuous with $L_W^k = \|H_k H_k^\top + \lambda P_k\|$. Hence, from (6.15) and the descent lemma (see [81, Section 2.1]),

$$f(W, H_k) \leq u_{W_k}(W) := \langle \nabla \widetilde{f}_{W_k}(W_k), W \rangle + \frac{L_W^k}{2}\|W - W_k\|_F^2 + C_2, \tag{6.16}$$

where C_2 is a constant depending on W_k. We use the surrogate $u_{W_k}(W)$ defined in (6.16) to update W_k. As TITAN recovers Nesterov-type acceleration for the update of each block of variables [51, Section 4.2], we have the following update for W:

$$\begin{aligned} W_{k+1} &= \underset{W \in \mathcal{X}_W}{\operatorname{argmin}} \langle \nabla \widetilde{f}_{W_k}(\overline{W_k}), W \rangle + \frac{L_W^k}{2}\|W - \overline{W_k}\|_F^2, \\ &= \left[\overline{W_k} + \frac{(X - \overline{W_k} H_k) H_k^\top - \lambda \overline{W_k} P}{L_W^k} \right]_{\Delta^{m \times r}}, \end{aligned} \tag{6.17}$$

where $[.]_{\Delta^{m \times r}}$ performs column wise projections onto the unit simplex as in [21] in order to satisfy the constraint on W in (6.2), and where $\overline{W_k}$ is an extrapolated point, that is, the current point W_k plus some momentum,

$$\overline{W_k} = W_k + \beta_W^k (W_k - W_{k-1}), \tag{6.18}$$

where the extrapolation parameter β_W^k is chosen as follows

$$\beta_W^k = \min \left(\frac{\alpha_k - 1}{\alpha_{k+1}}, 0.9999 \sqrt{\frac{L_W^{k-1}}{L_W^k}} \right), \tag{6.19}$$

$\alpha_0 = 1$, $\alpha_k = (1 + \sqrt{1 + 4\alpha_{k-1}^2})/2$. This choice of parameter satisfies the conditions to have a subsequential convergence of TITAN, see Section 6.3.1.3.

Surrogate function and update of H Since

$$\nabla_H f(W_{k+1}, H) = W_{k+1}^\top (W_{k+1} H - X),$$

the gradient of f according to H is L_H^k-Lipschitz continuous with $L_H^k = \|W_{k+1}^\top W_{k+1}\|$. Hence, we use the following Lipschitz gradient surrogate to update H_k:

$$u_{H_k}(H) = \langle \nabla_H f(W_{k+1}, H_k), H \rangle + \frac{L_H^k}{2}\|H - H_k\|_F^2 + C_3, \tag{6.20}$$

where C_3 is a constant depending on H_k. We derive our update rule for H by minimizing the surrogate function from Equation (6.20) embedded with extrapolation,

$$\begin{aligned} H_{k+1} &= \underset{H \in \mathcal{X}_H}{\operatorname{argmin}} \langle \nabla_H f(W_{k+1}, \overline{H_k}), H \rangle + \frac{L_H^k}{2}\|H - \overline{H_k}\|_F^2, \\ &= \left[\overline{H}_k + \frac{1}{L_H^k} W_{k+1}^\top (X - W_{k+1} \overline{H}_k) \right]_+, \end{aligned} \tag{6.21}$$

where $[.]_+$ denotes the projector setting all negative values to zero, and $\overline{H_k}$ is the extrapolated H_k:

$$\overline{H_k} = H_k + \beta_H^k (H_k - H_{k-1}), \tag{6.22}$$

where, as for the update of W,

$$\beta_H^k = \min \left(\frac{\alpha_k - 1}{\alpha_{k+1}}, 0.9999 \sqrt{\frac{L_H^{k-1}}{L_H^k}} \right). \tag{6.23}$$

6.3.1.2 Algorithm

Note that the update of W in (6.17) and H in (6.21) was described when the cyclic update rule is applied. Since TITAN also allows the essentially cyclic rule [51, Section 5], we can update W several times before switching updating H, and vice versa. Doing so allows to pre-compute some matrix operations before updating the factors. For instance, $XH^\top$ can be computed before updating W. The result can then be used several times during the update, which will save some computation time. Note that this pre-computing trick only works when there are no missing entries, due to the Hadamard product with M. This leads to our proposed method TITANized MinVol, see Algorithm 6.1 for the pseudocode. The stopping criteria in Algorithm 6.1 are the same as in [63]. The way λ and δ are computed is also identical to [63]. Let us mention that technically the main difference with [63] resides in how the extrapolation is embedded. In [63] the Nesterov sequence is restarted and evolves in each inner loop to solve each subproblem corresponding to each block. In our algorithm, the extrapolation parameter β_W (and β_H) for updating each block W (and H) is updated continuously without restarting. It means we are accelerating the global convergence of the sequences rather than trying to accelerate the convergence for the subproblems. Moreover, TITAN allows to update the surrogate function at each step, while the algorithm from [63] can only update it before each subproblem is solved, as it relies on Nesterov's acceleration for convex optimization.

6.3.1.3 Convergence guarantee

In order to have a convergence guarantee, TITAN requires the update of each block to satisfy the nearly sufficiently decreasing property (NSDP), see [51, Section 2]. By [51, Section 4.2.1], the update for H of TITANized MinVol satisfies the NSDP condition since it uses a Lipschitz gradient surrogate for $H \mapsto f(W, H)$ combined with the Nesterov-type extrapolation; and the bounds of the extrapolation parameters in the update of H are derived similarly as in [51, Section 6.1]. However, it is important noting that the update for W of TITANized MinVol does not directly use a Lipschitz gradient surrogate for $W \mapsto f(W, H)$. We thus need to verify NSDP condition for the update of W by another method that is presented in the following.

The function $u_{W_k}(W)$ is a Lipschitz gradient surrogate of $\tilde{f}_{W_k}(W)$, and we apply the Nesterov-type extrapolation to obtain the update in (6.17). Note that the feasible

Algorithm 6.1: TITANized MinVol

Input: $W_0, H_0, \lambda, \delta$

$\alpha_1 = 1$, $\alpha_2 = 1$, $W_{old} = W_0, H_{old} = H_0$, $L_H^{prev} = \|W_0^\top W_0\|$, $L_W^{prev} = \|H_0 H_0^\top + \lambda(W_0^\top W_0 + \delta I)^{-1}\|$

Output: W, H

while *stopping criteria not satisfied* **do**

  while *stopping criteria not satisfied* **do**

    $\alpha_0 = \alpha_1, \alpha_1 = (1 + \sqrt{1 + 4\alpha_0^2})/2$

    $P \leftarrow (W^\top W + \delta I)^{-1}$

    $L_W \leftarrow \|HH^\top + \lambda P\|$

    $\beta_W = \min\left((\alpha_0 - 1)/\alpha_1, 0.9999\sqrt{L_W^{prev}/L_W} \right)$

    $\overline{W} \leftarrow W + \beta_W(W - W_{old})$

    $W_{old} \leftarrow W$

    $W \leftarrow \left[\overline{W} + \frac{(XH^\top - \overline{W}(HH^\top + \lambda P))}{L_W}\right]_{\Delta^{m \times r}}$

    $L_W^{prev} \leftarrow L_W$

  $L_H \leftarrow \|W^\top W\|$

  while *stopping criteria not satisfied* **do**

    $\alpha_0 = \alpha_2, \alpha_2 = (1 + \sqrt{1 + 4\alpha_0^2})/2$

    $\beta_H = \min\left((\alpha_0 - 1)/\alpha_2, 0.9999\sqrt{L_H^{prev}/L_H} \right)$

    $\overline{H} \leftarrow H + \beta_H(H - H_{old})$

    $H_{old} \leftarrow H$

    $H \leftarrow \left[\overline{H} + \frac{W^\top(X - W\overline{H})}{L_H}\right]_+$

    $L_H^{prev} \leftarrow L_H$

set of W is convex. Hence, it follows from [51, Remark 4.1] that

$$\tilde{f}_{W_k}(W_k) + \frac{L_W^k(\beta_W^k)^2}{2}\|W_k - W_{k-1}\|_F^2 \geq \tilde{f}_{W_k}(W_{k+1}) + \frac{L_W^k}{2}\|W_{k+1} - W_k\|_F^2. \quad (6.24)$$

Furthermore, we note that $\tilde{f}_{W_k}(W_k) = f(W_k, H_k)$, and $\tilde{f}_{W_k}(W_{k+1}) \geq f(W_{k+1}, H_k)$. Therefore, from (6.24) we have

$$f(W_k, H_k) + \frac{L_W^k(\beta_W^k)^2}{2}\|W_k - W_{k-1}\|_F^2 \geq f(W_{k+1}, H_k) + \frac{L_W^k}{2}\|W_{k+1} - W_k\|_F^2, \quad (6.25)$$

which is the required NSDP condition of TITAN. Consequently, the choice of β_W^k in (6.19) satisfy the required condition to guarantee subsequential convergence [51, Proposition 3.1].

On the other hand, we note that the error function $W \mapsto err_1(W) := u_{W_k}(W) - f(W, H_k)$ is continuously differentiable and $\nabla_W err_1(W_k) = 0$; similarly for the error function $H \mapsto err_2(H) := u_{H_k}(H) - f(W_{k+1}, H)$. Hence, it follows from [51, Lemma 2.3] that the Assumption 2.2 in [51] is satisfied. Applying [51, Theorem 3.2], we conclude that every limit point of the generated sequence is a stationary point of Problem (6.2). It is worth noting that as TITANized MinVol does not apply restarting step, [51, Theorem 3.5] for a global convergence is not applicable.

6.3.2 Numerical Experiments

In this section we compare TITANized MinVol to [63], an accelerated version of the method from [36] (for $p = 2$), on two NMF applications: hyperspectral unmixing and document clustering, which are dense and sparse datasets, respectively. All tests are performed on MATLAB R2018a, on a PC with an Intel® Core™ i7 6700HQ and 24 GB RAM. The code is available on an online repository[2].

The datasets used are shown in Table 6.1. For each data set, each algorithm is launched with the same random initializations, for the same amount of wall-clock time. In order to derive some statistics, for both hyperspectral unmixing and document clustering, 20 random initializations are used (each entry of W and H are drawn from the uniform distribution in [0,1]). The wall-clock time used for each data set is adjusted manually, and corresponds to the maximum displayed value on the respective time axes in Figure 6.1; see also Table 6.2.

For display purposes, for each data set, we compare the average of the scaled objective functions according to time, that is, the average of $(f(W, H) - e_{\min})/\|X\|_F$ where $e_{\min}$ is the minimum obtained error among the 20 different runs and among both methods. The results are presented in Figure 6.1. On both hyperspectral and document datasets, TITANized MinVol converges on average faster than [63] except for the San Diego data set (although TITANized MinVol converges initially faster). For most tested datasets, MinVol [63] cannot reach the same error as TITANized MinVol within the allocated time.

[2] `https://gitlab.com/vuthanho/titanized-minvol`

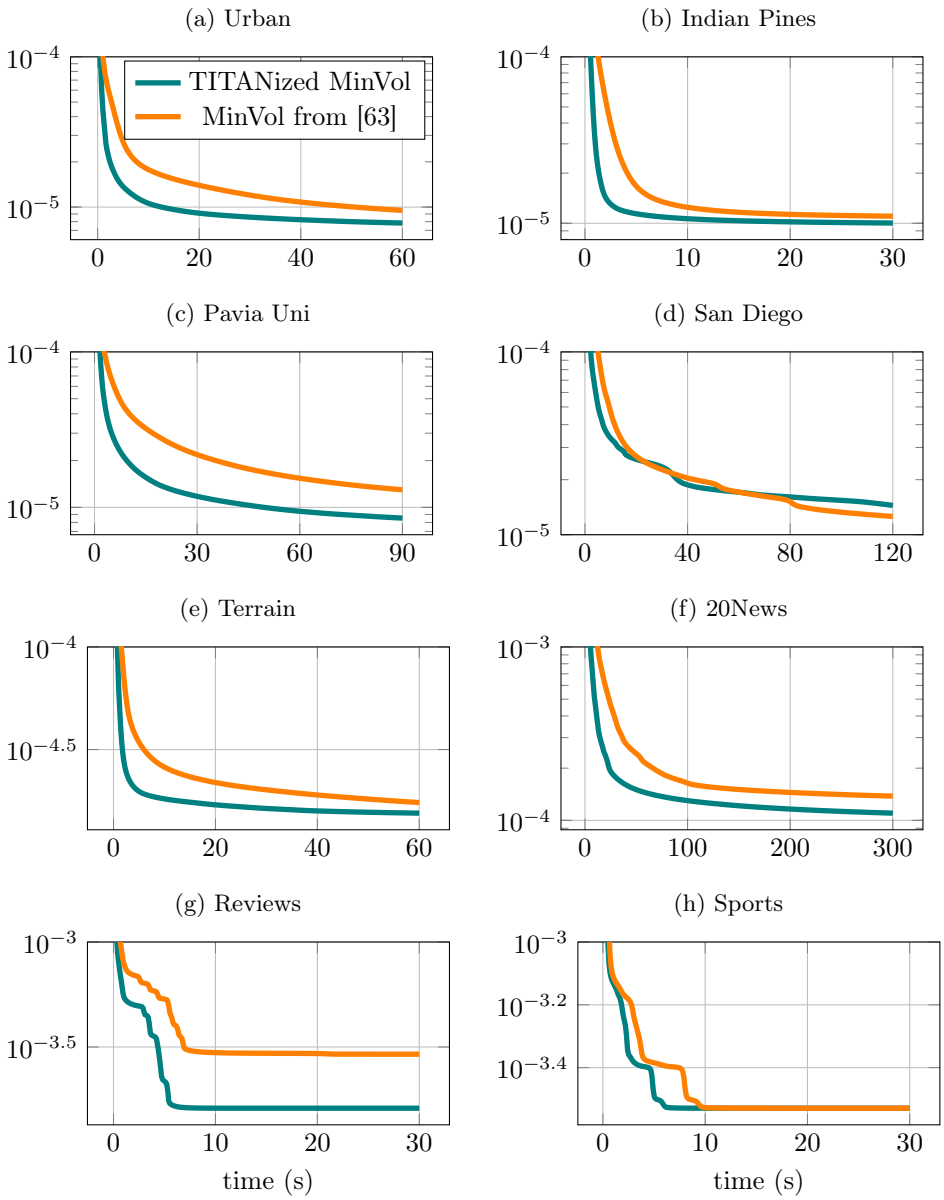


Figure 6.1: Evolution w.r.t. time of the average of $(f(W,H) - e_{\min})/\|X\|_F$ for the different datasets.

Data set	m	n	r
Urban	162	94249	6
Indian Pine	200	21025	16
Pavia Univ.	103	207400	9
San Diego	158	160000	7
Terrain	166	153500	5
20 News	61188	7505	20
Sports	14870	8580	7
Reviews	18483	4069	5

Table 6.1: Datasets used in our experiments and their respective dimensions

Data set	Our method's lead time (s)	wall-clock time for [63]	Saved wall-clock time
Urban	44	60	73%
Indian Pines	25	30	83%
Pavia Univ.	68	90	76%
San Diego	NaN	120	0%
Terrain	44	60	73%
20News	221	300	74%
Reviews	26	30	80%
Sports	15	30	50%

Table 6.2: TITANized MinVol's lead time over MinVol [63] to obtain the same minimum error.

Algorithm	ranking	
	Hyperspectral unmixing	Document clustering
TITANized MinVol	$(94, 6)$	$(55, 5)$
MinVol [63]	$(6, 94)$	$(5, 55)$

Table 6.3: Ranking among the different runs depending on the algorithm and the kind of data set

The ranking among all the tests has been reported in Table 6.3, where the i-th entry denotes how many times the corresponding algorithm was in the i-th place. We also reported in Table 6.2 TITANized MinVol's lead time over [63] when the latter reaches its minimum error after the maximum allotted wall-clock time. The lead time is the time saved by TITANized MinVol to achieve the error of the method from [63] using the maximum allotted wall-clock time. On average, TITANized MinVol is twice faster than [63], with an average gain of wall-clock time above 50%.

To summarize, our experimental results show that TITANized MinVol has a faster convergence speed and smaller final solutions than [63].

6.4 Minimum-volume Nonnegative Matrix Completion

Given a data matrix $X \in \mathbb{R}^{m \times n}$, there exist many scenarios where only a few entries of X are observed, e.g., in recommender systems illustrated by the famous Netflix problem [57]. Recovering these missing entries is often tackled by assuming that the fully observed data follow a certain structure. If the structuring assumption is meaningful, by fitting a model that follows the same structure on the observed entries, it is possible to recover the missing entries; see, e.g., [15, 16, 47]. The low-rank assumption is meaningful in many scenarios [94]. If $X \in \mathbb{R}^{m \times n}$ is low-rank, we can express it as the product of two smaller matrices, $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$, as $X = WH$ where $r \ll \min(m, n)$. Let us denote $\Omega \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$ the set containing the indices of the observed entries in X. If the rank of X is equal to r, we can look for $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$ such that $X(i, j) = W(i, :)H(:, j)$ for all $(i, j) \in \Omega$. Then, for every missing entry at $(i, j) \in \overline{\Omega}$, $X(i, j)$ can be estimated by computing $W(i, :)H(:, j)$. If X is noisy and does not follow the low-rank assumption, it might still be relevant to approximate it through a low-rank structure, because low-rank matrix approximations can identify patterns in the data via the extraction of common features among data points.

When the rank is unknown, a common tractable strategy is to minimize the nuclear norm, that is the sum of the singular values, of the estimation $\tilde{X}$ of X:

$$\min_{\tilde{X}} \|\tilde{X}\|_* \quad \text{such that} \quad \mathcal{P}_\Omega(\tilde{X}) = \mathcal{P}_\Omega(X),$$

where $\mathcal{P}_\Omega(Y)$ sets $Y(i, j)$ to zero if $(i, j) \notin \Omega$, or does not change it otherwise.

In this section, we consider the rank to be known, and our goal is not only to recover the missing entries in X, but also to recover the unique matrices W and H that generated the data $X = WH$. This could be useful in hyperspectral unmixing with missing data for instance, where the columns of W are expected to be the spectral signatures of the underlying materials, and where the j-th column of H contains the abundance in the j-th pixel of each extracted material. In this scenario, it is of course preferable to recover a unique set (W, H). To perform this task, it is possible to first use a data completion algorithm, and then use a constrained matrix factorization algorithm to estimate the sought factors W and H. Here, we focus on performing both tasks together, since estimating correctly W and H on Ω implies a correct recovery of the missing entries in $X = WH$. We assume that the data and the factors are nonnegative, that is, $X \geq 0$, $W \geq 0$ and $H \geq 0$, where $\geq$ is applied element wise. Hence, our goal is to perform NMF with missing data while recovering a unique decomposition. To do so, MinVol NMF is a relevant option, and its performances on matrix completion have never been explored before. In this chapter, we show that when correctly tuned, MinVol NMF performs well on the matrix completion task and is also able to retrieve the true underlying factors using only a few observed entries.

6.4.1 Motivation

In this section, we justify the choice of the minimum-volume criterion for the task of nonnegative matrix completion. Matrix completion in general has been well studied, especially by the compressed sensing community [15]. Among the techniques to perform matrix completion, the low-rank approach often arises, because the low-rank structure has been observed to be quite powerful in this setting, as it is able to identify hidden (linear) features in data. However, minimizing the rank of the estimation matrix while guaranteeing the equality constraints on the set of observed entries is NP-hard in general. A good convex relaxation that promotes low-rank structures is the nuclear norm minimization; see [85]. This is coming from the fact that the rank is the ℓ_0 norm of the vector of the singular values, while the nuclear norm is the ℓ_1 norm of this vector. Still, this requires to store the whole estimation $\tilde{X}$ of X, and it also becomes harder to impose additional structuring constraints. When the rank is known, we can fully exploit the low-rank structure by working with the low-rank factors W and H instead. It is then easier to add some structuring constraints on W and H. Also, this allows one to deal with larger problems. Since

$$\|X\|_* = \min_{X=WH} \frac{1}{2}\left(\|W\|_F^2 + \|H\|_F^2\right),$$

a good alternative to the nuclear norm regularization is then the regularizer $\frac{1}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$ [90]. If the rank is unknown, an overestimated rank coupled with a proper penalization of $\frac{1}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$ can yield state-of-the-art results. For example, in [87], a properly tuned matrix factorization model using the above regularizer can outperform deep neural networks on recommendation systems. In [69], they showed that the sightly different regularizer $\|W\|_* + \frac{1}{2}\|H\|_F^2$ yields better results than $\frac{1}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$, both with uniform or non-uniform samplings. Going back to our point of interest, it is interesting to observe that the MinVol regularizer provides more adaptability as a (non-convex) relaxation of the rank [64], since $\operatorname{logdet}(W^\top W + \delta I) = \sum_i \log(\sigma_i^2(W) + \delta)$. As it can be seen in Fig. 6.2, $\operatorname{logdet}(W^\top W + \delta I)$ approximates a range of behaviors between the ℓ_0 and the ℓ_1 norms. In particular, as δ goes to zero, $\operatorname{logdet}(W^\top W + \delta I)$ converges to the ℓ_0 norm of the vector of singular values of X, up to a constant factor. Hence the MinVol criterion $\operatorname{logdet}(W^\top W + \delta I)$ is clearly a good candidate as a regularizer for NMC.

Let us now propose two models to tackle NMC. The first one is to adapt (6.2) to the NMC problem, which yields

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \frac{1}{2}\|\mathcal{P}_\Omega(X - WH)\|_F^2 + \frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I) \\ \text{subject to} \quad & W \in \Delta^{m\times r},\ H \in \mathbb{R}_+^{r\times n}. \end{aligned} \tag{6.26}$$

Theorem 6.1 does not extend to the case where some values are missing. If the matrix completion is not unique, then it is impossible to guarantee a unique recovery of the matrices W and H. Hence, a trivial way to adapt Theorem 6.1 to missing values is to add the condition that matrix completion under MinVol NMF should be unique. However, better conditions than standard low-rank matrix completion theory under which solving (6.26) recovers a unique completion are, up to now, unknown.

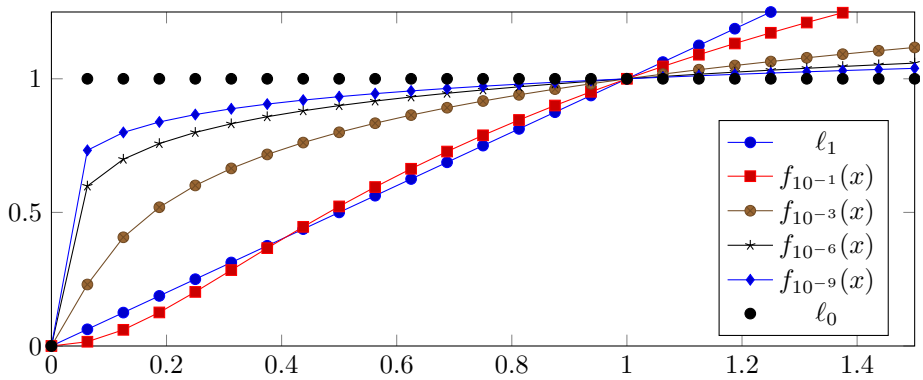


Figure 6.2: Function $f_\delta(x) = \frac{\ln(x^2+\delta)-\ln(\delta)}{\ln(1+\delta)-\ln(\delta)}$ for various values of δ, along the ℓ_0 and ℓ_1 norm.

The second one introduces a new variant of MinVol NMF which is not simplex structured. Inspired by the regularizer $\|W\|_* + \frac{1}{2}\|H\|_F^2$ and motivated by the link between the behavior of the nuclear norm and the MinVol criterion, here we consider $\operatorname{logdet}(W^\top W + \delta I) + \|H\|_F^2$ as a regularizer. The resulting new MinVol NMF adapted for NMC is

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \frac{1}{2}\|\mathcal{P}_\Omega(X - WH)\|_F^2 + \frac{\lambda}{2}\operatorname{logdet}(W^\top W + \delta I) + \frac{\gamma}{2}\|H\|_F^2 \\ \text{subject to} \quad & W \in \mathbb{R}_+^{m\times r},\ H \in \mathbb{R}_+^{r\times n}, \end{aligned} \tag{6.27}$$

where $\lambda \geq 0$ and $\gamma \geq 0$ balance the regularizers. Note that neither W nor H is simplex structured. The scaling ambiguity coupled with the volume penalization is counter balanced by the penalization of $\|H\|_F^2$. In fact, in the exact case and when $\delta = 0$, every row of H has the same norm at optimality. Consider a feasible (W, H) for (6.27) such that $X = WH$ and let $f(D) = \frac{\lambda}{2}\operatorname{logdet}(D^{-1}W^\top W D^{-1}) + \frac{\gamma}{2}\|DH\|_F^2$ where $D = \operatorname{Diag}(d_1, \ldots, d_r)$ is a positive diagonal matrix that can be seen as the scaling ambiguity between W and H. Nullifying the gradient of f relatively to each d_i, we have that $d_i^2 = \frac{\lambda}{\gamma\|H(i,:)\|_F^2}$, meaning that at optimality $\|H(i,:)\|_F^2 = \frac{\lambda}{\gamma}$ for all i.

6.4.2 Algorithms

In Section 6.4.3, we compare NMF, MinVol (6.26) and new MinVol (6.27). For a fair comparison, these models are fit with the same algorithmic scheme, adapted from [98], which is an extrapolated alternating block majorization-minimization method already described in Section 6.3. Our adaptation is described in Algorithm 6.2, where $\mathcal{P}_{\Delta^{m\times r}}$ (respectively $\mathcal{P}_{\mathbb{R}_+^{m\times r}}$) projects a matrix of size $m \times r$ onto $\Delta^{m\times r}$ (respectively $\mathbb{R}_+^{m\times r}$). See [22] for the details on the projection onto $\Delta^{m\times r}$. Essentially, the updates for W and H are several projected gradient descent steps, performed with a step size equal to the inverse of the Lipschitz constant. The updates for each model and each factor,

as well as the corresponding Lipschitz constant, are given in Table 6.4 and Table 6.5. The used Lipschitz constants are deliberately not tight. Consider the MinVol NMF update of H for instance. Let $M \in \{0,1\}^{m\times n}$ be such that $M(i,j) = 1$ if $(i,j) \in \Omega$, $M(i,j) = 0$ otherwise. A tighter Lipschitz constant is $\max_j \left\|W^\top \operatorname{Diag}(M(:,j))W\right\|$; see the paragraph in Section 3.2.1 on the choice of Lipschitz constant for the details. We deliberately keep $\|W^\top W\|$ as it is less costly to compute and the additional cost might not be worth it. Moreover, if at least one column of X is fully observed, then $\max_j \left\|W^\top \operatorname{Diag}(M(:,j))W\right\| = \|W^\top W\| = \|W\|^2$.

Algorithm 6.2: Main algorithm scheme

input: data matrix $X \in \mathbb{R}^{m\times n}$, initial factors $W \in \mathbb{R}_+^{m\times r}$ and $H \in \mathbb{R}_+^{r\times n}$

$\alpha_1 = \alpha_2 = 1,\ W_o = W,\ H_o = H$
while *stopping criteria not satisfied* **do**
  while *stopping criteria not satisfied* **do**
    $\alpha_0 = \alpha_1, \quad \alpha_1 = \frac{1}{2}(1 + \sqrt{1 + 4\alpha_0^2})$
    $\overline{W} = W + \frac{\alpha_0 - 1}{\alpha_1}(W - W_o)$
    $W_o = W$
    Update W according to Table 6.4
  while *stopping criteria not satisfied* **do**
    $\alpha_0 = \alpha_2, \quad \alpha_2 = \frac{1}{2}(1 + \sqrt{1 + 4\alpha_0^2})$
    $\overline{H} = H + \frac{\alpha_0 - 1}{\alpha_2}(H - H_o)$
    $H_o = H$
    Update H according to Table 6.5

	Update
MinVol	$\mathcal{P}_{\Delta^{m\times r}}\left(\overline{W} - \frac{1}{L}\nabla_W\right)$
new MinVol / NMF (with $\lambda = 0$)	$\mathcal{P}_{\mathbb{R}_+^{m\times r}}\left(\overline{W} - \frac{1}{L}\nabla_W\right)$

Table 6.4: Updates for W according to the model, where $P = (\overline{W}^\top\overline{W} + \delta I)^{-1}$, $L = \|HH^\top + \lambda P\|$ and $\nabla_W = \mathcal{P}_\Omega\left(\overline{W}H - X\right)H^\top + \lambda WP$.

6.4.3 Experiments

The goal of this section is to highlight the performance of the MinVol criterion for NMC. All experiments are run with Julia on a PC with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and 16GB RAM. All displayed measurements are averaged out of 20 runs. The code is available at `https://gitlab.com/vuthanho/minvol-nmc`. The compared models are NMF (to provide a baseline of a non-regularized model),

	L	Update
MinVol / NMF	$\|W^\top W\|$	$\mathcal{P}_{\mathbb{R}_+^{r\times n}}\left(\overline{H} - \frac{1}{L}\nabla_H\right)$
new MinVol	$\|W^\top W + \gamma I\|$	$\mathcal{P}_{\mathbb{R}_+^{r\times n}}\left(\frac{L-\gamma}{L}\overline{H} - \frac{1}{L}\nabla_H\right)$

Table 6.5: Updates for H according to the model, where $\nabla_H = W^\top \mathcal{P}_\Omega\left(W\overline{H} - X\right)$.

MinVol (6.26), and the new proposed MinVol (6.27). For all models, the stopping criteria of the **while** loop in Algorithm 6.2 is just a number of outer iterations equal to 50, and the stopping criteria of the two **while** loops in Algorithm 6.2 is a number of inner iterations equal to 20. All models are also initialized with the same warm start (W_0, H_0), which is the output of 500 iterations of NMF where the columns of W are simplex-structured. In this setting, all methods converge. For both MinVols, λ is first set to $\frac{\max(\|\mathcal{P}_\Omega(X-W_0H_0)\|_F^2, 10^{-6})}{|\operatorname{logdet}(W_0^\top W_0+\delta I)|}$. For the new proposed MinVol, γ is first set to $0.01\frac{\max(\|\mathcal{P}_\Omega(X-W_0H_0)\|_F^2, 10^{-6})}{\|H_0\|_F^2}$. On the hyperparameters λ and γ, we adapt the automatic tuning method developed in [82]. The automatic tuning does not introduce a significant additional cost and is triggered when the difference between the current and the last objective values divided by $\|\mathcal{P}_\Omega(X)\|_F^2$ is less than 10^{-3}.

First experiment: noiseless synthetic data The first experiment focuses on both data completion and recovery of the exact generating factors in a noiseless case. For this experiment, for a given rank r, we randomly generate two factors $(W, H) = [0,1]^{200\times r} \times [0,1]^{r\times 200}$ following a uniform distribution. Then, 80% random values of H are set to zeros. This is a reasonable assumption in real scenarios such as hyperspectral unmixing. For the explored range of ranks, this will provide almost surely a sufficiently scattered H. Then, we generate the full data matrix X simply by computing WH. The average of the elements of X is always set to 1, dividing X by its average. Finally, we create the observed data $\tilde{X}$ by removing a certain percentage of the entries in X. We vary the rank from 5 to 10, and the percentage of missing values from 80% to 90%. We report the root-mean-squared error (RMSE) of the missing values according to Definition 6.1 and the maximum subspace angle between the factor W that took part in generating the data X and its estimation $\tilde{W}$ according to Definition 6.2.

Definition 6.1 (RMSE) *The RMSE on the unobserved set $\overline{\Omega}$ is defined as follows*

$$RMSE(\tilde{X}, WH) = \sqrt{\frac{1}{|\overline{\Omega}|}\|\mathcal{P}_{\overline{\Omega}}(\tilde{X} - WH)\|_F^2}.$$

Definition 6.2 (Subspace angle [11, 107]) *Let USV and $\tilde{U}\tilde{S}\tilde{V}$ respectively be the singular value decomposition of W and $\tilde{W}$. Then the angle between the two subspaces*

specified by the columns of W *and* $\tilde{W}$ *is defined as follows*

$$Angle(W, \tilde{W}) = \arcsin(\min(1, \|\tilde{U} - UU^\top \tilde{U}\|_2)).$$

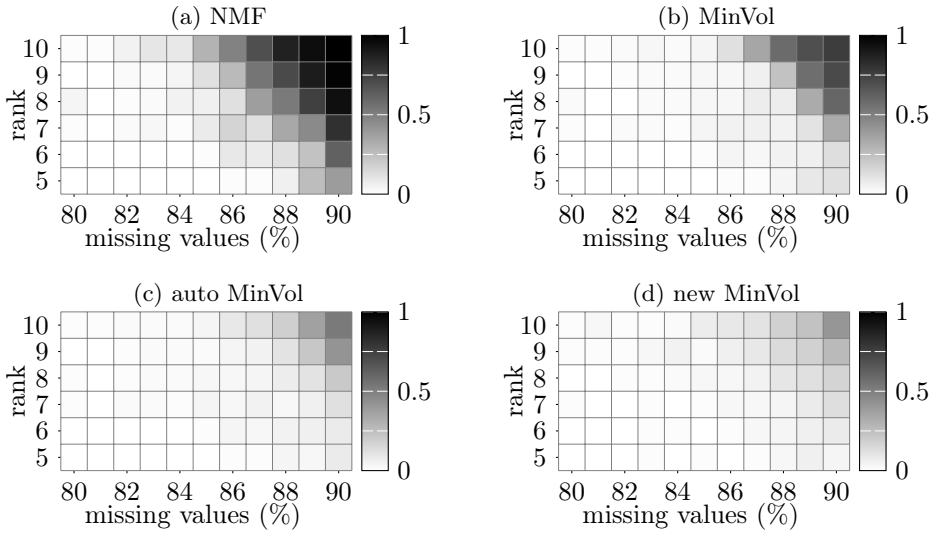


Figure 6.3: Average RMSE according to the rank r and to the percentage of missing values over 20 runs.

The RMSEs are reported in Fig. 6.3 and the subspace angles in Fig. 6.4. MinVol NMF coupled with the proposed auto-tuning proposed in [82] clearly outperforms the vanilla MinVol NMF with a fixed λ. The auto-tuned MinVol NMF is itself outperformed by our new proposed variant of MinVol NMF. For 90% missing values and a rank equal to 10 for instance, the average RMSE of the auto-tuned Minvol is 0.52 while it is 0.41 for the new MinVol.

Second experiment: noisy synthetic data We keep the same settings as in the first experiment, while fixing the rank to 10, and adding some uniformly distributed noise. The noise level corresponds to the RMSE between the clean data and the noisy data. We vary the noise level from 0 to 1 and the percentage of missing values from 80% to 90%. We report the RMSE in Fig. 6.5. It is not necessary to report the subspace angle since it is degrading too fast. Perfect matrix completion is a necessary condition to retrieve a low subspace angle, which is already not possible starting from a noise level equal to 0.2. Results in Fig. 6.5 show that our proposed variant of MinVol NMF is more consistent relatively to the percentage of missing values and more precise than vanilla MinVol NMF in the presence of noise.

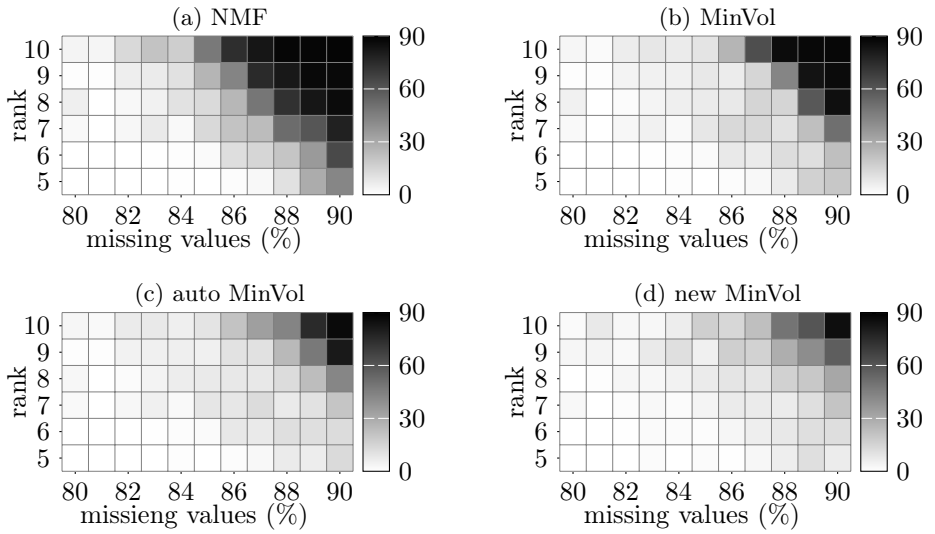


Figure 6.4: Average angle according to the rank r and to the percentage of missing values over 20 runs.

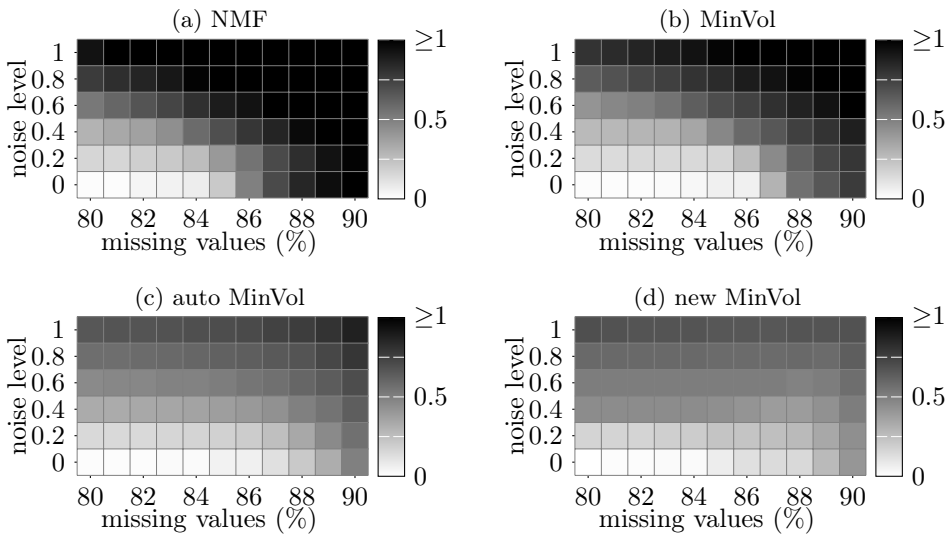


Figure 6.5: Average RMSE according to the noise level and to the percentage of missing values over 20 runs.

6.5 Identifiability of MinVol NMF with ℓ_1 penalty

In the previous section, we studied the model

$$\begin{array}{ll} \underset{W,H}{\text{minimize}} & \frac{1}{2}\operatorname{logdet}(W^\top W) + \frac{1}{2}\|H\|_F^2 \\ \text{subject to} & X = WH, \\ & W \in \mathbb{R}_+^{m\times r},\ H \in \mathbb{R}_+^{r\times n} \end{array} \tag{6.28}$$

for missing data. Additionally, we mentioned that the identifiability of (6.28) remains unknown with conditions milder than Theorem 2.1. For the sake of completion, let us mention in this section that the model

$$\begin{array}{ll} \underset{W,H}{\text{minimize}} & \frac{1}{2}\operatorname{logdet}(W^\top W) + \|H\|_1 \\ \text{subject to} & X = WH, \\ & W \in \mathbb{R}_+^{m\times r},\ H \in \mathbb{R}_+^{r\times n} \end{array} \tag{6.29}$$

is just as identifiable as vanilla MinVol NMF.

Theorem 6.2 *Let $X = WH$ be an ℓ_1-MinVol NMF of X of size $r = \operatorname{rank}(X)$, in the sense of* (6.29)*. If H satisfies SSC as in Definition 2.8, then ℓ_1-MinVol NMF (W, H) of X is essentially unique.*

This follows from two key points:

- Consider a feasible (W, H) for (6.29) such that $X = WH$ and let $f(D) = \frac{1}{2}\operatorname{logdet}(D^{-1}W^\top WD^{-1}) + \|DH\|_1$ where $D = \operatorname{Diag}(d_1, \ldots, d_r)$ is a positive diagonal matrix that can be seen as the scaling ambiguity between W and H. Nullifying the gradient of f relatively to each d_i, we have that $d_i = \frac{1}{\|H(i,:)\|_1}$, meaning that at optimality $\|H(i,:)\|_1 = 1$ for all i, since $d_i = 1$ at optimality (otherwise one can improve the solution by scaling, which would therefore not be globally optimal). Or more compactly, $He = e$.
- If H is SSC, MinVol NMF with $He = e$ is identifiable [32].

6.6 Conclusion

In this chapter, we developed a new algorithm to solve MinVol NMF based on the inertial block majorization-minimization framework of [51]. This framework, under some conditions that hold for our method, guarantees subsequential convergence. Experimental results show that this acceleration strategy performs better than the state-of-the-art accelerated MinVol NMF algorithm from [63]. Then, we argued on the favor of using more the MinVol criterion in the domain of matrix completion, which has never been explored before. Not only the MinVol criterion can emulate a broad of behaviors going from the rank minimization to the nuclear minimization,

but it also acts in favor of recovering the unique decomposition of a low-rank matrix if it exists. This paper also introduced a new variant of MinVol NMF which is not simplex-structured. Experiments show that a properly tuned MinVol NMF provides encouraging results, both on the task of matrix completion and unique factors recovery. Last but not least, experiments show that our new proposed variant of MinVol NMF outperforms vanilla MinVol NMF. Future work should focus on the potential identifiability of this new variant and on comparing with other matrix completion algorithms.

Chapter 7

Maximum-Volume Nonnegative Matrix Factorization

Hélène Vogelsinger - Reminiscence

In this chapter, we present a new volume regularized NMF, dubbed MaxVol NMF for Maximum-Volume Nonnegative Matrix Factorization. Compare to MinVol NMF, MaxVol NMF maximizes the volume of H instead of minimizing the volume of W. To the best of our knowledge, MaxVol MF (without nonnegativity) has only been briefly discussed in [92] as the sparse nonnegative case of their framework. Its behavior on HU has not been explored, and their proposed algorithm is in fact using an algorithm designed to solve MinVol MF coming from [35]. However, we will see that in the inexact case MinVol NMF and MaxVol NMF behaves differently. In particular MaxVol NMF is much more effective to extract sparse factors and does not generate rank-deficient solutions; see below for more details.

Outline and contribution of the chapter In Section 7.1 and Section 7.2, we motivate, introduce and analyze MaxVol NMF. In Section 7.3, we propose two algorithms to solve MaxVol NMF. In Section 7.4, we present a normalized variant of MaxVol NMF that exhibits better performance than MinVol NMF and MaxVol NMF in the context of HU. Finally, we conclude and discuss future works in Section 7.7.

7.1 Motivation

In the previous chapter, we highlighted the strengths of MinVol NMF. Let us also highlight two of its main weaknesses in the context of Hyperspectral Unmixing. In the remaining of this chapter, the MinVol NMF we are referring to is the one where the simplex structure is imposed on the columns of H, that is, $H \in \Delta^{r \times n}$.

First, the MinVol criterion introduces a bias that can reduce the quality of the unmixing. Let us illustrate this with the Samson dataset. The three main endmembers present in Samson are water, soil and tree. Due to the spectral signature of water having a low magnitude relatively to the spectral signature of soil and tree, a bad

λ	0 (NMF)	1	5	10	50	1 with autotuning
l_2 norm of water	0.73	0.35	0.36	0.35	0.29	0.31
l_0 norm of water	152	132	130	128	120	123

Table 7.1: l_2 and l_0 of the spectral signature of the water retrieved by MinVol for the Samson dataset, which is of size 156×9025.

estimation of the water spectral signature does not increase significantly the reconstruction error. Consider the MinVol penalty on top of that, decreasing the norm of the spectral signature of water is an easy way to decrease to volume of W, and it can be done at a very small "reconstruction price". This can be seen on Figure 7.1b, where the spectral signature of water (in red) for MinVol NMF with $\lambda = 1$ contains $156 - 132 = 24$ zeros (reported in Table 7.1), while there should not be any zeros because there is not a wavelength at which water absorbs completely electromagnetic energy. Here, increasing λ will only worsen this behavior, as it can be seen with $\lambda = 50$ on Figure 7.1 and with the l_0 norms reported in Table 7.1.

Second, the sparsity of the decomposition is implicit and depends on the quality of the data. In the presence of noise, at some point, increasing the weight λ of the volume criterion will not particularly increase the sparsity of H and improve the decomposition. See Figure 7.1. With λ increasing, the corresponding abundance map gets a little bit crispier. Still, the improvement in terms of sparsity is not that significant, and at the price of a worse spectral signature for the water. Now consider some data of better quality, like the Moffett dataset for instance. We can see on Figure 7.2 that the abundance map for NMF is not perfect, but it is already a better decomposition than what NMF could provide for Samson on Figure 7.1. Adding the MinVol criterion with $\lambda = 1$ improves the decomposition and, as a consequence, the sparsity. Still, the water and tree extraction are not right, as there are some detected water within the lands where it should in fact be trees. Increasing λ to 10 slightly improves this, but the wrong water artifacts are still here. Then, increasing again λ does not improve the unmixing. With $\lambda = 50$, one of the columns of W just collapses to zero. If a practitioner has some a priori knowledge on the sparsity of the decomposition, MinVol NMF cannot explicitly control sparsity, though sparsity is often desired in unmixing.

In this chapter, we will see how MaxVol NMF preserves the spirit of MinVol NMF without the aforementioned weaknesses.

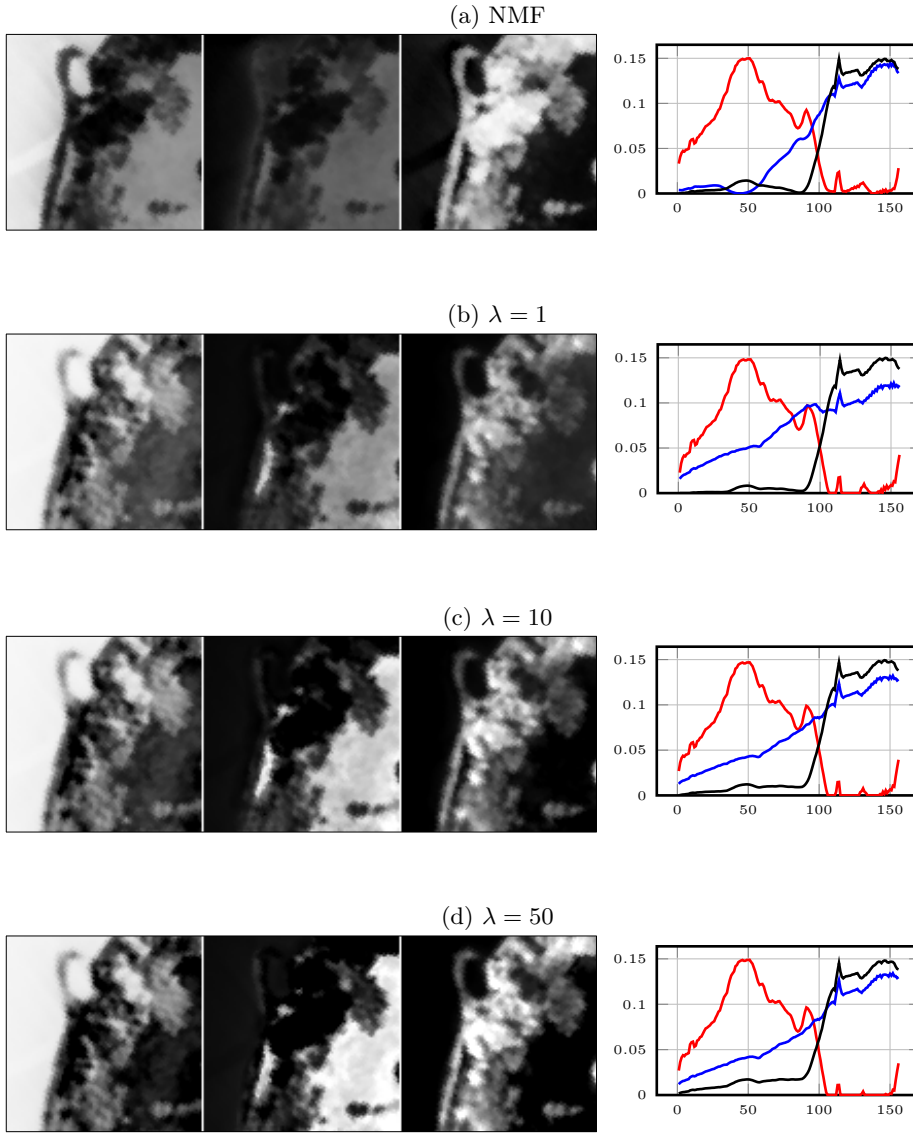


Figure 7.1: Abundance maps and normalized endmembers (from the left to the right: water, soil and tree) for MinVol on the Samson dataset with $\delta = 1$.

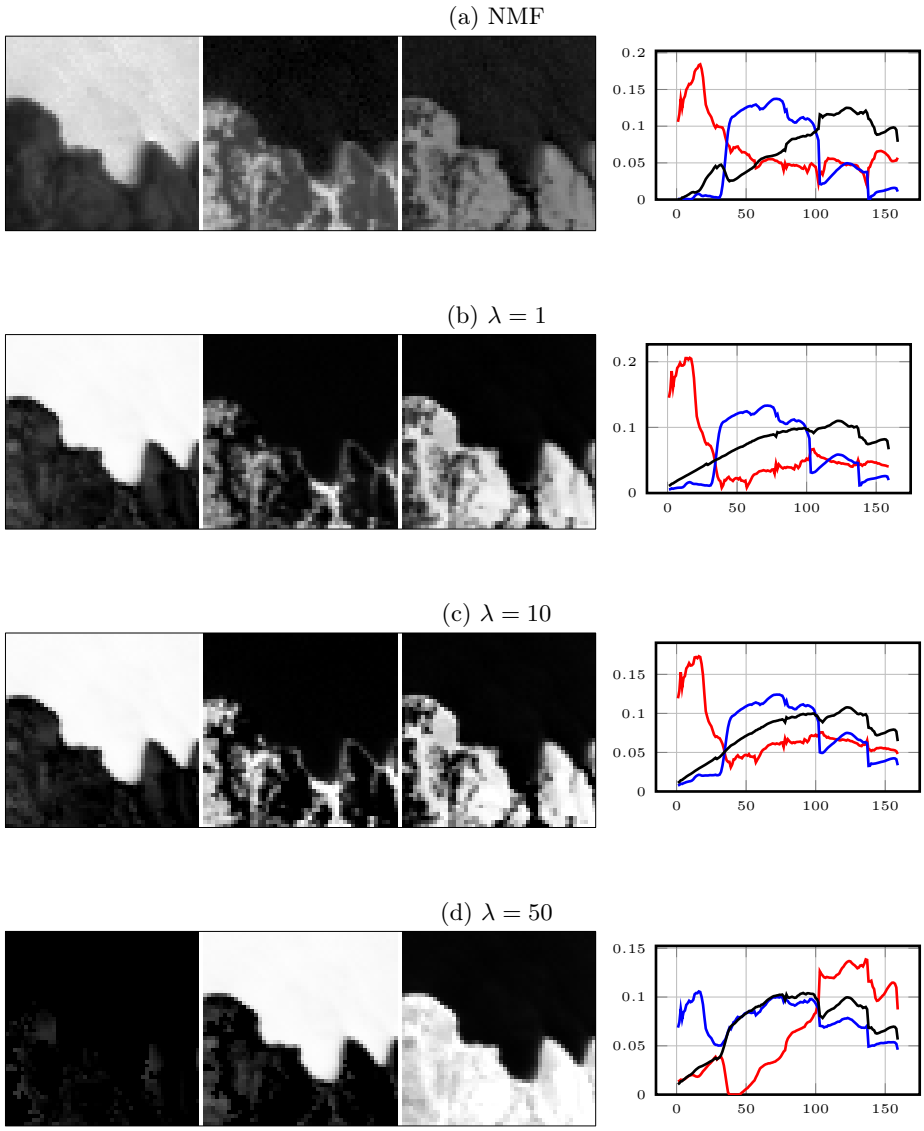


Figure 7.2: Abundance maps and normalized endmembers (from the left to the right: water, tree and soil, except for $\lambda = 50$) for MinVol on the Moffett dataset with $\delta = 0.1$.

7.2 MaxVol NMF

Let us introduce MaxVol NMF through its equivalence with MinVol NMF in the exact case. Consider the full rank SSNMF $X = \overline{W}\overline{H}$. For any full column rank matrix W of the same size as $\overline{W}$, there exists an invertible matrix Q such that $W = \overline{W}Q$. Then,

$$\det(W^\top W) = \det(Q^\top \overline{W}^\top \overline{W} Q) = \det(Q)^2 \det(\overline{W}^\top \overline{W}).$$

Minimizing $\det(W^\top W)$ is equivalent to minimizing $\det(Q)^2 \det(\overline{W}^\top \overline{W})$ relatively to Q. Hence, computing the exact MinVol NMF of X is equivalent to solving

$$\begin{array}{ll} \min\limits_{Q} & \det(Q)^2 \\ \text{s.t.} & \overline{W}Q \geq 0, Q^{-1}\overline{H} \in \Delta^{r\times n}. \end{array} \tag{7.1}$$

An obvious MinVol NMF of X is then $(\overline{W}Q, Q^{-1}\overline{H})$. Minimizing the quantity $\det(Q)^2$ is equivalent to maximizing the quantity $\det(Q^{-2})$. To sum up, in the exact case, minimizing the volume of W is equivalent to maximizing the volume of H. Here is the exact MaxVol NMF formulation:

$$\begin{array}{ll} \max\limits_{W,H} & \det(HH^\top) \\ \text{s.t.} & X = WH, \\ & W \geq 0, H \in \Delta^{r\times n}. \end{array} \tag{7.2}$$

7.2.1 Identifiability of MaxVol NMF

MaxVol NMF is just as identifiable as MinVol NMF. Actually, the proof is almost exactly the same as the one for MinVol NMF.

Theorem 7.1 *Let $X = WH$ be a MaxVol NMF of X of size $r = \operatorname{rank}(X)$, in the sense of (7.2). If H satisfies SSC as in Definition 2.8, then MaxVol NMF (W, H) of X is essentially unique.*

Proof 7.1 *Let $Q \in \mathbb{R}^{r\times r}$ be an invertible matrix such that (WQ^{-1}, QH) is another feasible solution of (7.2). There exists a right inverse $H^\dagger$ such that $HH^\dagger = I$ because $\operatorname{rank}(H) = r$. Since $e^\top H = e^\top$ and $e^\top QH = e^\top$ because (WQ^{-1}, QH) is feasible, we have*

$$e^\top Q = e^\top QHH^\dagger = e^\top H^\dagger = e^\top HH^\dagger = e^\top. \tag{7.3}$$

For the same reasons as in the proof of Theorem 6.1, that is, from (6.4) to (6.7), we have

$$\begin{aligned} |\det(Q)| &\leq \prod_{i=1}^{r} \|Q(i,:)\|_2 \\ &\leq \prod_{i=1}^{r} Q(i,:)e \\ &\leq \left(\frac{\sum_{i=1}^{r} Q(i,:)e}{r}\right)^r = \left(\frac{e^\top Qe}{r}\right)^r = 1, \end{aligned} \tag{7.4}$$

where the first inequality is coming from the Hadamard's inequality, the second from (6.7)*, and the last one from the arithmetic-geometric mean inequality and that* $e^\top Q = e^\top$*.*

Suppose now that (WQ^{-1}, QH) *is also an optimal solution to* (7.2)*. Then,*

$$\det(QHH^\top Q^\top) = \det(HH^\top) \tag{7.5}$$
$$\Leftrightarrow \quad |\det(Q)|^2 \det(HH^\top) = \det(HH^\top) \tag{7.6}$$
$$\Leftrightarrow \quad |\det(Q)| = 1. \tag{7.7}$$

The remainder of the proof is exactly like in Theorem 6.1.

7.2.2 Behavior of MaxVol NMF

In the inexact case, we consider the following MaxVol NMF formulation:

$$\begin{aligned} \min_{W,H} \quad & f(W,H) := \frac{1}{2}\|X - WH\|_F^2 - \lambda \operatorname{logdet}(HH^\top + \delta I) \\ \text{s.t.} \quad & W \geq 0, H \in \Delta^{r\times n}. \end{aligned} \tag{7.8}$$

It should be noted that, unlike MinVol NMF, from an optimization perspective, the δ term in the logdet is not needed anymore. Maximizing the logdet will prevent H from being rank deficient. Still, we keep δ in our model as it has some physical meaning. This is discussed in Section 7.4.

To understand the main difference between MinVol NMF and MaxVol NMF, consider the asymptotic case when λ goes to infinity. For MinVol NMF, W will just converge to 0. For MaxVol NMF, H will converge to a matrix whose rows are mutually orthogonal and such that the l_2 norm of each row are as close to each other as possible. Let us justify this intuition by considering the problem

$$\begin{aligned} \underset{X \in \mathbb{S}^r}{\text{minimize}} \quad & f_0(X) = \log\det X^{-1} \\ \text{subject to} \quad & e^\top X e \leq a, \\ & X \geq 0, \end{aligned} \tag{7.9}$$

where $a > 0$ and $\operatorname{dom} f_0 = \mathbb{S}^r_{++}$. We want to prove that $X = \frac{a}{r} I$ is the unique minimizer of (7.9). We solve this problem through its dual using the conjugate of f_0, like in [13, Section 5.1.6].

Definition 7.1 *The conjugate* f^* *of a function* $f : \mathbb{R}^r \to \mathbb{R}$ *is given by*

$$f^*(y) = \sup_{x \in \operatorname{dom} f} \left(y^\top x - f(x)\right).$$

Considering the optimization problem with linear inequality and equality constraints

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & Ax \leq b, \\ & Cx = d, \end{aligned} \tag{7.10}$$

the conjugate of f_0 can be used to write the dual function for (7.10) as

$$g(\lambda,\nu) = \inf_x \left(f_0(x) + \lambda^\top(Ax-b) + \nu^\top(Cx-d)\right) \tag{7.11}$$

$$= -b^\top\lambda - d^\top\nu + \inf_x \left(f_0(x) + (A^\top\lambda + C^\top\nu)^\top x\right) \tag{7.12}$$

$$= -b^\top\lambda - d^\top\nu - f_0^*(-A^\top\lambda - C^\top\nu). \tag{7.13}$$

The domain of g follows from the domain of f_0^*:

$$\operatorname{dom} g = \{(\lambda,\nu) | -A^\top\lambda - C^\top\nu \in \operatorname{dom} f_0^*\}.$$

Let us go back to the conjugate function of f_0, which is defined as

$$f_0^*(Y) = \sup_{X\succ 0} \left(\langle Y, X\rangle + \log\det X\right).$$

We first show that $\langle Y, X\rangle + \log\det X$ is unbounded above unless $Y \prec 0$. If $Y \not\prec 0$, then Y has an eigenvector v, with $\|v\|_2 = 1$, and eigenvalue $\lambda \geq 0$. Taking $X = I + tvv^\top$ we find that

$$\langle Y, X\rangle + \log\det X = \operatorname{tr} Y + t\lambda + \log\det(I + tvv^\top) = \operatorname{tr} Y + t\lambda + \log(1+t),$$

which is unbounded above as $t \to \infty$. Now consider the case $Y \prec 0$. We can find the maximizing X by setting the gradient with respect to X equal to zero:

$$\nabla_X(\langle Y, X\rangle + \log\det X) = Y + X^{-1} = 0,$$

which leads to $X = -Y^{-1}$. Therefore, we have

$$f_0^*(Y) = \log\det(-Y)^{-1} - r \tag{7.14}$$

with $\operatorname{dom} f_0^* = -\mathbb{S}^r_{++}$.

Applying the result in (7.13), the dual function for problem (7.9) is given by

$$g(\lambda,\nu) = \begin{cases} \log\det\left(\lambda J - \sum_{i,j}\nu_{i,j}E_{i,j}\right) + r - \lambda a & \text{if } \lambda J - \sum_{i,j}\nu_{i,j}E_{i,j} \succ 0, \\ \infty & \text{otherwise,} \end{cases} \tag{7.15}$$

with $\lambda \in \mathbb{R}_+$ and $\nu \in \mathbb{R}_+^{r\times r}$. Let $\lambda^* = \frac{r}{a}$, $\nu^*_{i,j} = \frac{r}{a}$ if $i \neq j$, $\nu^*_{i,j} = 0$ if $i = j$ and $X^* = \frac{a}{r}I$. We have $f_0(X^*) = g(\lambda^*, \nu^*)$, meaning that there is no duality gap and that X^* is a solution of (7.9). Finally, X^* is the unique solution because f_0 is strongly convex.

Due to this result and to the fact that $e^\top HH^\top e = n$, if $n = dr$ where $d \in \mathbb{N}^*$, then increasing λ will make $HH^\top$ converge to a diagonal whose elements are all equal to d. In other words, the rows of H will be mutually orthogonal. The simplex constraint on the columns of H and the fact that the rows are mutually orthogonal will impose that $H(i,j) \in \{0,1\}$. The norm of each row is then just the square root of the number of

non-zero elements in the corresponding row. From the HU point of view, one pixel will be assigned to only one material. This is equivalent to a hard clustering where every cluster should be of the same size. On a side note, if n is not a multiple of r, then the norm of each row will have to be different. The clustering behavior of MaxVol NMF is interesting and offers more control over the sparsity of the decomposition than MinVol NMF. Also, maximizing the volume of H indirectly minimizes the volume of W without the drawback of potentially setting a useful endmember to zero due to its low reflectance. However, the fact that increasing λ tends to an even clustering is a clear weakness. See the experiment on Figure 7.3 on Samson. Increasing λ intensifies the clustering, until a hard clustering is achieved with $\lambda = 50$. Increasing λ removes some of the false positives for water, but not all of them. This is probably because there are more pixels containing trees than water or stone in this dataset. Correctly assigning the water false positives to tree will unbalance even more the size of the clusters, though MaxVol NMF favors clusters of the same size. Also, the improvement of the abundance map of the water is at the cost of a hard clustering, while a soft clustering would be preferable to properly unmix soil and tree.

In Section 7.4, we present an improved variant of MaxVol NMF where the volume of the row wise normalized H is maximized instead. This variant is an improvement as it is not biased towards clusters of the same size.

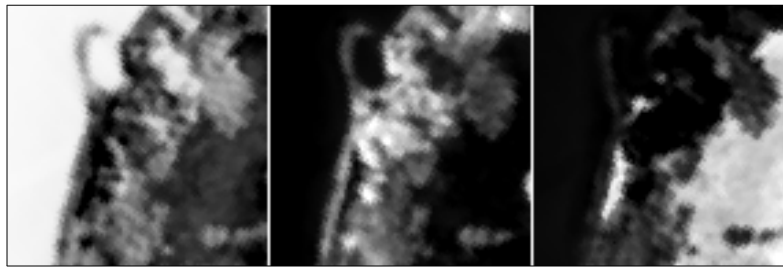
Remark 7.1 *About the results in Figure 7.3:*

- *λ is tuned using [82], in the same way we used it in Section 6.4.3.*
- *In Section 7.3 we show two different algorithms to solve MaxVol NMF. The abundance maps displayed on Figure 7.3 are the same regardless of the used algorithm, except for $\lambda = 50$ where the adaptive gradient method crashes, probably due to some numerical issues. The ADMM based algorithm still works well with $\lambda = 50$.*

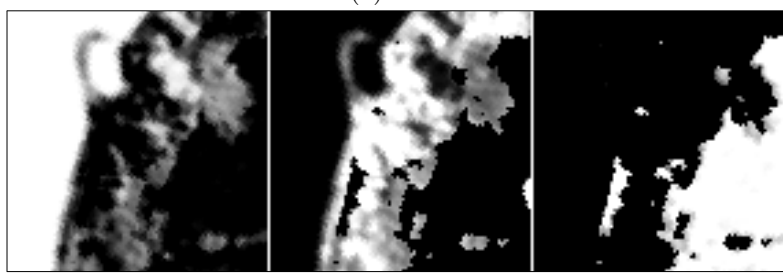
7.3 Solving MaxVol NMF

The most common strategy to solve problems like Eq. (7.8) is to use an alternated block optimization scheme. Consider blocks of variables, while updating one block, fix the others. When the update is finished, repeat the same process for the next block. Here, we only consider two blocks: W and H. The main difficulty in solving Eq. (7.8) holds in the $-\lambda \operatorname{logdet}$ term. Since $X \to \operatorname{logdet}(X)$ is concave, it is easy to derive a surrogate for Eq. (6.2) relatively to W whose gradient is Lipschitz continuous. The first-order Taylor approximation at the current iterate W^k is enough, as it has been seen in Section 6.3.1.1. It is then possible to update W by minimizing the obtained Lipschitz surrogate. This is exactly equivalent to performing a projected gradient step with a step size equal to the inverse of the Lipschitz constant of the gradient of the surrogate, that is $\frac{1}{\|HH^\top + \lambda(W^{k\top} W^k + \delta I)^{-1}\|}$. Since $-\operatorname{logdet}(.)$ is not concave, it prevents us from using for Eq. (7.8) the same update strategy that has been derived for Eq. (6.2). In this section, we propose several algorithms to solve Eq. (7.8). The

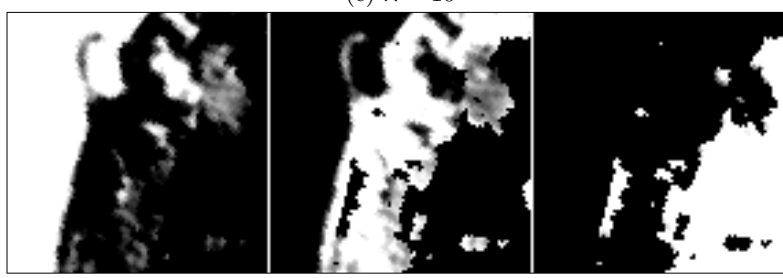
(a) $\lambda = 0.5$



(b) $\lambda = 5$



(c) $\lambda = 10$



(d) $\lambda = 50$

Figure 7.3: Abundance maps of MaxVol NMF on Samson, depending on λ.

first algorithm in Section 7.3.1 is adapted from [73]. Its core idea is to approximate the local Lipschitzness by using the previous iterate and to compute the corresponding Lipschitz gradient descent. The second algorithm is based on the Alternating Direction Method of Multipliers (ADMM).

Let us note that relatively to W, another choice could be to consider each of its columns as a block, also known as HALS [40]. As the update of W is not the main concern while solving Eq. (7.8) with an alternated block scheme, we will not explore this possibility. Note also that HALS could not be used to update H. It would alternatively update the rows of H, although they depend on each other because of the probability simplex constraint $H \in \Delta^{r \times n}$.

7.3.1 Adaptive accelerated gradient descent

Our first proposed algorithm for Eq. (7.8) relies on [73, Alg. 2]. This algorithm uses the previous iterate to approximate the local Lipschitzness and derive an appropriate step size. The previous iterate is also used to induce some extrapolation. The only knowledge that is needed from f is its gradient. It should be noted that [73, Alg. 2] is only designed for a one block variable, that is, all variables are updated at the same time. In our case, it would mean that $[W^\top, H] \in \mathbb{R}^{r \times (m+n)}$ should be updated all at once. Most gradient based algorithm for constrained matrix factorization are using a two block alternating strategy, performing several updates on W and then several updates on H. By doing so, a gradient based two block alternating algorithm can save computation time by precomputing some matrix operations that remain unchanged during the update of one block. We will follow this common two block strategy and, as it has been said before, all we need are the gradients:

$$\begin{aligned} \nabla f(W) &= \frac{\partial f}{\partial W}(W) \\ &= (WH - X)H^\top, \end{aligned} \tag{7.16}$$

$$\begin{aligned} \nabla f(H) &= \frac{\partial f}{\partial H}(H) \\ &= W^\top(WH - X) - 2\lambda(HH^\top + \delta I)^{-1}H. \end{aligned} \tag{7.17}$$

Our adaptation of [73, Alg. 2] with a two block strategy is given in Algorithm 7.1.

Remark 7.2 *The adaptive part is mostly useful for the update of H. In order to update W any other algorithm could be used instead of the* ***while*** *loop in Algorithm 7.1.*

Algorithm 7.1: Adgrad2

Input: data matrix $X \in \mathbb{R}^{m\times n}$, initial factors $W_o \in \mathbb{R}_+^{m\times r}$ and $H_o \in \Delta^{r\times n}$

$\Gamma_{W_o} = \|H_o H_o^\top\|, \gamma_{W_o} = {\Gamma_{W_o}}^{-1}, \theta_W = \Theta_W = 10^9, \overline{W}_o = W_o, \overline{W} = W = [W_o - 10^{-6}\nabla f(W_o)]_+$

$\Gamma_{Ho} = \|W_o^\top W_o\|, \gamma_{Ho} = {\Gamma_{Ho}}^{-1}, \theta_H = \Theta_H = 10^9, \overline{H}_o = H_o, \overline{H} = H = [H_o - 10^{-6}\nabla f(H_o)]_{\Delta^{r\times n}}$

for $k = 1, 2, \ldots$ **do**

  while *stopping criteria not satisfied* **do**

    $\gamma_W = \min\left(\gamma_{W_o}\sqrt{1+\frac{\theta_W}{2}}, \frac{\|\overline{W}-\overline{W}_o\|_F}{2\|\nabla f(\overline{W})-\nabla f(\overline{W}_o)\|_F}\right)$

    $\Gamma_W = \min\left(\Gamma_{W_o}\sqrt{1+\frac{\Theta_W}{2}}, \frac{\|\nabla f(\overline{W})-\nabla f(\overline{W}_o)\|_F}{2\|\overline{W}-\overline{W}_o\|_F}\right)$

    $W = [\overline{W} - \gamma_W \nabla f(\overline{W})]_+$

    $\theta_W = \gamma_W/\gamma_{W_o}, \Theta_W = \Gamma_W/\Gamma_{W_o}$

    $\overline{W}_o = \overline{W}$

    $\overline{W} = W + \frac{1-\sqrt{\gamma_W\Gamma_W}}{1+\sqrt{\gamma_W\Gamma_W}}(W - W_o)$

    $W_o = W$

    $\gamma_{W_o} = \gamma_W, \Gamma_{W_o} = \Gamma_W$

  while *stopping criteria not satisfied* **do**

    $\gamma_H = \min\left(\gamma_{H_o}\sqrt{1+\frac{\theta_H}{2}}, \frac{\|\overline{H}-\overline{H}_o\|_F}{2\|\nabla f(\overline{H})-\nabla f(\overline{H}_o)\|_F}\right)$

    $\Gamma_H = \min\left(\Gamma_{H_o}\sqrt{1+\frac{\Theta_H}{2}}, \frac{\|\nabla f(\overline{H})-\nabla f(\overline{H}_o)\|_F}{2\|\overline{H}-\overline{H}_o\|_F}\right)$

    $H = [\overline{H} - \gamma_H \nabla f(\overline{H})]_{\Delta^{r\times n}}$

    $\theta_H = \gamma_H/\gamma_{H_o}, \Theta_H = \Gamma_H/\Gamma_{H_o}$

    $\overline{H}_o = \overline{H}$

    $\overline{H} = H + \frac{1-\sqrt{\gamma_H\Gamma_H}}{1+\sqrt{\gamma_H\Gamma_H}}(H - H_o)$

    $H_o = H$

    $\gamma_{H_o} = \gamma_H, \Gamma_{H_o} = \Gamma_H$

7.3.2 Alternating direction method of multipliers (ADMM) for the MaxvolMF problem

Let us consider the following ADMM reformulation of Eq. (7.8):

$$\begin{aligned} \min_{W,H,Y,\Lambda} \quad & \mathcal{L}(W,H,Y,\Lambda) := \frac{1}{2}\|X - WH\|_F^2 - \lambda \operatorname{logdet}(Y + \delta I) + \langle Y - HH^\top, \Lambda \rangle \\ & \qquad + \frac{\rho}{2}\|Y - HH^\top\|_F^2 \\ \text{s.t.} \quad & W \geq 0, H \in \Delta^{r \times n}. \end{aligned} \tag{7.18}$$

According to [9], the ADMM algorithm consists of the following updates:

$$W^{k+1} = \operatorname*{argmin}_{W \geq 0} \mathcal{L}(W, H^k, Y^k, \Lambda^k) \tag{7.19}$$

$$H^{k+1} = \operatorname*{argmin}_{H \geq \Delta^{r \times n}} \mathcal{L}(W^{k+1}, H, Y^k, \Lambda^k) \tag{7.20}$$

$$Y^{k+1} = \operatorname*{argmin}_{Y} \mathcal{L}(W^{k+1}, H^{k+1}, Y, \Lambda^k) \tag{7.21}$$

$$\Lambda^{k+1} = \Lambda^k + \rho(Y^{k+1} - H^{k+1}{H^{k+1}}^\top) \tag{7.22}$$

Updating W Like in Section 7.3.1, the update for W can be computed through any algorithm for constrained convex problems, as Eq. (7.19) is equivalent to

$$W^{k+1} = \operatorname*{argmin}_{W \geq 0} \frac{1}{2}\|X - WH^k\|_F^2,$$

where $W \to \frac{1}{2}\|X - WH^k\|_F^2$ is convex. Here we propose to use TITAN [51] with a Lipschitz surrogate, like in Section 6.3. The resulting update for W^{k+1} is detailed in Algorithm 7.2

Algorithm 7.2: Update of W with TITAN

Input: α_1, X, H^k, W, W_o
Output: W

$L_W = \|H^k {H^k}^\top\|$
while *stopping criteria not satisfied* **do**
  $\alpha_0 = \alpha_1$
  $\alpha_1 = \frac{1}{2}(1 + \sqrt{1 + 4\alpha_0^2})$
  $\beta = \frac{\alpha_0 - 1}{\alpha_1}$
  $\overline{W} = W + \beta(W - W_o)$
  $W_o = W$
  $W = [\overline{W} + \frac{1}{L_W}(XH^\top - \overline{W}H^k{H^k}^\top)]_+$

Updating H We propose two ways of updating H. The first one consists of solving directly Eq. (7.20) with the adaptive accelerated gradient descent algorithm described in Section 7.3.1. The second one, that we will describe here, consists of deriving a non-Euclidean gradient method. Basically, we find a Bregman surrogate of $H \to \mathcal{L}(W^{k+1}, H, Y^k, \Lambda^k) := \mathcal{L}(H)$ and update H by minimizing this surrogate instead. The main motivation to use such a surrogate is that there does not exist a Lipschitz surrogate of $\mathcal{L}$ relatively to H. The gradient of $H \to \mathcal{L}(W^{k+1}, H, Y^k, \Lambda^k)$ is clearly not Lipschitz continuous because the gradient of $\|Y - HH^\top - \delta I\|_F^2$ relatively to H is cubic. Although $H \to \mathcal{L}(H)$ is not L-smooth, using the framework of [7], we can show that it is smooth relatively to the quartic norm kernel proposed in [28].

Definition 7.2 (Bregman distance)

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

with h a properly chosen convex function, dubbed a distance kernel.

Note that D_h is not a proper distance as it is asymmetric.

Definition 7.3 (Relative smoothness [7]) *We say that a differentiable function $f : \mathbb{R}^{r \times n} \to \mathbb{R}$ is L-smooth relatively to the distance kernel h if there exists $L > 0$ such that for every $X, Y \in \mathbb{R}^{r \times n}$,*

$$f(X) \leq f(Y) + \langle \nabla f(Y), X - Y \rangle + L D_h(X, Y).$$

If f is twice differentiable, L-smoothness relatively to h is equivalent to

$$\nabla^2 f(X)[U, U] \leq L \nabla^2 h(X)[U, U] \quad \forall X, U \in \mathbb{R}^{r \times n},$$

where $\nabla^2 f(X)[U, U]$ denotes the second derivative of f at X in the direction U.

First, we focus on the relative smoothness of the quartic term. According to [28] we have

$$\frac{1}{2}\|Y - HH^\top\|_F^2 := g(H) \leq g(H^k) + \langle \nabla g(H^k), H - H^k \rangle + D_h(H, H^k) \tag{7.23}$$

where $\nabla g(H^k) = 2(H^k {H^k}^\top - Y)H^k$, and $h(H) = \frac{\alpha}{4}\|H\|_F^4 + \frac{\sigma}{2}\|H\|_F^2$ with $\alpha = 6$ and $\sigma = 2\|Y\|_2$. Substituting (7.23) in (7.18),

$$\mathcal{L}(H) \leq u_{H^k}(H) := \frac{1}{2}\|X - WH\|_F^2 - \langle HH^\top, \Lambda \rangle + \rho \langle \nabla g(H^k), H \rangle + \rho h(H) - \rho \langle \nabla h(H^k), H \rangle + C_H \tag{7.24}$$

where C_H is a constant relatively to H. Compute the second directional derivative of u_{H^k}

$$\begin{aligned} \nabla^2 u_{H^k}(H)[U, U] =& \langle (W^\top W - 2\Lambda^\top)U, U \rangle + \rho\sigma\|U\|_F^2 + \rho\alpha(\|H\|_F^2\|U\|_F^2 + 2\langle H, U \rangle^2), \\ \leq& \rho\alpha(\|H\|_F^2\|U\|_F^2 + 2\langle H, U \rangle^2) + (\|W^\top W - 2\Lambda^\top\|_2 + \rho\sigma)\|U\|_F^2, \\ =& \nabla^2 \left(\frac{\tilde{\alpha}}{4}\|H\|_F^4 + \frac{\tilde{\sigma}}{2}\|H\|_F^2 \right)[U, U], \end{aligned} \tag{7.25}$$

where $\tilde{\alpha} = \rho\alpha$ and $\tilde{\sigma} = \rho\sigma + \|W^\top W - 2\Lambda^\top\|_2$. From (7.25) and (7.24), $H \to \mathcal{L}(H)$ is 1-smooth relatively to the kernel $\tilde{h} : H \to \frac{\tilde{\alpha}}{4}\|H\|_F^4 + \frac{\tilde{\sigma}}{2}\|H\|_F^2$. More explicitly,

$$\mathcal{L}(H) \leq u_{H^k}(H^k) + \langle \nabla u_{H^k}(H^k), H - H^k \rangle + D_{\tilde{h}}(H, H^k). \tag{7.26}$$

The update for H is then obtained by minimizing the aforementioned surrogate

$$\begin{aligned} H^{k+1} &= \underset{H \in \Delta^{r\times n}}{\operatorname{argmin}} \left\{ \langle \nabla u_{H^k}(H^k), H \rangle + \tilde{h}(H) - \langle \nabla \tilde{h}(H^k), H \rangle \right\}, \\ &= \underset{H \in \Delta^{r\times n}}{\operatorname{argmin}} \left\{ t_k(H) := \tilde{h}(H) - \langle Q^k, H \rangle \right\}, \end{aligned} \tag{7.27}$$

where $Q^k = \nabla \tilde{h}(H^k) - \nabla u_{H^k}(H^k)$. This is equivalent to the Bregman proximal iteration map described in [28] with a step size equal to 1.

Corollary 7.1 *The solution of* (7.27) *is of the form*

$$H^{k+1} = \frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2 + \tilde{\sigma}} [Q^k - e\nu^\top]_+,$$

where $\nu \in \mathbb{R}^n$.

Proof 7.2 *Consider the Lagrangian of* (7.27)

$$\mathcal{L}_{t_k}(H, \Lambda, \nu) = t_k(H) - \langle H, \Lambda \rangle + \langle H^\top e - e, \nu \rangle$$

where $\Lambda \in \mathbb{R}_+^{r\times n}$ *and* $\nu \in \mathbb{R}^n$. *According to the KKT optimality conditions:*

$$\begin{cases} H^{k+1} \in \Delta^{r\times n}, & (7.28) \\ \langle \Lambda^*, H^{k+1} \rangle = 0, & (7.29) \\ \nabla t_k(H^{k+1}) - \Lambda^* + e{\nu^*}^\top = 0, & (7.30) \end{cases}$$

$$\Leftrightarrow \begin{cases} H^{k+1} \in \Delta^{r\times n}, & (7.31) \\ \langle \nabla \tilde{h}(H^{k+1}) - Q^k + e{\nu^*}^\top, H^{k+1} \rangle = 0, & (7.32) \\ \nabla \tilde{h}(H^{k+1}) - Q^k + e{\nu^*}^\top \geq 0, & (7.33) \end{cases}$$

where (7.32) *is coming from substituting* (7.30) *in* (7.29), *and* (7.33) *is coming from the fact that* $\Lambda^* \geq 0$. *First, combining* (7.32) *and* (7.33), *we have*

$$(\nabla \tilde{h}(H^{k+1}) - Q^k + e{\nu^*}^\top) \circ H^{k+1} = 0 \tag{7.34}$$

where $\circ$ *is the Hadamard product. For all* p *in* $1, \ldots, r$, *for all* j *in* $1, \ldots, n$,

1. *if* $Q^k(p,j) - \nu_j^* < 0$, $\nabla \tilde{h}(H^{k+1})(p,j) - (Q^k(p,j) - \nu_j^*) > 0$ *because* $\nabla \tilde{h}(H) = (\tilde{\alpha}\|H\|_F^2 + \tilde{\sigma})H \geq 0$, *then* (7.34) $\Rightarrow H^{k+1}(p,j) = 0$,
2. *if* $Q^k(p,j) - \nu_j^* = 0$, $\nabla \tilde{h}(H^{k+1})(p,j) = (\tilde{\alpha}\|H^{k+1}\|_F^2 + \tilde{\sigma})H^{k+1}(p,j)$ *so* (7.34) $\Rightarrow H^{k+1}(p,j) = 0$,

3. *if* $Q^k(p,j) - \nu_j^* > 0$, $\nabla\tilde{h}(H^{k+1})(p,j) = (\tilde{\alpha}\|H^{k+1}\|_F^2 + \tilde{\sigma})H^{k+1} > 0$ *by* (7.33), *then* (7.34) $\Rightarrow$ $\nabla\tilde{h}(H^{k+1})(p,j) - (Q^k(p,j) - \nu_j^*) = 0 \Leftrightarrow H^{k+1}(p,j) = \frac{Q^k(p,j)-\nu_j^*}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}$.

In the end, $H^{k+1} = \frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}[Q^k - e\nu^{*\top}]_+$.

In particular, ν in Corollary 7.1 is such that $e^\top[Q^k - e\nu^\top]_+ = (\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma})e^\top \in \mathbb{R}^n$ since $e^\top H^{k+1} = e^\top$. In other words, $[Q^k - e\nu^\top]_+$ projects Q on a scaled probability simplex where the scaling is equal to $\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}$. How do we find ν since it depends on H^{k+1}? We propose to solve this inexactly with a simple fixed point algorithm where $\|H^{k+1}\|_F^2$ is the variable to optimize. The idea is that when $\|H^{k+1}\|_F^2$ is fixed, ν has a closed form solution. So for a fixed $\|H^{k+1}\|_F^2$ we compute ν, then we update $\|H^{k+1}\|_F^2$ according to the new ν and repeat this process. The algorithm is described in Algorithm 7.3. When $\|H^{k+1}\|_F^2$ is fixed, there are several algorithms that can compute exactly ν. In [49], the proposed algorithm requires to sort the entries of each column of Q^k. The complexity of this algorithm is mainly due to this sorting. Once the sorting is completed, ν is found just by computing n times the max between r entries, which is linear. There exist faster algorithms like [22] that do not rely on sorting. However, note that Q^k is not changing in Algorithm 7.3. Hence, using [49] to compute ν in Algorithm 7.3 only has a linear complexity if Q^k is sorted only once before the **while** loop. In our code, ϵ is fixed to 10^{-6} and the **while** loop cannot exceed 100 iterations.

Algorithm 7.3: Algorithm for (7.27)

Input: $Q^k, \tilde{\alpha}, \tilde{\sigma}$
init: $\|H^{k+1}\|_F^2, \nu$
Output: H^{k+1}

while $\frac{\|H^{k+1}\|_F^2 - \left\|\frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}[Q^k - e\nu^\top]_+\right\|_F^2}{\|H^{k+1}\|_F^2} > \epsilon$ **do**
  compute ν such that $\frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}[Q^k - e\nu^\top]_+ \in \Delta^{r\times n}$
  update $\|H^{k+1}\|_F^2$ to $\left\|\frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}[Q^k - e\nu^\top]_+\right\|_F^2$
$H^{k+1} = \frac{1}{\tilde{\alpha}\|H^{k+1}\|_F^2+\tilde{\sigma}}[Q^k - e\nu^\top]_+$

Updating Y Recall the ADMM update of Y^{k+1}, that is

$$Y^{k+1} = \underset{Y\succ -\delta I}{\operatorname{argmin}} -\lambda\operatorname{logdet}(Y+\delta I) + \langle Y, \Lambda\rangle + \frac{\rho}{2}\|Y - HH^\top\|_F^2. \tag{7.35}$$

Consider the change of variable $Z = Y + \delta I$,

$$Y^{k+1} + \delta I = \underset{Z\succ 0}{\operatorname{argmin}} -\lambda\operatorname{logdet}(Z) + \frac{\rho}{2}\left\|Z - \left(HH^\top + \delta I - \frac{1}{\rho}\Lambda\right)\right\|_F^2. \tag{7.36}$$

According to [105, Lemma 2.1], (7.36) has a closed form solution which is

$$\Phi^+_{\frac{\lambda}{\rho}}\left(HH^\top + \delta I - \frac{1}{\rho}\Lambda\right)$$

where $\Phi^+_\gamma(x) = \frac{1}{2}(\sqrt{x^2+4\gamma}+x)$ and for a symmetric A with an eigen value decomposition $A = PDP^\top$ and $D = \mathrm{Diag}(d)$, $\Phi^+_\gamma(A) = P\,\mathrm{Diag}(\Phi^+_\gamma(d))P^\top$ where $\Phi^+_\gamma(d)$ is applied element-wise. In the end,

$$Y^{k+1} = \Phi^+_{\frac{\lambda}{\rho}}\left(HH^\top + \delta I - \frac{1}{\rho}\Lambda\right) - \delta I.$$

7.3.3 Comparison of the two algorithms

Here, we compare the different proposed algorithms for MaxVol NMF, both on synthetic datasets and real datasets. The results are averaged over 10 runs and are presented on Figure 7.5. For the synthetic dataset, W is drawn following a uniform distribution in $[0, 1]$ and H is such that each of its column is drawn following a Dirichlet distribution where the concentration parameters are all equal to 0.2. The input matrix is then just $X = WH$. A different X is drawn at each run. The compared algorithms are Adgrad2 (Section 7.3.1), ADMM (Section 7.3.2) and ADMM+Adgrad. ADMM+Adgrad has the same formulation as in (7.18), but the update for H (7.20) is performed using the adaptive gradient descent method instead of minimizing the proposed Bregman surrogate. Regardless of the dataset and of the algorithm, the number of iterations is fixed to 500, the number of inner iterations[1] is fixed to 20, λ and δ are fixed to 1, the automatic tuning of λ is switched off because it changes the cost function. In Figure 7.5, on both synthetic data and Moffett, ADMM with $\rho = 0.01$ has the best convergence speed and the lowest error. Still on synthetic data and Moffett, we can see how the proposed Bregman surrogate provides a nice approximation of the original ADMM formulation (7.18). For equal ρ's, ADMM always converges faster and to a lower error than ADMM+Adgrad. This experimentally justifies our choice for the use of a Bregman surrogate to update H in the ADMM formulation of MaxVol NMF. However, this is at the cost of a higher computation time, due to Algorithm 7.3, as it can be seen in the reported average times in Table 7.2. One can always increase the tolerance threshold ϵ in Algorithm 7.3, but should remain careful. Let us increase ϵ to 10^{-3}. The computation time of ADMM with $\rho = 0.01$ is greatly improved, as a run on the synthetic dataset only lasts 2.44s in average. However, for $\rho = 0.1$ the algorithm diverges, as it can be seen on Figure 7.4, and the computation time is increased to 6.90s in average. Finally, ADMM is not always better than Adgrad2, like with Samson on Figure 7.5c. Reasons as to why one algorithm would be better than the other are, up to now, unknown.

[1] This value represents how many times H is updated in a row before updating W, and vice-versa.

Alg.	Adgrad2	ADMM+Adgrad $\rho = 0.01$	ADMM+Adgrad $\rho = 0.1$	ADMM $\rho = 0.01$	ADMM $\rho = 0.1$
Time (s)	3.67	2.88	2.39	5.33	23.5

Table 7.2: Average time per run on synthetic datasets

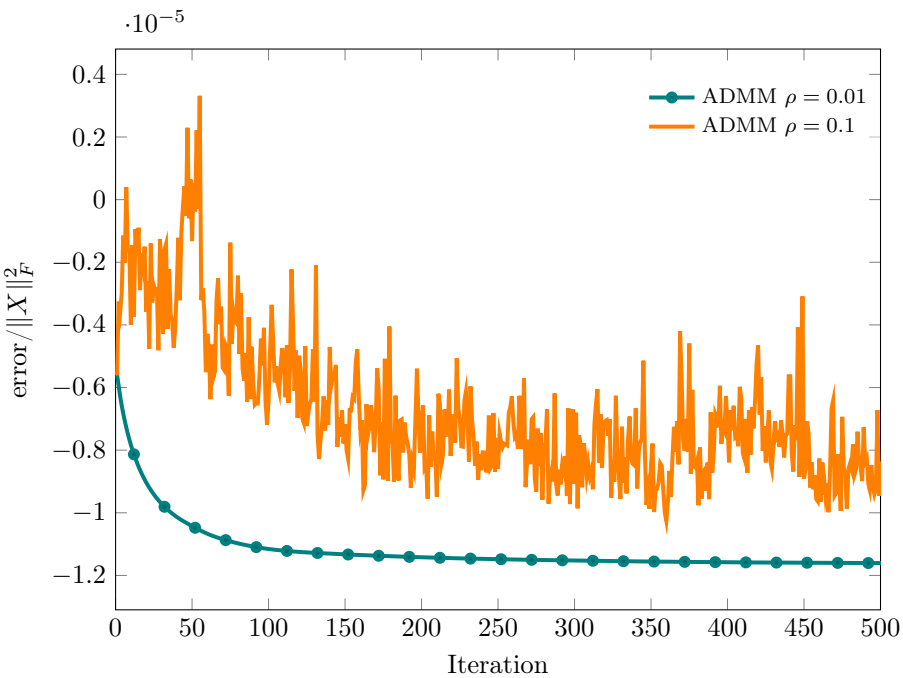


Figure 7.4: ADMM on synthetic dataset with $\epsilon = 10^{-3}$

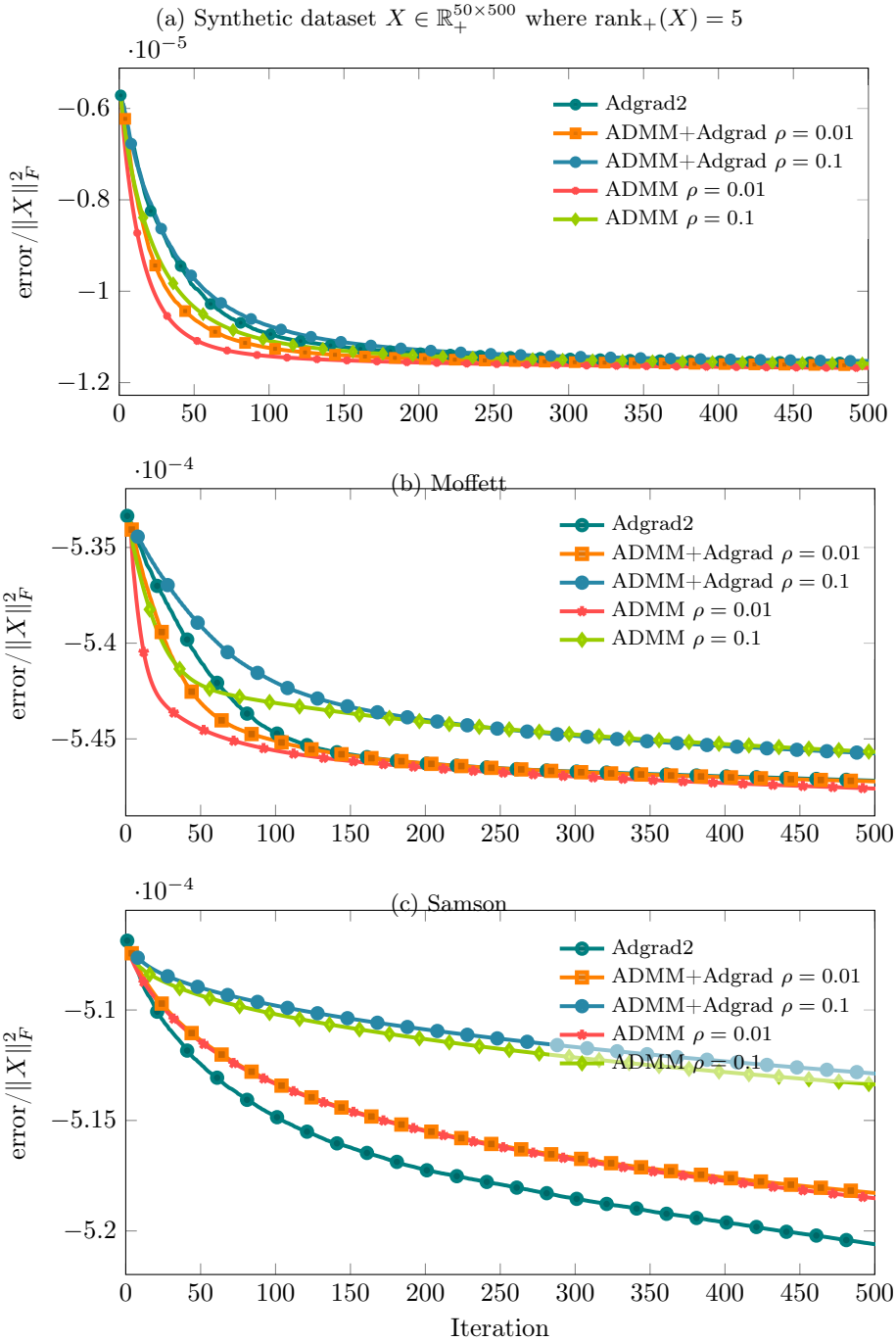


Figure 7.5: Comparison of algorithms for MaxVol NMF on various datasets

7.4 Normalized MaxVol NMF

In Section 7.2, we mentioned that a drawback of MaxVol (7.8) is its bias towards an even clustering. Here, we introduce a normalized variant of MaxVol NMF, where the volume of the row wise normalized H is maximized instead of the standard volume:

$$\begin{aligned} \min_{W,H} \quad & f(W,H) := \frac{1}{2}\|X - WH\|_F^2 - \lambda \operatorname{logdet}(\widetilde{H}\widetilde{H}^\top + \delta I) \\ \text{s.t.} \quad & W \geq 0, H \geq 0, \\ & \widetilde{H} = S^{-1}H \text{ where } S = \operatorname{Diag}(\|H(1,:)\|_2, \ldots, \|H(r,:)\|_2). \end{aligned} \tag{7.37}$$

This model is interesting for several reasons.

When λ is increasing, $\widetilde{H}\widetilde{H}^\top$ converges to the identity. In other words, increasing λ acts in favor of mutually orthogonal rows of H. Unlike MaxVol NMF, the norm of the rows of H can be anything since it is $\widetilde{H}\widetilde{H}^\top$ that converges to the identity and not $HH^\top$. In fact, Normalized MaxVol NMF can be viewed as a continuum between NMF and Orthogonal NMF (ONMF) . With $\lambda = 0$, NMF is retrieved. Increasing λ progressively retrieves ONMF. Let us prove this asymptotic behavior of Normalized MaxVol. To do so, we show that the problem

$$\begin{aligned} \underset{X \in \mathbb{S}^r}{\text{minimize}} \quad & f_0(X) = \log\det X^{-1} \\ \text{subject to} \quad & \operatorname{Diag}(X) = e, \\ & 0 \leq X \leq 1, \end{aligned} \tag{7.38}$$

where $\operatorname{dom} f_0 = \mathbb{S}^r_{++}$ has $X = I$ as a unique minimizer. Again, we solve this problem through its dual using the conjugate of f_0, which has already been computed in (7.14). First, (7.38) can be reformulated as

$$\begin{aligned} \underset{X \in \mathbb{S}^r}{\text{minimize}} \quad & f_0(X) = \log\det X^{-1} \\ \text{subject to} \quad & \langle E_{ii}, X\rangle = 1 \text{ for all } i, \\ & \langle -E_{ij}, X\rangle \leq 0 \text{ for all } i,j, \\ & \langle E_{ij}, X\rangle \leq 1 \text{ for all } i,j. \end{aligned} \tag{7.39}$$

Using again (7.13), we can write the dual of (7.38) with the conjugate of f_0:

$$g(\lambda,\gamma,\nu) = \begin{cases} \log\det\big(\operatorname{Diag}(\nu) + \gamma - \lambda\big) + r - \langle J, \gamma\rangle - e^\top \nu & \text{if } \operatorname{Diag}(\nu) + \gamma - \lambda \succ 0, \\ \infty & \text{otherwise,} \end{cases} \tag{7.40}$$

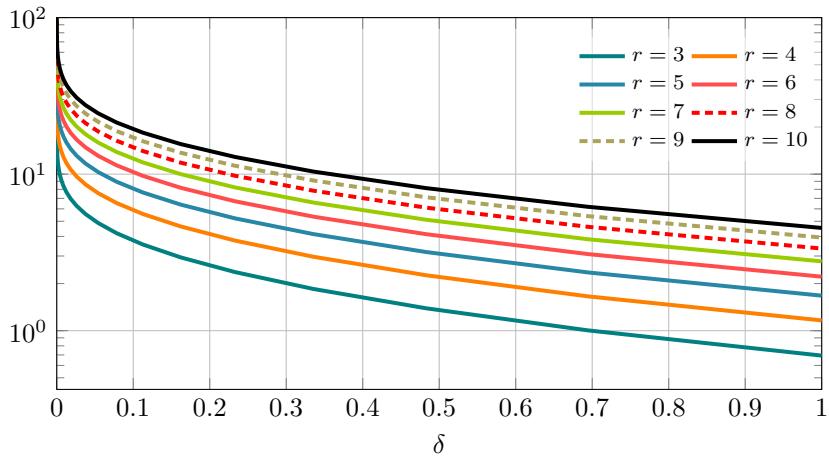
where $\lambda \in \mathbb{R}^{r\times r}_+$, $\gamma \in \mathbb{R}^{r\times r}_+$ and $\nu \in \mathbb{R}^r$. Let $\lambda^* = 0, \gamma^* = 0, \nu^* = e$ and $X^* = I$. We have $f_0(X^*) = g(\lambda^*, \gamma^*, \nu^*) = 0$, meaning that there is no duality gap and that X^* is a solution of (7.38). Finally, X^* is the unique solution because f_0 is strongly convex.

It is possible to control the range of the volume criterion via δ. When $\delta > 0$, we have

$$\operatorname{logdet}(\widetilde{H}\widetilde{H}^\top + \delta I) \in \left[\log(1 + r\delta^{-1}) + r\log\delta, r\log(1+\delta)\right]$$

where the minimum and maximum are respectfully reached when $\widetilde{H}\widetilde{H}^\top = J$ and $\widetilde{H}\widetilde{H}^\top = I$. The parameter δ then controls how larger can the volume of $\widetilde{H}$ be, while λ still balances the reconstruction error and the volume criterion. By increasing δ, the dynamical range is reduced, as it can be seen on Figure 7.6. With respect to λ and the reconstruction error, it is then harder to increase the volume of $\widetilde{H}$. In the context of HU, δ can be seen as a mixture tolerance parameter, while λ is more like a noise level estimation parameter.

Figure 7.6: Value $r\log(1+\delta) - \log(1+r\delta^{-1}) - r\log\delta$ depending on δ for various r's.



Another advantage of the normalized formulation of MaxVol NMF is the removal of the simplex structure on H. Let us remind that this simplex structure is not without loss of generality. In HU for instance, if there are two pure pixels of tree but one of them receives more light than the other, then a perfect unmixing would require a different grass endmember for each one. In other words, the simplex structure might require a larger rank. Also, the projection on the probability simplex is costly.

In spite of the benefits the normalized variant brings, we "lose" two aspects of the vanilla MaxVol NMF. The most notable one is the identifiability. It remains unknown if Normalized MaxVol NMF is identifiable or not. We also lose the possibility to solve Normalized MaxVol NMF with the same ADMM formulation that we used for MaxVol NMF. We could not find a kernel that would provide us with a Bregman surrogate. Even if we did, the considered Bregman surrogate would then need to be nice enough to be easily solved. This is not a big issue since we can still solve it with the adaptive accelerated gradient descent method, which is described in Section 7.5.

7.5 Solving Normalized MaxVol NMF

Here, we propose to solve Normalized MaxVol NMF (7.37) with the adaptive accelerated gradient descent method, already introduced in Section 7.3.1. Like it has been said in Section 7.3.1, we only need to know the gradient. Hence, in this section, we only describe the computation of the gradient. The algorithm is exactly the same as Algorithm 7.1, except for the projected gradient step in Algorithm 7.1 that should be replaced by $H = [\overline{H} - \gamma_H \nabla f(\overline{H})]_+$ because there is no simplex structure in the normalized variant. Let us now compute the gradient of f in (7.37) relatively to H.

Knowing that

$$\frac{\partial \widetilde{H}(k,:)}{\partial H(k,j)} = \left(-\frac{H(k,1)H(k,j)}{\|H(k,:)\|^3} \quad \dots \quad \frac{\|H(k,:)\|^2 - H(k,j)^2}{\|H(k,:)\|^3} \quad \dots \quad -\frac{H(k,n)H(k,j)}{\|H(k,:)\|^3} \right) \tag{7.41}$$

$$= \frac{1}{\|H(k,:)\|^3} \left(\|H(k,:)\|^2 e_j^\top - H(k,j)H(k,:) \right), \tag{7.42}$$

and using the chain rule, we have that

$$\frac{\partial \operatorname{logdet}(\widetilde{H}\widetilde{H}^\top + \delta I)}{\partial H(k,j)} = \left\langle \frac{\partial \operatorname{logdet}(\widetilde{H}\widetilde{H}^\top + \delta I)}{\partial \widetilde{H}}, \frac{\partial \widetilde{H}}{\partial H(k,j)} \right\rangle \tag{7.43}$$

$$= \left\langle 2(\widetilde{H}\widetilde{H}^\top + \delta I)^{-1}\widetilde{H}, \frac{1}{\|H(k,:)\|} E_{k,j} - \frac{H(k,j)}{\|H(k,:)\|^3} e_k H(k,:) \right\rangle \tag{7.44}$$

$$= \frac{1}{\|H(k,:)\|} \langle 2(\widetilde{H}\widetilde{H}^\top + \delta I)^{-1}\widetilde{H}, E_{k,j} \rangle \tag{7.45}$$

$$- \frac{1}{\|H(k,:)\|} \langle 2(\widetilde{H}\widetilde{H}^\top + \delta I)^{-1}, e_k \widetilde{H}(k,:)\widetilde{H}^\top \rangle \widetilde{H}(k,j).$$

In the end,

$$\frac{\partial \operatorname{logdet}(\widetilde{H}\widetilde{H}^\top + \delta I)}{\partial H} = 2S^{-1} \left[(\widetilde{H}\widetilde{H}^\top + \delta I)^{-1} - \operatorname{Diag}\left((\widetilde{H}\widetilde{H}^\top + \delta I)^{-1}\widetilde{H}\widetilde{H}^\top \right) \right] \widetilde{H} \tag{7.46}$$

and

$$\frac{\partial f}{\partial H} = W^\top (WH - X) - 2\lambda S^{-1} \left[(\widetilde{H}\widetilde{H}^\top + \delta I)^{-1} - \operatorname{Diag}\left((\widetilde{H}\widetilde{H}^\top + \delta I)^{-1}\widetilde{H}\widetilde{H}^\top \right) \right] \widetilde{H}. \tag{7.47}$$

7.6 Performance of Normalized MaxVol NMF on Hyperspectral Unmixing

In this section, we evaluate the performance of Normalized MaxVol NMF on famous hyperspectral datasets. Results can be compared with [112] where some ground-truths for a variety of known hyperspectral datasets are proposed. Even if these are called ground-truths, hyperspectral ground-truths do not exist except if the measurements are performed in a controlled environment. Consider the proposed abundance maps for Urban with four endmembers in [112]. Clearly, some trees are detected where in fact it should be a mixture of grass and soil. Some rooftops are also detected where it should be soil. Still, the author used as many a priori knowledge as possible to provide these abundance maps and endmembers that are probably close to reality. Our message here is that ground-truths for these hyperspectral datasets should be interpreted with caution. On Moffett and on Samson, our model clearly outperforms MinVol NMF and MaxVol NMF, see Figures 7.7 and 7.8. Water, soil and tree are correctly separated and their spectral signatures are very close to the expected ones in [112].

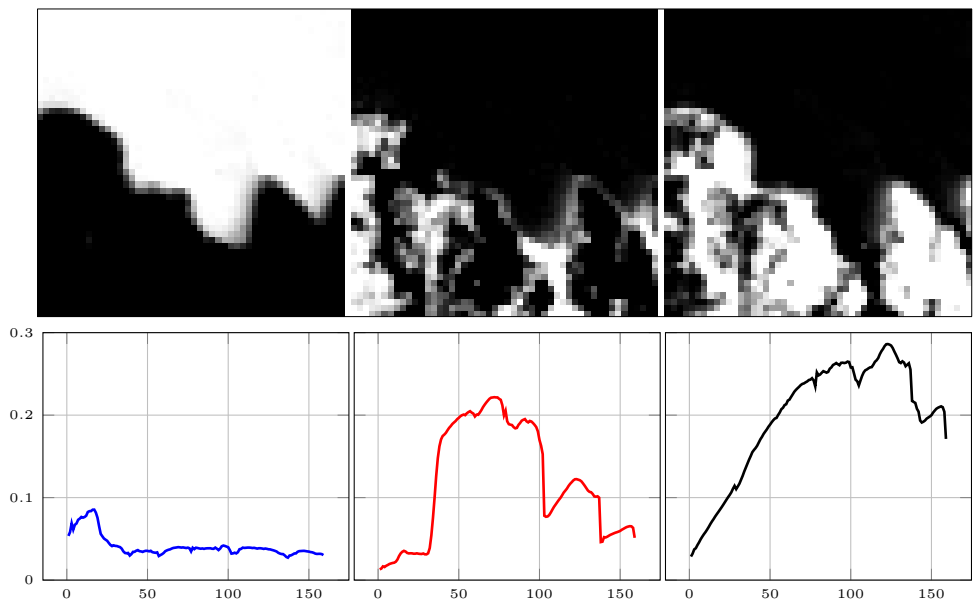


Figure 7.7: Abundance maps and endmembers (water, tree and soil) by Normalized MaxVol NMF on Moffett, with $\lambda = 1$ and $\delta = 0.5$.

Now that we very briefly assured that our model works on simple datasets, let us comment on one of its interesting features. Consider the Samson dataset again, but with $r = 6$. With MinVol NMF, the excessive endmembers should be brought to zero by tuning λ and δ. Otherwise, MinVol NMF would be over-parameterized and would

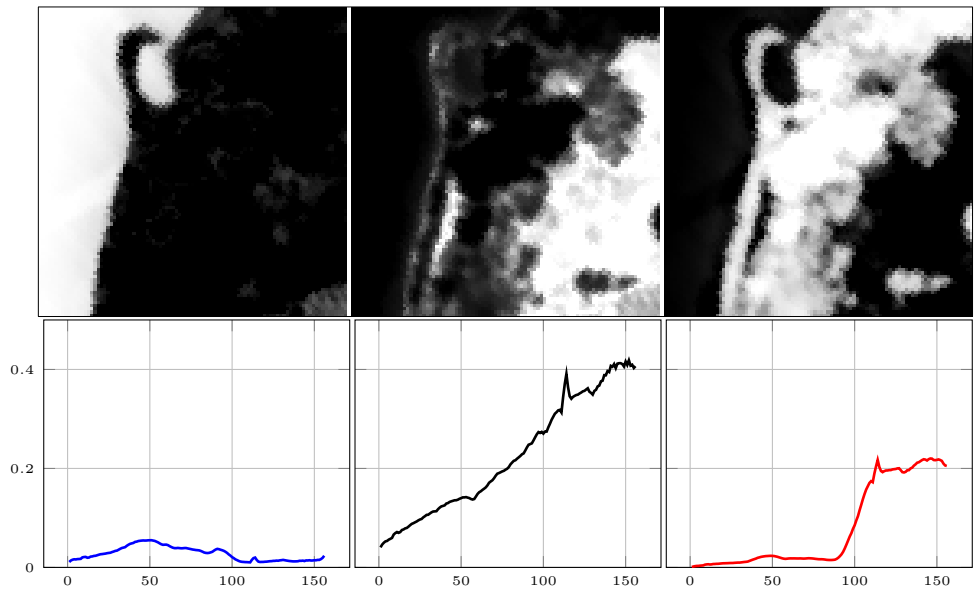


Figure 7.8: Abundance maps and endmembers (water, soil and tree) by Normalized MaxVol NMF on Samson, with $\lambda = 1$ and $\delta = 0.5$.

just learn the noise, which would output a non-interpretable matrix factorization. On Figure 7.9, we can see that increasing the rank above the standard $r = 3$ for Samson allowed MaxVol NMF to learn more spectral varieties. We can see three different kinds of tree and two different kinds of soil. We can then add together the rows of H that corresponds to varieties of the same endmembers. The resulting merged abundance maps are available on Figure 7.10. One can notice how close are the abundance maps on Figure 7.10 and Figure 7.8 to each other. Actually, results with $r = 6$ are more satisfying for the water unmixing. On Figure 7.8, some small artifacts of false-positives can be seen, especially in the bottom right corner of the abundance map corresponding to water. These artifacts are not visible on Figure 7.10. It would seem that MaxVol NMF allows to increase the number of parameters in order to improve the results in a controlled manner, at least in the context of HU.

Let us confirm this behavior on the Urban dataset. This dataset is particularly insightful in this case because it is known for having meaningful unmixing results for $r = 4, 5$ and 6. Results are displayed on Figure 7.11. With $r = 4$, we have roof, grass, a combination of asphalt and soil, and tree. With $r = 5$, the distinction is being made between asphalt and dirt. With $r = 6$, the distinction is being made between grass and dry grass. Typical ground-truths with $r = 6$ rather suggest a distinction between two kinds of roof, instead of grass and dry grass. Here our model propose another interpretation for $r = 6$ which still makes sense.

The last experiment is on the Jasper dataset, where ground-truths suggest four endmembers: tree, water, soil and road. Unmixing algorithms often struggle to cor-

rectly separate water and road on Jasper. On Figure 7.12, we can see that our model achieves not ideal but nonetheless decent results. The issue with our model here is that in order to improve the distinction between water and road, λ should be increased. However, increasing λ might not be the best option here. There are many areas where tree and soil are heavily mixed, and increasing λ will converge to ONMF, which cannot properly unmix these areas. One way to circumvent that is to increase the rank. Results with $r = 5$ are displayed on Figure 7.13. The additional endmember is in fact a combination of tree and soil. With this trick, water and road are properly identified without compromising the quality of the other endmembers.

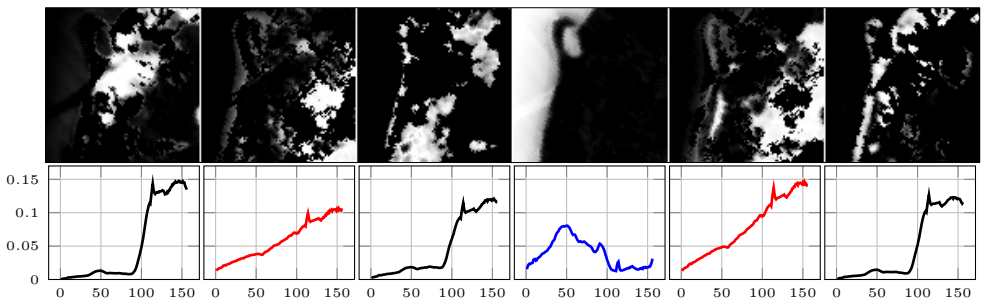


Figure 7.9: Abundance maps and endmembers (tree, soil and water) by Normalized MaxVol NMF with $r = 6$ on Samson, with $\lambda = 0.5$ and $\delta = 0.5$.

Remark 7.3 *On the results of our model, only the shape of the spectral signatures should be considered when comparing results, while the amplitude of the spectral signatures are to be considered with a pinch of salt due to the scaling ambiguity between W and H. Let us remind that Normalized MaxVol NMF is not simplex structured. In fact, we could take advantage of this scaling ambiguity to improve the condition number when updating a block, but this is not the goal of this chapter.*

7.7 Conclusion

In this chapter, we introduced MaxVol NMF, an analogue version of MinVol NMF where the volume of H is maximized instead of the volume of W being minimized. Just like MinVol NMF, this new model is identifiable. We also developed two different algorithms to solve MaxVol NMF. We introduced Normalized MaxVol NMF, a variant where the volume of the row wise normalized H is maximized. This model creates a continuum between NMF and ONMF and exhibits better results than MinVol NMF on hyperspectral unmixing. Its identifiability remains an open question. Similarly, a normalized version of MinVol NMF could be interesting. This could be seen as a minimum aperture NMF. One could say that this already exists through MinVol NMF with the constraint $e^\top W = e^\top$, which is partially true. For a fixed aperture, the volume of W is changing depending on where the columns of W are on the probability simplex. In other words, there is a little bias drawing the columns of W towards e, which is not the case with a normalized version of MinVol NMF. Normalized

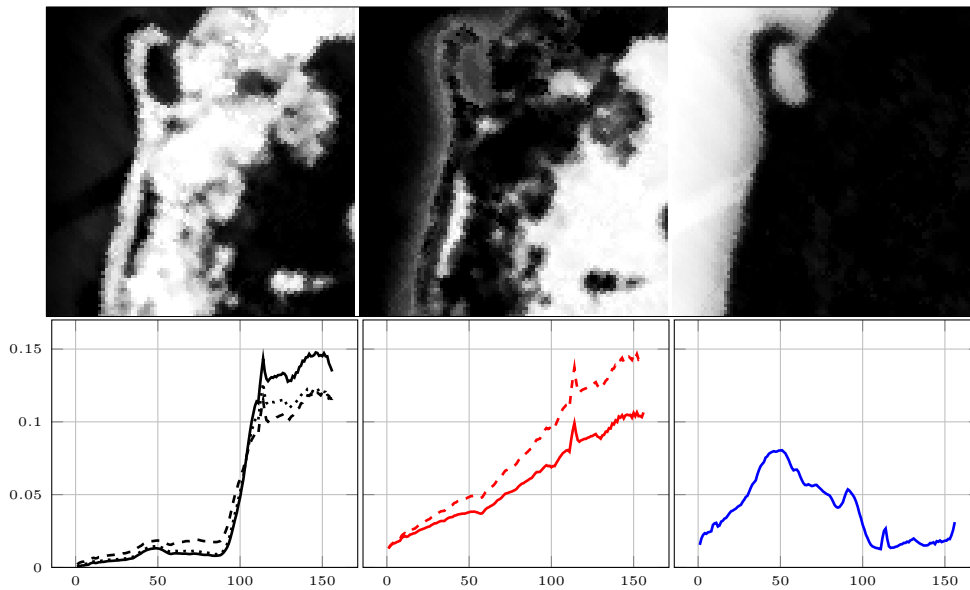


Figure 7.10: Abundance maps grouped by endmember varieties and endmembers (tree, soil and water) by Normalized MaxVol NMF with $r = 6$ on Samson, with $\lambda = 0.5$ and $\delta = 0.5$.

MaxVol NMF and Normalized MinVol NMF could be combined to control the spectral variability when increasing the rank. Finally, the performance of Normalized MaxVol NMF should be evaluated on other kinds of data.

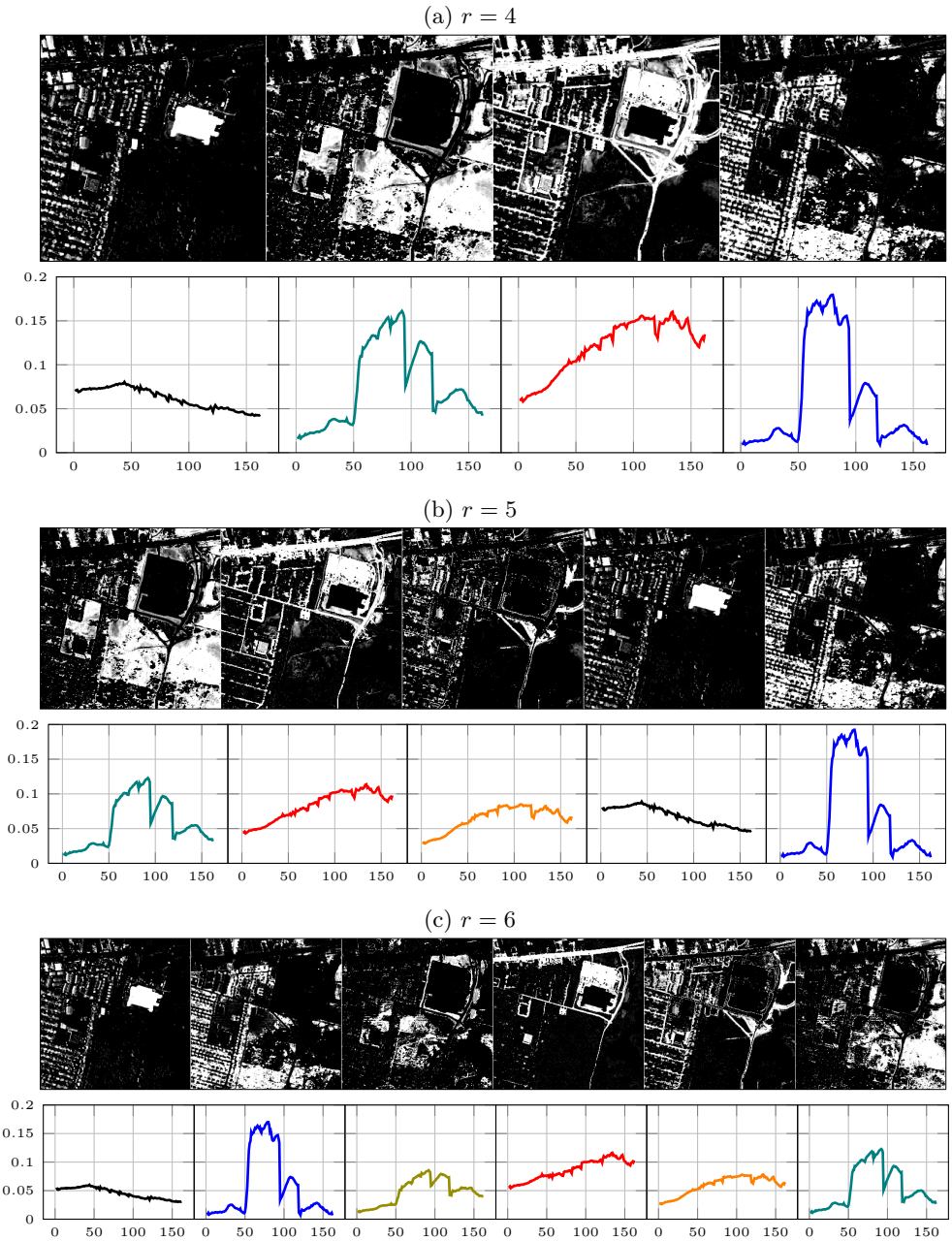


Figure 7.11: Abundance maps and endmembers (roof, tree, dry grass, asphalt, soil, grass) by Normalized MaxVol NMF on Urban, with $\lambda = 0.5$ and $\delta = 0.5$, depending on r.

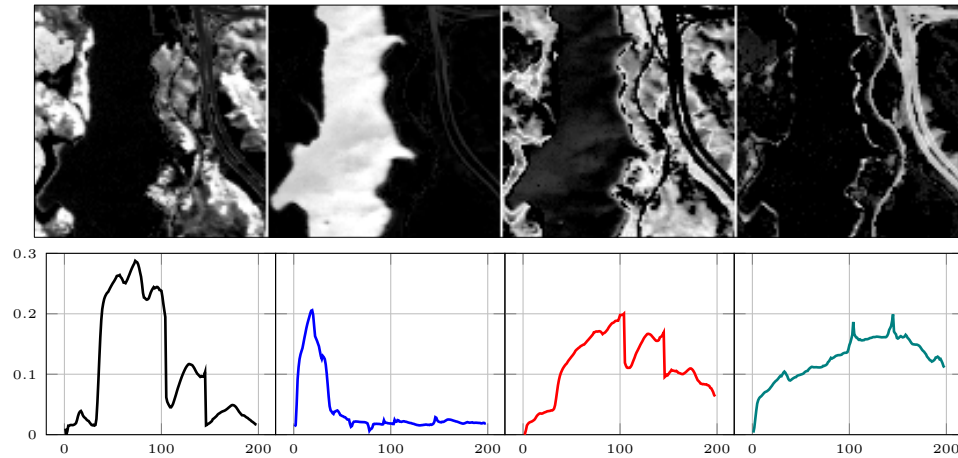


Figure 7.12: Abundance maps and endmembers (tree, water, soil, road) by Normalized MaxVol NMF with $r = 4$ on Jasper, with $\lambda = 2$ and $\delta = 1$.

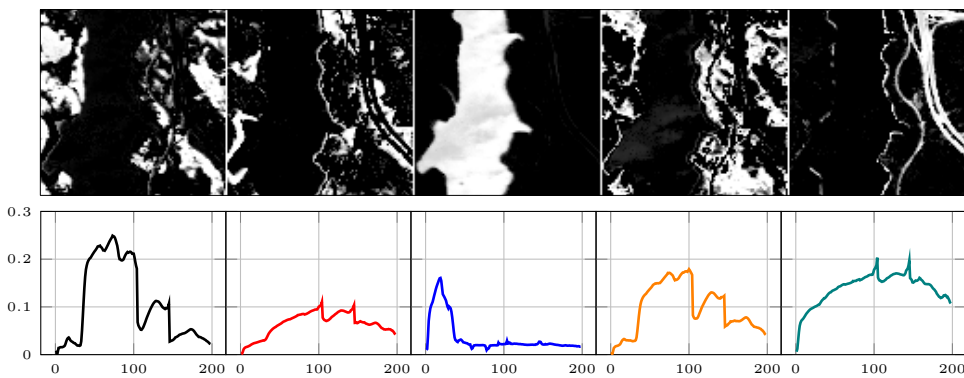


Figure 7.13: Abundance maps and endmembers (tree, soil, water, tree+soil, road) by Normalized MaxVol NMF with $r = 5$ on Jasper, with $\lambda = 0.5$ and $\delta = 0.5$.

Chapter 8

Highlight of the contributions and discussions

Andrew Prahlow - Final Voyage

In the conclusion, we first summarize the contributions of this thesis. We then discuss perspectives that could follow this work.

Summary

The three main motivations of this thesis were the applications, algorithms and theory related to identifiable volume-based regularized matrix factorization models. Except for separable NMF in Chapter 5, all studied models can be seen as volume-regularized models. With BSSMF, in Chapter 3, the columns of W lie in a chosen hyperrectangle and the columns of H lie in the probability simplex. With PMF, in Chapter 4, the rows of W and the columns of H lie in respective chosen polytopes. For BSSMF and PMF, the volume regularization is in fact a hard constraint. Such constraint can always be translated to a regularization term, using an indicator function that is added to the cost function. For instance, the constraint $H \in \Delta^{r\times n}$ can be replaced by adding $\iota_{\Delta^{r\times n}}(H)$ to the cost function, where $\iota_{\Delta^{r\times n}}(H)$ outputs 0 if every column of H lies in Δ^r, and ∞ otherwise. With MinVol NMF, in Chapter 6, the volume of the convex hull formed by the columns of W and the origin is minimized. With MaxVol NMF, in Chapter 7, it is the volume of the convex hull formed by the rows of H and the origin that is maximized.

In terms of applications, each model is useful for different reasons:

- BSSMF retrieves datalike features when the data is naturally bounded. In a way, to rephrase the well known "NMF learns parts of objects", we can say that "BSSMF learns meaningful objects". We also showed how BSSMF is a better basis for recommender systems than NMF.
- Separable NMF is useful when the looked for features are assumed to be data

points themselves, also called the *separability assumption*. This is the case in some blind source unmixing applications when each source is observed purely at least once, like it can be for hyperspectral unmixing for instance.

- When the separability assumption does not hold anymore, either due to the noise and outliers or due to the absence of pure endmembers among the data points, MinVol NMF is often a good alternative. It has also been used for other applications where NMF already showed its capabilities, like blind source separation problems [65], facial feature extraction [111] or community detection [53], to cite a few. Additionnaly, we showed how the MinVol criterion is promising as a regularizer for the task of matrix completion.

- MaxVol NMF creates a continuum between NMF and ONMF. It inherits from the same behaviors as MinVol NMF, but with more control on the sparsity of the decomposition. Actually, in the inexact case, MinVol NMF directly regularizes the volume of W, which indirectly affects the volume of H and offers little control on the sparsity of the decomposition. On the contrary, MaxVol NMF directly regularizes the volume of H, offering more control on the sparsity of the decomposition, and indirectly affects the volume of W. In the context of HU, for datasets close enough to the separability assumption due to noise, MaxVol NMF seems to exhibit better results than MinVol NMF. It should be noted that MaxVol NMF is probably less powerful for datasets composed only of mixtures. MaxVol NMF where the rank is overestimated also shows interesting results to take into account spectral variability.

Except for PMF[1], we developed fast algorithms for every studied model:

- For BSSMF and MinVol NMF, we derived instances from an inertial block majorization minimization framework for nonsmooth nonconvex optimization, called TITAN [51].

- For separable NMF, we developed RandSPA. It creates a continuum between SPA and VCA, which are fast greedy algorithms for column subset selection. RandSPA uses the best from both worlds if tuned accordingly, that is, the robustness of SPA and the randomness of VCA.

- For MaxVol NMF, the algorithm developed for MinVol NMF could not be used anymore. Hence, we developed two algorithms. One is Adgrad2, based on [73], and the other is based on ADMM, combined with an appropriate Bregman surrogate adapted from [28].

[1]The main reason for not developping an algorithm for PMF is that this model is too generic. It is totally possible to use one of the many Frank-Wolfe based algorithms to derive an alternated block scheme for any PMF. However, these algorithms are not very fast. For specific polytopes, it is probably faster to compute the projection on the polytope after a gradient descent step. We empirically noticed this when the polytopes are the probability simplex, where projected gradient descents were faster than alternating with Polyhedral Coordinate Descent method with Away steps [76] for instance.

Further research

Applications Recommender systems were the main motivation for creating BSSMF. Even if we showed that BSSMF performs better than NMF, the question remains on the competitivity of BSSMF against the state-of-the-art algorithms used for recommender systems. Of course, BSSMF would need to be customized, e.g., by adding some wisely chosen regularizers.

Normalized MaxVol NMF has only been used for HU. This model could be useful in other applications, like document clustering and recommender systems. Also, it should be noted that the maximum-volume criterion was originally thought as a regularizer for Bilinear NMF. This combination still needs to be explored. Normalized MaxVol NMF also opened the path to normalized MinVol NMF. The difference with vanilla MinVol NMF is that instead of minimizing the volume formed by the endmembers, the aperture between the endmembers is minimized. We mentioned that this behavior coupled with an overranked normalized MaxVol NMF would be able to control the spectral variability in HU. This still needs to be addressed properly, as well as other potential applications for normalized MinVol NMF.

Algorithms We showed that by taking the best run among several, RandSPA outperforms SPA and VCA. However, RandSPA could be further improved if we could learn a good Q matrix. A first idea would be to use an internal loop, instead of running the algorithm several times and saving the best run. The reason is that a good RandSPA run needs r successive good draws of random Q's. If at each selection step we could draw several Q's directly and chose the best one based on a proper criterion, this would probably improve the performance of one run of RandSPA. The main question is then which criterion would be a good one.

Theory All the studied models were either known to be identifiable (separable NMF and MinVol NMF), or proven in this thesis to be identifiable (BSSMF, PMF, ℓ_1-MinVol NMF, MaxVol NMF). We also studied/mentioned some other variants of MinVol NMF and MaxVol NMF, namely

- normalized MaxVol NMF (7.2),
- normalized MinVol NMF, whose identifiability result should be similar to that of normalized MaxVol NMF,
- MinVol NMF with a Frobenius penalty and without simplex structure. This model was defined for the inexact and missing data case in (6.27). Even when no data are missing, that is when $\mathcal{P}_\Omega$ is the identity (6.28), better conditions than Theorem 2.1 are unknown. Experiments in Section 6.4.3 strongly suggest that there exist milder conditions.

Due to their nonnegative nature, these models are obviously identifiable under Theorem 2.1 that requires both factors to satisfy the SSC. However, it is an open question to come up with milder conditions than Theorem 2.1 to retain identifiability, like Theorem 6.1 (that requires only one factor to be SSC).

In general, the identifiability of many CLRMFs with missing data remains unknown and quite challenging. Without success, we tried during this thesis to find reasonable conditions under which separable NMF with missing data would be identifiable. Of course, it would be possible to use a two-step approach: first complete the data using the low-rank assumptions (the completion is unique if sufficiently many entries are observed in random locations, see, e.g., [16]), then apply an identifiable NMF algorithms on the completed data. However, in general, single-step approaches perform significantly better in practice.

Bibliography

1. Abbas, K., Puigt, M., Delmaire, G. & Roussel, G. Locally-Rank-One-Based Joint Unmixing and Demosaicing Methods for Snapshot Spectral Images. Part I: a Matrix-Completion Framework. *IEEE Transactions on Computational Imaging* (2024).
2. Abdolali, M. & Gillis, N. Simplex-structured matrix factorization: Sparsity-based identifiability and provably correct algorithms. *SIAM Journal on Mathematics of Data Science* **3,** 593–623 (2021).
3. Ang, A. M. S. & Gillis, N. Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12,** 4843–4853 (2019).
4. Araújo, M. C. U., Saldanha, T. C. B., Galvao, R. K. H., Yoneyama, T., Chame, H. C. & Visani, V. The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. *Chemometrics and intelligent laboratory systems* **57,** 65–73 (2001).
5. Arora, S., Ge, R., Kannan, R. & Moitra, A. *Computing a nonnegative matrix factorization-Provably* in *44th Annual ACM Symposium on Theory of Computing, STOC'12* (2012), 145–161.
6. Bakshi, A., Bhattacharyya, C., Kannan, R., Woodruff, D. P. & Zhou, S. *Learning a Latent Simplex in Input Sparsity Time* in *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).
7. Bauschke, H. H., Bolte, J. & Teboulle, M. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research* **42,** 330–348 (2017).
8. Berman, A. & Plemmons, R. J. *Nonnegative Matrices in the Mathematical Sciences* (SIAM, 1994).
9. Bertsekas, D. P. Nonlinear programming: 3rd Edition (2016).
10. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review* **59,** 65–98 (2017).
11. Bjorck, A. & Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of computation* **27,** 579–594 (1973).

12. Bolte, J., Sabach, S. & Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* **146,** 459–494 (2014).

13. Boyd, S., Boyd, S. P. & Vandenberghe, L. *Convex optimization* (Cambridge university press, 2004).

14. Cai, J.-F., Candès, E. J. & Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization* **20,** 1956–1982 (2010).

15. Candes, E. & Recht, B. Exact matrix completion via convex optimization. *Communications of the ACM* **55,** 111–119 (2012).

16. Candes, E. J. & Plan, Y. Matrix completion with noise. *Proceedings of the IEEE* **98,** 925–936 (2010).

17. Candès, E. J., Li, X., Ma, Y. & Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)* **58,** 11 (2011).

18. Chandrasekaran, V., Sanghavi, S., Parrilo, P. A. & Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization* **21,** 572–596 (2011).

19. 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* (2007), 169–176.

20. Cichocki, A., Zdunek, R., Phan, A. H. & Amari, S.-i. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation* (John Wiley & Sons, 2009).

21. Condat, L. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming* **158,** 575–585 (2016).

22. Condat, L. Fast projection onto the simplex and the l 1 ball. *Mathematical Programming* **158,** 575–585 (2016).

23. Craig, M. D. Minimum-volume transforms for remotely sensed data. *IEEE Trans. Geosci. Remote Sens.* **32,** 542–552 (1994).

24. Cruces, S. Bounded component analysis of linear mixtures: A criterion of minimum convex perimeter. *IEEE Transactions on Signal Processing* **58,** 2141–2154 (2010).

25. d'Aspremont, A., El Ghaoui, L., Jordan, M. I. & Lanckriet, G. R. G. A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM Review* **49,** 434–448 (2007).

26. Deville, Y. From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms. *Digital Signal Processing* **87,** 21–33 (2019).

27. Donoho, D. & Stodden, V. *When does non-negative matrix factorization give a correct decomposition into parts?* in *Advances in Neural Information Processing Systems (NIPS)* (2004), 1141–1148.

28. Dragomir, R.-A., d'Aspremont, A. & Bolte, J. Quartic first-order methods for low-rank minimization. *Journal of Optimization Theory and Applications* **189,** 341–363 (2021).

29. Eckart, C. & Young, G. The approximation of one matrix by another of lower rank. *Psychometrika* **1,** 211–218 (1936).

30. Erdogan, A. T. A class of bounded component analysis algorithms for the separation of both independent and dependent sources. *IEEE Transactions on Signal Processing* **61,** 5730–5743 (2013).

31. Févotte, C. & Idier, J. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation* **23,** 2421–2456 (2011).

32. Fu, X., Huang, K. & Sidiropoulos, N. D. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters* **25,** 328–332 (2018).

33. Fu, X., Huang, K., Sidiropoulos, N. D. & Ma, W.-K. Nonnegative Matrix Factorization for Signal and Data Analytics: Identifiability, Algorithms, and Applications. *IEEE Signal processing magazine* **36,** 59–80 (2019).

34. Fu, X., Huang, K., Sidiropoulos, N. D. & Ma, W.-K. Nonnegative Matrix Factorization for Signal and Data Analytics: Identifiability, Algorithms, and Applications. *IEEE Signal Process. Mag.* **36,** 59–80 (2019).

35. Fu, X., Huang, K., Yang, B., Ma, W.-K. & Sidiropoulos, N. D. Robust volume minimization-based matrix factorization for remote sensing and document clustering. *IEEE Transactions on Signal Processing* **64,** 6254–6268 (2016).

36. Fu, X., Huang, K., Yang, B., Ma, W.-K. & Sidiropoulos, N. D. Robust volume minimization-based matrix factorization for remote sensing and document clustering. *IEEE Trans. Signal Process.* **64,** 6254–6268 (2016).

37. Fu, X., Ma, W.-K., Huang, K. & Sidiropoulos, N. D. Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain. *IEEE Transactions on Signal Processing* **63,** 2306–2320 (2015).

38. Full, W. E., Ehrlich, R. & Klovan, J. EXTENDED QMODEL–Objective definition of external end members in the analysis of mixtures. *Journal of the International Association for Mathematical Geology* **13,** 331–344 (1981).

39. Gabriel, K. R. & Zamir, S. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics* **21,** 489–498 (1979).

40. Gillis, N. & Glineur, F. Accelerated Multiplicative Updates and Hierarchical ALS Algorithms for Nonnegative Matrix Factorization. *Neural Computation* **24,** 1085–1105 (2012).

41. Gillis, N. *Nonnegative Matrix Factorization* (SIAM, Philadelphia, 2020).

42. Gillis, N. *Nonnegative matrix factorization* (SIAM, 2020).

43. Gillis, N. & Glineur, F. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal on Matrix Analysis and Applications* **32,** 1149–1165 (2011).

44. Gillis, N. & Kumar, A. Exact and heuristic algorithms for semi-nonnegative matrix factorization. *SIAM Journal on Matrix Analysis and Applications* **36,** 1404–1424 (2015).

45. Gillis, N. & Ma, W.-K. Enhancing pure-pixel identification performance via preconditioning. *SIAM Journal on Imaging Sciences* **8,** 1161–1186 (2015).

46. Gillis, N. & Vavasis, S. A. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36,** 698–714 (2013).

47. Gross, D. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory* **57,** 1548–1566 (2011).

48. Guan, N., Tao, D., Luo, Z. & Yuan, B. NeNMF: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing* **60,** 2882–2898 (2012).

49. Held, M., Wolfe, P. & Crowder, H. P. Validation of subgradient optimization. *Mathematical programming* **6,** 62–88 (1974).

50. Hien, L. T. K., Gillis, N. & Patrinos, P. *Inertial Block Proximal Methods for Non-Convex Non-Smooth Optimization* in *Proceedings of the 37th International Conference on Machine learning (ICML)* (2020).

51. Hien, L. T. K., Phan, D. N. & Gillis, N. An inertial block majorization minimization framework for nonsmooth nonconvex optimization. *Journal of Machine Learning Research* **24,** 1–41 (2023).

52. Honeine, P. An eigenanalysis of data centering in machine learning. *arXiv preprint arXiv:1407.2904* (2014).

53. Huang, K. & Fu, X. *Detecting Overlapping and Correlated Communities without Pure Nodes: Identifiability and Algorithm* in *Proceedings of the 36th International Conference on Machine Learning* (2019), 2859–2868.

54. Huang, K., Sidiropoulos, N. D. & Swami, A. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing* **62,** 211–224 (2013).

55. Kannan, R., Ishteva, M. & Park, H. Bounded matrix factorization for recommender system. *Knowledge and Information Systems* **39,** 491–511 (2014).

56. Kim, J. & Park, H. *Toward faster nonnegative matrix factorization: A new algorithm and comparisons* in *2008 Eighth IEEE International Conference on Data Mining* (2008), 353–362.

57. Koren, Y., Bell, R. & Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **42,** 30–37 (2009).

58. Kueng, R. & Tropp, J. A. Binary component decomposition Part I: the positive-semidefinite case. *SIAM Journal on Mathematics of Data Science* **3,** 544–572 (2021).

59. Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K. & Jensen, S. H. Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience* **2008** (2008).

60. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86,** 2278–2324 (1998).

61. Lee, D. D. & Seung, H. S. *Algorithms for non-negative matrix factorization* in *Advances in Neural Information Processing Systems (NIPS)* (2001), 556–562.

62. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401,** 788–791 (1999).

63. Leplat, V., Ang, A. M. S. & Gillis, N. *Minimum-volume Rank-deficient Nonnegative Matrix Factorizations* in *ICASSP* (2019), 3402–3406.

64. Leplat, V., Ang, A. M. & Gillis, N. *Minimum-volume rank-deficient nonnegative matrix factorizations* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2019), 3402–3406.

65. Leplat, V., Gillis, N. & Ang, M. S. Blind Audio Source Separation with Minimum-Volume Beta-Divergence NMF. *IEEE Transactions on Signal Processing* **68,** 3400–3410 (2020).

66. Leplat, V., Gillis, N. & Idier, J. Multiplicative updates for NMF with β-divergences under disjoint equality constraints. *SIAM Journal on Matrix Analysis and Applications* **42,** 730–752 (2021).

67. Liang, D., Krishnan, R. G., Hoffman, M. D. & Jebara, T. *Variational autoencoders for collaborative filtering* in *Proceedings of the 2018 world wide web conference* (2018), 689–698.

68. Lin, C.-H., Ma, W.-K., Li, W.-C., Chi, C.-Y. & Ambikapathi, A. Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Trans. Geosci. Remote Sens.* **53,** 5530–5546 (2015).

69. Liu, G., Liu, Q. & Yuan, X. *A New Theory for Matrix Completion* in *Advances in Neural Information Processing Systems* (2017).

70. Liu, K., Li, X., Zhu, Z., Brand, L. & Wang, H. Factor-Bounded Nonnegative Matrix Factorization. *ACM Trans. Knowl. Discov. Data* **15.** ISSN: 1556-4681 (2021).

71. Ma, W.-K., Bioucas-Dias, J. M., Chan, T.-H., Gillis, N., Gader, P., Plaza, A. J., Ambikapathi, A. & Chi, C.-Y. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine* **31,** 67–81 (2013).

72. Ma, W.-K., Bioucas-Dias, J. M., Chan, T.-H., Gillis, N., Gader, P., Plaza, A. J., Ambikapathi, A. & Chi, C.-Y. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Process. Mag.* **31,** 67–81 (2014).

73. Malitsky, Y. & Mishchenko, K. *Adaptive Gradient Descent without Descent* in *Proceedings of the 37th International Conference on Machine Learning* (JMLR.org, 2020).
74. Man Shun Ang, A., Cohen, J. E., Gillis, N. & Thi Khanh Hien, L. Accelerating block coordinate descent for nonnegative tensor factorization. *Numerical Linear Algebra with Applications* **28,** e2373 (2021).
75. Mansour, A., Ohnishi, N. & Puntonet, C. G. Blind multiuser separation of instantaneous mixture algorithm based on geometrical concepts. *Signal Processing* **82,** 1155–1175 (2002).
76. Mazumder, R. & Wang, H. A Cyclic Coordinate Descent Method for Convex Optimization on Polytopes. *arXiv preprint arXiv:2303.07642* (2023).
77. Miao, L. & Qi, H. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Trans. Geosci. Remote Sens.* **45,** 765–777 (2007).
78. Moussaoui, S., Brie, D. & Idier, J. *Non-negative source separation: range of admissible solutions and conditions for the uniqueness of the solution* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* **5** (2005), v–289.
79. Nadisic, N., Gillis, N. & Kervazo, C. Smoothed separable nonnegative matrix factorization. *preprint arXiv:2110.05528* (2021).
80. Nascimento, J. M. & Dias, J. M. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **43,** 898–910 (2005).
81. Nesterov, Y. *Lectures on Convex Optimization* 2nd. ISBN: 3319915770 (Springer Publishing Company, Incorporated, 2018).
82. Nguyen, D. T. & Chi, E. C. *Towards tuning-free minimum-volume nonnegative matrix factorization* in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)* (2024), 217–225.
83. Ozerov, A. & Févotte, C. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Trans. Audio Speech Lang. Process.* **18 (3),** 550–563 (2009).
84. Prévost, C. & Leplat, V. Data Fusion and Unmixing with the Regularized Non-Negative Block-Term Decomposition: Joint Problems, Blind Approach and Automatic Model Order Selection (2023).
85. Recht, B., Fazel, M. & Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* **52,** 471–501 (2010).
86. Rendle, S., Krichene, W., Zhang, L. & Koren, Y. Revisiting the Performance of iALS on Item Recommendation Benchmarks. *arXiv preprint arXiv:2110.14037* (2021).
87. Rendle, S., Krichene, W., Zhang, L. & Koren, Y. *Revisiting the performance of ials on item recommendation benchmarks* in *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), 427–435.

88. Sørensen, M., Sidiropoulos, N. D. & Swami, A. Overlapping Community Detection via Semi-Binary Matrix Factorization: Identifiability and Algorithms. *IEEE Trans. Signal Process.* **70,** 4321–4336 (2022).

89. Srebro, N. & Jaakkola, T. *Weighted low-rank approximations* in *Proceedings of the 20th International Conference on Machine learning (ICML)* **3** (2003), 720–727.

90. Srebro, N., Rennie, J. & Jaakkola, T. S. *Maximum-margin matrix factorization* in *Advances in Neural Information Processing Systems (NIPS)* (2005), 1329–1336.

91. Tatli, G. & Erdogan, A. T. *Generalized Polytopic Matrix Factorization* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2021), 3235–3239.

92. Tatli, G. & Erdogan, A. T. Polytopic matrix factorization: Determinant maximization based criterion and identifiability. *IEEE Trans. Signal Process.* **69,** 5431–5447 (2021).

93. Todd, M. J. *Minimum-volume ellipsoids: Theory and algorithms* (SIAM, 2016).

94. Udell, M. & Townsend, A. Why Are Big Data Matrices Approximately Low Rank? *SIAM Journal on Mathematics of Data Science* **1,** 144–160 (2019).

95. Vaswani, N., Chi, Y. & Bouwmans, T. Rethinking PCA for Modern Data Sets: Theory, Algorithms, and Applications [Scanning the Issue]. *Proceedings of the IEEE* **106,** 1274–1276 (2018).

96. Vavasis, S. A. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization* **20,** 1364–1377 (2010).

97. Vavasis, S. A. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization* **20,** 1364–1377 (2010).

98. Vu Thanh, O., Ang, A., Gillis, N. & Hien, L. T. K. *Inertial majorization-minimization algorithm for minimum-volume NMF* in *European Signal Processing Conference (EUSIPCO)* (2021), 1065–1069.

99. Vu Thanh, O. & Gillis, N. *Identifiability of Polytopic Matrix Factorization* in *2023 31st European Signal Processing Conference (EUSIPCO)* (2023), 1290–1294.

100. Vu Thanh, O. & Gillis, N. *Minimum-Volume Nonnegative Matrix Completion* in *European Signal Processing Conference (EUSIPCO)* (2024).

101. Vu Thanh, O., Gillis, N. & Lecron, F. *Bounded Simplex-Structured Matrix Factorization* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2022), 9062–9066.

102. Vu Thanh, O., Gillis, N. & Lecron, F. Bounded Simplex-Structured Matrix Factorization: Algorithms, Identifiability and Applications. *IEEE Transactions on Signal Processing* **71,** 2434–2447 (2023).

103. Vu Thanh, O., Nadisic, N. & Gillis, N. *Randomized Successive Projection Algorithm* in *GRETSI'22, XXVIIIème Colloque Francophone de Traitement du Signal et des Images* (2022).

104. Vu Thanh, O., Puigt, M., Yahaya, F., Delmaire, G. & Roussel, G. *In Situ Calibration of Cross-Sensitive Sensors in Mobile Sensor Arrays Using Fast Informed Non-Negative Matrix Factorization* in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), 3515–3519.

105. Wang, C., Sun, D. & Toh, K.-C. Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM Journal on Optimization* **20,** 2994–3013 (2010).

106. Wang, J., Guan, S., Liu, S. & Zhang, X.-L. Minimum-volume multichannel nonnegative matrix factorization for blind audio source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **29,** 3089–3103 (2021).

107. Wedin, P.-A. *On angles between subspaces of a finite dimensional inner product space* in *Matrix Pencils: Proceedings of a Conference Held at Pite Havsbad, Sweden, March 22–24, 1982* (2006), 263–285.

108. Wu, R., Ma, W.-K. & Fu, X. *A stochastic maximum-likelihood framework for simplex structured matrix factorization* in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* (2017), 2557–2561.

109. Xu, Y. & Yin, W. A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update. *Journal of Scientific Computing* **72,** 700–734 (2017).

110. Zhang, Z.-Y., Li, T., Ding, C., Ren, X.-W. & Zhang, X.-S. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery* **20,** 28–52 (2010).

111. Zhou, G., Xie, S., Yang, Z., Yang, J.-M. & He, Z. Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts. *IEEE Transactions on Neural Networks* **22,** 1626–1637 (2011).

112. Zhu, F. Hyperspectral unmixing: ground truth labeling, datasets, benchmark performances and survey. *arXiv preprint arXiv:1708.05125* (2017).

113. Zhuang, L., Lin, C.-H., Figueiredo, M. A. T. & Bioucas-Dias, J. M. Regularization Parameter Selection in Minimum Volume Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.* **57,** 9858–9877 (2019).