Nonnegative Matrix Factorization with Sum-of-Norms Regularization for Hyperspectral Unmixing

by

Waqas Bin Hamed

A research project report
presented to the University of Waterloo
in partial fulfillment of the
research paper requirement for the degree of
Master's of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

We introduce a Nonnegative Matrix Factorization (NMF) model with a regularization function that encourages a low-rank representation of data. We apply our method to hyperspectral unmixing, where we estimate a set of endmembers and their corresponding abundances from a hyperspectral image. Furthermore, we explore two acceleration approaches to improve the convergence of our proposed model. Our numerical experiments demonstrate the model's ability to automatically determine the model order and produce meaningful decompositions on real-world hyperspectral images. We provide the implementation in Python [1].

Acknowledgements

I would like to thank Professor Hans De Sterck and Andersen Man Shun Ang for their guidance and support during the research term.

Dedication

This is dedicated to my family and friends.

[1] https://github.com/waqasbinhamed/seminmf.git

Table of Contents

Author's Declaration ii

Abstract ii

Acknowledgements ii

Dedication ii

List of Figures vi

List of Tables viii

1 Introduction 1

1.1 Nonnegative Matrix Factorization 1

1.2 Hyperspectral Unmixing 3

2 Background 5

2.1 Sum-of-norms clustering 5

2.2 Block Coordinate Descent 6
2.3 Alternating Direction Method of Multipliers 7

3 Algorithm Development **8**

3.1 Related works/Literature Review 8
3.2 Problem Formulation 11
3.3 Solution Approach 12
3.3.1 Subproblem on h^j 12
3.3.2 Subproblem on w_j 13
3.3.3 Accelerated version 17

4 Experiments **20**

4.1 Dataset 20
4.2 Algorithm Setup 22
4.3 ADMM vs Subgradient vs Nesterov Smoothing 22
4.4 Acceleration Methods 24
4.5 NMF-SON vs NMF 26
4.6 Challenges 31

5 Conclusion **34**

5.1 Discussion 34
5.2 Further work 35

References 39

Appendices 40

A Derivation of ADMM Iterations 41

List of Figures

1.1 Nonnegative Matrix Factorization of a matrix. 1

1.2 Example of hyperspectral unmixing[1]. 3

4.1 Two versions of the Urban dataset. 21

4.2 Jasper dataset. 21

4.3 Convergence comparison of multiple approaches used to solve the w_j subproblem. The Subgradient approximation fails to regularize, and the most effective algorithm is ADMM. 23

4.4 Convergence comparison of acceleration methods. Both versions of Anderson Acceleration failed to improve convergence speed and exhibited unexpected behavior. HER was able to improve convergence speed. 25

4.5 Process of creating abundance maps and endmember spectra from hyperspectral images. 26

4.6 Results of basic NMF, with rank $r = 2$, on the small Urban dataset. Components 1 and 2 represent the roof and trees, respectively. 27

4.7 Results of basic NMF, with rank $r = 6$, on the small Urban dataset. Components 2 and 3 correspond to trees and the roof, respectively. All other components are not useful results. 27

4.8 Results of NMF-SON, with rank $r = 6$ and $\lambda = 3$, on the small Urban dataset. NMF-SON returns pairs of duplicate components: 3 and 4 represent trees, and 5 and 6 represent noise. Components 1 and 2 represent parts of the roof. 28

4.9 Results of NMF-SON, with rank $r = 6$ and $\lambda = 50$, on the small Urban dataset. All the components are identical. 29

4.10 Results of basic NMF, with rank $r = 4$, on the Jasper dataset. Components 1, 3, and 4 correspond to soil, trees and road. Component 2 is an unexpected result and most likely represents sand. 30

4.11 Results of basic NMF, with rank $r = 8$, on the Jasper dataset. Components 1, 3, and 4 represent trees, road, and soil. Other components are unclear. . 32

4.12 Results of NMF-SON, with rank $r = 8$ and $\lambda = 1$, on the Jasper dataset. The rank is reduced to $r = 7$ since components 6 and 7 are the same. Components 1, 3, and 5 represent trees, roads, and soil. Other components are unclear. 33

List of Tables

4.1 Runtimes of approaches to solve the w_j subproblem. ADMM took the longest time, followed by Subgradient Approximation and Nesterov Smoothing. 23

4.2 Runtimes of acceleration methods. Both versions of Anderson Acceleration took longer to complete. Meanwhile, HER reduced the runtime. 25

Chapter 1

Introduction

1.1 Nonnegative Matrix Factorization

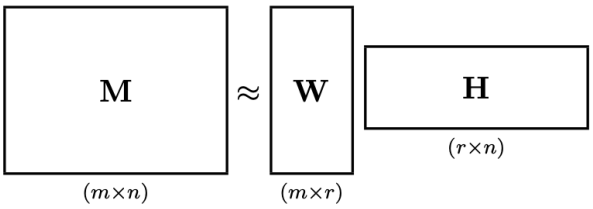


Figure 1.1: Nonnegative Matrix Factorization of a matrix.

Nonnegative Matrix Factorization[11][22][12][4][7] is a method of obtaining a low-rank representation of the original matrix. Given a matrix, $M \in \mathbb{R}_+^{m \times n}$ the aim is to find factor matrices $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ whose product is an approximation of the original matrix. The rank r is a parameter for the model specified by the user, and $r \leq \min(m, n)$. Figure 1.1 pictorially demonstrates NMF.

NMF can be rewritten as

$$m_j \approx \sum_{i=1}^{r} H_{ij} w_i, \tag{1.1}$$

where m_j and w_j are the j-th column of M and i-th column of W respectively. Equation 1.1 shows that the columns in M are approximated by a nonnegative linear combination of the columns of W multiplied by corresponding components of H[11].

NMF was introduced in 1994 by Paatero and Tapper[22] as Positive Matrix Factorization. It gained popularity after Lee and Seung's 1999 article[12] in which they compared NMF to Principal Component Analysis (PCA) and Vector Quantization (VQ) on facial images and text documents. Their work demonstrated NMF's ability to generate more meaningful parts-based representation of data as compared to PCA and VQ. They argue that the ability to learn parts-based representations is due to the additive nature of the model, as no negative values are possible. NMF has inherent clustering properties, and with additional constraints, it is equivalent to k-means clustering, a popular unsupervised clustering method[4]. For more information about NMF, the recommended reference is the "Nonnegative Matrix Factorization" book by Nicolas Gillis[7].

NMF models have been shown to perform well for various applications with nonnegative data. Xu, Liu, and Gong[32] demonstrate that NMF is an effective model for document clustering. In this case, the input matrix M contains term frequencies per document, the resulting matrix W represents term frequencies per topic (or cluster), and H represents the composition of topics per document. NMF has also been used to predict movie ratings[34], in which case M is an incomplete matrix where each element represents a user rating for a movie. The columns of the resulting matrix W represent user communities' ratings, and columns of H represent a user's affinity for those communities. The product WH is a reconstruction of the original matrix with missing values filled in. This means that the reconstruction matrix has a user's rating for a movie the user has not watched. Modified versions of NMF have also been used for gene clustering[26] and facial expression recognition[37].

NMF models generally require the rank r of the resulting representations to be specified beforehand. Selecting the rank without prior knowledge about the data can be a challenging problem. The main contribution of this work is introducing a regularized NMF model that encourages a low-rank representation of the data. We demonstrate our model's ability to adaptively select the rank and create meaningful representations on an image processing task known as Hyperspectral Unmixing.

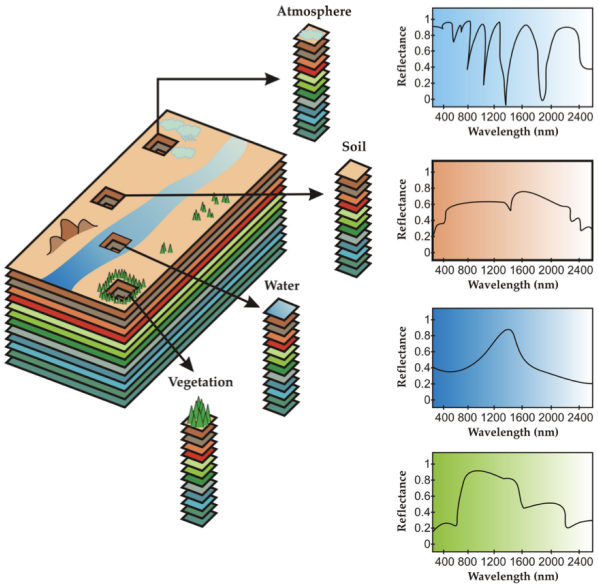


Figure 1.2: Example of hyperspectral unmixing[1].

1.2 Hyperspectral Unmixing

Hyperspectral sensors are built to function across many regions of the electromagnetic spectrum. Our work focuses on the images captured by sensors that operate at the near-infrared and shortwave infrared spectral bands[1]. In images captured by these cameras, a pixel represents the mixture of light reflected by materials in the field of view. These images are stored in a data cube, where each frontal slide of the data cube is an image corresponding to a specific wavelength. A vector of a particular pixel through all the planes represents the light reflected by the location for all spectral bands.

Hyperspectral unmixing is a technique to identify and separate the individual spectral signatures of different materials in a mixed pixel. It involves analyzing the spectral reflectance of a scene at many different wavelengths and using algorithms to identify and distinguish the unique signatures of different materials in the scene. This allows for identifying and mapping the materials in an area, which can be helpful for various applications, such as monitoring vegetation health or identifying mineral deposits. The complete process of hyperspectral unmixing generally includes:

1. **Collecting and preprocessing the hyperspectral data**: This involves acquiring

the data using hyperspectral cameras and preprocessing the data to remove noise.

2. **Identifying endmembers and estimating their abundances**: Endmembers are the spectral signatures of the materials in the scene. This step involves identifying and selecting a set of endmembers from the data that will be used in the unmixing process and using algorithms to estimate the proportion of each endmember present in each mixed pixel.

3. **Separating the endmembers**: Once the abundances of each endmember have been estimated, they can be used to separate the endmembers and produce a map of the materials present in the scene. Note that some hyperspectral unmixing methods, including NMF, perform step 2 and 3 concurrently.

4. **Postprocessing and interpretation**: The separated endmembers can be post-processed and interpreted to extract useful information about the materials in the scene.

Models for hyperspectral unmixing are categorized as either linear mixing models or non-linear mixing models[29]. Linear mixing models assume that the spectral signatures of the materials in a mixed pixel can be linearly combined to produce the observed spectrum. This is a relatively simple and computationally efficient approach, but it can be limited in its ability to accurately model complex mixing scenarios. Non-linear mixing models, on the other hand, can handle more complex mixing scenarios, but they are generally more computationally intensive.

In the remaining report, the background information for and approach to algorithm development is discussed, and numerical experiments are shared to demonstrate the effectiveness of our algorithms.

Chapter 2

Background

This chapter covers the preliminary information needed to formulate our model and develop the algorithm for solving it. Section 2.1 explains the Sum-of-norms clustering algorithm that can adaptively select the optimal number of clusters. Our model is inspired by this clustering method and shares key components. Section 2.2 elaborates on Block Coordinate Descent, a standard algorithm for solving Nonnegative Matrix Factorization. The last section 2.3 explains an algorithm for solving optimization problems in a distributed manner. It is used to solve a subproblem of our model.

2.1 Sum-of-norms clustering

Clustering is a fundamental area of unsupervised machine learning. It involves dividing data points into clusters based on their properties. Lindsten, Ohlsson, and Ljung[14] propose a clustering model called Sum-of-norms (SON) clustering with two advantages: the problem is convex, and the number of clusters does not have to be specified beforehand.

Their proposed model is a minimization problem

$$\min_{\mu_1 \cdots \mu_N} \sum_{j=1}^{N} \|x_j - \mu_j\|_2^2 + \lambda \sum_{j=2}^{N} \sum_{i<j} \|\mu_i - \mu_j\|_p, \tag{2.1}$$

where $\{x_j\}_{j=1}^N$ is the set of observations in $\mathbb{R}^d$, $\{\mu_j\}_{j=1}^N$ are the centriods of the clusters, λ is the regularization parameter and $p > 1$. The solution to this optimization problem will result in $\|\mu_i - \mu_j\|_p = 0$, and the x_j near these centroids can be seen as belonging to the same cluster. Hence, reducing the number of clusters to optimal.

2.2 Block Coordinate Descent

Block Coordinate Descent(BCD) solves non-linear optimization problems by dividing elements into subgroups and iteratively minimizing the objective function for only the elements in the selected subgroup while keeping other variables constant. It is a common approach for solving NMF[10]. For a minimization problem of the form

$$\min_{W,H} f(W, H), \tag{2.2}$$

with matrices $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$, that can be solved iteratively; there are several choices for subgroups under the BCD framework:

- A single element of a matrix is updated independently
$$w_{ij}^{k+1} \leftarrow \operatorname*{argmin}_{w_{ij}} f(w_{ij}^k, W_{-ij}, H), \tag{2.3}$$
where W_{-ij} is the W matrix without the ij-th element.
- One of the matrices is updated independently
$$W^{k+1} \leftarrow \operatorname*{argmin}_{W} f(W^k, H). \tag{2.4}$$
- A column or row of a matrix is updated in each step
$$w_j^{k+1} \leftarrow \operatorname*{argmin}_{w_j} f(w_j^k, W_{-j}, H), \tag{2.5}$$
where $j \in [n]$, W_{-j} is the W matrix without the j-th column.

We use cyclic ordering for updating each column w_j as in equation 2.5. For the remainder of the report, the notation w_j is used to denote the j-th column of matrix W, w^j is used to denote the j-th row of matrix W, and k represents the iteration number.

2.3 Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM)[3] is an algorithm that solves optimization problems by separating them into smaller subproblems that are easier to handle. The solutions to small local subproblems are used to calculate the solution to the global problem.

Consider a problem in the form

$$\begin{aligned} &\min f(x) + g(z), \\ &\text{subject to } Ax + Bz = c, \end{aligned} \tag{2.6}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{r \times n}$, $B \in \mathbb{R}^{r \times m}$ and $c \in \mathbb{R}^r$, and the functions f and g are separable and convex, can be solved using ADMM.

The augmented Lagrangian of the minimization problem 2.6 is

$$L_\rho(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2, \tag{2.7}$$

where $y \in \mathbb{R}^r$ is the Lagrangian variable and $\rho > 0$ is a penalty paramter. Using the augmented Lagrangian, we can express the iterations of the ADMM algorithm as

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x L_\rho(x, z^k, y^k), \\ z^{k+1} &= \operatorname*{argmin}_z L_\rho(x^{k+1}, z, y^k), \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \end{aligned} \tag{2.8}$$

Chapter 3

Algorithm Development

This chapter focuses on formulating our model and the algorithm for solving it. Section 3.1 discusses existing approaches for hyperspectral unmixing, including NMF models. In the remaining sections, our model is explained in detail, and three approaches for solving it are discussed. This is followed by introducing two accelerated versions of our algorithm to improve convergence.

3.1 Related works/Literature Review

There are several factors to consider when choosing a model for hyperspectral unmixing, including:

- **Noise**: The data may contain noise due to atmospheric conditions and unexpected material in the field of view. Noise can affect the accuracy and reliability of the results.
- **Computation time**: Hyperspectral unmixing can be computationally intensive, especially for large datasets. It is essential to consider the computation time required by the model and choose one that is suitable for the available resources.
- **Endmembers**: The expected number of endmembers and their purity in the field of view can affect the accuracy and interpretability of the results. Some models

for unmixing assume that at least a single pure pixel exists in the data for each endmember.

Generally, the choice of model for hyperspectral unmixing depends on the data's specific characteristics and the analysis's goals. Many algorithms exist for this purpose, including N-FINDR, PPI, NMF, and deep learning models.

N-FINDR[30] is a two-step process. First, the endmembers are identified, and then their abundances are approximated. The model relies on the Linear Mixing Model

$$m_{ij} = \sum_k w_{ik} h_{kj} + \epsilon, \tag{3.1}$$

where m_{ij} is the i-th band of the j-th pixel, w_{ik} is the i-th band of the k-th endmember, h_{kj} is abundance of the k endmember for the j-th pixel and ϵ is Gaussian random error. Equation 3.1 is equivalent to $M = WH + E$, where matrix E represents noise. The model assumes at least one pure pixel in the image for each endmember. To find the endmembers, the data is first reduced to $K-1$ dimensions, where K denotes the number of endmembers we expect, through an orthogonal subspace projection. Then, the volume of the simplex created using the endmember vectors is calculated repeatedly by replacing the endmember vectors with pixel vectors until the maximum volume is reached. Now that the endmembers have been identified, a least squares problem is solved with the physical constraint that no values of H are negative to find the endmember abundances.

PPI[2] is similar to N-FINDR but uses an alternate way to identify the endmembers. The pixels from the data are projected onto random unit vectors, and an extremity score is calculated for each of them. The cumulative extremity records for the pixels are used to identify the extreme pixels corresponding to pure endmembers.

Deep learning approaches have also been considered for unmixing. An early attempt by Licciardi and Del Frate[13] proposed a neural network architecture with an auto-associative neural network for dimensionality reduction and multilayer perceptron as a fuzzy classifier to predict endmember abundances. More recently, a convolution neural network architecture was proposed for unmixing[35]. Both these models rely on labeled data and only predict the endmember abundances, not the endmember spectra. Guo, Wang and Qi[9] proposed an unsupervised model that uses two autoencoder models, first to denoise the data and the second (with nonnegative and sparsity constraints) to learn the endmember's

spectra and predict abundances.

From equation 3.1, we can see that NMF is equivalent to the linear mixture model if we disregard the noise, so it is an appropriate model for hyperspectral unmixing. NMF also benefits from not needing labeled data and having an inherent nonnegativity constraint. The general Nonnegative Matrix Factorization model can be expressed as an optimization problem by employing an appropriate matrix norm. The Frobenius norm is widely used for this purpose and leads to

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\|M - WH\|_F^2. \tag{3.2}$$

Modified versions of the NMF model 3.2 have been proposed to improve the basic model's performance for hyperspectral unmixing by adding additional constraints or changing the structure of the basic model[6]. Constrained versions of NMF can be expressed as

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\|M - WH\|_F^2 + \lambda_1 g_1(W) + \lambda_2 g_2(H), \tag{3.3}$$

where g_1 and g_2 are regularization terms and λ_1 and λ_2 are their corresponding parameters. The additional constraints improve upon the general model by addressing its limitations. A minimum volume constraint $g_1(W) = vol(W)$ can improve endmember extraction[20]. Sparsity constraints, such as $g_2(H) = \|H\|_{1/2}$[23], are also common as they better represent real-world data where we do not expect to find endmembers everywhere in the field of view. Our model 3.5 also falls under the constrained NMF category.

Structured NMF models change the structure of the problem 3.2 rather than incorporating additional constraints. Weighted NMF[16] falls in this category and can be expressed as

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\|(M - WH)B\|_F^2, \tag{3.4}$$

where B is a diagonal matrix with weights calculated using k-means clustering analysis. This model tends to perform well with imbalanced datasets.

Lastly, there are multilayer NMF and deep NMF models that can learn hierarchical features in the data[6]. The survey[6] elaborates on many NMF models for Hyperspectral Unmixing.

As far as we know, no NMF models for Hyperspectral Unmixing focus on adaptively determining the rank of the decompositions. There are, however, NMF algorithms for other use cases that can automatically select the rank. Some of these methods rely on supplementary techniques like clustering to determine the rank beforehand. For example, Rank-Adaptive NMF algorithm[24] uses affinity propagation (AP) clustering to determine the number of components before utilizing NMF. Alternatively, AFRS-NMF[33] (Adaptive Factorization Rank Selection - Nonnegative Matrix Factorization) uses multiple sparsity constraints on the composition matrix H to determine the rank. The model's ability to select the rank was demonstrated on a tumor detection task.

3.2 Problem Formulation

Similar to Sum-of-norms clustering[14], we introduce a regularization term to equation 3.2 to minimize the Euclidean distance between columns of the matrix W. The resulting model is

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\|M - WH\|_F^2 + \lambda \sum_{(i,j)\in E} \|w_i - w_j\|_2, \tag{3.5}$$

where r is the specified rank, λ is a regularization parameter, $M \in \mathbb{R}_+^{m\times n}$, $W \in \mathbb{R}_+^{m\times r}$, $H \in \mathbb{R}_+^{r\times n}$, w_j and w_i are the i-th and j-th columns of W, respectively, and E is the set of all pair-wise (i, j), $i \neq j$ combinations of columns of W. The key challenge in formulating the solution for the minimization problem 3.5 is that the $\|w_i - w_j\|_2$ term is not differentiable when $w_i = w_j$. Note that the second term in equation 3.5 is equivalent to the l_1 norm of the difference between the columns of W.

Consider the ground truth factorization is $M = W_{true}H_{true}$ where $W \in \mathbb{R}_+^{m\times r_{true}}$, $H \in \mathbb{R}_+^{r_{true}\times n}$ and r_{true} is the ideal rank. Solving equation 3.5 with an appropriate λ and $r \geq r_{true}$ would lead to duplicate columns of W and effectively reduce the rank r to r_{true}.

3.3 Solution Approach

We use a Block Coordinate Descent algorithm called Hierarchical Alternating Least Squares to define subproblems for a column of W and a row of H. The subproblems are then solved independently in an iterative manner. Equation 3.2 can be restructured as

$$\|M - \sum_{i=1}^{r} w_i h^i\|_F^2 = \|M - \sum_{i=1, i\neq j}^{r} w_i h^i - w_j h^j\|_F^2, \tag{3.6}$$

where w_j is the j-th column of W and h^j is the j-th row of H. We can further simplify it to

$$\|M_j - w_j h^j\|_F^2, \tag{3.7}$$

where $M_j = M - \sum_{i=1, i\neq j}^{r} w_i h^i$.

3.3.1 Subproblem on h^j

Equation 3.7 can be expressed as a quadratic equation on h_j[18]:

$$\|M_j\|_F^2 - 2\langle M_j h^j, w_j\rangle + \|h^j\|_2^2 \|w_j\|_2^2 + c. \tag{3.8}$$

With this formulation and taking into consideration that the regularization term in equation 3.5 is not dependent on h^j, the subproblem for h^j is a minimization problem of the form

$$\min_h f(h^j) = \min_h \frac{1}{2}\|w_j\|_2^2 \|h^j\|_2^2 - \langle w_j^T M_j, h^j\rangle. \tag{3.9}$$

The nonnegativity constraint on h^j is ignored for the moment and is addressed later. To solve equation 3.9, we use gradient descent with an update step

$$h^j = h^j - \frac{1}{L} \nabla_h f(h^j), \tag{3.10}$$

with gradient $\nabla_h f(h^j) = \|w_j\|_2^2 h^j - w_j^T M_j$ which has a Lipschitz constant $L = \|w_j\|_2^2$. This leads to a closed-form solution

$$h^j = \frac{w_j^T M_j}{\|w_j\|_2^2}. \tag{3.11}$$

Finally, the nonnegativity constraint is applied by taking the nonnegative projection $[.]_+ = \max\{0, .\}$ of the numerator in 3.11. The complete solution the row h^j of matrix H is

$$h^j = \frac{\left[w_j^T M_j\right]_+}{\|w_j\|_2^2}. \tag{3.12}$$

Note that this is the gradient descent step, Newton's method update, and the exact solution to the h^j subproblem.

3.3.2 Subproblem on w_j

Similar to h_j, the subproblem on w_j can be expressed as a quadratic minimization problem

$$\min_w \frac{1}{2} \|h^j\|_2^2 \|w\|_2^2 - \langle M_j h^{j^T}, w \rangle + \lambda \sum_{i \neq j} \|w - w_i\|_2 + c, \tag{3.13}$$

where $w = w_j$ for simplicity. The key challenge in solving 3.13 stems from the regularization term $\|w - w_i\|_2$, which is not differentiable. We explore three algorithms for solving the w_j subproblem.

Subgradient method

The sub-differential of the l_2 norm is

$$\partial\|x\|_2 = \begin{cases} \frac{x}{\|x\|_2} & x \neq 0 \\ \tau, \|\tau\|_2 \leq 1 & x = 0 \end{cases}, \tag{3.14}$$

where τ is any vector that meets the $\|\tau\|_2 \leq 1$ condition. We use the sub-differential to approximate the norm in 3.13 and solve the subproblem using gradient descent. This leads to the following heuristic update step

$$w^{k+1} = \left[w^k - \alpha \left(\|h^j\|_2^2 w^k - M_j h^{j^T} + \lambda \sum_{i \neq j} \partial\|w^k - w_i\|_2 \right) \right]_+ \tag{3.15}$$

To find the step size α, we use line search as suggested by [31].

Nesterov Smoothing Approximation

Given any vector $a \in \mathbb{R}^m$, and paramter $\mu > 0$, the Nesterov smoothing approximation[21] of the Euclidean norm $\|x - a\|_2$ is

$$\|x - a\|_2 \approx \frac{1}{2\mu}\|x - a\|_2^2 - \frac{\mu}{2}\left[d(\frac{x-a}{\mu}; \mathbb{B}) \right]^2, \tag{3.16}$$

where $d(.;.)$ is the Euclidean distance and $\mathbb{B}$ is the closed unit ball of $\mathbb{R}^m$. Furthermore,

$$\partial\|x - a\|_2 = \text{Proj}_{\mathbb{B}}(\frac{x-a}{\mu}), \tag{3.17}$$

where $\text{Proj}_{\mathbb{B}}(.)$ denotes the projection on unit l_2 norm ball. The projection of $z \in \mathbb{R}^m$ on unit l_2 norm ball[17] is

$$\text{Proj}_{\mathbb{B}}(z) = \begin{cases} z & \|z\|_2 \leq 1 \\ \frac{z}{\|z\|_2} & \|z\|_2 > 1 \end{cases}. \tag{3.18}$$

Using this approximation of the Euclidean norm, the equation 3.13 can be expressed as

$$\min_w \frac{1}{2}\|h^j\|_2^2\|w\|_2^2 - \langle M_j h^j, w\rangle + \lambda \sum_{i\neq j} \frac{1}{2\mu}\|w - w_i\|_2^2 - \frac{\mu}{2}\left[d(\frac{w-w_i}{\mu}; \mathbb{B})\right]^2 + c. \tag{3.19}$$

This leads to the gradient descent step

$$w^{k+1} = \left[w^k - \alpha\left(\|h^j\|_2^2 w^k - M_j {h^j}^T + \lambda \sum_{i\neq j} \text{Proj}_{\mathbb{B}}(\frac{w-w_i}{\mu})\right)\right]_+. \tag{3.20}$$

Note that μ is the same for all i's. Similar to the Subgradient approach, we find the step size α using line search as suggested by [31].

ADMM

To use ADMM, we first introduce local variables w_f, w_0, w_i and a central variable z to represent w_j, and express the equation 3.13 in a separable form

$$\min_{w_f, w_0, \{w_i\}, z} f(w_f) + g_0(w_0) + \sum_{i, i\neq j} g_i(w_i), \tag{3.21}$$

where

$$f(w) = \frac{1}{2}\|h^j\|_2^2\|w\|_2^2 - \langle M_j {h^j}^T, w\rangle, \tag{3.22}$$

$$g_0(w) = i_+(w), \tag{3.23}$$

$$g_i(w) = \lambda \|w - c_i\|_2 \ , \ c_i = w_i \text{ such that } i \neq j, \tag{3.24}$$

and $w_f = z$, $w_0 = z$, $w_i = z \ \forall i$. The indicator function

$$i_+(x) = \begin{cases} +\infty & x < 0 \\ 0 & x \geq 0 \end{cases}, \tag{3.25}$$

which maps from $\mathbb{R}^m$ to $\mathbb{R}$, is included in the ADMM formulation to impose the nonnegativity constraint. The solution to the local problems (3.22, 3.23, 3.24) can now be used to solve the global minimization problem. The augmented Lagrangian of equation 3.19 is

$$\begin{aligned} \underset{w_f,w_0,w_i,z}{\operatorname{argmin}} \ \underset{y_f,y_0,y_i}{\operatorname{argmax}} L_\rho(w_f, w_0, w_i, z, y) &= f(w_f) + g_0(w_0) + \sum_{i,i\neq j} g_i(w_i) + \langle y_f, w_f - z\rangle \\ &+ \langle y_0, w_0 - z\rangle + \sum_{i,i\neq j} \langle y_i, w_i - z\rangle + \frac{\rho}{2}\|w_f - z\|_2^2 + \frac{\rho}{2}\|w_0 - z\|_2^2 + \sum_{i,i\neq j} \frac{\rho}{2}\|w_i - z\|_2^2 \end{aligned} \tag{3.26}$$

where y_f, y_0, y_i are Lagrangian variables and $\rho > 0$ is a penalty parameter. In a similar manner to section 2.3, the augment Lagrangian 3.26 leads to the following iterations:

$$w_f^{k+1} = \underset{w_f}{\operatorname{argmin}} L_\rho(w_f) = \frac{M_j(h^j)^T - y_f^k + \rho z^k}{\rho + \|h^j\|_2^2}, \tag{3.27}$$

$$w_0^{k+1} = \underset{w_0}{\operatorname{argmin}} L_\rho(w_0) = \left[z^k - \frac{y_0^k}{\rho} \right]_+, \tag{3.28}$$

$$w_i^{k+1} = \underset{w_i}{\operatorname{argmin}} L_\rho(w_i) = \begin{cases} \zeta - \lambda(\frac{\frac{\zeta}{\lambda} - c_i}{\|\frac{\zeta}{\lambda} - c_i\|_2}) & \|\frac{\zeta}{\lambda} - c_i\|_2 > 1 \\ \zeta - \lambda(\frac{\zeta}{\lambda} - c_i) & \|\frac{\zeta}{\lambda} - c_i\|_2 \leq 1 \end{cases} \text{ where } \zeta = z^k - \frac{y_i^k}{\rho}, \tag{3.29}$$

$$z^{k+1} = \frac{\rho(w_f^{k+1} + w_0^{k+1}) + \rho \sum_{i,i\neq j} w_i^{k+1} + y_f^k + y_0^k + \sum_{i,i\neq j} y_i^k}{\rho(2 + |E|)}, \tag{3.30}$$

$$y_f^{k+1} = y_f^k + \rho(w_f^{k+1} - z^{k+1}), \tag{3.31}$$
$$y_0^{k+1} = y_0^k + \rho(w_0^{k+1} - z^{k+1}), \tag{3.32}$$
$$y_i^{k+1} = y_i^k + \rho(w_i^{k+1} - z^{k+1}). \tag{3.33}$$

The complete derivation of the ADMM iterations is shown in Appendix A.

3.3.3 Accelerated version

We consider two acceleration procedures to improve the convergence speed of our ADMM solution for the subproblem of w_j: Anderson Acceleration and Heuristic Extrapolation with Restarts.

Anderson Acceleration

Anderson Acceleration[27] is a technique for improving the convergence of iterative algorithms. It works by incorporating information from previous iterations of the algorithm into the current iteration to improve the rate of convergence and reduce the number of iterations required to reach the solution. This can make the algorithm more efficient. For a fixed point (FPI) method of the form

$$x^{k+1} = f(x^k), \tag{3.34}$$

where f is an iterative function and $x \in \mathbb{R}^n$. Anderson acceleration improves the convergence by using the update formula

$$x^{k+1} = f(x^k) + \sum_{i=0}^{m_k-1} \beta_i^k \left(f(x^{k-i}) - f(x^{k-i-1}) \right), \tag{3.35}$$

where m is the window size, k is the iteration number, and $m_k = \min\{m, k\}$. The β_i^k coefficients are computed by solving the minimization problem

$$\beta_i^k = \underset{\beta_i}{\operatorname{argmin}} \left\| r(x^k) + \sum_{i=0}^{m_k-1} \beta_i (r(x^{k-i}) - r(x^{k-i-1})) \right\|_2^2, \tag{3.36}$$

where $r(x^k) = x^k - f(x^k)$ is the residual of equation 3.32 for the k-th iteration. In our implementation, the unconstrained linear least-squares problem 3.34 is solved using QR decomposition. We explore two Anderson accelerated versions of our ADMM solution for the suproblem of w_j: applying it to equation 3.30 for z^{k+1}, and applying it to equations 3.27, 3.28, 3.29 and 3.30 for w_f^{k+1}, w_0^{k+1}, w_i^{k+1} and z^{k+1} respectively.

Heuristic Extrapolation with Restarts

The Heuristic Extrapolation with Restarts (HER) paper[19] proposes a strategy for accelerating the convergence of Block Coordinate Descent methods for Nonnegative Tensor Factorization methods. It involves using a heuristic to make informed guesses about the next steps in an algorithm and then restarting the algorithm from an intermediate step if the convergence does not improve. We apply their procedure to our algorithm by modifying the column-wise BCD to calculate matrices W separately and H.

Algorithm 1: NMF-SON ADMM Approach with HER

Data: $M, W_0, H_0, \lambda, \beta_0 \in (0,1), \bar{\beta}_0 = 1, \eta, \bar{\gamma}, \gamma$
Result: W, H
$k = 1$;
$\hat{W}, \hat{H} = W_0, H_0$;
while *criteria not met* **do**
  $H_{k+1} = \text{update\_H\_func}(\hat{W}_k, H_k)$;
  $\hat{H}_{k+1} = [H_{k+1} + \beta_k(H_{k+1} - H_k)]_+$;
  $W_{k+1} = \text{update\_W\_func}(\hat{H}_{k+1}, W_k)$;
  $\hat{W}_{k+1} = [W_{k+1} + \beta_k(W_{k+1} - W_k)]_+$;
  $e^{k+1} = F(\hat{W}, H_{k+1})$;
  if $e^{k+1} > e^k$ **then**
    $\bar{\beta}_{k+1} = \beta_k$;
    $\beta_{k+1} = \beta_k/\eta$;
    $\hat{H}_{k+1}, \hat{W}_{k+1} = H_{k+1}, W_{k+1}$;
  else
    $\bar{\beta}_{k+1} = \min\{1, \bar{\beta}_k\bar{\gamma}\}$;
    $\beta_{k+1} = \min\{\bar{\beta}_k, \beta_k\gamma\}$;
    $H_{k+1}, W_{k+1} = \hat{H}_k, \hat{W}_k$;
  end
end

Chapter 4

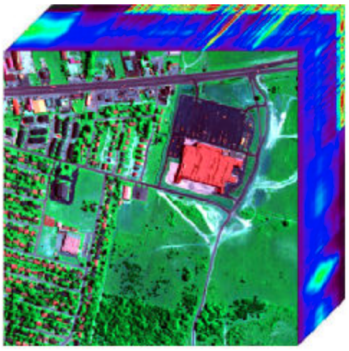
Experiments

4.1 Dataset

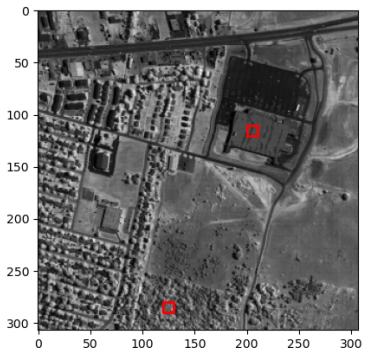
The Urban hyperspectral dataset[25][1] shows a Walmart store in Copperas Cove, Texas. The raw version of the dataset is a $210 \times 307 \times 307$ cube, where each of the 210 planes corresponds to wavelengths ranging from 400nm to 2500nm. A single pixel in an image represents a $2 \times 2m^2$ area. For our work, we use a preprocessed version of the dataset, where some of the 210 channels are removed due to water vapor and atmospheric effects. The preprocessed dataset is a $168 \times 307 \times 307$ cube. For this dataset, we expect to detect six materials (asphalt, grass, tree, roof, metal, and dirt).

Applying our NMF-SON method to the entire Urban dataset is a computationally expensive task, so we create a smaller dataset by selecting subimages from the full images and merging them. The subimages, each of size 10×10, are chosen purposefully to only contain only two endmembers (trees and roof) and then concatenated to form a $168 \times 20 \times 10$ hyperspectral cube. The areas for the subimages have been outlined in figure 4.1b.

Another dataset we use for our experiments is the Jasper Ridge hyperspectral cube[25]. The original dataset has the shape $224 \times 512 \times 614$. The wavelength range is 380 nm to 2500 nm. The large image size makes it challenging to work with this dataset. We utilize a preprocessed subimage of 100×100 pixels. Similar to the Urban dataset, several images were removed from the dataset to reduce noise. The resulting dataset has the shape $198 \times 100 \times 100$. We expect to detect four endmembers: road, soil, water, and tree.



(a) Complete Urban dataset.



(b) Urban image outlining areas used to create the small Urban dataset.

Figure 4.1: Two versions of the Urban dataset.

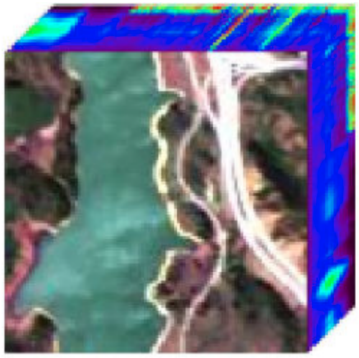


Figure 4.2: Jasper dataset.

4.2 Algorithm Setup

For all experiments except Section 4.3, we use the ADMM approach described in section 3.3.2. The penalty parameter $\rho = 1$ for ADMM was fixed for all tests. The experiments were run until the maximum number of iterations was reached or the stopping criteria 4.1 was met. The initial matrices W_0 and H_0 for the tests were randomly initialized to have values between 0 and 1.

$$\frac{|F(W^k, H^k) - F(W^{k-1}, H^{k-1})|}{F(W^{k-1}, H^{k-1})} \leq 10^{-5}, \tag{4.1}$$

where

$$F(W, H) = \underbrace{\frac{1}{2}||M - WH||_F^2}_{f(W,H)} + \underbrace{\lambda \sum_{(i,j)\in E} ||W_i - W_j||_2}_{g(W)}. \tag{4.2}$$

The λ hyperparameter is scaled before each iteration k using the formula

$$\lambda_k = \lambda \frac{f(W_{k-1}, H_{k-1})}{g(W_{k-1})}. \tag{4.3}$$

Due to this feature, $F(W, H)$ is a homotopy. Scaling λ helps avoid circumstances when the regularization term is too large or too small.

4.3 ADMM vs Subgradient vs Nesterov Smoothing

We compare the three approaches to solve the w_j subproblem: the Subgradient approximation 3.3.2, Nesterov Smoothing 3.3.2 and ADMM 3.3.2. All three algorithms ran for a maximum of 3000 iterations with $\lambda = 2$ and rank $r = 6$ on the small Urban dataset. The tests were carried out on a laptop with an Intel Core i7-8550U CPU and 16GB RAM.

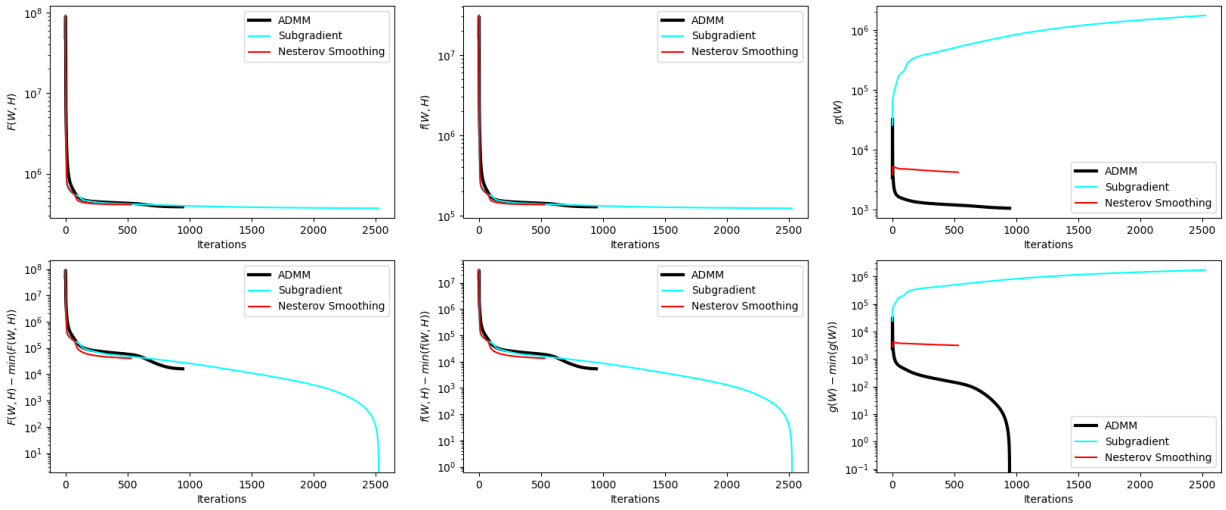


Figure 4.3: Convergence comparison of multiple approaches used to solve the w_j subproblem. The Subgradient approximation fails to regularize, and the most effective algorithm is ADMM.

Runtimes

Method	**Time Taken**
ADMM	7min 38s
Subgradient	6min 9s
Nesterov Smoothing	2min 6s

Table 4.1: Runtimes of approaches to solve the w_j subproblem. ADMM took the longest time, followed by Subgradient Approximation and Nesterov Smoothing.

From figure 4.3, it is evident that the subgradient method fails to reduce the regularization term $g(W)$. This means the method is inappropriate for our model as it does not promote duplicate vectors. The poor performance was expected as the algorithm is a very naive approach. The Nesterov Smoothing method performs better than the subgradient approach in terms of regularization but not as well as the ADMM approach. Due to this, we choose to use the ADMM approach for our numerical experiments.

A drawback of the ADMM algorithm is that it is the most computationally expensive approach, as shown in table 4.1. We could reduce the computation time by first running the Nesterov Smoothing approach and using the results to initialize the ADMM algorithm.

4.4 Acceleration Methods

We tested two acceleration methods, Anderson Acceleration and Heuristic Extrapolation with Restarts (HER), to improve our algorithm's convergence. Both acceleration methods are initialized with $\lambda = 2$ and rank $r = 6$ on the small Urban dataset. All tests ran for a maximum of 1000 iterations. The tests were carried out on a laptop with an Intel Core i7-8550U CPU and 16GB RAM.

Two versions of Anderson Acceleration were considered: one where only the z variable of the ADMM approach is accelerated, and another where w_f, w_0, w_i, z were accelerated. Both versions used a window size of two, so only the values of the previous two iterations were used for acceleration. Figure 4.4 shows that both versions of Andersen Acceleration fail to improve convergence and exhibit odd oscillating behavior. This is unexpected as Anderson Acceleration has shown to improve the convergence of multiple ADMM algorithms[28]. Another drawback is that the accelerated algorithms took longer than the base algorithm, as shown in 4.2 with the same setting.

On the other hand, HER was able to improve the convergence of our algorithm when compared to the baseline implementation. Furthermore, the HER version has a significantly shorter runtime even though more matrix operations are involved in the implementation.

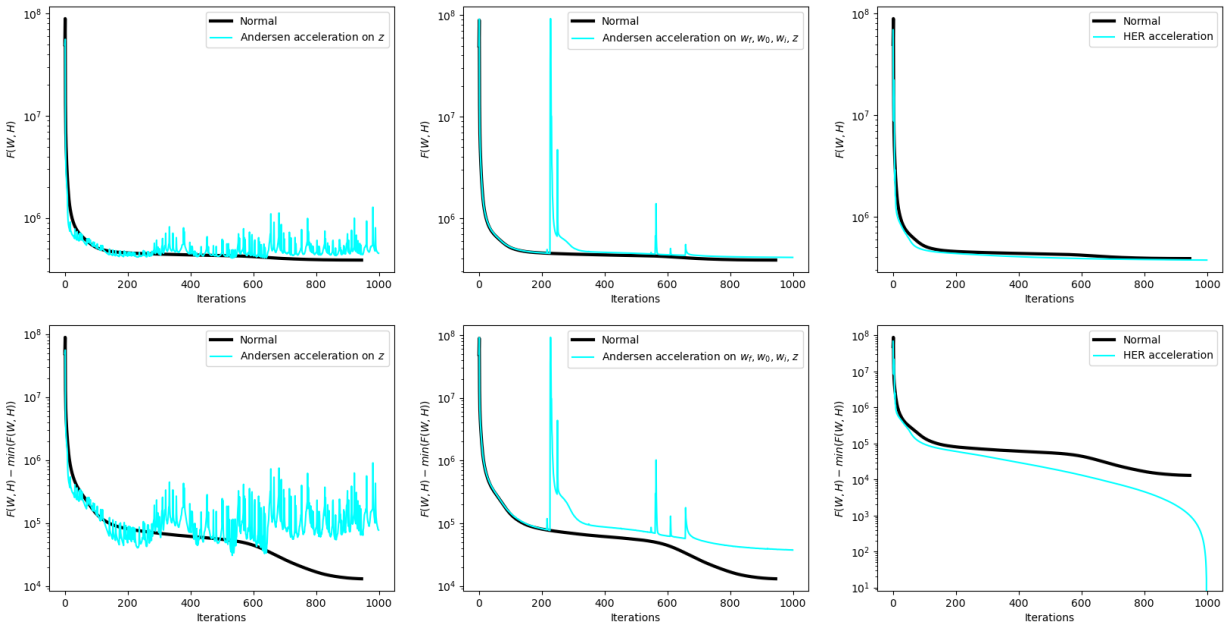


Figure 4.4: Convergence comparison of acceleration methods. Both versions of Anderson Acceleration failed to improve convergence speed and exhibited unexpected behavior. HER was able to improve convergence speed.

Runtimes

Method	**Time Taken**
Basline	7min 15s
Andersen acceleration on z	11min 28s
Andersen acceleration on w_f, w_0, w_i, z	40min 32s
HER	2min 19s

Table 4.2: Runtimes of acceleration methods. Both versions of Anderson Acceleration took longer to complete. Meanwhile, HER reduced the runtime.

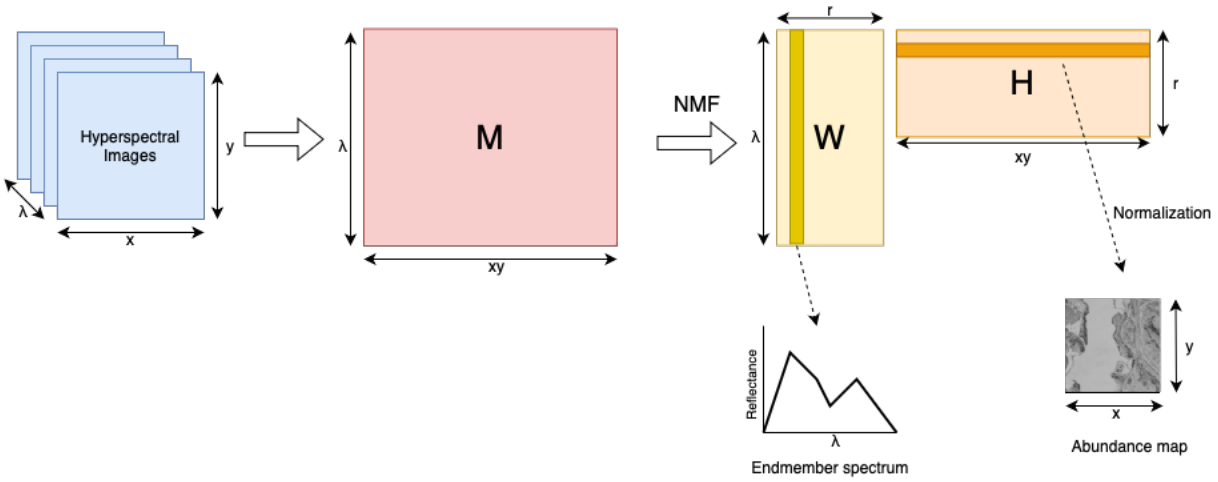


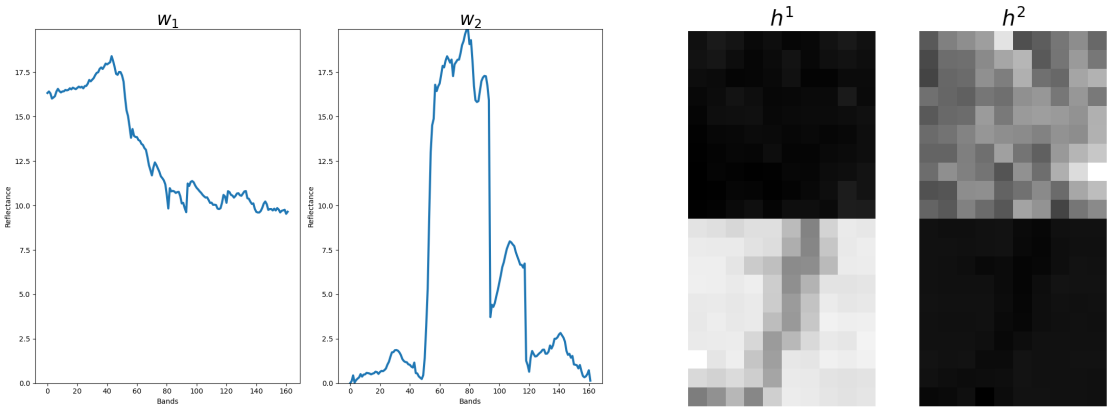
Figure 4.5: Process of creating abundance maps and endmember spectra from hyperspectral images.

4.5 NMF-SON vs NMF

This section compares our model NMF-SON and basic NMF for hyperspectral unmixing on the small Urban and Jasper datasets. Figure 4.5 illustrates the process of creating the endmember spectra and abundance maps presented in this section. The Hyperspectral dataset is resized into a matrix M, and NMF decomposes M into smaller representations W and H. Each column of W is the endmember spectrum of a material, and the corresponding row of H contains the abundance values for that endmember in the field of view. The rows of H are normalized and reshaped into images called abundance maps. The normalization leads to clearer images. In these images, lighter regions indicate a higher abundance of the corresponding endmember.

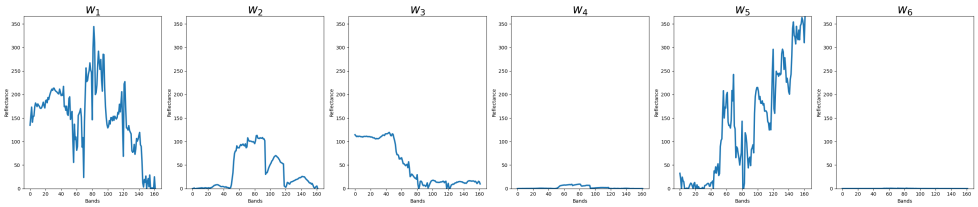
For the small Urban dataset, we ran the basic NMF with rank $r = 2$ to show the ideal decomposition, and with rank $r = 6$ to compare it with NMF-SON. NMF-SON was initialized with rank $r = 6$ with $\lambda = 3$ and $\lambda = 50$. All tests ran for a maximum of 3000 iterations.

The endmember spectra and abundance maps in figures 4.6a and 4.6b, respectively, show the decomposition we expect. The column w_1 is the endmember spectra for the roof of the building, and it corresponds to the abundance map on the left in figure 4.6b,

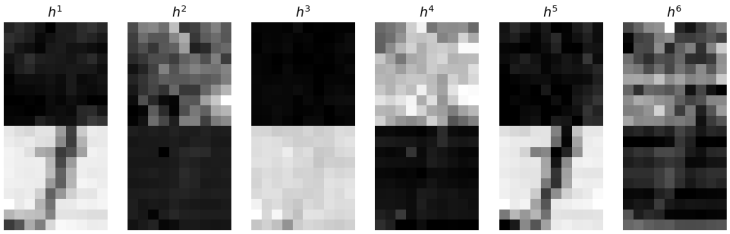


(a) Endmember Spectra (b) Abundance maps

Figure 4.6: Results of basic NMF, with rank $r = 2$, on the small Urban dataset. Components 1 and 2 represent the roof and trees, respectively.

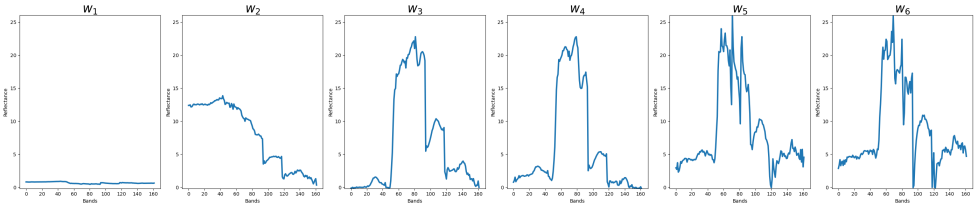


(a) Endmember Spectra

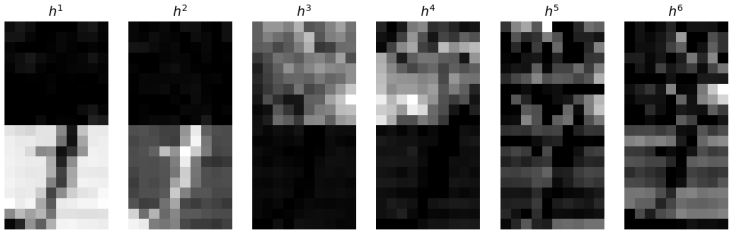


(b) Abundance maps

Figure 4.7: Results of basic NMF, with rank $r = 6$, on the small Urban dataset. Components 2 and 3 correspond to trees and the roof, respectively. All other components are not useful results.



(a) Endmember Spectra



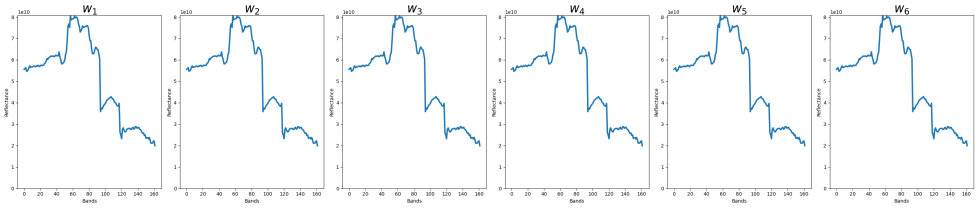
(b) Abundance maps

Figure 4.8: Results of NMF-SON, with rank $r = 6$ and $\lambda = 3$, on the small Urban dataset. NMF-SON returns pairs of duplicate components: 3 and 4 represent trees, and 5 and 6 represent noise. Components 1 and 2 represent parts of the roof.

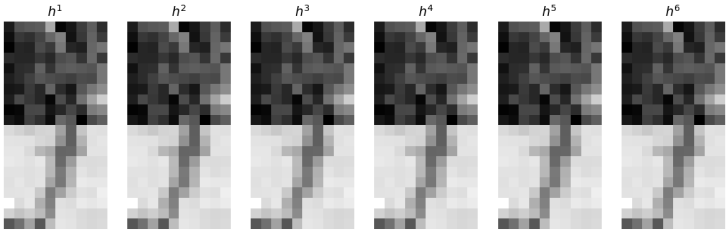
where the bottom half has higher abundance values. Similarly, w_2 represents trees, and the corresponding abundance map has higher values in the top half. The basic NMF with a large rank $r = 6$ fails to generate meaningful results, as shown in figures 4.7a and 4.7b. Only components 2 and 3, which correspond to trees and roof, respectively, are helpful in this decomposition.

On the other hand, our NMF-SON model reduced the rank of the resulting matrices and generated more meaningful representations than the basic NMF with rank $r = 6$. Figures 4.8a and 4.8b shows that components 3 and 4, and 5 and 6 are pairs of duplicate endmember spectra and abundance maps. w_3 and w_4 spectras match the signature of trees, and w_5 and w_6 probably correspond to noise. Components 1 and 2 both highlight parts of the roof. Considering the duplicate components, our method reduced the rank from $r = 6$ to $r = 4$ and provided better results than basic NMF with rank $r = 6$. Further λ tuning could reduce the rank and improve the decomposition to represent the roof endmember better.

To validate our model, we run the same experiment with a larger $\lambda = 50$. Figures 4.9a and 4.9b show that all the endmember spectra and the abundance maps are identical. We



(a) Endmember Spectra



(b) Abundance maps

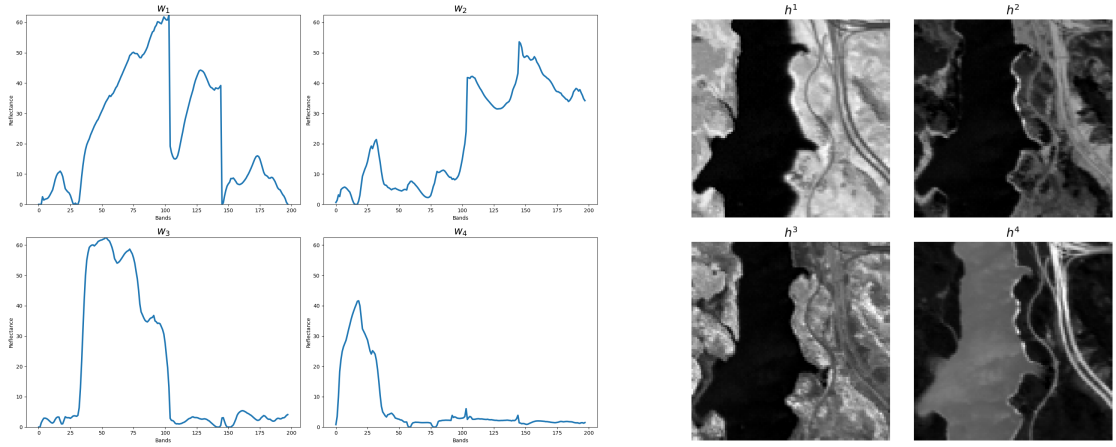
Figure 4.9: Results of NMF-SON, with rank $r = 6$ and $\lambda = 50$, on the small Urban dataset. All the components are identical.

expect the model to generate this, as a large enough λ should reduce the solution to rank 1.

Similar numerical experiments were performed with the Jasper dataset. We ran the basic NMF with rank $r = 4$ (the expected number of endmembers) and rank $r = 8$ to compare it with NMF-SON. NMF-SON was initialized with rank $r = 8$ and $\lambda = 1$. All tests ran for a maximum of 3000 iterations.

The decomposition results of basic NMF with rank $r = 4$, in figure 4.10, are challenging to interpret. Only three of the model's expected endmembers (road, soil, water, and trees) are shown, and one component is an unexpected endmember. The three correctly shown endmembers are soil, trees, and road, represented by components 1, 3, and 4, respectively. Component 2 returns an unexpected endmember with a high abundance near the water. This could represent sand on the shoreline. None of the components can represent the water endmember clearly. Water has a relatively higher abundance in component 4 but is less significant than the road. This indicates that the endmember spectra for water and road are too similar to separate using basic NMF.

The results for basic NMF with rank $r = 8$, shown in 4.11, are also difficult to understand. Components 1, 3, and 4 correspond to trees, roads, and soil. The endmember



(a) Endmember Spectra

(b) Abundance maps

Figure 4.10: Results of basic NMF, with rank $r = 4$, on the Jasper dataset. Components 1, 3, and 4 correspond to soil, trees and road. Component 2 is an unexpected result and most likely represents sand.

signatures for these materials are similar to their signatures in the $r = 4$ decomposition, even though the scale is different. Components 2 and 5 highlight the same areas (near the water), but their signatures are not similar; this could indicate two unexpected endmembers. Components 6, 7, and 8 most likely represent noise due to light reflecting off multiple materials since they highlight the same area as other endmembers but do not have similar endmember signatures, and the reflectance is low. Even with a larger rank, basic NMF cannot separate water from other endmembers and is highlighted in the abundance map for the road. This supports our previous claim that the endmember of water and road are too similar to separate.

The results for NMF-SON are shown in figure 4.12. Our model is able to reduce the rank from $r = 8$ to $r = 7$, but the representations are unclear. Based on the abundance maps for this decomposition and the ones in figure 4.11, components 1, 3, and 5 represent trees, roads, and soil, respectively. Components 6 and 7 are duplicates and highlight an unexpected endmember close to water. Component 8 also has high abundances near the shoreline but a different pattern than components 6 and 7; this is also an unexpected endmember. Components 2 and 4 are probably noise.

All models used for Jasper struggle to identify the endmembers correctly. The end-

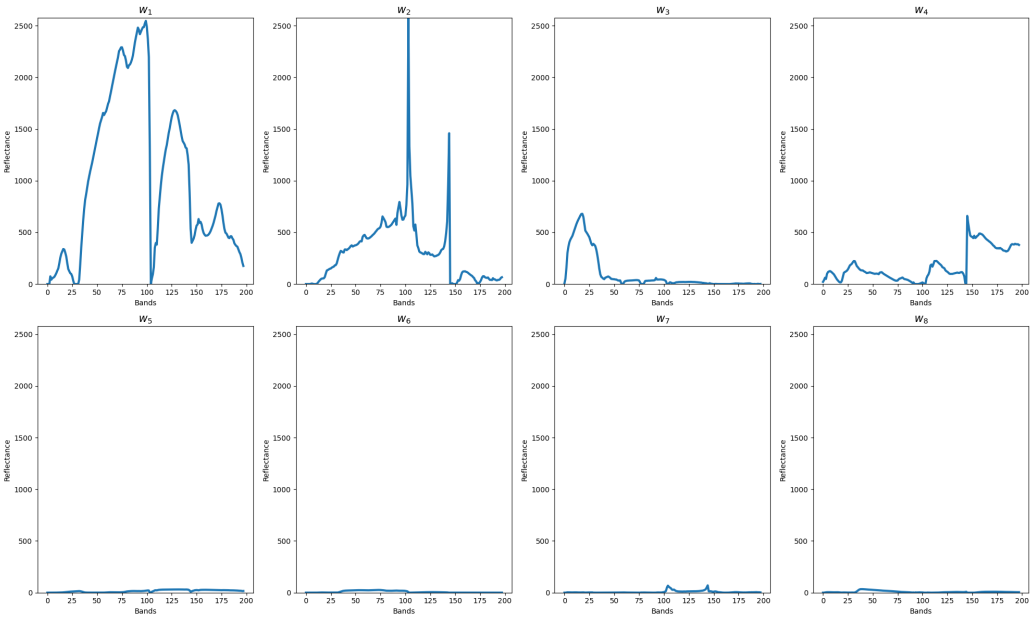
member for water is not returned separately due to the similarity between the spectra of water and road. The models also return unexpected endmembers near the shoreline that have different endmember signatures; this means that there are multiple materials near the water that are not reported in the data source[25].

4.6 Challenges

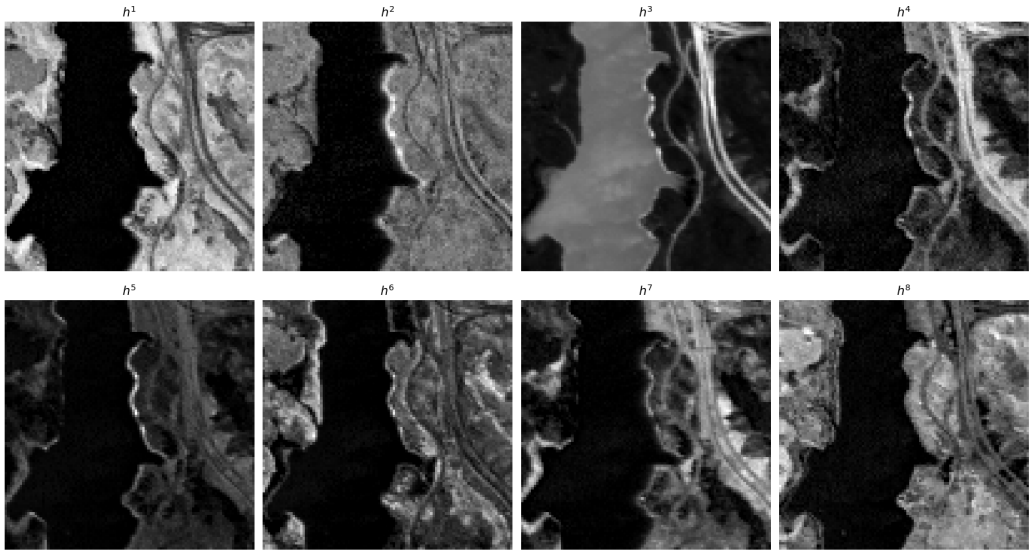
The minimization problem for NMF-SON is non-convex, which means there are multiple local minima that the algorithm could converge to. The initial matrices W and H affect the algorithm's performance and the local minimum reached upon completion.

The hyperparameter λ can not be directly learned from the data and must be set before running the algorithm. Finding the optimal values for the λ is challenging because the search space for λ is all nonnegative real numbers, making the optimization process computationally expensive and time-consuming. Furthermore, the optimal values can vary depending on the specific data and selected rank r. There are heuristic approaches to improve hyperparameter tuning, but no method exists to find the optimal value.

Lastly, the NMF model's patterns may not always align with our expectations, making it difficult to interpret and understand the results. We face this issue for the Jasper dataset, for which all the decompositions return unexpected endmembers not mentioned in the data source[25]. Even by simply examining the figure 4.2, we can see a different material near the shoreline which is not explicitly listed in the data source. Better information about the endmembers in a field of view could improve our understanding and help in accurately determining the usefulness of the NMF decompositions. Another reason for the unexpected endmembers could be that NMF models fit the Linear Mixing Model, which is a simple approach. For real-world data, light may be reflected off multiple materials (endmembers) before being recorded by the sensors.

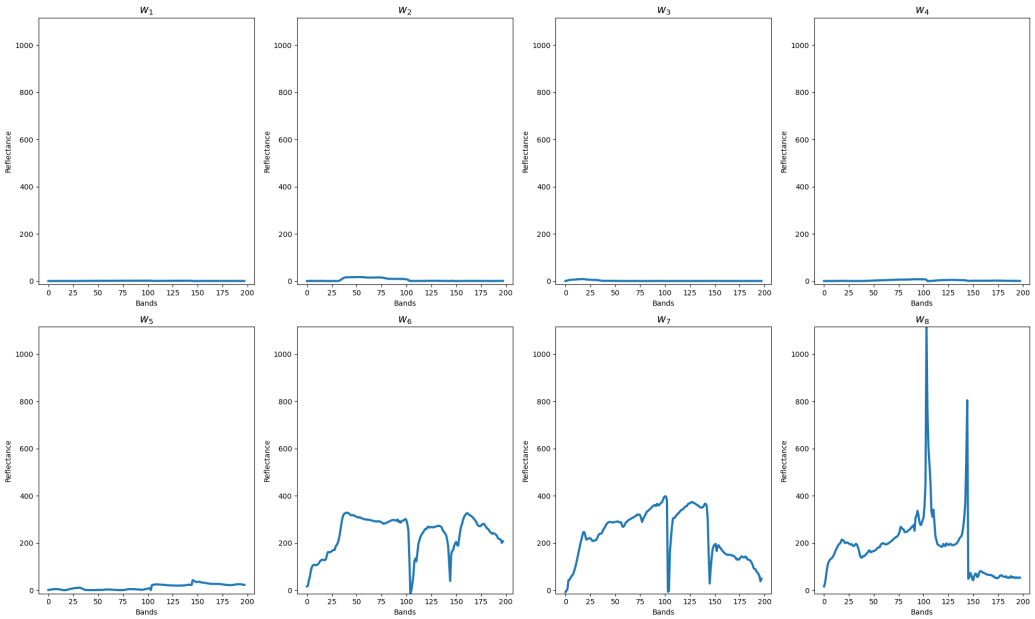


(a) Endmember Spectra

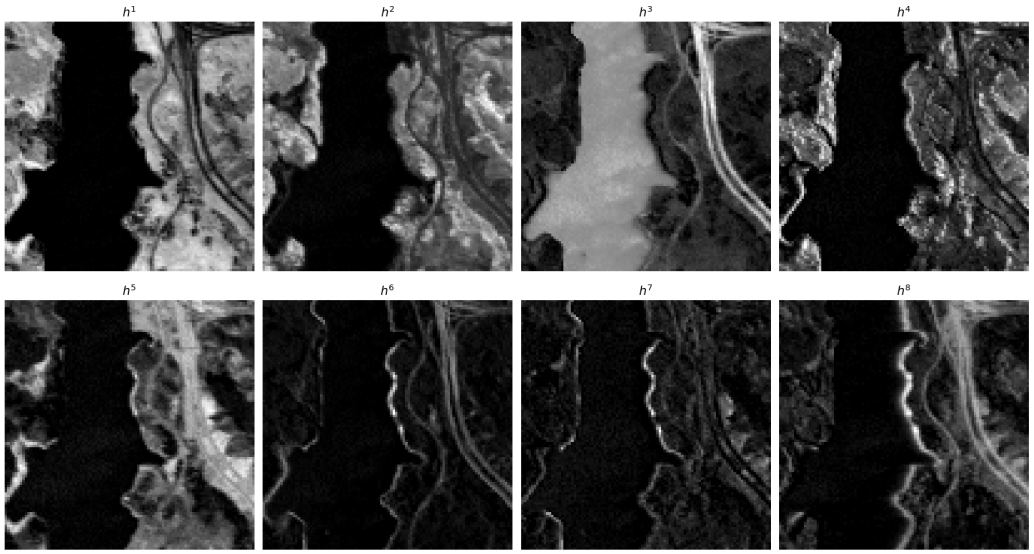


(b) Abundance maps

Figure 4.11: Results of basic NMF, with rank $r = 8$, on the Jasper dataset. Components 1, 3, and 4 represent trees, road, and soil. Other components are unclear.



(a) Endmember Spectra



(b) Abundance maps

Figure 4.12: Results of NMF-SON, with rank $r = 8$ and $\lambda = 1$, on the Jasper dataset. The rank is reduced to $r = 7$ since components 6 and 7 are the same. Components 1, 3, and 5 represent trees, roads, and soil. Other components are unclear.

Chapter 5

Conclusion

5.1 Discussion

In this report, we introduced a new NMF model that can automatically determine the rank of the smaller representation matrices. This was achieved by using a regularization term inspired by Sum-of-norms clustering. The model encourages a low rank representation of data, as long as the specified rank is larger than the true rank, $r \geq r_{true}$, and the appropriate hyperparameter λ is used. The model's ability to automatically detect the appropriate rank and generate meaning decompositions was assessed on the hyperspectral unmixing task. The numerical results comparing our model with the basic NMF model show that for a rank r such that $r \geq r_{true}$, our model can reduce the number of components and generate better representations than basic NMF.

Furthermore, we compared three alternatives for solving our model: Subgradient approximation of the norm, Nesterov Smoothing of the norm, and Alternating Direction Method of Multipliers (ADMM). Our numerical results demonstrate that ADMM is the most appropriate choice as it effectively minimizes the regularization function. The performance comes at the cost of computational resources as it is also the most elaborate algorithm. We explore two acceleration approaches to improve the convergence of our algorithm: Anderson Acceleration and Heuristic Exploration with Restarts (HER). The Anderson Acceleration approach fails to improve the convergence compared to the unaccelerated version of our algorithm and exhibits an odd convergence pattern. On the other

hand, HER can improve convergence and reduce our algorithm's runtime.

There are, however, some challenges that affect the performance of our algorithm. These include noise in the data, tuning λ, and initializing the model. Initializing NMF models is a well-known challenge, and there are ways to make better guesses[5], but this is an open problem. There is no way to initialize the matrices to guarantee convergence to the global minimum for non-convex problems. Similarly, tuning λ or hyperparameters, in general, is a challenging problem with no solution.

5.2 Further work

There are multiple directions for improving our model. A parallel implementation of our algorithm could be developed. Our ADMM approach is well suited for this as all the local variables and functions for w_j can be calculated on separate machines and used to calculate the resulting value. This would reduce the runtime of our algorithm and make it more suitable for larger datasets.

Another possible direction is to explore Multigrid implementations of our model. Multigrid NMF algorithms speed up such models' convergence by reducing the data's dimension to a coarser grid and solving the coarse grid problem. Then, the coarse grid solution is translated to the original dimension using interpolation[8] and used as initialization for the fine (original) algorithm. Initializing the original algorithm using such an approach can reduce the number of iterations required.

A common restriction imposed on NMF models for Hyperspectral Unmixing is that the columns of the abundance matrix sum to one[6]. This would include adding $\mathbf{1}_r^T H = \mathbf{1}_n^T$ where $\mathbf{1}_r$ and $\mathbf{1}_n$ are all-one vectors with of size r and n respectively. This restriction could result in more understandable abundance maps, and that may eliminate the need to normalize the rows of H after the NMF algorithm is complete. Further restrictions to the model, such as sparsity constraints, may also improve endmember extraction.

So far, we have tested our model for hyperspectral unmixing only. However, it can also be applied to other use cases since the model has no hyperspectral unmixing specific restriction. Numerical experiments for tasks including document clustering or facial expression recognition could be conducted to assess the versatility of our model.

References

[1] José Bioucas-Dias et al. "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches". In: *IEEE journal of selected topics in applied earth observations and remote sensing* 5.2 (2012), pp. 354–379.

[2] Joseph Boardman, Fred Kruse, and Robert Green. "Mapping target signatures via partial unmixing of AVIRIS data". In: *Summaries Proceedings of the Fifth JPL Airborne Earth Science Workshop* (1995).

[3] Stephen Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[4] Chris Ding, Xiaofeng He, and Horst Simon. "On the equivalence of nonnegative matrix factorization and spectral clustering". In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM. 2005, pp. 606–610.

[5] Flavia Esposito. "A review on initialization methods for nonnegative matrix factorization: towards omics data experiments". In: *Mathematics* 9.9 (2021), p. 1006.

[6] Xin-Ru Feng et al. "Hyperspectral unmixing based on nonnegative matrix factorization: A comprehensive review". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2022).

[7] Nicolas Gillis. *Nonnegative matrix factorization*. SIAM, 2020.

[8] Nicolas Gillis and François Glineur. "A multilevel approach for nonnegative matrix factorization". In: *Journal of Computational and Applied Mathematics* 236.7 (2012), pp. 1708–1723.

[9] Rui Guo, Wei Wang, and Hairong Qi. "Hyperspectral image unmixing using autoencoder cascade". In: *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE. 2015, pp. 1–4.

[10] Jingu Kim, Yunlong He, and Haesun Park. "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework". In: *Journal of Global Optimization* 58.2 (2014), pp. 285–319.

[11] Daniel Lee and Sebastian Seung. "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems* 13 (2000).

[12] Daniel Lee and Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.

[13] Giorgio Licciardi and Fabio Del Frate. "Pixel unmixing in hyperspectral data by means of neural networks". In: *IEEE transactions on Geoscience and remote sensing* 49.11 (2011), pp. 4163–4172.

[14] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. "Clustering using sum-of-norms regularization: With application to particle filter output computation". In: *2011 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE. 2011, pp. 201–204.

[15] Xin Luo et al. "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems". In: *IEEE Transactions on Industrial Informatics* 10.2 (2014), pp. 1273–1284.

[16] Xiaochen Lv, Wenhong Wang, and Hongfu Liu. "Cluster-wise weighted NMF for hyperspectral images unmixing with imbalanced data". In: *Remote Sensing* 13.2 (2021), p. 268.

[17] Andersen Man Shun Ang. *Projection onto unit L2 ball*. 2020. URL: https://angms.science/doc/CVX/Proj_l2.pdf (visited on 12/14/2022).

[18] Andersen Man Shun Ang. *Solving Nonnegative Matrix Factorization using column-wise Block Coordinate Descent*. 2020. URL: https://angms.science/doc/NMF/nmf_cd.pdf (visited on 12/14/2022).

[19] Andersen Man Shun Ang et al. "Accelerating block coordinate descent for non-negative tensor factorization". In: *Numerical Linear Algebra with Applications* 28.5 (2021), e2373.

[20] Lidan Miao and Hairong Qi. "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.3 (2007), pp. 765–777.

[21] Nguyen Nam et al. "Nesterov's smoothing technique and minimizing differences of convex functions for hierarchical clustering". In: *Optimization Letters* 12.3 (2018), pp. 455–473.

[22] Pentti Paatero and Unto Tapper. "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". In: *Environmetrics* 5.2 (1994), pp. 111–126.

[23] Yuntao Qian et al. "Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization". In: *IEEE Transactions on Geoscience and Remote Sensing* 49.11 (2011), pp. 4282–4297.

[24] Dong Shan et al. "Rank-adaptive non-negative matrix factorization". In: *Cognitive Computation* 10.3 (2018), pp. 506–515.

[25] Le Sun. *Hyperspectral Data Set*. URL: http://lesun.weebly.com/hyperspectral-data-set.html (visited on 12/14/2022).

[26] Leo Taslaman and Björn Nilsson. "A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data". In: *PloS one* 7.11 (2012), e46331.

[27] Homer Walker and Peng Ni. "Anderson acceleration for fixed-point iterations". In: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735.

[28] Dawei Wang, Yunhui He, and Hans De Sterck. "On the asymptotic linear convergence speed of Anderson acceleration applied to ADMM". In: *Journal of Scientific Computing* 88.2 (2021), pp. 1–35.

[29] Jiaojiao Wei and Xiaofei Wang. "An overview on linear unmixing of hyperspectral data". In: *Mathematical Problems in Engineering* 2020 (2020).

[30] Michael Winter. "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data". In: *Imaging Spectrometry V*. Vol. 3753. SPIE. 1999, pp. 266–275.

[31] Stephen Wright, Jorge Nocedal, et al. "Numerical optimization". In: *Springer Science* (1999), pp. 30–66.

[32] Wei Xu, Xin Liu, and Yihong Gong. "Document clustering based on non-negative matrix factorization". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 267–273.

[33] Xiaohui Yang et al. "Adaptive factorization rank selection-based NMF and its application in tumor recognition". In: *International Journal of Machine Learning and Cybernetics* 12.9 (2021), pp. 2673–2691.

[34] Sheng Zhang et al. "Learning from incomplete ratings using non-negative matrix factorization". In: *Proceedings of the 2006 SIAM international conference on data mining*. SIAM. 2006, pp. 549–553.

[35] Xiangrong Zhang et al. “Hyperspectral unmixing via deep convolutional neural networks”. In: *IEEE Geoscience and Remote Sensing Letters* 15.11 (2018), pp. 1755–1759.

[36] Lihong Zhao, Guibin Zhuang, and Xinhe Xu. “Facial expression recognition based on PCA and NMF”. In: *2008 7th World Congress on Intelligent Control and Automation*. IEEE. 2008, pp. 6826–6829.

[37] Ruicong Zhi et al. “Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.1 (2010), pp. 38–52.

Appendices

Appendix A

Derivation of ADMM Iterations

The minimization problem for our model NMF-SON is

$$\min_{w} \frac{1}{2}\|h^j\|_2^2\|w\|_2^2 - \langle M_j {h^j}^T, w\rangle + \lambda \sum_{i\neq j} \|w - w_i\|_2 + c, \tag{A.1}$$

For ADMM, we introduce local variables w_f, w_0, w_i and a central variable z to represent w_j, and express the equation A.1 in a separable form

$$\min_{w_f, w_0, \{w_i\}, z} f(w_f) + g_0(w_0) + \sum_{i\neq j} g_i(w_i), \tag{A.2}$$

where

$$f(w) = \frac{1}{2}\|h^j\|_2^2\|w\|_2^2 - \langle M_j {h^j}^T, w\rangle, \tag{A.3}$$

$$g_0(w) = i_+(w), \tag{A.4}$$

$$g_i(w) = \lambda\|w - c_i\|_2 \ , \ c_i = w_i \text{ such that } i \neq j, \tag{A.5}$$

and $w_f = z$, $w_0 = z$, $w_i = z \ \forall i$. The indicator function

$$i_+(x) = \begin{cases} +\infty & x < 0 \\ 0 & x \geq 0 \end{cases}, \tag{A.6}$$

maps from $\mathbb{R}^m$ to $\mathbb{R}$. The augmented Lagrangian of equation A.3 is

$$\begin{aligned} \underset{w_f,w_0,w_i,z}{\operatorname{argmin}} \underset{y_f,y_0,y_i}{\operatorname{argmax}} L_\rho(w_f, w_0, w_i, z, y) &= f(w_f) + g_0(w_0) + \sum_{i \neq j} g_i(w_i) + \langle y_f, w_f - z \rangle \\ &+ \langle y_0, w_0 - z \rangle + \sum_{i \neq j} \langle y_i, w_i - z \rangle + \frac{\rho}{2}\|w_f - z\|_2^2 + \frac{\rho}{2}\|w_0 - z\|_2^2 + \sum_{i \neq j} \frac{\rho}{2}\|w_i - z\|_2^2, \end{aligned} \tag{A.7}$$

where y_f, y_0, y_i are langragian variables and ρ is a penalty paramter. Using the augment Lagrangian A.7, we get the following iterations, where k is the iteration number:

$$\begin{aligned} w_f^{k+1} &= \underset{w_f}{\operatorname{argmin}} L_\rho(w_f) \\ &= \underset{w_f}{\operatorname{argmin}} f(w_f) + \langle y_f, w_f - z \rangle + \frac{\rho}{2}\|w_f - z\|_2^2 \\ &= \underset{w_f}{\operatorname{argmin}} \frac{1}{2}\|h^j\|_2^2\|w_f\|_2^2 - \langle M_j {h^j}^T, w_f \rangle + \langle y_f, w_f - z \rangle + \frac{\rho}{2}\|w_f - z\|_2^2 \\ &\Leftrightarrow \|h^j\|_2^2 w_f - M_j {h^j}^T + y_f + \rho w_f - \rho z = 0 \\ &\Leftrightarrow w_f^{k+1} = \frac{M_j (h^j)^T - y_f^k + \rho z^k}{\rho + \|h^j\|_2^2} \end{aligned} \tag{A.8}$$

$$
\begin{aligned}
w_0^{k+1} &= \underset{w_0}{\operatorname{argmin}}\, L_\rho(w_0) \\
&= \underset{w_0}{\operatorname{argmin}}\, g_0(w_0) + \langle y_0, w_0 - z\rangle + \frac{\rho}{2}\|w_0 - z\|_2^2 \\
&= \underset{w_0}{\operatorname{argmin}}\, i_+(w_0) + \langle y_0, w_0 - z\rangle + \frac{\rho}{2}\|w_0 - z\|_2^2 \\
&= \left[w_0 - \frac{1}{\rho}(y_0 + \rho(w_0 - z))\right]_+ \\
&= \left[z^k - \frac{y_0^k}{\rho}\right]_+
\end{aligned}
\tag{A.9}
$$

$$
\begin{aligned}
w_i^{k+1} &= \underset{w_i}{\operatorname{argmin}}\, L_\rho(w_i) \\
&= \underset{w_i}{\operatorname{argmin}}\, g_i(w_0) + \langle y_i, w_i - z\rangle + \frac{\rho}{2}\|w_i - z\|_2^2 \\
&= \underset{w_i}{\operatorname{argmin}}\, \lambda\|w_i - c_i\|_2^2 + \langle y_i, w_i - z\rangle + \frac{\rho}{2}\|w_i - z\|_2^2 \\
&= \operatorname{Prox}_{\lambda\|.-c_i\|_2}(w_i - \frac{1}{\rho}(y_i + \rho(w_i - z))) \\
&= \operatorname{Prox}_{\lambda\|.-c_i\|_2}(\zeta) \text{ where } \zeta = z - \frac{y_i}{\rho} \\
&= \zeta - \operatorname{Prox}_{\lambda\|.-c_i\|_2\le 1}(\frac{\zeta}{\lambda}) \\
&= \begin{cases} \zeta - \lambda(\frac{\frac{\zeta}{\lambda} - c_i}{\|\frac{\zeta}{\lambda} - c_i\|_2}), & \|\frac{\zeta}{\lambda} - c_i\|_2 > 1 \\ \zeta - \lambda(\frac{\zeta}{\lambda} - c_i), & \|\frac{\zeta}{\lambda} - c_i\|_2 \le 1 \end{cases} \text{ where } \zeta = z^k - \frac{y_i^k}{\rho},
\end{aligned}
\tag{A.10}
$$

where c_i are columns w_i such that $i = j$. Note that $\operatorname{Prox}_{\lambda\|.-c_i\|_2}(\zeta) = \operatorname{argmin}_u \frac{1}{2}\|u - \zeta\|_2^2 + \lambda\|u - \zeta\|$ which we solve using Moreau's decomposition

$$
\operatorname{Prox}_{\lambda g}(v) = v - \lambda \operatorname{Prox}_{\frac{1}{\lambda}g*}(\frac{v}{\lambda}), \tag{A.11}
$$

where $g*$ is the conjugate of g, which is the unit norm ball of the dual of l_2 norm. So, $\operatorname{Prox}_{\frac{1}{\lambda}g*} = \operatorname{Prox}_{\lambda\|.-c_i\|_2\le 1}$ and

$$\text{Prox}_{\lambda\|.-c_i\|_2\leq 1} = \begin{cases} \frac{\zeta - c_i}{\|\zeta - c_i\|_2}, & \|\zeta - c_i\|_2 > 1 \\ \zeta - c_i & \|\zeta - c_i\|_2, \leq 1 \end{cases} \tag{A.12}$$

Now using the updated local variables w_f^{k+1}, w_0^{k+1} and w_i^{k+1}'s, we can update the central variable z.

$$\begin{aligned} z^{k+1} &= \underset{z}{\text{argmin}}\, L_\rho(z) \\ &= \underset{z}{\text{argmin}} \langle y_f, w_f - z\rangle + \langle y_0, w_0 - z\rangle + \sum_{i\neq j}\langle y_i, w_i - z\rangle + \frac{\rho}{2}\|w_f - z\|_2^2 \\ &+ \frac{\rho}{2}\|w_0 - z\|_2^2 + \sum_{i\neq j}\frac{\rho}{2}\|w_i - z\|_2^2 \\ &\Leftrightarrow -y_f - y_0 - \sum_i y_i + \rho(z - w_f) + \rho(z - w_0) + \sum_i \rho(z - w_i) = 0 \\ &\Leftrightarrow \rho(2 + |E|)z = \rho(w_f + w_0) + \rho\sum_i w_i + y_f + y_0 + \sum_{i\neq j} y_i \\ &= \frac{\rho(w_f^{k+1} + w_0^{k+1}) + \rho\sum_i w_i^{k+1} + y_f^k + y_0^k + \sum_{i\neq j} y_i^k}{\rho(2 + |E|)} \end{aligned} \tag{A.13}$$

Finally, the Lagrangian variables are updated.

$$y_f^{k+1} = y_f^k + \rho(w_f^{k+1} - z^{k+1}) \tag{A.14}$$

$$y_0^{k+1} = y_0^k + \rho(w_0^{k+1} - z^{k+1}) \tag{A.15}$$

$$y_i^{k+1} = y_i^k + \rho(w_i^{k+1} - z^{k+1}) \tag{A.16}$$