Unterschrift des Betreuers

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna | Austria

Diplomarbeit

Non-Negative Matrix Factorization

Ausgeführt am Institut für

Stochastik und Wirtschaftsmathematik

der Technischen Universität Wien

unter der Anleitung von

Univ.-Prof. Dipl.-Ing. Dr. techn. Peter Filzmoser

durch

El-Cheschin Carim, BSc
Thimiggasse 63-69/2/5, 1180 Vienna

Datum

Unterschrift

Acknowledgements

Firstly, I would like to express my gratitude to Prof. Filzmoser for his kind support and helpful advice in writing this thesis. I would also like to thank Irene for preparing the data and the helpful input as well as Prof. Varmuza for the chemometrical insights.

The numerical experiments in this work are based on data from meteorites. For this, the project P 26871-N20 of the Austrian Science Fund (FWF) is acknowledged.

I would also like thank my proofreaders, Prof. Filzmoser, my colleague Bauer P. and my friend Schuler I., who spent a significant amount of time to improve the readability of this thesis and corrected my spelling mistakes.

A special thanks to my girlfriend Forst J., who supported and motivated me all the time.

Furthermore, I would like to thank my friends and colleagues for their support and friendship during my time at university.

I dedicate this work to Michael Grutschnig, one of my best friends, who was like a brother to me, but unfortunately died far too early. I learned a lot from him and without him I would never have come this far.

Lastly, I would like to thank my big brother and father for supporting me throughout my studies and for never doubting my abilities.

Abstract

Non-negative matrix factorization - NMF is a Linear Dimensionality Reduction method, which approximates a high dimensional non-negative data matrix by a multiplication of two low-ranked matrices that preserves the non-negativity of the data. This property has proven to be beneficial as it allows for the approximated data to be interpreted in the same way as the original data. In addition, NMF leads to a part-based representation of the data, which supports easy identification of the essential parts/features.

The thesis starts with a short introduction of NMF, which includes a motivation behind the method, a detailed comparison to the well-known Principal Component Analysis and the possible generalizations of the ”standard NMF” problem. This is followed by a chapter presenting an overview of the wide range of NMF algorithms, which are separated into algorithms based on standard nonlinear optimization schemes and so called separable NMF. All algorithms of the first group are based on the two block gradient descent scheme. In contrast, the separable NMF is restricted to a subclass of matrices characterized by a practical geometrical interpretation which is exploited in many separable NMF algorithms. The last theoretical chapter focuses on the description of the key topics that should be considered when applying NMF such as initialization methods, rank estimation and quality measures to compare the performance of the algorithms.

The thesis concludes with the analysis of the NMF methods for a spectrometric dataset consisting of TOF-SIMS measurements taken from meteorites. The ability of NMF to separate spectra into two dissimilar spectra with one considered as the background and one as meteorite specific has been analyzed. The obtained results are promising and give reason to believe that NMF is an adequate method for such tasks. In addition, the robustness to noise of NMF methods in the context of spectral data has been tested and finally the task of defining an appropriate factorization rank has been discussed.

Contents

Abstract **v**

1 Introduction to NMF **1**
1.1 Motivation and Delimitation 1
1.1.1 PCA vs NMF 3
1.2 Generalizations of the "standard NMF" Problem 7

2 Algorithms **9**
2.1 Standard Nonlinear Optimization Schemes 10
2.1.1 Lee and Seung: Multiplicative Algorithms 10
2.1.2 ALS Algorithms 17
2.2 Separable NMF 22
2.2.1 Separability Assumption 22
2.2.2 Geometry of Exact NMF 23
2.2.3 Algorithms 24

3 Challenges Arising from NMF **29**
3.1 Initialization Methods 29
3.1.1 Random Initialization Strategies 29
3.1.2 Nonnegative Double Singular Value Decomposition 30
3.1.3 SVD-NMF 36
3.1.4 Clustering Methods 37
3.1.5 ICA Based Initialization 40
3.2 Quality Measures 48
3.2.1 Standard Measures 48
3.3 Rank Estimation 51
3.3.1 Rank Estimation Based on Quality Measures 51
3.3.2 SVD-based rule 52

4 Simulations **55**
4.1 Meteorite Data 55
4.2 Corn vs Substrate 58
4.3 MixUp Spectra with/without Noise 65
4.3.1 Mix - without Noise 66
4.3.2 Mix - with Additive Noise 68
4.4 Rank Estimation "allende" 75

4.5 Conclusion 79

Bibliography **97**

1 Introduction to NMF

This chapter focuses on the purpose, delimitation to other methods of matrix factorization and the basic concepts of **NMF - Non-Negative Matrix Factorization**. It should be noted that terms such as *Positive Matrix Factorization* or *Non-Negative Matrix Approximation* sometimes are used instead of NMF in different/other literature . In this thesis, only the term NMF is used in order to avoid possible misunderstandings.

1.1 Motivation and Delimitation

In today's world, an enormous amount of information is collected and stored in data sets which can be represented mathematically as matrices. Witch such huge amounts of data it can be difficult to keep track of the primary objective. Furthermore, in such cases data analysis methods, which are suitable for low-dimensional data, should be used with caution. As a result, methods for extracting essential or fundamental information from such large amounts of data (matrices) have become increasingly popular in recent years. These methods are referred to as **CLRMA - Constrained Low-Rank Matrix Approximation**, which corresponds to *Linear Dimensionality Reduction*.

Consider the task of finding a set of r basis vectors $w_l \in \mathbb{R}^p$ $(l = 1, 2 \ldots, r)$ and the corresponding weights h_{lj} for a given set of n data points $m_j \in \mathbb{R}^p$ $(j = 1, 2, \ldots, n)$ with the restrictions that <u>for all</u> j, $m_j \approx \sum_{l=1}^{r} h_{lj}\ w_l$ and also $r \ll \min(n, p)$. The low-rank approximation of matrix M, with

$$M = [m_1\ m_2\ \ldots\ m_n] \approx [w_1\ w_2\ \ldots\ w_r][h_1\ h_2\ \ldots\ h_n] = WH,$$

where each column of $M(\in \mathbb{R}^{p \times n})$ is a data point, each column of $W(\in \mathbb{R}^{p \times r})$ is a basis vector, and each column of $H(\in \mathbb{R}^{r \times n})$ provides the coordinates of the corresponding column of M in the basis W, can be considered equivalent to the problem described beforehand. In other words, linear combinations of the columns of W are used to approximate each column of M.

Due to the relatively small number of basis vectors compared to the large number of data points, a good approximation can only be achieved if the basis vectors detect a latent structure in the data. If dealing with such models in practice, two major choices emerge:

1. **Measure of the error** $M - WH$.
   The use of the standard least-squares error (or Frobenius norm), $||M - WH||^2 = \sum_{lj}(M - WH)^2_{lj}$ leads to the well-known principal component analysis (**PCA**). A brief explanation of the principal component analysis and a comparison with *NMF* will be given in subsection 1.1.1. In practice, it is common that some data is missing or weights are assigned to the entries of M. Thus, this problem can be interpreted as a weighted low-rank matrix approximation (**WLRA**) with error $\sum_{lj} U_{lj}(M - WH)^2_{lj}$ for some non-negative weight matrix U, where $U_{lj} = 0$ if the entry (l, j) is missing (see Srebro and Jaakkola, 2003). Moreover, the problem can be referred to PCA with missing data or to low-rank matrix completion with noise, if U contains only entries in $\{0, 1\}$. Another possibility is to use the sum of absolute values of the entries as error $\sum_{lj} |M - WH|_{lj}$, which is more robust to outliers and is sometimes referred to as **robust PCA** (see Candès et al., 2011). These few examples will be expanded upon in the following sections by introducing other measures/cost functions.

2. **Constraints that the factors** W **and** H **should satisfy**.
   The setting of these constraints leads to a meaningful interpretation of the factors and depends on the respective area of application. Consider for instance k-means[1], which equates to the requirement that the factor H in each column must have a single entry equal to 1, so that the columns of W can be determined as cluster centroids in this context. Another common variant, which is known as **sparse PCA**, is achieved by constraining the factors (W and/or H) to be sparse (see d'Aspremont et al., 2007), yielding to an easily interpretable and more compact decomposition (e.g., if H is sparse, it follows that each data point is the linear combination of only a few basis elements). If componentwise nonnegativity is required for both factors, W and H, the CLRMA is called **NMF - Nonnegative Matrix Factorization**.

In fact, the **standard NMF** problem for $M \in \mathbb{R}^{p \times n}_{+}$ can be formulated in the following way:

$$\min_{W \in \mathbb{R}^{p \times r}, H \in \mathbb{R}^{r \times n}} ||M - WH||^2 \quad \text{such that} \quad W, H \geq 0. \tag{1.1}$$

Basically, there are the two main motivations for the application of NMF:

1. **Interpretability**
   The non-negativity constraints allow the basis elements to be interpreted like the data, while the H entries can be interpreted as activation coefficients or weightings. Furthermore, non-negativity is a natural requirement for many practical problems. For example, non-negative entities such as color intensities, chemical concentrations, frequency counts, signal intensities of ions, etc., can still be displayed as non-negative measurements after the application of NMF,

---

[1] The k-means procedure outputs a set of centroids w_k that minimizes the sum of squared distances between each data point and the closest centroid.

allowing direct interpretation. In text mining, where each entry M_{lj} of the matrix M indicates the importance (e.g., the number of appearances) of word l in document j, the non-negativity allows the columns of the factor W to be interpreted as topics, and the columns of the factor H links the documents to these topics (see Lee and Seung, 1999).

2. **Sparseness of factors**
   The sparseness of factors is naturally achieved by the non-negativity constraints related to the first-order optimality conditions given for a $\min_{x \geq 0} f(x)$ problem. These conditions are:

   a) $x \geq 0, \nabla_k f(x) \geq 0 \quad \forall k$

   b) $\nabla_k f(x) x_k = 0 \quad \forall k$

   Hence, the stationary points of (1.1) are expected to have zero entries, which leads to better compression and interpretability of the data compared to unconstrained variants like **PCA**. The sparse representation of basic factors facilitates interpretation, because the resulting parts are structurally simple.

1.1.1 PCA vs NMF

Review of PCA

The principal component analysis is one of the most widely used linear reduction methods, which has a lot to do with its ability to provide the optimal solution of linear matrix approximation with respect to the standard least-squares error. The dimensionality reduction of the data matrix M is achieved by finding a few orthogonal linear combinations (the principal components-PCs) of the original data points, which are maximizing the variance in the data. Based on the eigenvalue decomposition of the covariance matrix $\Sigma = M^T M/(p-1)$ (of the standardized data matrix M):

$$\Sigma = \frac{1}{(p-1)} V D^2 V^T, \tag{1.2}$$

where $V = [v_1 \ldots v_n] \in \mathbb{R}^{n \times n}$ is the matrix of the orthogonal eigenvectors (in this context called *principal component directions*) in descending order and $D^2 = [d_1^2 \ldots d_n^2] \in \mathbb{R}^{n \times n}$ is the diagonal matrix with the corresponding eigenvalues $d_1^2 \geq d_2^2 \geq \cdots \geq d_n^2$, the principal components are given by

$$Y = MV. \tag{1.3}$$

Consequently, the first PC $y_1 = Mv_1$ is the linear combination of the original data with the largest variance ($Var(Mv_1) = \frac{d_1^2}{p-1}$); the second PC $y_2 = Mv_2$ is the linear combination of the data with the second largest variance ($Var(Mv_2) = \frac{d_2^2}{p-1}$) while being orthogonal to the first PC, and so forth.

Another possibility to find the principal components is to minimize the least-squares error (*reconstruction error*) between data points and their orthogonal projections. This problem can be solved by a singular value decomposition (**SVD**) of M:

$$M = UDV^T, \tag{1.4}$$

where $U \in \mathbb{R}^{p \times n}$ denotes an orthogonal Matrix, $D = \sqrt{D^2}$ with D^2 (given in (1.2)) and V consisting of the principal component directions (given in (1.2)).

The main components are the transformed data points of the orthogonal space spanned by the main component directions. Thus, if only the first $r < n$ principal component directions are used for the transformation, the data points are mapped to the r-dimensional orthogonal subspace. According to the **Schmidt and Eckart-Young theorem** (see Stewart and Sun, 1990), the transformed data matrix $Y^{(r)} = M[v_1 \ldots v_r]$ is the best least-squares rank-r approximation of the original data matrix M:

$$Y^{(r)} = \underset{rank(G) \leq r}{\arg\min} \|M - G\| . \tag{1.5}$$

In many situations, a big portion of the data variance can be explained by the first two (or three) PCs, which makes a simple visualization of the approximated data possible. Conversely, it can be difficult to interpret PCA's results accurately.

PCA vs NMF

In the following listings the key differences between the Principal Component Analysis and the Non-Negative Matrix Factorization are highlighted:

- **Uniqueness**
  While PCA is able to reach the global minimum of the corresponding optimization problem, algorithms of NMF generally converge only to local minima (and even this convergence is not guaranteed, since for many algorithms saddle points are also possible). For this reason, the set of principal components is unique, while NMF has several solutions concerning basis and weight matrices. In Chapter 2, the convergence of algorithms will be discussed in further detail. To minimize the possibility of being trapped with a bad solution and impose some sort of uniqueness, many strategies have been developed to find appropriate initial matrices for W and H. A selection of these initialization methods is described in Section 3.1.
- **Ranking**
  The principal components are naturally ranked according to the quantity of their explained variance (Naik, 2015). This ranking is not given in NMF as all factors

are considered equally important. In addition, the NMF factors do not provide an immediate indication of what an appropriate value of the rank parameter r could be. The application of PCA does not even require the r value to be specified, since all the eigenpairs are computed and then the most important principal components are selected according to the proportion of variance (common values are 80% and 90%) that should be achieved. Using the NMF procedure, the parameter r must be specified by the user as an input parameter. In practice, there are some strategies for assigning an advantageous ranking value r for a given matrix. This is usually done by running different factorizations with different rank values to evaluate their factorization performance with respect to the target matrix. In Section 3.3 some strategies to estimate the rank value are described.

- **Orthogonality**
  The principal directions, which, as already explained, maximize variance in the data, form an orthogonal basis. The factors obtained by NMF are positive vectors that better approximate non-negative data, however they are usually not orthogonal. The basis factors of NMF have a nice geometrical interpretation as they are the basis of the hypercone containing all data and they preseve local data structure in this subspace (see for details Huang et al., 2014). In Figure 1.1 a simple example is given to illustrate the difference between a basis extracted by PCA and NMF. In general, orthogonality is a desirable characteristic, but as this example shows, the elements of the principal components are not all positive due to forced orthogonality, even if PCA is applied to non-negative data. Moreover, it can be noted that the non-negativity restriction of PCA is always violated, which leads to a loss of interpretability of the data when moving from the original data space to the reduced subspace with low dimensionality.

As has been mentioned, one of the key advantages of NMF over PCA is the interpretability of the obtained factors. As was shown by Lee and Seung (1999), the parts-based representation is typical for NMF and it leads to more intuitive and understandable results than PCA. In Figure 1.2 a demonstrative example in the context of a facial image recognition problem is illustrated.

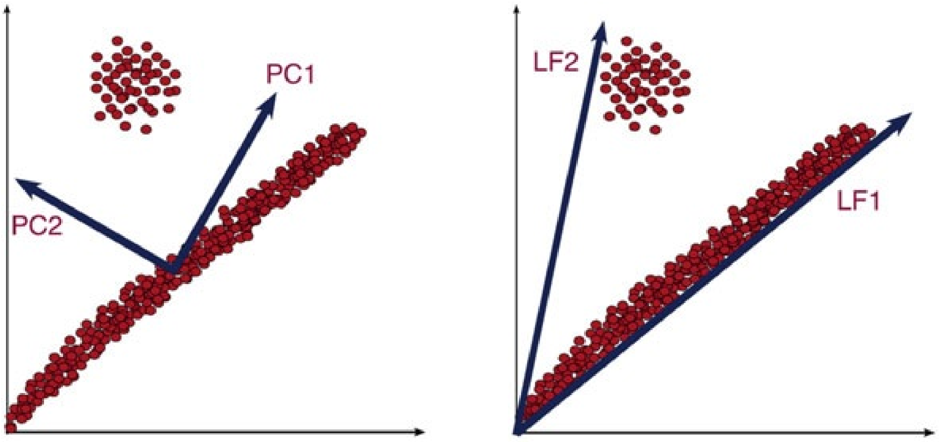

Figure 1.1: **Comparison PCA vs NMF** - In this example the basis components extracted with PCA (left panel) and those with NMF (right panel) of a non-negative two-dimensional data matrix are compared. After transforming the samples (all in the positive orthant) by PCA, all the data points belonging to the line assume negative values. The NMF basis preserves the non-negativity of data, which leads to a part-based representation (from Naik, 2015).

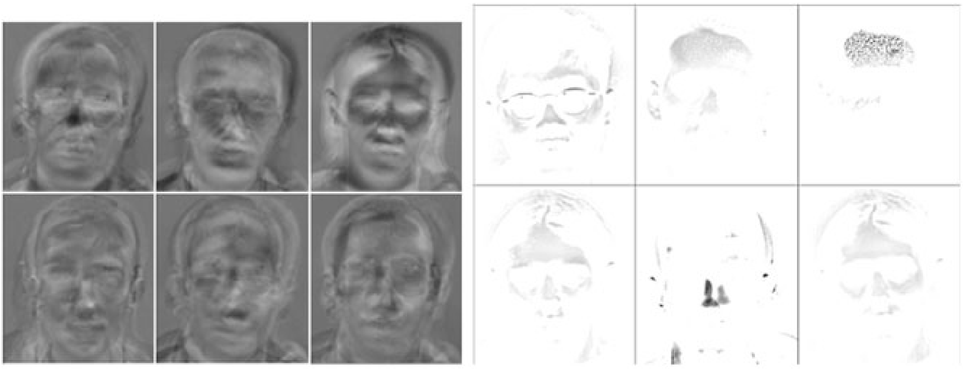

Figure 1.2: **Facial image recognition - PCA vs NMF** - In this example the basis components extracted with PCA (left panel) and those with NMF (right panel) for 6 images of faces are compared. The eigenfaces obtained by PCA are prototypical faces containing all kinds of facial traits, while the NMF basis vectors represent particular traits: different kind of eyes, noses, and mouths (from Naik, 2015).

1.2 Generalizations of the ”standard NMF” Problem

The decomposition of the original data as combinations of parts is the key feature of NMF, but without any constraint (”standard NMF”) the parts could lack intuitiveness and fail to provide the analyst with a clear understanding of the underlying data. Thus, generalizations of the ”standard NMF” problem (1.1) have the intention to achieve better interpretable results by imposing additional constraints on the basis matrix W and/or weight matrix H.

A more general formulation for the NMF problem is given as:

$$\min_{W \in \mathbb{R}^{p \times r}, H \in \mathbb{R}^{r \times n}} \|M - WH\|^2 + \alpha J_1(W) + \beta J_2(H) \quad \text{such that} \quad W, H \geq 0, \tag{1.6}$$

where the penalty terms $J_1(W)$ and $J_2(H)$ add constraints to the original problem, while α and β, so-called regularization parameters, balance the trade-off between the approximation error and additional constraints (see Naik, 2015). Typically, penalty terms are added to enforce sparseness (some examples are obtained in Chapter 2) or to enhance smoothness of the NMF factors.

This formulation could be further generalized by replacing the Frobenius norm $\|\cdot\|$ with an arbitrary divergence function (or loss function) $\mathcal{D}(\cdot\|\cdot)$:

$$\min_{W \in \mathbb{R}^{p \times r}, H \in \mathbb{R}^{r \times n}} \mathcal{D}(M\|WH) + \alpha J_1(W) + \beta J_2(H) \quad \text{such that} \quad W, H \geq 0, \tag{1.7}$$

The algorithms described in Chapter 2 either use the Frobenius norm or the Kullback-Leibler-divergence, but many other choices are possible depending on the area of application.

2 Algorithms

In this chapter a selection of algorithms for **NMF** are described, which are considered as relevant and useful. Besides the basic methodology of the algorithms also their important properties like the theoretical assumptions, computational efficiency and stability are shown.

Remark: In general the most widely used convex models are based on the approach of minimizing the nuclear norm of X:

$$\min_X \|M - X\| + \lambda \|X\|_*, \tag{2.1}$$

where $\lambda > 0$ is a penalty parameter and $\|X\|_* = \sum_{i=1}^{\min(n,p)} \sigma_i(X) = \|\sigma(X)\|_1$ is the nuclear norm, with $\sigma(X)$ being the vector of singular values of X. In case the matrix M satisfies some conditions depending on the model (M has to be close to a low-rank matrix), the optimal solution can be guaranteed to recover the solution of the original problem ($\min_X \|M - X\|$ s.t $rank(X) = r$). This problem (2.1) can be reformulated as a semidefinite program and in many cases any stationary point can be guaranteed to be a global minimum (see Boumal et al., 2016 and Li and Tang, 2016).

In contrast to other *CLRMA* varaints (such as robust PCA, sparse PCA, and PCA with missing data), there does not exist a successful convexification approach for NMF (Gillis, 2017). It has to be considered that the low-rank approximation $X = WH$ can not be used directly to define the nuclear norm of X. Since even if the best non-negative approximation X of non-negative rank r for M is given, it is generally still difficult to recover the exact NMF (W, H) of X. Due to the symmetry of the problem (permuting columns of W and rows of H accordingly leads to an equivalent solution) writing a direct convexification in variables (W, H) seems difficult. Breaking this symmetry seems nontrivial (see Gillis, 2011).

The NMF algorithms can be divided into two main classes, where one class requires the M input matrix to be separable (discussed and defined subsequently) and the other class does not impose this assumption.

2.1 Standard Nonlinear Optimization Schemes

These NMF algorithms are based on a *two-block coordinate descent scheme*, which is a straightforward and popular way used for CLRMA:

0. Initialize $(W, H) \geq 0$.
1. $W \leftarrow X$, where X solves exactly or approximately $\min_{X \geq 0} \|M - XH\|$.
2. $H \leftarrow Y$, where Y solves exactly or approximately $\min_{Y \geq 0} \|M - WY\|$.

The two sub-problems to be solved are known as non-negative least squares (**NNLS**). The NNLS sub-problems are convex in W or respectively H as $W \geq 0$ or respectively $H \geq 0$ are defining convex sets. Therefore, in each step the global minima for the sub-problem can be determined by applying standard nonlinear optimization techniques. Nonetheless, this proposed scheme only guarantees convergence, usually to a first-order stationary point.

The algorithms in this section differ by the approaches applied to solve the NNLS subproblems.

2.1.1 Lee and Seung: Multiplicative Algorithms

The pioneering papers of Lee and Seung (see Lee and Seung, 1999, 2001) popularized the NMF problem since they provided simple and efficient algorithms at that point in time.

The two algorithms of *Lee and Sueng* are based on iterative multiplicative updates of W and H, but differ in the cost function they minimize.

Cost Functions and optimization problems

To quantify the quality of the approximation, cost functions need to be defined. A measure of distance between two non-negative matrices A and B can be used to specify a cost function. The square of the Euclidean distance between A and B (see Paatero, 1997),

$$\|A - B\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2 \tag{2.2}$$

appears to be a useful measure and is considered to be the cost function for the NMF standard problem (1.1). This function fulfils two important properties, namely it is lower bounded by zero and achieves this bound only if $A = B$.

The measure used for the second algorithm is

$$D(A\|B) \quad = \quad \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right). \tag{2.3}$$

Due to the fact that this measure is not symmetric in A and B, it cannot be called a "distance" and it is therefore referred to as "divergence" of A from B in literature. The same two properties of the Euclidean distance are satisfied with zero as the lower bound, and vanishing if and only if $A = B$. When $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$, the matrices A and B can be regarded as normalized probability distributions and the measure reduces to the well-known Kullback-Leibler divergence, or relative entropy.

Consider this two formulations of NMF as optimization problems:

a) **NMF standard problem** (see (1.1))

b) $min_{W \in \mathbb{R}^{p\times r}, H \in \mathbb{R}^{p\times n}} D(M\|WH)$ such that $W, H \geq 0$.

Despite the functions $||M - WH||^2$ and $D(M||WH)$ being convex only in W or H, the convexity is not given in respect to both variables together. Hence, expecting an algorithm to solve the two optimization problems in the sense of finding global minima, is problematic. Nevertheless, a variety of techniques from numerical optimization can be used to find local minima. A simple technique to implement is *gradient descent*, which can have a slow convergence rate. Other methods such as *conjugate gradient* are proven to convergence faster, but are more complicated to implement than *gradient descent* (Lee and Seung, 2001).

Update rules - relation to gradient descent

Lee and Seung proposed the following "multiplicative update rules", which are easy to implement and are used as the basis for many other NMF algorithms.

Theorem 2.1.1 (*Lee and Seung, 2001*):
The Euclidean distance $||M - WH||$ is non-increasing under the update rules

$$W_{lj} \leftarrow W_{lj} \frac{(MH^T)_{lj}}{(WHH^T)_{lj}}, \qquad H_{lj} \leftarrow H_{lj} \frac{(W^T M)_{lj}}{(W^T WH)_{lj}}. \tag{2.4}$$

The Euclidean distance is invariant under these updates if and only if W and H are at a stationary point of the distance.

Theorem 2.1.2 (*Lee and Seung, 2001*):
The divergence $D(M\|WH)$ is non-increasing under the update rules

$$W_{lj} \leftarrow W_{lj} \frac{\sum_k H_{jk} \frac{M_{lk}}{(WH)_{lk}}}{\sum_m H_{jm}}, \qquad H_{lj} \leftarrow H_{lj} \frac{\sum_k W_{kl} \frac{M_{kj}}{(WH)_{kj}}}{\sum_m W_{ml}}. \tag{2.5}$$

The divergence is invariant under these updates if and only if W and H are at a stationary point of the divergence.

Before the proofs to these theorems are outlined, the characteristics of these update rules are mentioned and compared to the update rules of the gradient decent (method). As already can be suggested by the name of the update rules, each update is obtained as the multiplication by a factor. Especially, in case of an exact factorization $M = WH$ the multiplicative factor is equal to 1, which ensures the perfect reconstruction of M to be a fixed point of these update rules.

A simple additive update for H, which reduces the squared distance can be written as

$$H_{lj} \leftarrow H_{lj} + \eta_{lj} \; [(W^T M)_{lj} - (W^T WH)_{lj}]. \tag{2.6}$$

Considering that η_{lj} are all set to some sufficiently small positive number this additive update is equivalent to conventional gradient descent and should reduce $\|M - WH\|$. The update rule for H which is given in Theorem 2.1.1 can be obtained with (2.6) by diagonally rescaling the variables and setting

$$\eta_{lj} = \frac{H_{lj}}{(W^T WH)_{lj}}. \tag{2.7}$$

Note that this rescaling leads to a multiplicative factor with the positive component of the gradient in the denominator and the absolute value of the negative component in the numerator of the factor.

In a very similar way, the relation to the gradient descent is also shown for the divergence. It can be explained with the diagonally rescaled gradient descent taking the form

$$H_{lj} \leftarrow H_{lj} + \eta_{lj} \left[\sum_k W_{kl} \frac{M_{kj}}{(WH)_{kj}} - \sum_k W_{kl} \right]. \tag{2.8}$$

This update should reduce $D(M\|WH)$, if the η_{lj} are all set to some small positive number. By setting

$$\eta_{lj} = \frac{H_{lj}}{\sum_k W_{kl}}, \tag{2.9}$$

and insert it in (2.8) the update rule for H can be obtained. Consequently, this rescaling can be viewed as a multiplicative rule with the positive component of the gradient in the denominator and the absolute value of the negative component in the numerator of the multiplicative factor.

The reason why such a rescaled gradient descent converges and leads to a decrease of the cost function is not obvious, since the chosen η_{lj} are not small. In the next section the proof of convergence is given.

Proofs of update rules convergence

In order to prove the Theorems 2.1.1 & 2.1.2 the concept of **auxiliary functions** (see Lee and Seung, 2001) is applied similar to the one used in the Expectation-Maximization algorithm (see Dempster et al., 1977).

Definition 2.1.3 [Auxiliary function]

A function $G(h, h^{'})$ is called an auxiliary function for $F(h)$ if:

1. $G(h, h) \ = F(h)$, and
2. $G(h, h^{'}) \ \geq F(h)$ for all $h^{'}$.

The following lemma illustrates why the concept of auxiliary function can be useful to minimize $F(h)$ and to find a local minimum.

Lemma 2.1.4 (*Iterative minimization, Lee and Seung, 2001*):
If G is an auxiliary function, then F is nonincreasing under the update

$$h^{t+1} = arg \min_{h} G(h, h^t) \tag{2.10}$$

Proof: $F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$ ■

Only in case of $F(h^{t+1}) = F(h^t)$ a local minimum of $G(h, h^t)$ is obtained at h^t. This implies that the derivatives $\nabla F(h^t) = 0$, if differentiability and local continuity at h^t of F is given. Thus by iterative application of the update in Eq. (2.10) a sequence of estimates that converge to a local minimum $h_{min} = argmin_h F(h)$ of the objective function is constructed:

$$F(h_{min}) \leq \ldots \leq F(h^{t+1}) \leq F(h^t) \leq \ldots \leq F(h_2) \leq F(h_1) \leq F(h_0)$$

With this knowledge the update rules in Theorem 2.1.1 and Theorem 2.1.2 follow from Eq. (2.10) by defining the appropriate auxiliary functions $G(h, h^t)$ for both $\|M - WH\|$ and $D(M\|WH)$.

Lemma 2.1.5 (*Lee and Seung, 2001*):
If $K(h^t)$ is the diagonal matrix

$$K_{ab}(h^t) = \delta_{ab}\frac{(W^TWh^t)_a}{h^t_a}$$

then

$$G(h, h^t) = F(h^t) + (h - h^t)^T \nabla F(h^t) + \frac{1}{2}(h - h^t)^T K(h^t)(h - h^t) \tag{2.11}$$

is an auxiliary function for

$$F(h) = \frac{1}{2}\sum_k (m_k - \sum_l W_{kl}h_l)^2$$

Proof: $G(h, h) = F(h)$ follows by definition of $G(.,.)$, so only the inequality $G(h, h^t) \geq F(h)\ \forall h^t$ needs to be proven (see Lee and Seung, 2001 for a possible proof). ■

With all this preliminary work the proof of the Theorem 2.1.1 can be demonstrated:

Proof (*of Theorem 2.1.1*): Proof from Lee and Seung (2001). Using the auxiliary function, which was defined in the previous Lemma and solving the Eq. (2.10) by setting the gradient to zero leads to the update rule:

$$h^{t+1} = h^t - K(h^t)^{-1}\nabla F(h^t)$$

According to the Iterative minimization Lemma, F is non-increasing under this update rule. It can be obtained

$$h^{t+1}_a = h^t_a \frac{(W^Tm)_a}{(W^TWh^t)_a}$$

by applying the explicit structure of $K(h^t)^{-1}$ and $\nabla F(h^t) = W^T(Wh^t - m)$. Rewritten in matrix form this is equivalent to the update rule in Eq. (2.4). In an analogous way the update rule for W can be shown by reversing the roles of W and H. ■

For the divergence cost function the following auxiliary function need to be considered:

Lemma 2.1.6 (*Lee and Seung, 2001*):
Define

$$G(h,h^t) = \sum_k (m_k \log m_k - m_k) + \sum_{ka} W_{ka} h_a \\ - \sum_{ka} m_k \frac{W_{ka} h_a^t}{\sum_b W_{kb} h_b^t} \left(\log W_{ka} h_a - \log \frac{W_{ka} h_a^t}{\sum_b W_{kb} h_b^t} \right)$$

This is an auxiliary function for

$$F(h) = \sum_k m_k \log \left(\frac{m_k}{\sum_a W_{ka} h_a} \right) - m_k + \sum_a W_{ka} h_a \tag{2.12}$$

Proof: See Lee and Seung (2001) ■

Applying this Lemma and the Iterative minimization Lemma for the proof of Theorem 2.1.2:

Proof (*of Theorem 2.1.2*): Proof from Lee and Seung (2001). Again the minimum of $G(h,h^t)$ with respect to h is obtained by setting the gradient to zero:

$$\frac{dG(h,h^t)}{dh_a} = -\sum_k m_k \frac{W_{ka} h_a^t}{\sum_b W_{kb} h_b^t} \frac{1}{h_a} + \sum_k W_{ka} = 0.$$

Considering that the update rule of Eq. (2.10) takes the form

$$h_a^{t+1} = \frac{h_a^t}{\sum_b W_{ba}} \sum_k \frac{m_k}{\sum_b W_{kb} h_b^t} W_{ka}.$$

Rewritten in matrix form, this is equivalent to the update rule in Eq. (2.5) and as known from the Iterative minimization Lemma, F in Eq. (2.12) is nonincreasing under this update. The update rule for W can similarly be shown to be nonincreasing, by reversing the roles of W and H. ■

Weaknesses and modifications

The statements given in Theorem 2.1.1 and Theorem 2.1.2 to achieve convergence to a local minimum by applying the multiplicative update rules have been proven to be incorrect (see for instance Lin, 2007b). Actually, the proof of Lee and Seung shows only a continuous descending characteristic, which does not exclude the descent to a saddle point. To get a better understanding why this is the case it is enough to consider two basic observations involving the Karush-Kuhn-Tucker optimality conditions. Only the case of the Euclidean distance $f(W,H) = \|M - WH\|$ is discussed, as the divergence (2.3) has very similar issues.

Firstly, if the initial matrices (W, H) are strictly positive these matrices remain positive throughout the iterations due to the multiplicative form of the update rules. Secondly, if for example $H_{lj} > 0$ for all iterations and if this element converges to a limit value of 0 with $[\frac{\partial f}{\partial H}]_{lj} \geq 0$ at (W^*,H^*) then a stationary point (W^*,H^*) with H^*_{lj} can be obtained according to the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\begin{aligned} W \geq 0, \qquad H \geq 0, \\ \frac{\partial f}{\partial H} = 2(WH - M)H^T \geq 0, \qquad \frac{\partial f}{\partial W} = W^T(WH - M) \geq 0, \\ (WH - M)H^T * W = 0 \qquad W^T(WH - M) * H = 0. \end{aligned}$$

In this scenario the corresponding complementary slackness condition $(\frac{\partial f}{\partial H})(W^*, H^*) \geq 0$ must hold. However, it is not obvious how to use the multiplicative update rules to verify this. When (W, H) converge to (W^*, H^*) and $W^* > 0$ and $H^* > 0$, then $(\frac{\partial f}{\partial W})(W^*, H^*) = 0$ and $(\frac{\partial f}{\partial H})(W^*, H^*) = 0$. This can be achieved by using the additive form of the multiplicative updates specified in Eq. 2.6, hence the KKT conditions of optimality are satisfied and (W^*, H^*) turns out to be a stationary point. The rectified statement about the convergence of the Lee and Seung multiplicative update algorithms can be summarized as follows:

When the algorithm has converged to a limit point in the interior of the feasible region, this point is a stationary point. This stationary point may or may not be a local minimum. When the limit point lies on the boundary of the feasible region, its stationarity cannot be determined (Berry et al., 2007).

It has to be noted that in comparison to the newer algorithms the multiplicative algorithms are outperformed, since even if they converge (which occurs often in practice) it has been repeatedly shown that the convergence is notoriously slow. They require many more iterations than alternatives like ALS algorithms (see Subsection 2.1.2) and geometrical algorithms (see Subsection 2.2.3) by having high computational costs of $O(pnr)$ per iteration. With clever implementations of the occurring matrix multiplications this can be decreased a bit.

A number of modifications to the original Lee and Seung algorithms have been introduced with the objective to overcome these shortcomings. For instance, Lin (see Lin, 2007a) proposed a modification that is guaranteed to converge to a stationary point, however, this algorithm requires more work per iteration than the already slow Lee and Seung algorithm.

A simple modification from **Brunet** (see Brunet et al., 2004, for more details) is based on the multiplicative updates for the Kullback-Leibler divergence (KL) (as in (2.5)) with an additional stabilisation step to shift up all entries from zero every 10 iterations, to a very small positive value ϵ. This approach shows to be appealing as it avoids the so-called *locking phenomenon* , i.e. in the case of an element becoming 0 (or is 0 from the start) during the iterative process of the classical multiplicative update algorithm it remains 0.

2.1.2 ALS Algorithms

The Alternating Least Squares - **ALS** algorithms were first introduced by Paatero (see Paatero and Tapper, 1994) , who initially invented the whole NMF theory.

NMF algorithms of the ALS class can be further separated into two groups, one group that solves the **NNLS** problems exactly and the other group which solves the **NNLS** approximately by computing the solution of the ordinary least square (**LS**) and then afterwards applies a function to enforce non-negativity to the LS solution.

The first group typically uses the fast non-negativity constrained least squares (NNLS) algorithm (see Van Benthem and Keenan, 2004), which is improved upon by the active set based NNLS method. As an example of this group, the sparse NMF (SNMFL/L or R) algorithm (see Kim and Park, 2007) will be introduced later on.

The first algorithm that is considered as the basic ALS algorithm of the second group is shown:

Algorithm 1 Basic ALS for NMF

```
Initialize $W$ by a initialization method (see Section 3.1).
for $j \leq$ maxiter do
    Solve for $H$ the LS equation $W^TWH = W^TM$.
    Set all negative elements in $H$ to 0.
    Solve for $W$ the LS equation $HH^TW^T = HM^T$.
    Set all negative elements in $W$ to 0.
end for
Output $W$ and $H$
```

The projection step was implemented as a simple method to insure non-negativity. It sets all negative elements resulting from the least squares computation to 0. Even though it's simple, there are a few benefits as it aids sparsity and it avoids the *locking phenomena*, which is a major drawback of most multiplicative algorithms. Since locking of elements to 0 is restrictive, i.e. as soon as the algorithm starts to go down a path to a fixed point, regardless of whether it is a bad fixed point or not, it must continue in this direction.

In the following subsections two modifications (see Langville et al., 2014), which are adding sparsity restrictions to the NMF problem, are described.

ACLS

The difference between the Alternating Constrained Least Squares - **ACLS** algorithm and the basic ALS algorithm is the objective function that must be solved at each alternating step:

$$\min_{h_j \geq 0, \lambda_H \geq 0} \left\| m_j - W h_j \right\|^2 + \lambda_H \left\| h_j \right\|^2 , \tag{2.13}$$

where m_j and h_j are columns of M and H, respectively. Due to the lower computational costs it was suggested to solve these non-negative constrained least squares problems approximately, such as for the basic ALS, by first running a standard (unconstrained) least squares step and then set all negative elements of the LS solution to 0. The ACLS algortihm is shown below.

Algorithm 2 ACLS for NMF

```
1: Input from user $\lambda_W$, $\lambda_H$
2: Initialize $W$ by a initialization method (see Section 3.1).
3: for $j \leq$ maxiter do
4:     Solve for $H$ the CLS equation $(W^T W + \lambda_H I)H = W^T M$.
5:     Set all negative elements in $H$ to 0.
6:     Solve for $W$ the CLS equation $(HH^T + \lambda_W I)W^T = HM^T$.
7:     Set all negative elements in $W$ to 0.
8: end for
9: Output $W$ and $H$
```

The two sparsity parameters λ_H and λ_W have to be set by the user. By increasing the values the sparsity of the two NMF factors is increased, but as no upper-bounds on these parameters are given, the best values for λ_H and λ_W have to be obtained by trial and error. As a consequence, the application of ACLS is more challenging.

AHCLS

The more advanced Alternating Hoyer-Constrained Least Squares - **AHCLS** provides better sparsity parameters with more intuitive bounds. AHCLS replaces the crude measure $\|x\|^2$ to approximate the sparsity of a vector x by a more sophisticated measure, $spar(x)$, invented by Hoyer (see Hoyer, 2004). This measure is based on the relationship between the L_1 norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ and the Euclidean norm:

$$spar(x) = \frac{\sqrt{n} - \|x\|_1 \,/\, \|x\|}{\sqrt{n} - 1} \quad \text{for} \quad x \in \mathbb{R}^n$$

In case of x containing only a single non-zero component this function evaluates at 1, and takes a value of zero if and only if all components are equal (up to signs). Otherwise it is interpolating smoothly between these two extremes.

For AHCLS the user needs to define two scalars α_W and α_H in addition to λ_H and λ_W of ACLS. The two additional scalars $0 \leq \alpha_W, \alpha_H \leq 1$ represent a user's desired sparsity in each column of the factors. As they range from 0 to 1 they match nicely with the ordinary interpretation of sparsity as a percentage. The positive parameters λ_W, λ_H measure how important it is to the user that $spar(W_{j*}) = \alpha_W$ and $spar(H_{j*}) = \alpha_H$. As a guideline it was recommended in Langville et al. (2014) to use $0 \leq \lambda_W, \lambda_H \leq 1$.
In the following the AHCLS algorithm as proposed in Langville et al. (2014) is shown, where E is the matrix with all elements equal to one.

Algorithm 3 AHCLS for NMF

```
Input from user $\lambda_W$, $\lambda_H$, $\alpha_W$, $\alpha_H$
Initialize $W$ by a initialization method (see Section 3.1).
Define $\beta_H = ((1-\alpha_H)\sqrt{r}+\alpha_H)^2$ and $\beta_W = ((1-\alpha_W)\sqrt{r}+\alpha_W)^2$
for $j \leq$ maxiter do
    Solve for $H$ the CLS equation $(W^T W + \lambda_H \beta_H I - \lambda_H E)H = W^T M$.
    Set all negative elements in $H$ to 0.
    Solve for $W$ the CLS equation $(HH^T + \lambda_W \beta_W I - \lambda_W E)W^T = HM^T$.
    Set all negative elements in $W$ to 0.
end for
Output $W$ and $H$
```

According to the experiments from Langville et al. (2014), AHCLS enforces sparsity better than ACLS and moreover the four AHCLS parameters are easier to set.

Properties of ACLS and AHCLS

The ACLS and AHCLS are very fast NMF algorithms, even faster than current truncated SVD algorithms due to the fact that each CLS step solves only a small $r \times r$ matrix system. Furthermore, they converge quickly and give very accurate NMF factors as was shown in Langville et al. (2014).

In general, the convergence of ALS algorithms to a fixed point is proven, but this fixed point may be a local extremum or a saddle point (see Langville et al., 2014). Furthermore, it is known that the ACLS and AHCLS algorithms with properly enforced non-negativity, for example by the NNLS algorithm (as used for SNMF/L or SNMF/R), convergence to a local minimum (see Lin, 2007b). Nonetheless, it is practical to use the ad-hoc enforcement of non-negativity (setting all negative values to 0) as it speeds up the algorithm and improves sparsity.

Theoretically, the ad-hoc enforcement of non-negativity is unattractive, hence some alternatives should be considered. For instance, the alternating least squares approach could be converted to an alternating linear programming approach, but this has the same problem of lengthy execution time as the NNLS algorithm. The improvement to incorporate sparsity for both NMF factors by the user leads in many applications to better interpretability of the results and reduction of storage.

SNMF/L or R with NNLS algorithm

These two algorithms adapt the objective function of the standard NMF problem (1.1) to enforce either sparseness on either W or H. The sparse NMF - **SNMF/L** algorithm is applied to impose sparsity on W (where 'L' denotes the sparsity imposed on the left factor). In contrast to this, **SNMF/R** is applied to impose sparsity on H (where 'R' denotes the sparsity imposed on the right factor). To enforce sparsity on a factor of NMF, the L_1-norm minimization is utilized. The usage of the L_1-norm instead of the Euclidean norm as measure of sparseness is motivated by the fact that a quadratic penalty corresponds to Gaussian priors and does not encourage sparsity, but rather scales the result and gives non-sparse low values.

Formulation of the SNMF/R optimization problem:

$$\min_{W,H\geq 0} \frac{1}{2}\{\|M - WH\|^2 + \eta \, \|W\|^2 + \beta \sum_{j=1}^{n} \|h_j\|_1^2\}, \tag{2.14}$$

where h_j is the j-th column vector of H, $\eta > 0$ is a parameter to suppress $\|W\|^2$ and $\beta > 0$ is a regularization parameter to balance the trade-off between the the sparseness of H and the accuracy of the approximation. The two input parameters η and β have to be chosen by trial and error and are dependent on the application. The SNMF/R algorithm is shown below, where $e_{1\times r}$ is a row vector with all components equal to one, $0_{1\times n}$ is a zero vector, I_r is an identity matrix of size $r \times r$ and $0_{r\times p}$ is a zero matrix of size $r \times p$:

Algorithm 4 SNMF/R for NMF

```
Input from user $\eta$, $\beta$
Initialize $W$ by a initialization method(see Section 3.1).
for $j \leq$ maxiter do
    Solve for $H$ by running the NNLS algorithm on
        $\min_{H\geq 0} \| \begin{pmatrix} W \\ \sqrt{\beta} e_{1\times r} \end{pmatrix} H - \begin{pmatrix} M \\ 0_{1\times n} \end{pmatrix} \|^2$
    Solve for $W$ by running the NNLS algorithm on
        $\min_{W\geq 0} \| \begin{pmatrix} H^T \\ \sqrt{\eta} I_r \end{pmatrix} W^T - \begin{pmatrix} M^T \\ 0_{r\times p} \end{pmatrix} \|^2$
end for
Output $W$ and $H$
```

The SNMF/L algorithm optimizes the following objective function in order to impose sparseness on W:

$$\min_{W,H\geq 0} \frac{1}{2}\{\|M - WH\|^2 + \eta \|H\|^2 + \alpha \sum_{i=1}^{p} \|w_i\|_1^2\}, \tag{2.15}$$

where w_i is the i-th row vector of W, $\eta > 0$ is a parameter to suppress $\|H\|^2$ and $\alpha > 0$ is a regularization parameter to balance the trade-off between the accuracy of the approximation and the sparseness of W. The algorithm for SNMF/L is very similar to SNMF/R and it is therefore not stated explicitly.

The convergence properties of SNMF/L and SNMF/R are essentially the same, which is why it is enough to examine the properties of either SNMF/L or SNMF/R. Here, the case of SNMF/R is discussed shortly. In order to clarify that the objective function (2.14) is differentiable with respect to W or H, it can be rewritten as

$$\min_{W,H\geq 0} \frac{1}{2}\{\|M - WH\|^2 + \eta \|W\|^2 + \beta \sum_{j=1}^{n} (\sum_{q=1}^{r} h_{qj})^2\}, \tag{2.16}$$

if $\eta > 0$, $\beta > 0$, and $h_{qj} = |h_{qj}|$, which is the case for $h_{qj} \geq 0$. The differentiability and the existence of accumulation points (see Kim and Park, 2007) imply that the assumptions of Grippo and Sciandrone's Corollary (see Grippo and Sciandrone, 2000), which showed that the two-block coordinate method does not require each sub-problem to have an unique solution for convergence, are satisfied. Therefore, it can be stated that the algorithm converges to a critical point of the problem shown in (2.16).

As for other ALS algorithms, the convergence speed is superior to that of multiplicative update rule based algorithms. The computational costs per iteration have been the major weakness of the ALS algorithms, which solved exactly the NNLS sub-problems, but by further improving the NNLS algorithm the computational costs are acceptable.

2.2 Separable NMF

For separable NMF, the standard NMF problem was restricted by only considering a subclass of matrices, the so called *separable* matrices. The paper Arora et al. (2012) showed that this leads to an easier problem than that of the standard NMF. In this section, an overview of the theoretical background and algorithms to solve separable NMFs is given.

2.2.1 Separability Assumption

It is first necessary to introduce the non-negative rank of a matrix M before defining the separability property of matrices. The non-negative rank of M is defined as:

Definition 2.2.1 [non-negative rank (Gillis, 2017)]
The non-negative rank, $rank_+(M)$, of $M \in \mathbb{R}_+^{p\times n}$ is the minimum r such that an "exact NMF" exists.

The "exact NMF" problem refers to the problem of finding an exact factorization, i.e. finding $W \geq 0$ and $H \geq 0$ such that $M = WH$. In general, it can be stated that $rank(M) \leq rank_+(M) \leq \min(p, n)$, since $M = MI = IM$ (I is the identity matrix).

Then the separable matrices can be defined as follows.

Definition 2.2.2 [separable matrix (Gillis, 2017)]
A matrix M is separable if there exists a subset $\mathcal{K} \subseteq \{1, \ldots, n\}$ of cardinality r with $r = rank_+(M)$ and a non-negative matrix H such that $M = M(:, \mathcal{K})H$, where $M(:, \mathcal{K})$ are the columns of M whose column index is included in the subset $\mathcal{K}$.

Therefore, separability requires each column of the basis matrix W in an NMF decomposition to be present in the input matrix M, and the weight matrix H must contain the identity matrix as a sub-matrix. In conclusion, the separable NMF problem can be reduced to the problem of identifying the subset $\mathcal{K}$. If noise is considered (as obtained in almost every measurement or observation) on the input matrix M, the subset $\mathcal{K}$ is obtained by minimizing

$$\min_{H \geq 0} \|M - M(:, \mathcal{K})H\| . \tag{2.17}$$

Even though this condition/assumption is not easily met, it is advantageous to satisfy it in some situations as shown in the following examples:

- Blind hyperspectral unmixing: For each endmember (basis vector), a pixel exists which contains only that endmember. These pixels are called "pure" pixels and this assumption has been used since the 1990s in that community known as pure-pixel assumption. Several algorithms in this context are based on this assumption.
- Document classification: for each topic, the existence of a "pure" (or "anchor") word used only by that topic is required (see Arora et al., 2013)
- Time-resolved Raman spectra analysis: for each substance only a peak in its spectrum exists, while the other spectra are (close to) zero (see Luce et al., 2016) .

The separability assumption (after normalization of M and W) can be reformulated by the geometric interpretation of the "exact NMF" given in the next section.

2.2.2 Geometry of Exact NMF

The geometric interpretation of the "exact NMF" can be used to develop efficient algorithms as will be discussed in Subsection 2.2.3.

In case of the "exact NMF" the following two preprocessing steps can be made without loss of generality:

- Remove the zero columns of M and W.
- Normalize the entries of M and W so that the entries of each column sum up to one:
  $$MD_M^{-1} = WD_W^{-1} D_W H D_M^{-1},$$
  where D_M and D_W are diagonal matrices with $D_M(j,j) = \|m_j\|_1$ and $D_W(j,j) = \|w_j\|_1$, respectively. This normalization implies also the normalization of H ($\|h_j\|_1$ for all j) due to the "exact NMF" equation $m_j = \sum_{k=1}^{r} w_k h_{kj} = Wh_j$.

After this normalization the columns of M belong to the convex hull of the columns of W:

$$m_j \in conv(W) \subseteq \triangle^p = \{x \in \mathbb{R}^p | x \geq 0, \|x\|_1 = 1\} \quad \forall j, \tag{2.18}$$

where $conv(W) = \{Wx | x \geq 0, \|x\|_1 = 1\}$.

Consequently, the "exact NMF" problem is equivalent to finding a polytope, $conv(W)$, nested between two given polytopes, $conv(M)$ and the unit simplex $\triangle^p$ (Gillis, 2017). The inner polytope, $conv(M)$, has a dimension of $rank(M) - 1$, while the dimension of the outer polytope, $\triangle^p$, is $p - 1$. The dimension of the nested polytope $conv(W)$ is unknown in advance. If the three polytopes (inner, nested, and outer) have the same dimension, this problem is referred to as the nested polytope problem - **NPP** (see Das and Joseph, 1990), which is well known in computational geometry.

If $rank(M) = rank(W)$ is imposed explicitly as an additional constraint on the exact NMF problem, it can be proven that NPP is equivalent to this restricted variant of exact NMF (see Gillis, 2017).

From a geometric point of view, the separability assumption is equivalent to $conv(W) = conv(M)$ considering the "exact NMF" with the proposed preprocessing steps.

2.2.3 Algorithms

Due to the geometrical interpretation, the separable NMF problem reduces to identifying the vertices of the convex hull of the columns of M, what is considered a relatively easy geometric problem. If noise is added to the separable matrix the estimation becomes more challenging.

Many of the algorithms for separable NMF have been developed within the blind hyperspectral unmixing community (sometimes referred to as pure-pixel search algorithms) and are based on the geometrical interpretation.

Geometric algorithms

Most of the geometric algorithms are not robust to noise, but the fast and robust recursive algorithms introduced by Gillis and Vavasis (2014) proved to handle noise appropriately. Therefore, this approach and its characteristics are described in this section.

The algorithm is based on the fact that over a polytope, a strongly convex function is always maximized at a vertex. Consequently, the column of M maximizing the strongly convex function is selected for W (since we assume that the columns of M are normalized such that $conv(W) = conv(M)$ under the separability assumption). Then M is updated by projecting each column onto the orthogonal complement of the selected column (such that this particular column projects onto 0), this amounts to applying a linear transformation to the polytope. In the case that W has full rank (geometrically meaning the polytope is a simplex, which is usually the case in practice), then the other vertices do not project onto 0, and these two steps can be applied recursively.

The complete algorithm is shown below:

Algorithm 5 Recursive algorithm for separable NMF

```
Input from user: $M$, $r$ and a strongly convex function $f$.
Set $R = M = (m_1, \dots, m_n) = (r_1, \dots, r_n)$ and $J = \{\}$, $j = 1$.
while $R \neq 0$ and $j \leq r$ do
    Set
        $$j^* = \underset{1 \leq l \leq n}{\arg\max} f(r_l). \dagger$$
    Set $w_j = r_{j^*}$.
    Project $R$ to the orthogonal complement of $w_j$: $R \leftarrow \big(I - \frac{w_j w_j^T}{\|w_j\|^2}\big)R$.
    Set $J = J \cup \{j^*\}$.
    Set $j = j + 1$.
end while
Output $W = M(:, J)$.
    † In case of a tie, pick the index $j$ such that the corresponding column of the
original matrix $M$ maximizes $f$. In case of another tie, pick one of these columns
randomly.
```

The algorithm requires the following two additional assumptions to obtain a suitable factorization:

- The separable matrix $M \in \mathbb{R}^{p \times n}$ can be written as $M = WH = W[I_r, H^{'}]$, where $W \in \mathbb{R}^{p \times r}$ has rank r, $H \in \mathbb{R}_+^{r \times n}$, and $\left\| h_j^{'} \right\|_1 \leq 1 \ \forall j$.
- The function $f : \mathbb{R}^p \to \mathbb{R}_+$ is strongly convex with parameter $\mu > 0$, its gradient is Lipschitz continuous with constant L, and its global minimizer is the all-zero vector with $f(0) = 0$.

The first assumption can be made without loss of generality by permuting the columns of M, and by a specific normalization scheme of M it can be achieved that $\left\| h_j^{'} \right\|_1 \leq 1 \ \forall j$. It was verified that $f(x) = \|x\|^2$ is the optimal choice to reduce the error bounds if noise is added to M (see Gillis and Vavasis, 2014).

Under these assumption the algorithm recovers a set of indices J such that $M(: J) = W$ up to a permutation. If noise N is added to the noiseless separable matrix M, then for $M^{'} = M + N$ still the endmembers (basis vectors) can be extracted up to an error bound (see for details Gillis and Vavasis, 2014). Furthermore, the robustness of noise can be improved by using strategies such as:

- applying dimensionality reduction, such as PCA, to the columns of M in order to filter the noise
- perform a precondition based on minimum-volume ellipsoid
- check whether the r identified vertices still maximize $f(\cdot)$ once projected onto the orthogonal complement of the other vertices and

- considering the non-negativity constraints in the projection step.

The proposed algorithm has some appealing properties. For example, the computational costs of the algorithm are very low since the algorithm is linear in n. Another advantage is the simple implementation and that no parameter needs to be chosen a priori nor has to be tuned. Even if the input matrix M is not approximately separable, r columns of M are identified whose convex hull has large volume. This is a key advantage over other separable NMF algorithms. In general, geometric algorithms are sensitive to outliers and for this reason, Gillis proposed a simple post-processing strategy to identify outliers (see Gillis and Vavasis, 2014).

Convex models

In contrary to the general NMF problem it is possible to construct convex models to solve the separable NMF problem. The separability of M is equivalent to the existence of an $n \times n$ non-negative matrix X with r nonzero rows such that $M = MX$ with $X(\mathcal{K}, :) = H$. This consideration leads to the following formulation of the separable NMF problem

$$\min_{X \geq 0} \|X\|_{row,0} \quad \text{s.t. } M = MX,$$

where $\|X\|_{row,0}$ counts the number of nonzero rows of X. In order to achieve a convex model a standard approach is to use the ℓ_1 norm instead of $\|X\|_{row,0}$. Other typical approaches are $\sum_{i=1}^{n} \|X(i,:)\|_k$ for some k, for example $k = \infty$ and $k = 2$.

Consider the case of M being normalized, then the entries of H are bounded above by one, since the columns of W are vertices. Hence, another formulation for separable NMF can be obtained:

$$\min_{X \geq 0} \|diag(X)\|_0 \quad \text{s. t. } M = MX \text{ and } x_{ij} \leq x_{ii} \leq 1 \quad \forall i, j. \tag{2.19}$$

The conditions on X are ensuring that the diagonal entry has to be the largest on each row and since the goal is to minimize the number of nonzero entries of the diagonal of X, the optimal solution will contain r nonzero diagonal entries which implies r nonzero rows. Again by replacing the $\| \cdot |_0$ with the ℓ_1 norm a convex model is obtained (see Recht et al., 2012):

$$\min_{X \geq 0} trace(X) \quad \text{s. t. } M = MX \text{ and } x_{ij} \leq x_{ii} \leq 1 \quad \forall i, j, \tag{2.20}$$

where $trace(X)$ is equal to $\|diag(X)\|_1$, since X is non-negative. The noiseless case never occurs in practice, therefore it should be considered the preferable approach as to how these models need to be modified. The modification can be done by either replacing the equality term $M = MX$ with $\|M - MX\| \leq \epsilon$ for some appropriate norm (typically the ℓ_1, ℓ_2, or Frobenius norm) or by adding the equality term in the objective function as penalty. It was determined that the two convex models are essentially equivalent (see Gillis and Luce, 2018).

The computational costs of these models are very high as n^2 variables need to be optimized. As a consequence, their utilization in hyperspectral imaging for example, where n is referred to the number of pixels and is typically on the order of millions, is impractical.

3 Challenges Arising from NMF

This chapter provides an overview of the main aspects that need to be considered for the application of NMF methods. Moreover, a collection of quality measures is described to appropriately compare the wide range of NMF algorithms.

3.1 Initialization Methods

Many NMF algorithms, especially all which are applying the two-block coordinate descent scheme, need an initialization of the basis matrix W and the coordinate matrix H, but in this case only W has to be initialized as the initial H can be estimated by solving the **NNLS** sub-problem. The sensitivity to the initialization of W and H is a well-known fact (see Wild, 2004), which explains the intense research interest in finding a practical seeding method.

In general, two possibilities exist to assess a "good initialization strategy": (i) one that leads to rapid error reduction and faster convergence; (ii) one that leads to better overall error at convergence. Most of the initialization strategies focus on the first objective, since the second is more challenging as NMF algorithms typically converge to a local minimum. In this section, a number of seeding methods that have been considered beneficial are depicted. The initialization methods can be divided into two groups: (i) based on randomly chosen variables; (ii) based on deterministic strategies. These more sophisticated deterministic methods render replication of the NMF algorithm obsolete. Especially if dealing with big data loads, this feature turns out to be very helpful, since running an NMF algorithm can be time-consuming.

The strategies outlined in this section are intended to give a good overview on the wide range of initialization methods. Nevertheless, there are notable strategies, which do not fit into the framework of the described methods such as the strategy introduced by Janecek and Tan (2011), which proposes the use of population based algorithms - **PBA**.

3.1.1 Random Initialization Strategies

A simple method is given by drawing the entries of each factor from a uniform distribution over $[0, max(M)]$, where M is the input matrix (whose factorization is requested). The generated initial matrices of W and H are dense and take much storage ,even if M is sparse, since only the maximum is considered in this initialization strategy. In order to ensure reasonable results, the NMF algorithm has to be repeated around 1000 times, since different initial matrices could converge to different local minima. For further analyses only the factorization which performed best, is used.

Two other random strategies suggested by Langville et al. (2014) create sparse initials out of the given data. The first one, **random Acol**, creates each column of W by averaging p randomly chosen columns of M, which tends to maintain sparsity of the input matrix M. This proposed method is computationally very inexpensive and lies between pure random initialization and spherical cluster-based initialization (see Subsection 3.1.4) in terms of performance (Langville et al., 2014).

In contrary to random Acol, the second strategy, **random C**, restricts the choice to the q (this parameter has to be set) longest (in term of the Frobenius norm) columns of M instead of taking all columns in consideration. The longest columns are the densest columns, if the matrices considered are sparse, it would be more likely that these columns are centroids. Similar to Acol it is computationally inexpensive, but is considered to be not very effective according to Langville et al. (2014).

3.1.2 Nonnegative Double Singular Value Decomposition

The initialization method **Nonnegative Double Singular Value Decomposition (NNDSVD)** proposed by Boutsidis and Gallopoulos (2008) is based on **singular value decomposition (SVD)**. More precisely, two SVD processes are used: While the first one creates the rank-k approximation, the second one is considered as "small" SVD on each of the positive sections of the factors. The behaviour of unit rank matrices appears to be essential for the construction of the initial matrices W and H and will be explained in detail. The following definitions are required for the upcoming theorems:

Definition 3.1.1 [Positive and negative section]
Given any vector or matrix variable $X \in \mathbb{R}^{p \times n}$, the *positive section* $X_+ \geq 0$ of X is defined as:

$$X_+ = \left[1_{[x_{kl} \geq 0]} x_{kl}\right]_{k=1,\dots,p;\ l=1,\dots,n}$$

The *negative section* $X_- \geq 0$ of X will be the matrix

$$X_- = X_+ - X$$

By the use of this definition any vector or matrix can be written as $X = X_+ - X_-$ and particularly if $X \geq 0$ then the negative section $X_- = 0$.

Strategy

The *NNDSVD* method starts with the calculation of the SVD and takes advantage of SVD specific properties. Using SVD, every matrix $M \in \mathbb{R}^{p \times n}$ of rank $r \leq \min(p, n)$ can be expressed as the sum of r leading singular factors $M = \sum_{l=1}^{r} \sigma_l u_l v_l^T$, where $\sigma_1 \geq \ldots \geq \sigma_r > 0$ are the non-zero singular values of M and $\{u_l, v_l\}_{l=1}^{r}$ the corresponding left and right singular vectors. Then, considering the **Schmidt and Eckart-Young theorem** (see Stewart and Sun, 1990), for every $k \leq r$, the optimal rank-k approximation of M with respect to the Frobenius norm, say $M^{(k)}$, can be constructed out of the first k singular factors and their singular vectors as

$$M^{(k)} \overset{\text{def}}{=} \sum_{l=1}^{k} \sigma_l C^{(l)} = arg \min_{rank(G) \leq k} \|M - G\| \, , \tag{3.1}$$

where $C^{(l)} = u_l v_l^T$. For NMF the matrices M and $M^{(k)}$ need to be non-negative, but the singular vectors in general will have negative entries. Hence, a modification of (3.1) was developed that will produce a non-negative approximation of M. Approximating every unit rank matrix $C^{(l)}$ by its positive section $C_+^{(l)}$ leads to a non-negative approximation of M. These positive sections $C_+^{(l)}$ are possessing favourable characteristics that play a key role in the NNDSVD algorithm. In the following their properties will be shown (Boutsidis and Gallopoulos, 2008):

- Their rank is at most 2 because of the "set to zero with small rank increment" property (see Lemma 3.1.2)
- They are the best nonnegative approximations of $C^{(l)}$ in terms of the Frobenius norm (see Lemma 3.1.6)
- Corresponding singular vectors exist, which are non-negative and readily available from the singular triplets $\{\sigma_l, u_l, v_l\}$ of M (see Theorem 3.1.3).

These properties were proven by Boutsidis and Gallopoulos (2008).

Lemma 3.1.2 (*set to zero with small increment*):
Consider any matrix $C \in \mathbb{R}^{p \times n}$ such that rank(C) = 1, and write $C = C_+ - C_-$. Then rank(C_+), rank(C_-)≤ 2.

Proof: Since C has rank 1, it can be written as $C = xy^T = (x_+ - x_-)(y_+ - y_-)^T = (x_+ y_+^T + x_- y_-^T) - (x_+ y_-^T + x_- y_+^T)$. All four factors are non-negative and for each x, y the non-zero values of the positive section and the corresponding negative section are situated at locations that are complementary to one another. Hence, each non-zero element of C is obtained from exactly one term from the terms on the right, which leads to $C_+ = x_+ y_+^T + x_- y_-^T$ and $C_- = x_+ y_-^T + x_- y_+^T$ with the rank of each one being at most 2. ■

This property is called "set to zero with small rank increment" as it states that if all negative values of a unit rank matrix were set to zero, the resulting matrix will have rank 2 at most. A similar property for matrices of $rank(C) > 1$ cannot be achieved (see for an example, Boutsidis and Gallopoulos, 2008, p. 1353).

The Perron-Frobenius theory (see Catral et al., 2004) ensures the maximum left and right singular vectors of C_+ to be non-negative due to the non-negativity of C_+. In addition, the specific structure of C_+ guarantees even the remaining singular vectors to be non-negative as the following theorem shows.

Theorem 3.1.3
Let $C \in \mathbb{R}^{p \times n}$ have unit rank, so that $C = xy^T$ for some $x \in \mathbb{R}^p$, $y \in \mathbb{R}^n$. Define $\hat{x}_\pm \overset{\text{def}}{=} \frac{x_\pm}{\|x_\pm\|}$, $\hat{y}_\pm \overset{\text{def}}{=} \frac{y_\pm}{\|y_\pm\|}$ as the normalized positive and negative sections of x and y. Let also $\mu_\pm = \|x_\pm\| \, \|y_\pm\|$ and $\xi_\pm = \|x_\pm\| \, \|y_\mp\|$. Then the unordered singular value expansions of C_+ and C_- are

$$C_+ = \mu_+ \hat{x}_+ \hat{y}_+^T + \mu_- \hat{x}_- \hat{y}_-^T \quad \text{and} \tag{3.2}$$

$$C_- = \xi_+ \hat{x}_+ \hat{y}_-^T + \xi_- \hat{x}_- \hat{y}_+^T \tag{3.3}$$

The maximum singular triplet of C_+ is $(\mu_+, \hat{x}_+, \hat{y}_+)$ if $\mu_+ = \max(\|x_+\| \, \|y_+\|, \|x_-\| \, \|y_-\|)$, otherwise it is $(\mu_-, \hat{x}_-, \hat{y}_-)$. Similarly, the maximum singular triplet of C_- is $(\xi_+, \hat{x}_+, \hat{y}_-)$ if $\xi_+ = \max(\|x_+\| \, \|y_-\|, \|x_-\| \, \|y_+\|)$, otherwise it is $(\xi_-, \hat{x}_-, \hat{y}_+)$.

Proof: see Boutsidis and Gallopoulos (2008). ■

A direct connection to the concept of non-negative rank can be derived for this decomposition.

Definition 3.1.4 [non-negative rank Gregory and Pullman, 1983]
The non-negative rank, $rank_+(A)$, of $A \in \mathbb{R}_+^{p \times n}$ is the smallest number of non-negative unit rank matrices into which a matrix can be decomposed additively.

This alternative definition of the non-negative rank is equivalent to the definition given in Subsection 2.2.1. Combining the actually shown properties of $C_\pm$ with the ones known from previous sections, $rank(A) \leq rank_+(A) \leq \min(p, n)$ and if $rank(A) \leq 2$, then $rank_+(A) = rank(A)$, precise estimates regarding the nonnegative ranks of $C_\pm$ can be stated.

Corrolary 3.1.5
Consider the matrices $C_{\pm}$ from Theorem 3.1.3

1. $rank_{+}(C_{\pm}) \leq 2$.
2. $rank_{+}(C_{\pm}) = rank(C_{\pm})$.
3. If C contains both positive and negative elements, then $rank_{+}(C_{\pm}) = 2$.
4. If $C \geq 0$ (resp. $C \leq 0$) then $rank_{+}(C_{+}) = 1$ (resp. $rank_{+}(C_{-}) = 1$).

An explicit construction for the decomposition of C_{+} as well as a computationally inexpensive way to perform it is given in Theorem 3.1.3. The next lemma can be formulated as a direct consequence of the Frobenius norm definition.

Lemma 3.1.6
Let $C \in \mathbb{R}^{p \times n}$. Then $C_{+} = \arg\min_{G \in \mathbb{R}_{+}^{p \times n}} \|C - G\|$.

Considering this Lemma, the best (in terms of the Forbenius norm) nonnegative approximation of each unit rank term $C^{(l)} = u^{(l)}(v^{(l)})^{T}$ would be the corresponding $C_{+}^{(j)}$.

The necessary steps for the NNDSVD initialization can be summarized as follows:

1. Compute the k leading singular triplets of M
2. Extract the positive sections of the unit rank singular factor expansion of M as suggested in Theorem 3.1.3
3. Approximate each of these factors by its maximum singular triplet
4. Initialize (W, H) with the scalar multiples ($\sqrt{\sigma_j}$) of the factors from Step 3

Since the leading singular triplet is non-negative, it can be readily used to initialize the first column (respectively row) of W (respectively H) and Step 3 is applied from $j = 2$ onwards.

Error bound and variants

As the primary objective is to minimize the Frobenius norm of the residual $R = M - WH$, it is desirable to provide a bound for this error corresponding to the initial factors (W, H) obtained by *NNDSVD*. In the next proposition (Boutsidis and Gallopoulos, 2008) bounds can be determined with the use of the preceding results. Therefore following notations $\{\sigma_j\}_{j=1}^{r}$ for the nonzero singular values of M in non-increasing order and $\{\sigma_j(C_{+}), x_j(C_{+}), y_j(C_{+})\}$ for the singular triplets of C_{+} are used.

Proposition 3.1.7

Given $M = M^{(k)} + E^{(k)} \in \mathbb{R}_+^{p\times n}$ with $E^{(k)} \overset{\text{def}}{=} \sum_{j=k+1}^{r} \sigma_j u_j v_j^T$, and the pair (W, H) initialized by *NNDSVD*, then the Frobenius norm of $R = A - WH = E^{(k)} + \hat{E}$ with $\hat{E} \overset{\text{def}}{=} \sum_{j=2}^{k} \sigma_j \sigma_2(C_+^{(j)}) x_2(C_+^{(j)})(y_2(C_+^{(j)}))^T - \sum_{j=2}^{k} \sigma_j C_-^{(j)}$ is bounded as follows:

$$\|E^{(k)}\| \le \|R\| \le \|E^{(k)}\| + \|\hat{E}\|, \tag{3.4}$$

where

$$\|\hat{E}\| \le \sum_{j=2}^{k} (\sigma_2(C_+^{(j)}) + 1)\sigma_j \le 2\sum_{j=2}^{k} \sigma_j. \tag{3.5}$$

Proof: These inequalities can be derived from the properties of the singular vectors $x_1(C_+), y_1(C_+)$, by the application of the triangle inequality and the fact that $\left\|C_\pm^{(j)}\right\| \le 1$. For further details, see Boutsidis and Gallopoulos (2008). ■

For modified versions of *NNDSVD*, which generate an initial pair (W_f, H_f), where $W_f \overset{\text{def}}{=} W + E_W$, $H_f \overset{\text{def}}{=} H + E_H$, (W, H) from the noiseless case and E_W, E_H are structured perturbations such that their non-zero elements occur at positions complementary to those of W and H, respectively, and their Forbenius norm being small $\max(\|E_H\|, \|E_W\|) \le \epsilon$, the error can be bounded by applying the aforementioned proposition.

$$\|A - W_f H_f\| \le \|R\| + \epsilon(\|W\| + \|H\|) = \|R\| + 2\epsilon\sqrt{k}. \tag{3.6}$$

Except for the maximum singular vectors, each of the first k singular vectors probably contains both positive and negative elements, thus the positive sections of them (which are used for the initialization) are likely to contain a number of zeros. This effect would be preferred in some cases, e.g. if sparseness is requested, particularly as some **NMF** algorithms retain the same sparsity in the iterates that was present in the initial (W, H).

In case of dense matrices, a large number of zeros may become undesirable, as results from Boutsidis and Gallopoulos (2008) suggest that the basic *NNDSVD* algorithm provides rapid error reductions, but eventually leads to worse errors than *RANDOM* initialization. Therefore, modified versions of *NNDSVD* were developed that rely on structured perturbations as mentioned in the previous paragraph.

In particular for the variant *NNDSVDa* the zero values in the original (W, H) are perturbed by setting all zeros equal to the average of all elements of M (denoted as $mean(M)$). For variant *NNDSVDar* each zero element is replaced by a random value, which is drawn from an uniform distribution in $[0, \frac{mean(M)}{100}]$. These variants need no additional calculation for the basic initialization and lead to error bounds such as in Eq. (3.6).

When the computation of all leading k singular triplets of M turns out to be difficult or expensive, Boutsidis and Gallopoulos (2008) recommend a useful extension of *NNDSVD*, which uses the fact that not only the maximum but also the trailing singular triplet, $\{\sigma_2(C_+), x_2(C_+), y_2(C_+)\}$, has strictly non-negative components. This extension is called *2-step NNDSVD* and modifies *NNDSVD* as follows:

- For $j = 2, \ldots$ until all k columns and rows of (W, H) are filled
  - $\underline{\text{if } rank(C_+(j)) = 1:}$
    Initialize column j of W and row j of H with scalar multiples of the maximum left and right singular vectors of $C_+^{(j)}$ as it is done in the original algorithm.
  - $\underline{\text{if } rank(C_+(j)) = 2:}$
    Initialize the columns and rows $2j, 2j+1$ of W and H with scalar multiples of $x_1(C_+^{(j)}), x_2(C_+^{(j)})$ and $y_1(C_+^{(j)})^T, y_2(C_+^{(j)})^T$.

If k is odd and all $C^{(2)}, \ldots, C^{(k+1)/2}$ have rank-2, then these factors are sufficient to produce a non-negative initialization for (W, H). The reconstruction of $C_+^{(j)}$ is exact, as all singular vectors' which generate $C_+^{(j)}$, participate in the initialization. This leads to a different upper bound for the residual, if *2-step NNDSVD* is applied as initialization.

Corrolary 3.1.8

Given $M \in \mathbb{R}_+^{p \times n}$, and the pair (W, H) initialized as in *2-step NNDSVD*, then

$$\|E^{(k/2)}\| \leq \|R\| \leq \|E^{(k/2)}\| + \sum_{j=2}^{k/2} \sigma_j. \tag{3.7}$$

Computational costs

The two key computational steps of *NNDSVD* are computing the k largest singular triplets and computing the maximum singular triplet of the positive section of each singular factor in the singular expansion of M.

The runtime of *NNDSVD* can be reduced by any algorithmic improvement of these two key steps, whereby for dense M a rough estimate of the cost of the first step is $O(kpn)$. Exploiting the fact of $C_+^{(j)}$ that the singular triplets are readily available (see Theorem 3.1.3) the computational cost is rather low with $O(p+n)$. Therefore, the overall cost for *NNDSVD* on dense data is $O(kpn)$ (Boutsidis and Gallopoulos, 2008).

3.1.3 SVD-NMF

In 2015, Qiao introduced a new possibility to initialize the matrices (W, H), which is called **SVD-NMF** as it is based on the SVD of the target matrix M (see Qiao, 2015). Furthermore, a selection rule to compute the rank r of the factorization was suggested, but this is discussed later in the Subsection 3.3.2. In comparison to *NNDSVD*, the *SVD-NMF* method uses only once the utilization of singular triplet information and does not exploit properties of the unit rank matrices that are generated from the singular vectors. The *SVD-NMF* method has some major benefits as mentioned by Qiao (2015):

- cheap computational cost as the singular triplets are only computed once
- can reach fast convergence
- simple and easy to implement
- can be easily combined with many NMF algorithms

Strategy

First of all, for any given matrix $M \in \mathbb{R}_+^{p\times n}$ the respective SVD can be expressed as follows:

$$M = U\Sigma V^T, \tag{3.8}$$

where U is a $p \times p$ orthogonal matrix, V^T denotes the transposed of the $n \times n$ orthogonal matrix V (V^T orthogonal $\Leftrightarrow$ V orthogonal) and Σ is a $p \times n$ diagonal matrix with this specific structure

$$\Sigma = \begin{pmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{3.9}$$

The diagonal entries of $\Sigma_1 = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$ are sorted in descending order, i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, and are considered as the nonzero singular values of M (see also Section 3.1.2) with $r = rank(M)$.

The *SVD-NMF* initialization method computes the SVD of the non-negative matrix M and uses its singular triplets U, Σ, V to obtain good initial matrices for (W, H). The **Schmidt and Eckart-Young theorem** (see Stewart and Sun, 1990) ensures that a unique matrix $M^{(k)} = U\ diag(\sigma_1, \sigma_2, \ldots, \sigma_k, \underbrace{0, \ldots, 0}_{n-k})\ V^T$, with k corresponding to the chosen factorization rank ($1 \leq k \leq r \leq n$), can be found and is the global minimizer of the optimization problem (3.1). This matrix $M^{(k)}$ can be rewritten as:

$$M^{(k)} = U \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_k, \underbrace{0, \dots, 0}_{n-k}) \; V^T = \tilde{U}\tilde{\Sigma}V^T \tag{3.10}$$

$$\text{where} \quad \tilde{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1} & u_{p2} & \cdots & u_{pk} \end{pmatrix} \tag{3.11}$$

$$\text{and} \quad \tilde{\Sigma} = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k & \cdots & 0 \end{pmatrix}, \tag{3.12}$$

$\tilde{U} \in \mathbb{R}^{p \times k}$ and $\tilde{\Sigma} \in \mathbb{R}^{k \times n}$. If all entries of $\tilde{U}$ and $\tilde{\Sigma}V^T$ were non-negative, then the matrix $W = \tilde{U}$ and $H = \tilde{\Sigma}V^T$ could instantly be used to initialize W and H for an NMF algorithm, however, this is usually not the case as the entries of singular vectors obtained from SVD can be negative. Therefore, the negative elements of matrix $\tilde{U}$ are replaced by the absolute values of themselves, $|\tilde{U}|$ denotes the matrix, where all entries of $\tilde{U}$ are their absolute values. For $\tilde{\Sigma}V^T$ the same adjustments are made, thus the matrix $|\tilde{\Sigma}V^T|$ is obtained.

Finally, the following initialization formulas of W, H,

$$W_0 = |\tilde{U}|, \quad H_0 = |\tilde{\Sigma}V^T|,$$

are used for *SVD-NMF* with $W_0 H_0 \approx M^{(k)}$ (Qiao, 2015). It has to be mentioned that the replacement of the negative entries by their absolute values can lead to better or worse results, thus an improvement of this method would be achieved by finding a way to implement the negatives entries.

3.1.4 Clustering Methods

One of the most prominent clustering methods used for the initialization of NMF algorithms is **spherical k-means clustering**.Wild (2004) proposed this as a seeding method for *NMF*. In 2001, the *Spherical K-Means* method was first introduced by Dhillon and Modha (2001) for clustering large sets of sparse text data. A notation consistent with Wild et al. (2004) is used to depict the clustering method and its characteristics.
Remark:
$\{c_j\}_{j=1}^k \dots$ the k centroids (average of all the vectors in subset π_j) in $\mathbb{R}^p$
$\{\pi_j\}_{j=1}^k \dots$ the k disjoint subsets of the original $p \times n$ data matrix M.

$\Pi_k \equiv \{\pi_j\}_{j=1}^{k} \ldots$ the clustering defined by the k subsets.
$\iota(\Pi_k) \ldots$ an n-long indicator vector, where $\iota_i \equiv j$ if $m_i \in \pi_j$

It should be mentioned that also other clustering methods such as the *Fuzzy C-Means Clustering* have been suggested for initialization (see Rezaei and Boostani, 2011).

Strategy

As for the standard *K-Means* the method tries to find k centroids $\{c_j\}_{j=1}^{k}$ that represent k disjoint subsets of the columns of M with each subset containing all the vectors in M that are closest to their respective centroid. In contrast to standard *K-Means* the column vectors $m_1, m_2, \ldots, m_n$ of the original $p \times n$ data matrix M need to be normalized to unit length in the Euclidean (Frobenius) norm. This normalization has the effect that only the direction of each vector remains as important characteristic. Furthermore, the data set M is restricted entirely to non-negative elements, which already constitutes a necessary requirement for the application of NMF. As shown by Wild et al. (2004)[Theorem 1] any normalization of the basis matrix W has no influence on the convergence of a *NMF* algorithm.

Until now, it has not been clarified which measure is used to quantify the similarity of two vectors. Taking the above considerations into account, it is possible to use the *Cosine Similarity measure*

$$\cos(\theta_{x,y}) = \|x\| \, \|y\| \cos(\theta_{x,y}) = x^T y, \tag{3.13}$$

to quantify the similarity of two normalized vectors. The first equality holds due to the normalization of x and y, and the second is the standard definition of the inner product. The Cosine Similarity measure is bounded by $0 \leq \cos(\theta_{x,y}) \leq 1$ (given the non-negativity of x and y) and values closer to one indicate more similar vectors as perfect similarity is achieved if the two vectors point in the same direction and the angle $\theta_{x,y}$ between the two vectors x and y is equal to 0. Requiring that each centroid is also of unit length $\|c_j\| = 1$, the following objective function can be defined, which *Spherical K-Means* aims to maximize:

$$\Theta_{SKM}(\Pi_k) \equiv \sum_{j=1}^{k} \sum_{m_i \in \pi_j} m_i^T c_j = \sum_{i=1}^{n} m_i^T c_{\iota_i}. \tag{3.14}$$

Θ_{SKM} seeks to maximize the intracluster coherence while minimizing the intercluster coherence (Wild et al., 2004). The *Spherical K-Means* algorithm, which has been proven (Dhillon and Modha, 2001) to be non-decreasing in the objective function Θ_{SKM}, is summarized as follows:

Algorithm 6 Spherical K-Means

```
Initialize $k$ centroids $\{c_j\}_{j=1}^{k}$, set $t = 0$.
while clusters change from $t$ to $t+1$ do
    Compute $d_{ij}^{(t)} = m_i^T c_j^{(t)}$ for $1 \leq i \leq n$ and $1 \leq j \leq k$.
    Define the new partition $\Pi_k^{(t+1)}$ by updating each cluster:
        $\pi_j^{(t+1)} = \{m_i | j = arg \max_l d_{il}^{(t)}\}.$
    Recompute each centroid
        $c_j^{(t+1)} = \frac{\sum_{m_i \in \pi_j^{(t+1)}} m_i}{\left\| \sum_{m_i \in \pi_j^{(t+1)}} m_i \right\|}.$
end while
Output the final centroids and partition
    $(\{c_j^{(T)}\}_{j=1}^{k}, \Pi_k^{(T)})$
```

A straightforward way to define an *NMF* initial based on the clustering Π_k obtained with *Spherical K-Means* would be to use the final centroids $\{c_j\}_{j=1}^{k}$ and the indicator vector $\iota(\Pi_k)$ to define a $p \times k$ **concept matrix** C, whose columns correspond to the k centroids, and a sparse $k \times n$ indicator matrix Δ:

$$\Delta_{i,j} \equiv \begin{cases} 1 & \text{if } \iota_j = 1 \\ 0 & \text{else} \end{cases} \text{for } j = 1, \ldots, n, \ i = 1, \ldots, k. \tag{3.15}$$

This approximation $M \approx C\Delta$ represents a non-negative rank k approximation of M, which replaces every column vector m_i with its associated cluster's centroid c_j, where $m_i \in \pi_j$. It is assumed that the k centroids $\{c_j\}_{j=1}^{k}$ are linearly independent (Dhillon and Modha, 2001). This initialization with $W = C$ and $H = \Delta$ will result in a fixed point for Euclidean multiplicative update *NMF* algorithms (Wild et al., 2004)[Lemma 3], which is the reason why this initialization should be discarded.

Therefore, and as some of the n vectors in M are located near the intersection of two or more clusters, the idea to use linear (instead of binomial) combinations of the centroids was considered. This means that as before the concept matrix C is used as W and the initial H is the solution of the *NNLS* sub-problem (see Section 2.1) as it minimizes $\|M - CZ\|$ under the restriction of $Z \geq 0$. This strategy has been proven to provide *NMF* initializations that lead to faster convergence of the NMF algorithms than other clusterings and random initialization (Wild et al., 2004).

Computational considerations

This initialization leads to sparser basis vectors than the use of random initializations as the centroids that result from Spherical K-Means reflect the sparsity of the original data, however, they are usually less sparse than a typical data vector since they are obtained by averaging several data vectors. As a result, a slight reduction in computational expenses for the *NMF* algorithm can be achieved for those algorithms, in which the *locked-in* phenomena occurs. The two major reasons to prefer *Spherical K-Means* over its more general predecessor, *K-Means* in this context are:

1. The efficiency and robustness for very large data sets of the *Spherical K-Means* algorithm, with refinements (Dhillon and Modha, 2001).
2. Due to normalization inherent to the *NMF* multiplicative update schemes, the centroids need to be as different as possible when normalized. As already stated, the *Spherical K-Means* is only concerned with the direction of data vectors, which leads to centroids that considered to be more linearly independent than those from *K-Means* (Wild et al., 2004).

In contrast to the *SVD* based initialization methods, the computational requirements per iteration of *Spherical K-Means* is with $O(pnk)$ clearly higher, since for instance the overall costs of *NNDSVD* (see Section 3.1.2) are of the same magnitude, and considering that many iterations could be performed until an optimal clustering is obtained. To lower the costs, as in general it is the desire to not increase the overall computational complexity of the initialization-factorization process, it was suggested to not take the final clustering and instead use the clustering (Concept matrix) resulted after a fixed number of iterations. The simulations performed by Wild et al. (2004) indicate that such a stopped clustering results in similar errors for the initials as the optimal clustering.

Another drawback of the *Sperical K-Means* clustering is the need of initializing the centroids, hence to avoid the possibility of not converging to a local minimum, the clustering should be repeated with different random initial centroids. Every clustering algorithm used to define initials for (W, H) has its major disadvantage in the computational expensiveness.

3.1.5 ICA Based Initialization

The motivation for using the independent component analysis - **ICA** (a detailed introduction of ICA is given in Hyvärinen and Oja, 2000) for the initialization of *NMF* lies in its defining property of estimating bases, through which the weights for the corresponding bases (sources in terminology of *ICA*) become independent from each other (or at least as independent as possible). In contrast, initialization methods using *SVD* (or respectively *PCA*) produce orthogonal bases to represent the data matrix M.

Furthermore, it has been shown that optimal *NMF* bases are along the edges of a *convex polyhedral cone*, which is defined by the observed points in M, in a p-dimensional space (Huang et al., 2014). In Figure 3.1 an example of various *NMF* bases is given when $p = r = 2$. This simple example illustrates that orthogonal bases taken from an SVD (or PCA) may not be a good choice as initial *NMF* bases, since they possibly represent a meaningless area. All data points are represented by the optimal bases, which probably are ICA bases since they tend to be dissimilar but not orthogonal and the tight bases have struggle to represent all data points due to the non-negative constraint of the weights.

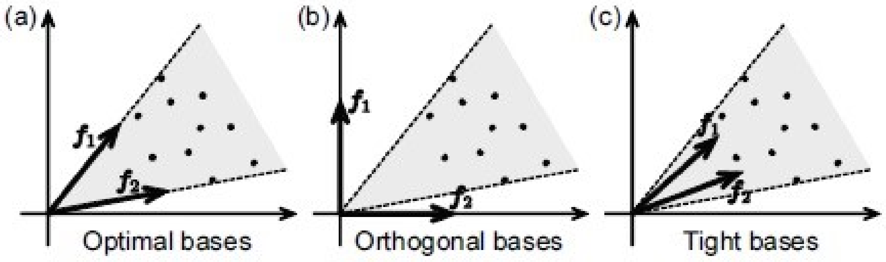

Figure 3.1: **Example of NMF Bases** - Geometry of (a) optimal, (b) orthogonal, and (c) tight bases, where the observed data points are represented by the black dots, the gray area is indicating the cone defined by the data points, the broken lines are indicating the edges of cone, f_k denotes kth NMF basis, $p = k = 2$ and $n = 10$ (Kitamura and Ono, 2016).

Before the strategy to establish estimates for the initials of (W, H) is described, a short explanation of the basic concept of ICA is given. ICA belongs to the unsupervised learning methods, particularly to the blind source separation - **BSS** methods like *NMF* and has the goal of finding a linear representation of non-Gaussian data, so that the components are statistically independent, or as independent as possible. The ICA model for $M \in \mathbb{R}^{k \times n}$ can be defined as:

$$M = AS, \tag{3.16}$$

with the unknown mixing matrix $A = (a_1, \ldots, a_k)$, where a_k is the $k \times 1$ kth ICA basis, $S = (s_1, \ldots, s_k)^T$, where s_k is the $n \times 1$ kth unknown independent component (source signal). The assumption for the source signals s_k of having a non-Gaussian distribution is essential to estimating the ICA model, as without non-Gaussianity the mixing matrix is not identifiable (Hyvärinen and Oja, 2000). The restriction to non-Gaussianity can be justified as in practice the non-Gaussian components of the underlying data tend to be more interesting than the Gaussian components, which often are obtained as noise or some sort of perturbation.

If the mixing matrix A was known, the components could be easily calculated by $S = TM$ with $T = A^{-1}$, but since this is not the case, a good estimator of T has to be found. The basic concept of ICA relies on the classical **Central Limit Theorem**, as it states that the distribution of a sum of independent random variables tends towards a Gaussian distribution under certain conditions, which basically means that any sum of independent random variables is more Gaussian than the original variables. Therefore,the estimator of one of the independent components can be obtained by maximizing the non-Gaussianity of $t_j^T M$ for an arbitrary $j \in \{1, \ldots, k\}$, since the different independent components are uncorrelated, the next one could be found in the same manner with the only difference of restricting the search to the orthogonal complement of the previously estimated components.

For ICA the following two quantitative measures of non-Gaussianity of a random variable, say y, are used :

(a) absolute value of the kurtosis defined as $|kurt(y)| = |\mathbb{E}(y^4) - 3(\mathbb{E}(y^2))^2|$

(b) negentropy $J(y) = H(y_{gauss}) - H(y)$, where y_{gauss} is a Gaussian random variable of the same covariance matrix as y and $H(.)$ the entropy of a random vector.

As mentioned by Hyvärinen and Oja (2000) the negentropy is in some sense the optimal estimator of non-Gausianity, as far as statistical properties are concerned, but computationally very difficult to calculate as an estimate (possibly non-parametric) of the pdf is required. This issue was solved by using simpler and faster approximations of negentropy, which also have appealing statistical properties, especially robustness. Such an approximation is given by

$$J(y) \propto [\mathbb{E}(G(y)) - \mathbb{E}(G(v))]^2, \tag{3.17}$$

where v is a standardized Gaussian variable and $G(\cdot)$ a non-quadratic function. The following choices of $G(\cdot)$ were suggested by Hyvärinen and Oja (2000):

(a) $G(u) = \log\cosh(u)$ for general purpose.

(b) $G(u) = -\exp(-\frac{u^2}{2})$ if robustness is a major objective.

A fast and efficient algorithm to estimate the independent components, which uses the above approximation of the negentropy, is the *FastICA* algorithm as introduced by Hyvärinen and Oja (2000). For the FastICA, the matrix M needs to be centered and whitened, so that the independent component can be obtained at the extrema of $J(t^T M)$ given in (3.17). The update rule of FastICA for a column t of the requested matrix $T \in \mathbb{R}^{k \times k}$ is as follows

$$t \longleftarrow \mathbb{E}(Mg(t^T M)^T) - \mathbb{E}(g^{'}(t^T M))t, \tag{3.18}$$

where $g(\cdot)$ is the derivative of $G(\cdot)$, and $g^{'}(\cdot)$ is the derivative of $g(\cdot)$. After every update (3.18), the vectors $t_1, \ldots, t_k$ are orthogonalized either in a Gram-Schmidt-like procedure or symmetrically. Finally, this algorithm converges to T (often called demixing matrix) and by matrix multiplying T and M the estimated independent components S are obtained, which are unique apart from permutations and signs. It has to be mentioned that the FastICA algorithm has many desirable properties compared to other ICA algorithms like for instance the cubic convergence (gradient descent methods only have linear convergence) and simple application as there is no necessity to choose a step size parameter.

In general, the estimated components sources s_k tend to be sparse, if a super-Gaussian distribution (random variables with positive kurtosis) is assumed, which is also the case for NMF as the sparsity is induced by the non-negative constraint of W.

Strategy

As the traditional ICA do not exclude negative values for the ICA bases a_k and the independent components s_k, the *non-negativity* assumption of the components (sources) and the ICA bases have to be added. This model is known as non-negative independent component analysis - **NICA** (Yuan and Oja, 2004). The NICA is combined with a PCA as a preprocess to reduce the dimensionality, this reduction can be represented as

$$\begin{cases} P_1 M = AS \\ P_2 M \approx 0 \end{cases}, \tag{3.19}$$

where $P = (P_1^T P_2^T)^T$ is the $p \times p$ transformation matrix of PCA with $P_1 \in \mathbb{R}^{k \times p}$ and $P_2 \in \mathbb{R}^{(p-k) \times p}$. The eigenvectors of the variance-covariance matrix MM^T are the row vectors in P and are arranged in descending order on the basis of their eigenvalues. Consequently, P_1 includes the top k eigenvectors of MM^T and P_2 the remaining ones. Moreover, $A = (a_1, \dots, a_k)$ is a mixing (ICA basis) matrix, $S = (s_1, \dots, s_k)^T$ is a source matrix, and 0 is the $(p-k) \times n$ zero matrix. Then the NICA is applied on $P_1 M$, which is the representation of the data matrix M corresponding to the first k principal components.

Plumbley (2002, 2003) has outlined an alternative approach for the ICA problem under the consideration of the non-negativity assumption. The following definition is needed for the further analysis:

Definition 3.1.9 [non-negative and well-grounded]
A source s_i is *non-negative* if $Pr(s_i < 0) = 0$, and *well-grounded* if $Pr(s_i < \delta) > 0$ for any $\delta > 0$; i.e. s_i has non-zero pdf all the way down to zero.

Using this definition, Plumbley (2002) has proven the following key result:

Theorem 3.1.10
Suppose that s is a vector of non-negative well-grounded independent unit-variance sources s_k, $i = 1, \dots, k$, and $y = Qs$ where Q is a square orthonormal rotation, i.e. $Q^T Q = I$. Then Q is a **permutation matrix**, i.e. the elements y_j of y are a permutation of the sources s_i, if and only if **all** y_j are **non-negative**.

Proof: See Plumbley (2002). ■

Based on this theorem, the NICA can be reduced to the issue of finding a square orthonormal rotation matrix T for the noncentered and whitened data for which the estimated (separated) sources become non-negative:

$$\begin{aligned} Y &= TZ, && (3.20) \\ Z &= VP_1M = VAS, && (3.21) \end{aligned}$$

where V is a whitening matrix, which adapts the matrix P_1M in the way that the covariance matrix $\Sigma_{P_1M} = P_1M(P_1M)^T - \mu\mu^T$ with $\mu = mean(P_1M)$ becomes the identity matrix, whereby it is important that this whitening process does not center the data. The whitening matrix V can be obtained by the eigenvalue decomposition of $\Sigma_{P_1M} = EDE^T$ with E being the orthonormal basis of eigenvectors and D the diagonal matrix of the corresponding eigenvalues, hence setting $V = \Sigma_{P_1M}^{-1/2} = ED^{-1/2}E^T$ provides the desired result. The centering is avoided since it would transform the non-negative data into data which possess also negative elements (all elements smaller than the mean would be negative). Moreover, $Y = (y_1, \ldots, y_k)^T$ is a matrix consisting of the estimated sources y_k. The following cost function was suggested as suitable to find the rotation matrix T (Plumbley, 2002):

$$J(T) = \mathbb{E}(\left\|Z - \hat{Z}\right\|^2) = \mathbb{E}(\|Y - Y_+\|^2) = \sum_{k,n} \min(0, y_{kn})^2, \qquad (3.22)$$

where $\hat{Z} = T^TY_+$ is a re-estimate of $Z = T^TY$ and the second equality holds since T has to be a square orthonormal rotation matrix. This cost function will be minimized, if all the y_k are positive, but due to the dimensionality reduction via PCA this global minimum is not likely to exist (P_1M probably has negative entries). A simple method to solve this minimization problem (3.22) is based on the gradient descent method (Oja and Plumbley, 2004) , which suggests the following update rule for a column of T:

$$t_k \longleftarrow t_k - 2\gamma \sum_n \min(0, y_{kn}) z_{kn}, \qquad (3.23)$$

where γ is the step-size parameter (a hyperparameter of this algorithm). Each of the k column vectors t_k of T is updated by (3.23) and then the following symmetrical decorrelation step is applied on $T = (t_1, \ldots, t_k)$:

$$T \longleftarrow \left(TT^T\right)^{-1/2} T, \qquad (3.24)$$

where the inverse square root $(TT^T)^{-1/2}$ is obtained from the eigenvalue decomposition of $TT^T = F\Lambda F^T$ as $(TT^T)^{-1/2} = F\Lambda^{-1/2}F^T$. These two steps are repeated until T converges. To assess the convergence the following stopping criterion was used (same as used in FastICA):

$$D(T_{old}, T_{new}) = \max\left(\left|diag(|T_{new}T_{old}^T|) - 1\right|\right), \qquad (3.25)$$

where T_{old} is the matrix before the update and T_{new} the one after the update. Another possibility to solve the minimization problem (3.22) would be the "fast NICA" algorithm (Yuan and Oja, 2004), which does not require a hyperparameter such as γ for the the estimation. Nonetheless, for this initialization method the gradient descent method is used.

For the initialization of the weight matrix H in the *NMF* model, the estimated independent components (sources) $Y = TZ$ are directly used. Moreover, the estimated orthonormal (demixing) matrix T can be taken to calculate the initial basis matrix W of the *NMF* model. If it is approximately assumed that $M = WH$, $S = Y$, and $A = (TV)^{-1}$, the following equation can be obtained from (3.19):

$$PWH \approx \begin{bmatrix} (TV)^{-1} \\ 0 \end{bmatrix} H. \tag{3.26}$$

Therefore, after multiplying the equation with the inverse matrix of P from the left side and the inverse matrix of H from the right side, the basis matrix W can be identified as

$$W \approx P^{-1} \begin{bmatrix} (TV)^{-1} \\ 0 \end{bmatrix}. \tag{3.27}$$

The non-negativity of these initial matrices of (W, H) can not be guaranteed, since the PCA applied for dimensionality reduction likely induced negative entries in the estimated weight matrix H and for the basis matrix W non-negativity was not considered in the cost function (3.22) or at any point of the estimation process. The matrices (W, H) can only be used as initials for a *NMF* algorithm, if the negative values of W and H are replaced by non-negative values. To perform this so called *nonnegativization* of W and H the following three methods were proposed (Kitamura and Ono, 2016):

- **Nonnegativization 1**: $W_0 = |W|$, $H_0 = |H|$,
- **Nonnegativization 2**: $W_0 = |W|$, $H_0 = \alpha_H W_0^T M$,
- **Nonnegativization 3**: $H_0 = |H|$, $W_0 = \alpha_W M H_0^T$,

where $|\cdot|$ denotes the entrywise absolute operator, and α_W and α_H are coefficients for fitting the scale of $W_0 H_0$ to M. These coefficients can be calculated from

$$\alpha_W = arg \min_{\alpha} \mathcal{D}(M \| \alpha M H_0^T H_0), \tag{3.28}$$

$$\alpha_H = arg \min_{\alpha} \mathcal{D}(M \| \alpha W_0 W_0^T M), \tag{3.29}$$

after the proposed initialization of H respectively W, where $\mathcal{D}(\cdot\|\cdot)$ denotes an arbitrary cost function (measure of error) for NMF. For typical cost functions like the Euclidean distance (EU) or the generalized Kullback-Leibler divergence (KL) the solutions of (3.28) and (3.29) can be described as follows:

- **For EU-NMF**: $\alpha_W = \frac{\sum_{p,n} m_{pn} b_{pn}}{\sum_{p,n} b_{pn}^2}, \quad \alpha_H = \frac{\sum_{p,n} m_{pn} c_{pn}}{\sum_{p,n} c_{pn}^2}$,
- **For KL-NMF**: $\alpha_W = \frac{\sum_{p,n} m_{pn}}{\sum_{p,n} b_{pn}}, \qquad \alpha_H = \frac{\sum_{p,n} m_{pn}}{\sum_{p,n} c_{pn}}$,

where $b_{pn} = \left[MH_0^T H_0\right]_{pn}$ and $c_{pn} = \left[W_0 W_0^T M\right]_{pn}$.

To get a compact overview of the proposed initialization strategy, the algorithm is depicted as shown below:

Algorithm 7 NICA for NMF

```
Run a PCA as illustrated in (3.19).
Whiten the matrix $P_1 M$ with $V = ED^{-1/2}E^T$, but do not center.
Initialize $T \in \mathbb{R}^{k \times k}$ by an arbitrary orthonormal rotation matrix
while $D(T_{old}, T_{new}) >$ tolerance and $j \leq$ maxiter do
    Set $Y = TZ$ with $Z = VP_1 M$
    for $u = 1$ to $k$ do
        Compute
            $$t_u \longleftarrow t_u - 2\gamma \sum_n \min(0, y_{un}) z_{un}$$
    end for
    Apply the symmetrical decorrelation for $T = (t_1, \dots, t_k)$:
        $$T \longleftarrow \left(TT^T\right)^{-1/2} T.$$
end while
Set
    $$H = Y = TZ$$
and
    $$W = P^{-1} \begin{bmatrix} (TV)^{-1} \\ 0 \end{bmatrix}$$
Perform one of the suggested Nonnegativizations for $W$ and $H$
Output $W$ and $H$
```

Computational considerations

The most computationally critical parts are the update rule based on gradient descent (3.23) and the symmetrical decorrelation step (3.24), since the convergence can be slow and depends on the choice of the hyperparameter γ . A replacement of the gradient descent procedure with the *fastNICA* (Yuan and Oja, 2004) algorithm could reduce the computational costs.

Nevertheless, the computational costs can be seen as not critical compared with the case of *NMF* iterations (Kitamura and Ono, 2016). As the experimental comparison conducted by (Kitamura and Ono, 2016) has shown, the NICA based initialization provides faster and deeper convergence of the *NMF* cost function than random initialization, NNDSVD or PCA-based initialization, but it is computational more intensive.

3.2 Quality Measures

Universal quality measures are required to compare the performance of NMF algorithms on a given data set. The performance can be evaluated under different aspects, such as approximation error, sparseness of the NMF factors, clustering performance and computational costs. It is common practice to quantify the computational costs by the CPU time required to perform the factorization.

The performance of the initialization methods used for NMF algorithms can be sufficiently evaluated by considering the possibly improved final approximation error and the convergence speed of the NMF algorithm, i.e how many iterations are performed until the NMF factors (W, H) converge.

3.2.1 Standard Measures

The approximation error is considered as the final value obtained from the objective function (of the NMF algorithm), but as the algorithms can differ in terms of the objective function, a comparison of the residuals is not appropriate.

In order to evaluate how well the NMF model reconstructs the original data, the **explained variance - evar** is used. Evar is defined as:

$$evar(WH) = 1 - \frac{RSS}{\sum_{i,j} m_{ij}^2},$$

where (W, H) are the computed NMF factors, m_{ij} are the entries of the data matrix M and $RSS \overset{\text{def}}{=} \|M - WH\|^2$. Due to the use of the Frobenius norm in the definition of RSS, a priori NMF algorithms that use the Frobenius norm to measure the error would be favoured, but on the other hand the results obtained by Pascual-Montano et al. (2006) show that algorithms not based on the Frobenius norm may still achieve better values of explained variance.

Sparseness

The sparseness of NMF factors can be an interesting matter as it improves their interpretability. A sparseness measure is a function, which quantifies how much energy of a vector is packed into a few components. Considering a normalized scale, the sparsest possible vector (only a single component is non-zero) should have a sparseness of one, whereas a vector with all elements equal should have a sparseness of zero. As already mentioned in Subsection 2.1.2, a useful sparseness measure was invented by Hoyer (2004):

$$spar(x) = \frac{\sqrt{n} - \|x\|_1 \,/\, \|x\|}{\sqrt{n} - 1} \quad \text{for} \quad x \in \mathbb{R}^n$$

This function interpolates smoothly between the two extremes described above.

Clustering performance

Every NMF can be interpreted as clustering into r clusters (r NMF basis vectors) as each observation (data point) m_j can be assigned to a cluster k by considering $k = \arg\max_l h_{lj}$, i.e. assigned to the basis vector which has the highest weight for the observation m_j. In cases where a prior knowledge of the class assignments is given, the following two measures (used in Kim and Park, 2007) can be useful to evaluate the quality of the clustering generated from NMF. The measure **Purity** is given by

$$Purity = \frac{1}{n} \sum_{q=1}^{r} \max_{1 \le j \le l} (n_q^j),$$

where n is the total number of observations, l the number of original classes and n_q^j is the number of samples in the cluster q that belong to original class j. If the clustering is perfectly reconstructed, purity has the value 1. Otherwise it can be stated the larger the value of purity is, the better the clustering performance.

The second measure is called **Entropy** and is defined as follows:

$$Entropy = -\frac{1}{n \log_2 l} \sum_{q=1}^{r} \sum_{j=1}^{l} n_q^j \log_2 \frac{n_q^j}{n_q},$$

where n_q is the size of cluster q and the other terms used equally to the definition of Purity. In contrast to Purity, Entropy reaches 0 for perfect reconstruction and cluster quality is considered better at a small value.

The well-known silhouette coefficient (see for details and definition Rousseeuw, 1987) specifies for an observation how good the assignment to the two nearest clusters is. A silhouette coefficient of almost 1 means that the observation is well clustered, while a small value (around 0) means that the observation lies between two clusters. If an observation has a negative silhouette coefficient, it is probably assigned to the wrong cluster.

Since the solution of the NMF problem is not unique and can depend on the initial matrices chosen for W and H, the stability of the clusters obtained by NMF can be of further interest. Consequently, some quality measures are needed to evaluate the stability of clustering. For every NMF run a connectivity matrix C of size $n \times n$ is defined by:

$$c_{ij} = \begin{cases} 1 & \text{if samples } i \text{ and} j \text{ belong to the same cluster} \\ 0 & \text{if samples } i \text{ and} j \text{ belong to different clusters.} \end{cases}$$

If multiple runs of the same NMF algorithm with different random initializations are done, the **consensus matrix** $\overline{C}$ can be computed as the average connectivity matrix over these NMF clustering runs. The consensus matrix $\overline{C}$ has entries that range from 0 to 1, which are reflecting the probability that samples i and j belong to the same cluster. In case of a stable clustering it would be expected that C will not vary a lot among runs, hence the entries of $\overline{C}$ will be close to 0 or 1.

The entries of $\overline{C}$ can be interpreted as similarity measures, which allows one to define the **Cophenetic Correlation Coefficient** $\rho_k(\overline{C})$ as Pearson correlation between the sample distances induced by the consensus matrix, and the cophenetic distances obtained by its hierarchical clustering (see Brunet et al., 2004). The Cophenetic Correlation Coefficient indicates the dispersion of the consensus matrix and equals 1 if a perfect consensus matrix (all entries equal 0 or 1) is obtained. In case of entries that are scattered between 0 and 1, the Cophenetic Correlation Coefficient is < 1.

Another possibility to measure the dispersion of a consensus matrix $\overline{C}$ is given by the dispersion coefficient (Kim and Park, 2007) :

$$disp = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} 4 * (c_{ij} - \frac{1}{2})^2.$$

As for the Cophenetic Correlation Coefficient, the Dispersion Coefficient becomes 1 in case of a perfect consensus matrix and otherwise $0 \leq disp < 1$.

These two cluster stability measures are used in the section to estimate an appropriate factorization rank r for a given data matrix.

3.3 Rank Estimation

The rank r of the factorization is a critical parameter for every *CLRMA*, in case of *NMF* it is linked with the non-negative rank of a matrix (see Definition 2.2.1). Every non-negative matrix has a non-negative rank, but up to now there has been no algorithm proposed to determine the non-negative rank in ploynomial time (see Gillis, 2017). Consequently, the factorization rank r has to be estimated.

In general, r should be small to reduce the dimension of $M \in \mathbb{R}_{+}^{p \times n}$, but on the other hand for the accuracy of the approximation, r should not be too small, since the approximation error is increasing by downsizing r. Thus, a method or strategy to choose r is demanded that considers these two objectives. Usually it is requested that r is smaller than $\min\{p, n\}$ and satisfies the basic rule $(p + n)r < pn$ (Qiao, 2015). In the following subsections some methods to find a suitable factorization rank r are introduced.

3.3.1 Rank Estimation Based on Quality Measures

Given an NMF method and the data matrix D, a common way of estimating the rank r is to try different values, and compute quality measures (see Section 3.2) of the calculated NMF in order to choose the best value according to these quality criteria. In order to get a robust estimate of the factorization rank r, about $30 - 50$ runs with random initialization of the given NMF method have to be performed. In the following, a list of proposed strategies to choose the optimal value of r is given:

- proposed in Brunet et al. (2004): Take the first value of r for which the cophenetic coefficient starts decreasing. A decrease of the cophenetic coefficient indicates an unstable clustering of NMF.
- proposed in Hutchins et al. (2008): Take the first value where the RSS curve presents an inflection point, such a point means that an additional rank does improve the reconstruction error to a smaller magnitude than the previous rank increasements.
- proposed in Frigyesi and Höglund (2008): Take the smallest value at which the decrease in the RSS is lower than the decrease of the RSS obtained from random data (a possibility to create random data is to randomly permutate the entries of the data matrix M). In contrast to the other two strategies, this approach prevents over-fitting of the data as an increase in r is considered relevant if the information captured by the factorization is greater than that obtained from random unstructured data, otherwise the increase in r is likely to result in over-fitting.

The best way to implement these strategies is achieved by plotting the relevant quality measures. A simple example of an NMF rank survey is given in Figure 3.2, wherein a randomly generated (non-negative) data matrix with a non-negative rank of 3 is analyzed.

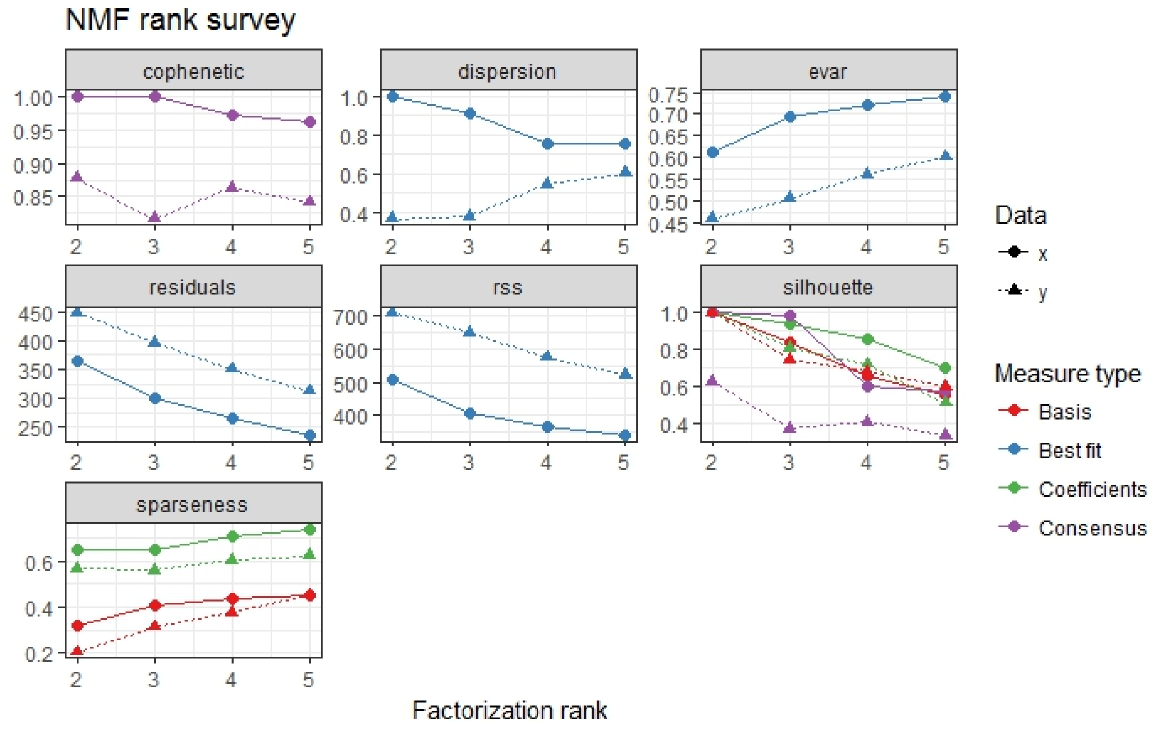

Figure 3.2: **Rank survey** - Here a rank survey for a randomly generated data matrix D with a non-negative rank of 3 is illustrated. Each point (x-Data) was obtained from 10 runs of the Brunet algorithm performed on D and each triangle (y-Data) was obtained from 10 runs of the Brunet algorithm performed on the randomly permutated data matrix. All the proposed strategies correctly estimate a factorization rank of 3. In addition, further quality measures are shown to give a good overview of the influence of the factorization rank on NMF clustering and the factors (W, H).

3.3.2 SVD-based rule

This rule was suggested by Qiao (2015) and is taking the singular values of M into account to define a rule for the factorization rank r. In particular, a certain amount of relatively larger singular values is used, and as the singular values of M obtained by SVD are sorted in descending order, the sum of first few singular values accounts for a large proportion of the sum of all singular values. According to Qiao (2015) 0.9 was chosen as the extracting proportion, since it contains enough information of singular values and avoids the factorization rank being too small to influence the accuracy of factorization.

At first the sum of all non-zero singular values are summed up, that is $\boldsymbol{sum_t} = \sigma_1 + \sigma_2 + \ldots + \sigma_t$, and after this the number of singular values which accounts for 90%

of all non-zero singular values is evaluated.
That is the r, which holds the following inequalities for $\boldsymbol{sum_r} = \sigma_1 + \sigma_2 + \ldots + \sigma_r$:

$$\boldsymbol{sum_r} / \boldsymbol{sum_t} < 90\% \quad \text{and} \quad \boldsymbol{sum_{r+1}} / \boldsymbol{sum_t} \geq 90\%. \tag{3.30}$$

As known from the singular value decomposition, the non-zero singular values are the square root of non-negative eigenvalues of matrix MM^T, hence it can be stated that $t \leq \min\{p, n\}$, where p, n are the number of rows respectively columns of matrix M. According to the numerical experiments conducted by Qiao (2015), it can be obtained that $r \ll t$ and the basic rule $r < \frac{pn}{p+n}$ are usually satisfied if this rule is applied.

4 Simulations

The open source programming environment for statistical computing **R** (see R-Core-Team, 2013) is used for all calculations in the subsequently described simulations. The **R** package **NMF** (see Gaujoux and Seoighe, 2010) provides a useful framework for the application of NMF methods with the possibility of extending the package by customized algorithms and initialization methods.

In the upcoming simulations the built-in NMF methods of Lee, Brunet and SNMF/L were applied in combination with the built-in initialization methods random (corresponds to the simple random method described in Subsection 3.1.1) and NNDSVD. In addition the proposed algorithms of ALS, ACLS, AHCLS and the recursive algorithm for separable NMF were extended to the functionality of the package in order to have a consistent framework for all algorithms. Moreover, the proposed initialization methods of random Acol, random C, SVD-NMF, Spherical k-Means, NICA were added to the list of initializations. For the analyses different settings of NICA were used, in reference to the utilized Nonnegativization $1-3$ the terms NICA1, NICA2 or NICA3 are used. All NMF algorithms (built-in or customized) can be applied to a given dataset by calling the **R**-function "nmf" , which allows to specify the initialization method by an optional parameter. A detailed description on the functionality of the **R**-function "nmf" is given in Gaujoux and Seoighe (2010). Since the recursive algorithm for separable NMF is the only geometric algorithm considered for comparison, it will be called the GEO algorithm in the coming sections. For AHCLS and SNMF/L, the hyperparameters were set to the following values after some test runs with the underlying data: (i) AHCLS: $\lambda_H = 0.05$, $\lambda_W = 0.01$, $\alpha_H = 0.6$, $\alpha_W = 0.8$ (ii) SNMF/L: $\beta = 0.01$ and $\eta = \max(D)$, where D denotes the data matrix.

In the subsequent simulations, the focus was on investigating the functionality and properties of the above-mentioned NMF algorithms (and their initializations) in the context of mass spectrometry. A description of the data set used for the simulations can be found in the next section.

4.1 Meteorite Data

The dataset used for the simulations originates from the CoMeCS-Project (2017) and contains spectral information of meteorites. It was provided by Brandstätter, Ferrière, and Koeberl from the Natural History Museum (Vienna, Austria). The preparation of the samples was done by Engrand from the Centre de Sciences Nucléaires et de Sciences de la Matière (Orsay, France). The **TOF-SIMS**[1] measurements were taken by Hilchenbach from the Max Planck Institute for Solar System Research (Göttigen, Germany). In spectrometry, the measurements are called spectra and are typically visualized in a mass spectrum, which represents the distribution of the ions by mass (to be more precise mass-to-charge ratio). An example of a mass spectrum is given in Figure 4.1.

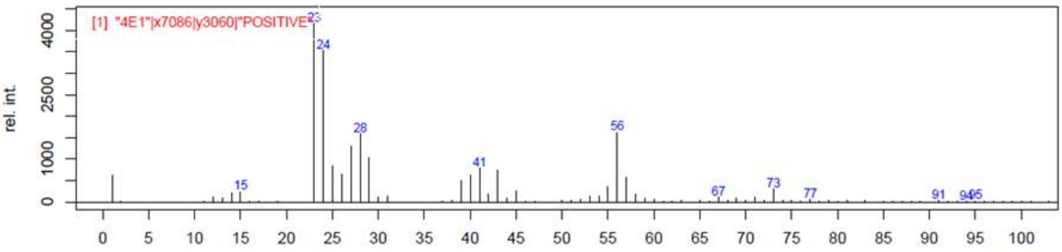

Figure 4.1: **Example of COSIMA TOF-SIMS spectrum** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensities on the y-axis. This spectra was measured for meteorite "Renazzo", taken from CoMeCS-Project (2017).

In the dataset, 1408 spectra with their 297 intensities of ions at a given mass number are stored. The mass number is the total number of nucleons (all protons and neutrons) in an atomic nucleus that uniquely defines an isotope of a chemical element. The intensities of the mass numbers $m23$, $m115$ and $m197$ were removed for chemical reasons, consequently only 297 of 300 inorganic mass bins were considered. Within the context of pre-processing steps, each spectrum was standardized in the sense that the sum of the relative intensities is always 100.

The data is classified into eleven different classes with ten different meteorites and the eleventh class, called "substrate", which consists of gold spectra. Since the spectral composition of Gold is distinctive, it should be distinguishable from the spectra of the meteorite corns. On every gold plate, or also known as target (in total 4), a number of corns from different meteorites are placed. **TOF-SIMS-**measurements of the targets with the placed corns are taken along a grid and resulted in the observed spectra of the dataset. As can be seen in Figure 4.2 the mesh size of the grid is bigger than the corns of the meteorites.

Therefore, it has to be clarified if the observed spectra belong to the gold-plate ("substrate") or to the meteorite corn, which essentially can be interpreted as a classification problem with two classes. In order to solve it, kNN (k nearest neighbour) and a method based on orthogonal distances in the substrate-PCA space were used.

---

[1] Time-of-Flight Secondary Ion Mass Spectrometry (TOF-SIMS) is the combination of the analytical technique SIMS (Secondary Ion Mass Spectrometry) with Time-of-Flight mass analysis. In SIMS a pulsed beam of primary ions is focused onto a sample surface to remove molecules from its atomic monolayers (secondary ions). For the Time-of-Flight mass analysis these particles are accelerated into a "flight tube" and the exact time at which they reach the ctor detector is measured to determine their mass.

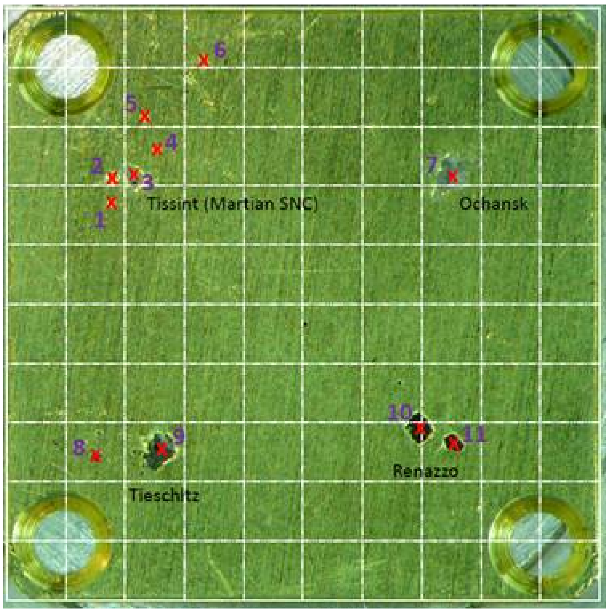

Figure 4.2: **Target 4E1** - Image of Target 4E1 (gold plate) with eleven corns placed upon it. The corns belong to different corns, taken from CoMeCS-Project (2017).

An intersection of these two estimated class assignments were taken to consider if a spectrum is corn specific or not. In Table 1 the class assignments for the different meteorite classes are listed.

Table 1: This table shows the number of total corns, the number of different gold plates (targets) they were placed on, the total observations for each of the elven classes and how many were classified by kNN and PCA as meteorite specific

	Nr. of corns	Nr. of targets	Total observations	on corn
allende	4	2	213	150
lance	3	2	94	77
mocs	6	2	106	71
murchison	2	2	97	84
ochansk	1	1	115	80
pultusk	11	2	199	107
renazzo	2	1	70	64
substrate	-	4	111	-
tamdakht	9	2	163	106
tieschitz	2	1	70	36
tissint	6	1	170	71

Since all the intensities of a spectrum are non-negative (positive) a NMF algorithm could be applied to estimate two basis spectra, where one basis spectrum should represent a typical meteorite spectrum and the second basis spectrum should be very similar to a Gold spectrum. The similarity to a substrate spectra can be determined by comparing the resulted NMF basis spectra with the spectra of the class "substrate", which are pure spectra of Gold and as mentioned before should be distinguishable to spectra from a meteorite. Taking the weight matrix into account each spectrum could be assigned by its respective weight for the two basis spectra to the corn or substrate class.

This idea motivates the subject of the next section, but at first it should be stated which similarity measures were used for the comparison of two spectra. The Pearson correlation coefficient r between two spectra and the correlation coefficient of the scaled spectra $I_m^s = m^2 * \sqrt{I_m}$ (where I_m is the intensity on mass number m), which puts more weight on the higher masses, were referred as popular similarity measures by Varmuza (2010). The correlation coefficient for the scaled spectra will be called *rscaled* in the upcoming analyses. Furthermore, the L_1- and $L2$- error were listed as typical similarity measures.

4.2 Corn vs Substrate

Since one of the key features of NMF is to separate non-negative data into parts, it is of interest to analyse how NMF performs on the task of distinguishing between the spectra of a corn and the substrate (gold-plate). As a reference, to evaluate the performance, the already performed classification based on kNN (k-nearest neighbour) and PCA was taken (see Table 1).

In this section, an example of such an analysis is given by considering only the spectra potentially taken from the meteorite ”ochansk”, where 115 spectra have been measured as stated in Table 1. In order to get a first insight, the chosen NMF algorithms with 1000 random initializations (seed = 123457) were applied and for each algorithm the best solution out of these 1000 initializations was kept. This high number of replications should ensure to achieve reasonable results.

In Table 2 the performance of this NMF algorithms are compared by the proposed quality measures (see Section 3.2) and their ability of estimating two dissimilar basis spectra (measured by r and *rscaled*). All algorithms achieved a very high value of explained variance in the range of 95% to 97%. Furthermore, the estimated basis matrices W preserved the sparseness of the target matrix (115 spectra), which is around 86.73%. In terms of their cluster ability, all performed pretty similar and reconstructed the reference classification to an acceptable level with purity between 0.70 and 0.76. In terms of their ability to estimate two dissimilar basis spectra, the *SNMF/L* method achieved the best result with a correlation coefficient less than 0.1 and even *rscaled* is considered low with 0.16. Moreover the computation time and convergence speed of the factorization obtained by *SNMF/L* was the lowest of all the methods except from *GEO*, which is the only non-iterative algorithm. The *GEO* algorithm performed surprisingly well in comparison to the other algorithms even though the separability assumption was not fulfilled. As a result of this first analysis, the factorization obtained by *SNMF/L* was considered as the favoured method for further investigations. In the following, the obtained basis matrix and weight matrix by the best method are denoted as W^*, respectively H^*.

Table 2: Quality measures - NMF for meteorite "ochansk" results are based on 1000 random initializations

	Quality measures							
Algorithms	Evar (%)	Sparsity W/H (%)	r	r scaled	purity	entropy	CPU time (seconds)	niter
LEE	97.53	88.95 / 59.09	0.21	0.41	0.73	0.57	0.61	420
BRUNET	96.57	88.50 / 60.57	0.25	0.51	0.76	0.54	0.67	440
ALS	97.53	88.95 / 61.78	0.21	0.41	0.70	0.65	1.52	600
AHCLS	97.52	88.49 / 63.29	0.28	0.52	0.76	0.53	1.49	600
SNMF/L	97.46	89.89 / 49.26	0.09	0.16	0.75	0.55	0.36	90
GEO	95.29	90.84 / 57.21	0.20	0.22	0.75	0.55	0.11	0

In order to verify, if a deterministic initialization method could achieve better or similar results for this issue, the proposed seeding methods in combination with the *SNMF/L* algorithm have been applied. The results are given in Table 3. To compare the results to (W^*, H^*), the relative difference in term of the SNMF/L objective function $\frac{f(W,H)-f(W^*,H^*)}{f(W^*,H^*)}$, where $f(\cdot,\cdot)$ is equal to (2.15), the relative Frobineus norm difference between W^* and W, respectively H^* and H have been used. The initialization with deterministic strategies resulted in all cases apart from NICA2 in an almost identical solution to the one based on 1000 random initializations. Furthermore, it is not obvious if the basis matrix obtained with NICA2 method could result in a better separation of the corn and substrate, but since the (scaled) correlation coefficient were identical to the correlation coefficients obtained by the solution (W^*, H^*), hence there is no reason to prefer this method. To summarize the results, every deterministic initialization apart from NICA2 performed well as it resulted in a solution very close to the best solution obtained from random initialization and the computation time of the algorithm was reduced, despite of the increased iteration steps until convergence.

Table 3: Comparison to results of random initialization - NMF for meteorite "ochansk" results for SNMF/L with different deterministic initializations

	Difference measures				
Initializations	$\frac{f(W,H)-f(W^*,H^*)}{f(W^*,H^*)}$	$\frac{\|W^*-W\|}{\|W^*\|}$	$\frac{\|H^*-H\|}{\|H^*\|}$	CPU time (seconds)	niter
NNDSVD	0.0007	0.0004	0.001	0.14	105
SVD	0.0006	0.0004	0.0009	0.14	105
SPHERICAL	0.0008	0.0005	0.001	0.18	125
NICA1	0.0008	0.0005	0.001	0.17	120
NICA2	0.0007	1.34	1.05	0.16	110
NICA3	0.0008	0.0005	0.001	0.18	130

The dissimilarity between the two basis spectra $W^* = (w_1^*, w_2^*)$ obtained by *SNMF/L* can be better understood by the illustration of the two spectra in a mass spectrum (see Figure 4.3). The $L1$-error and $L2$-error between the two basis spectra are computed and strengthen the opinion of dissimilarity. The high discrepancy between $L1$-error and $L2$-error is caused by the few high peaks, which were emphasized by the $L2$-error. As in Figure 4.3 depicted the highest intensities (so-called peaks) are exclusively obtained for the first 100 masses. Consequently, a second plot of the spectra restricting to the first 100 masses is given in Figure 4.4.

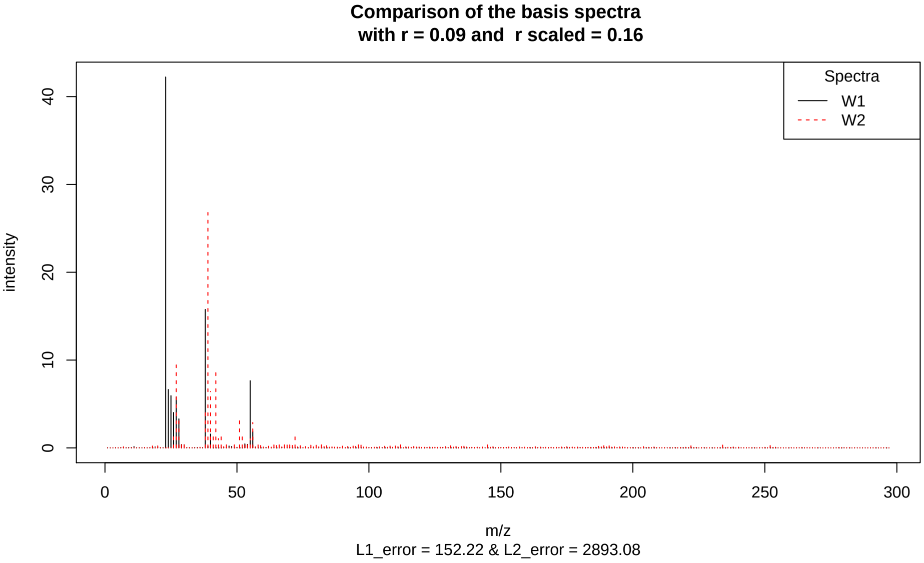

Figure 4.3: **Mass spectrum** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. This illustration enforces the already computed dissimilarity of the two basis spectra. In addition to the correlation coefficient the $L1$-error and $L2$-error are stated.

The 5 masses, which contribute the most to the spectrum of the respective basis showed an intersection only in one mass "m39"-Potassium, but for basis 2 the intensity is about 10% lower. The first basis consists mostly of the masses "m24","m25","m26","m39" and "m56", where the masses $24 - 26$ are all types of Magnesium with different numbers of neutrons, and mass 56 represents an iron isotope. In contrast, the second basis contains high intensities of "m28"-Silicon, "m39"-Potassium, "m40"-Argon, "m41"-Potassium and "m43"-Calcium.

The application of the NMF procedure would be considered a success if one of the basis spectra turns out be very similar to the spectra obtained from the substrate. Therefore, the basis spectra are compared to all substrates measured on the same target "4E1" (23 spectra in this sub-class of substrates). The results of this comparison are stated in Table 4 and indicate that the first basis w_1^* has little in common with all substrate spectra. In contrast, the second basis w_2^* can be considered similar to the substrates of the target "4E1" as it had to at least one of the substrate spectra a nearly perfect correlation of 0.98 respectively 0.94 for the scaled spectra. In addition, the $L1$- and $L2$-errors are at a low level for the second basis factor.

Table 4: Comparison to results of random initialization - NMF for meteorite "ochansk" basis spectra obtained by SNMF/L compared to substrate of target "4E1". Range denotes the interval between minimum and maximum of the respective similarity measure.

	Similarity measures			
Basis spectra	r range	r scaled range	L_1 range	L_2 range
W1	$[0.07, 0.19]$	$[0.02, 0.26]$	$[139.40, 157.70]$	$[2369, 2960]$
W2	$[0.83, 0.98]$	$[0.58, 0.94]$	$[25.96, 48.67]$	$[36.23, 295.02]$

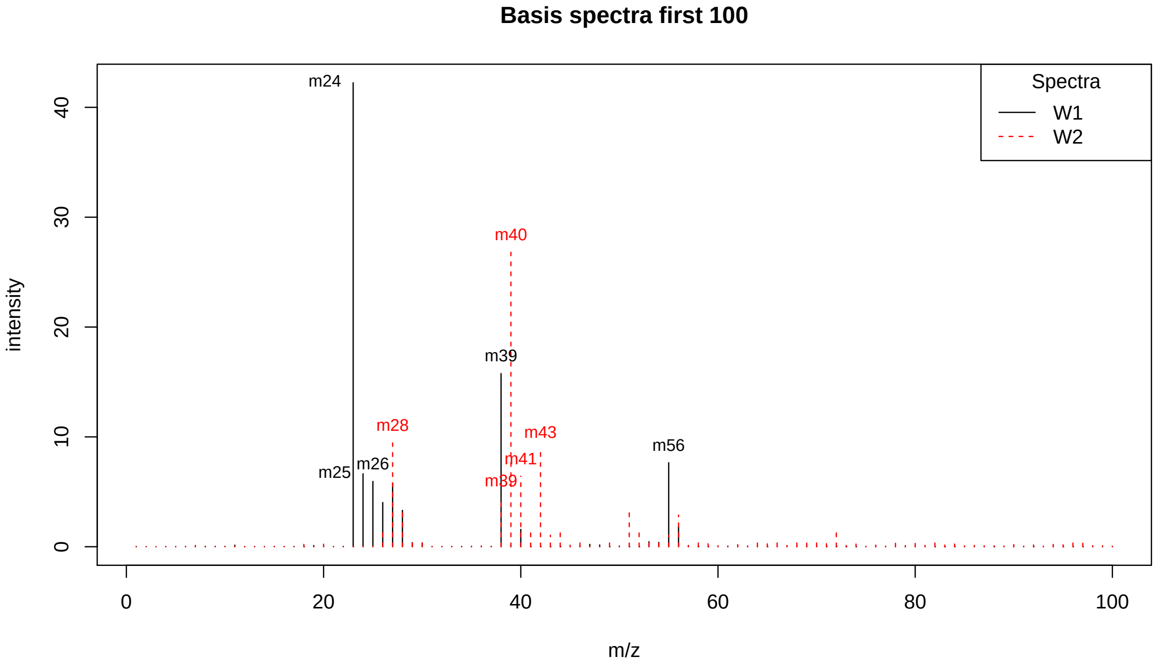

Figure 4.4: **Mass spectrum of first** 100 **masses** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. For every basis spectra the 5 ions, which contribute the most to the spectrum of the basis factor are labelled.

The mean spectra of the substrate sub-class "4E1" was used to illustrate the similarities respectively dissimilarities to the NMF basis spectra. In Figure 4.5 the mass spectrum of the NMF basis and the "mean" substrate were compared with restriction to the first 100 ion masses. As already noted within the first 100 ion masses the most peaks were observed. For the second basis w_2^* the peaks were observed on the identical position as the peaks of the mean substrate, hence the mass spectrum visually confirmed the quantitative results.

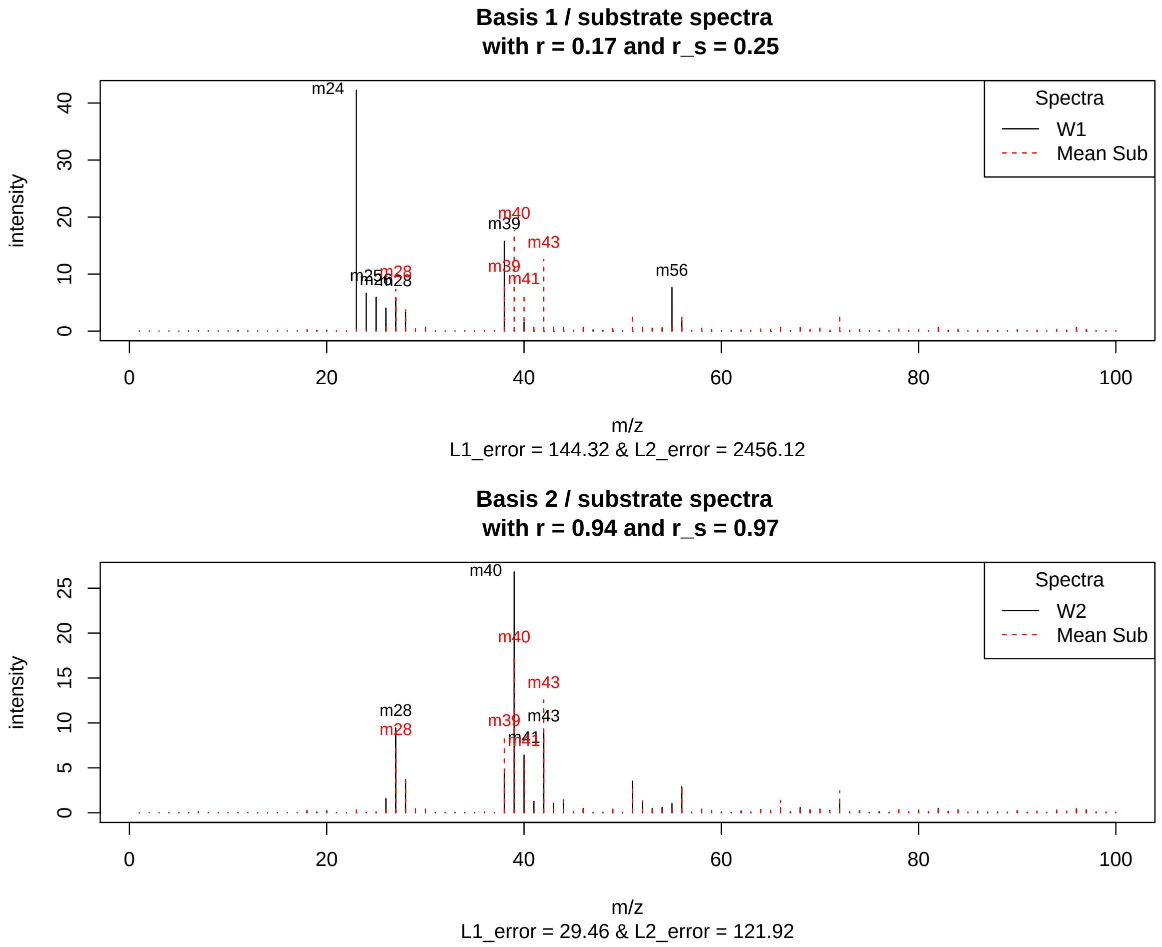

Figure 4.5: **Mass spectrum Basis vs Substrate** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. This illustration gives further justification on the high similarity between the second NMF basis and the mean substrate of the subclass "4E1". In addition to the correlation coefficient the $L1$-error and $L2$-error are stated, whereby r_s is denoted for *r scaled*.

Until now only the basis spectra have been analysed, but the weight matrix H^* is crucial for the definition of the cluster. The spectrum i belongs to the NMF cluster of the first basis, which is considered as the spectrum to represent the meteorite "ochansk", if the value of h_{1i} is bigger than the value of h_{2i}, the respective weight for spectrum i of the second basis. Otherwise, the spectrum i belongs to the NMF cluster of the second basis, which is considered as the spectrum to represent the substrate. In order to get further insight of the clustering structure the values of the optimal (normed) weight matrix H^* were plotted (see Figure 4.6). Due to the form of H^* it was revealed that some spectra such as the observations with id 25 and 45 could be reconstructed solely out of the second basis spectrum. For most cases a clearly dominating spectrum could be obtained, but for a few spectra almost equal weights of basis spectra 1 and 2 were obtained.

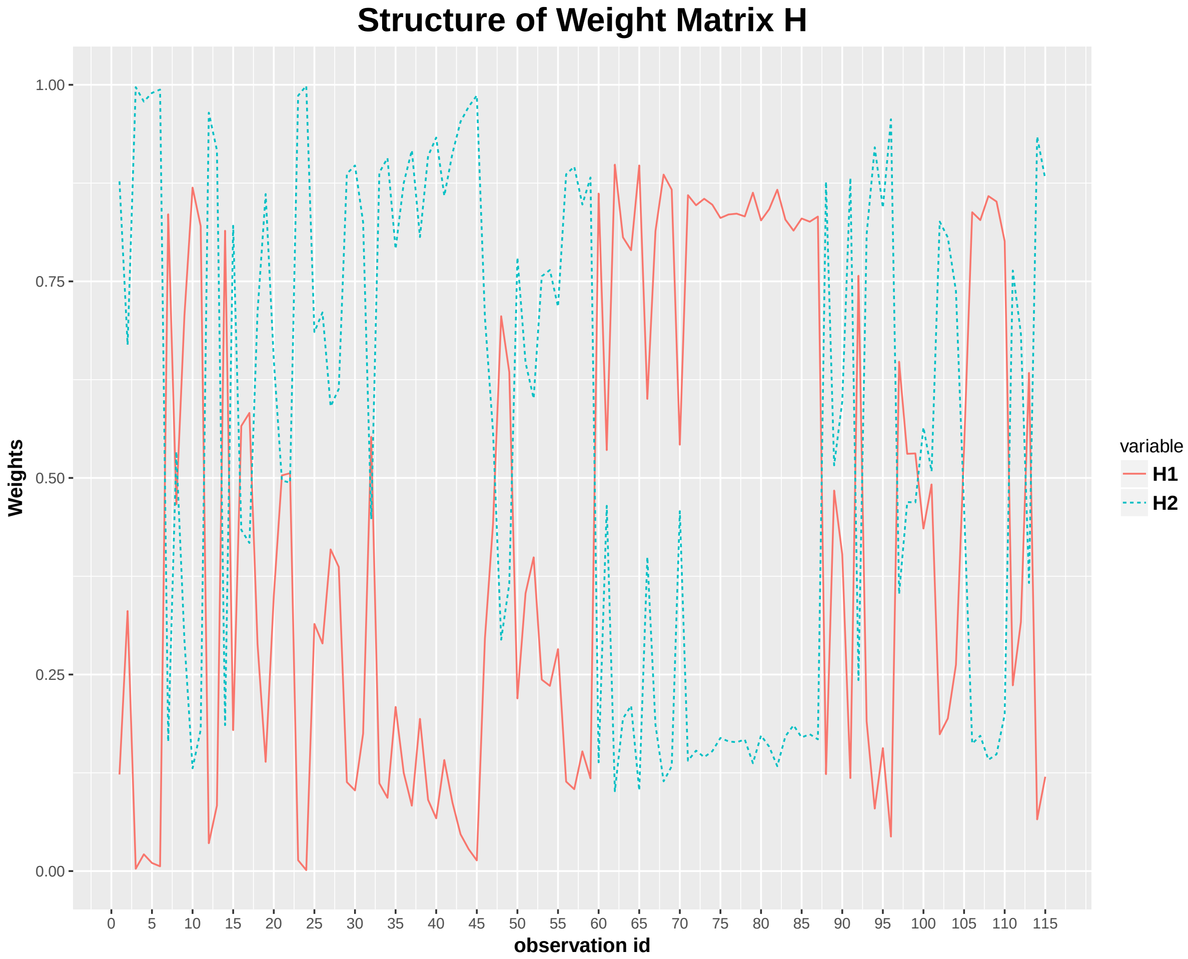

Figure 4.6: **Structure of the weight matrix** H^* - For every spectrum the two corresponding weights of the NMF basis spectra were pointed out. Considering this structure there are existing spectra, which are completely belonging to the corn and others that can be considered as substrate as they only consist of the second basis spectrum.

Since the exact x- and y-coordinates (on the specific target) of every measured spectrum are available the NMF clustering could be illustrated under the consideration of the spectra position. From the graphic of the target "4E1" (see Figure 4.2), the spectra belonging to the substrate should mainly be obtained in the upper right and lower left corner of the mesh grid. According to the graphs in Figure 4.7 the clustering imposed by the NMF factors achieved a better performance of locating the corn than the reference clustering (based on the intersection of PCA and kNN). The spectra with a high weight for the first basis are located in the center of the mesh grid and only low weights are considered for the spectra in the corners. In contrast to the reference cluster no corn specific spectra were located at the upper right corner as it should be considering the Figure 4.7.

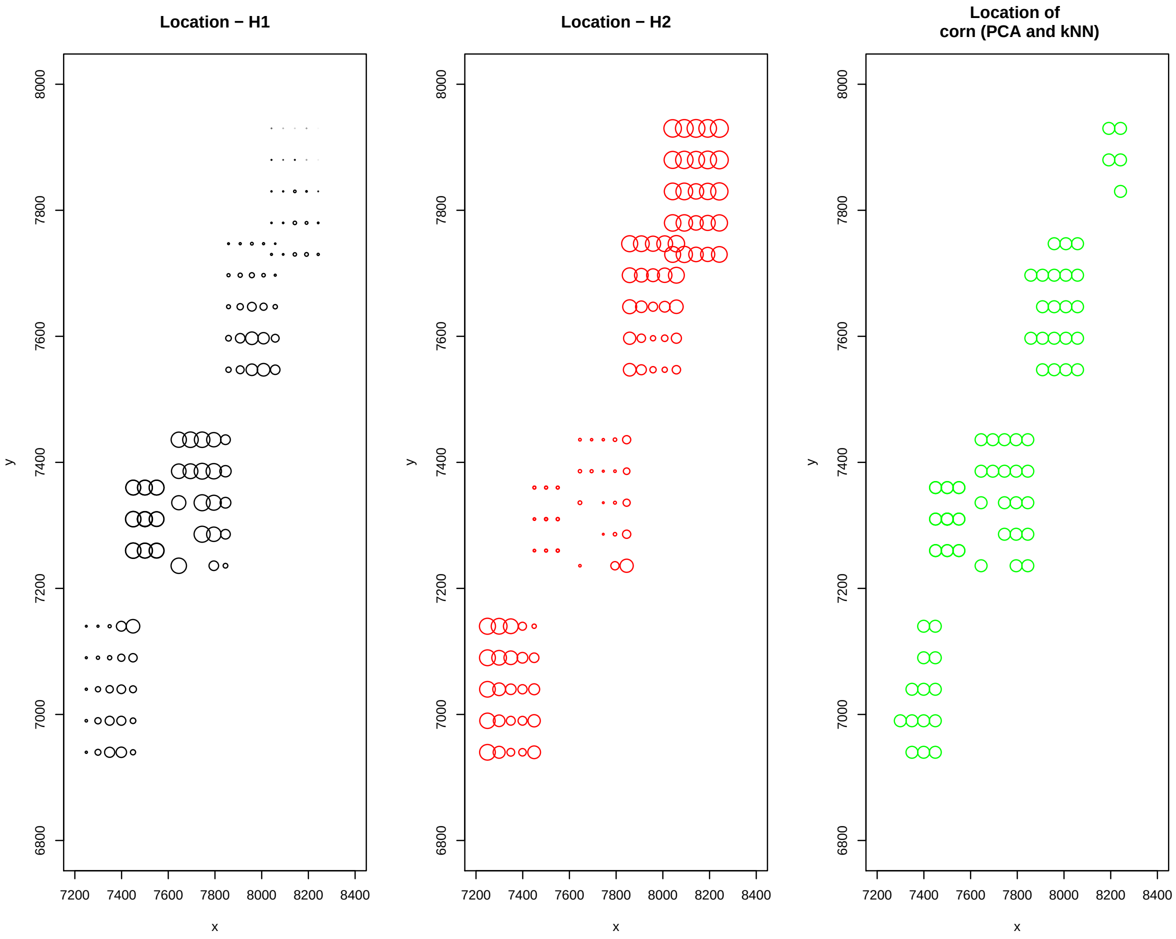

Figure 4.7: **Locating the corn "ochansk"** - The first plot depicted the x- and y- coordinate of the measured spectra by drawing a circle at the respective position, whereby the size of the circle depends on the weight of the first NMF basis spectrum assigned for this spectrum. The second plot can be interpreted in the same manner for the second NMF basis spectrum. In the last plot only the spectra, which are belonging to corn "ochansk" according to the clustering obtained by PCA and kNN, are marked with a circle.

4.3 MixUp Spectra with/without Noise

The NMF algorithms were applied to spectra, which have been constructed out of the addition of two known spectra. The NMF algorithms should identify these two known spectra as their basis spectra W and the weights of the positive linear combination should match the values of the weight matrix H.

The following two spectra were chosen to construct the spectra:

- ″**msub**″: the spectrum obtained by taking the mean of all ″substrates″ spectra, which will be referred to as in the subsequent analysis.
- ″**mall**″: the spectrum obtained by taking the mean of all spectra of meteorite ″allende″, which belong to a measurement on the corn according to the clustering obtained by PCA and kNN.

As these two spectra will be of further interest their mass spectra were plotted in Figure 4.8. This first impression indicated that the spectra can be regarded as similar in some way, which makes the separation of these spectra more complicated. The high *r scaled* is influenced by the fact that the major peak differences occur for the low masses.

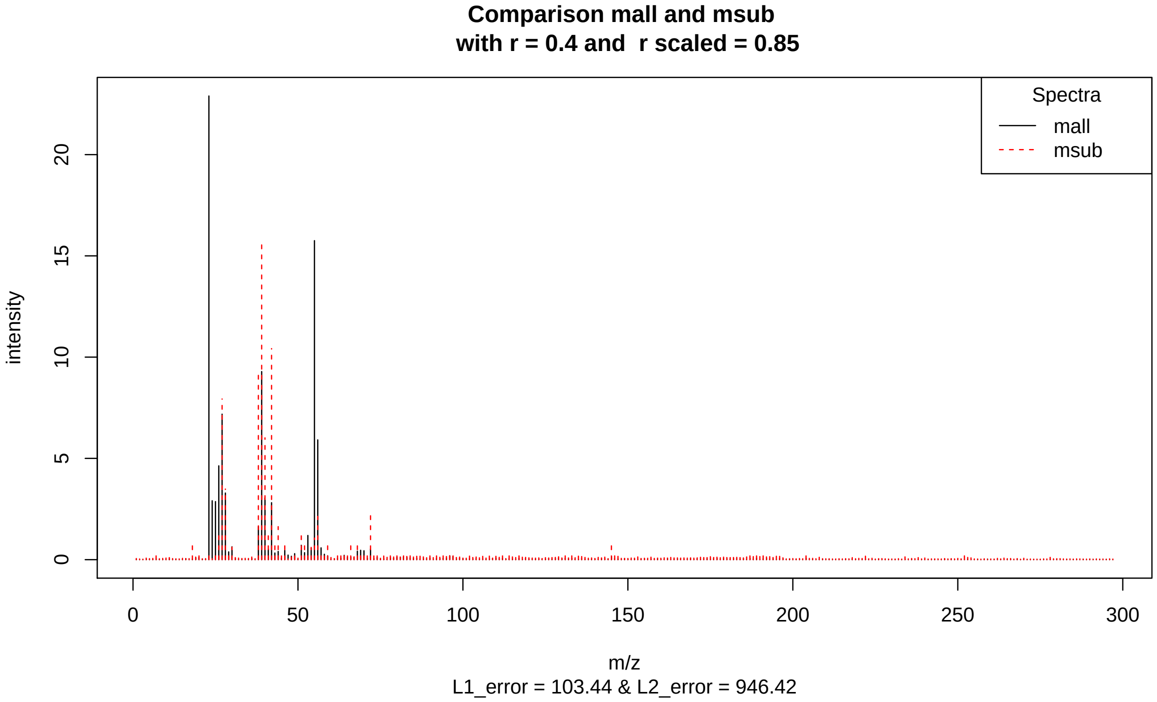

Figure 4.8: **Mass spectrum mall vs msub** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. In addition to the correlation coefficient the $L1$-error and $L2$-error are stated.

In order to examine the major peaks the mass spectra with restriction to the first 100 masses were plotted in Figure 4.9. The masses ″m28″-Silicon and ″m40″-Argon appeared to a high intense in both spectra. For the **mall** spectrum other high peaks are obtained at the ″m24″-Magnesium and ″m56″/″m57″-iron (different number of neutrons) isotopes. The other most dominant peaks of the **msub** spectrum are at ″m39/m41″-Potassium (different number of neutrons) and ″m43″-Calcium isotopes.

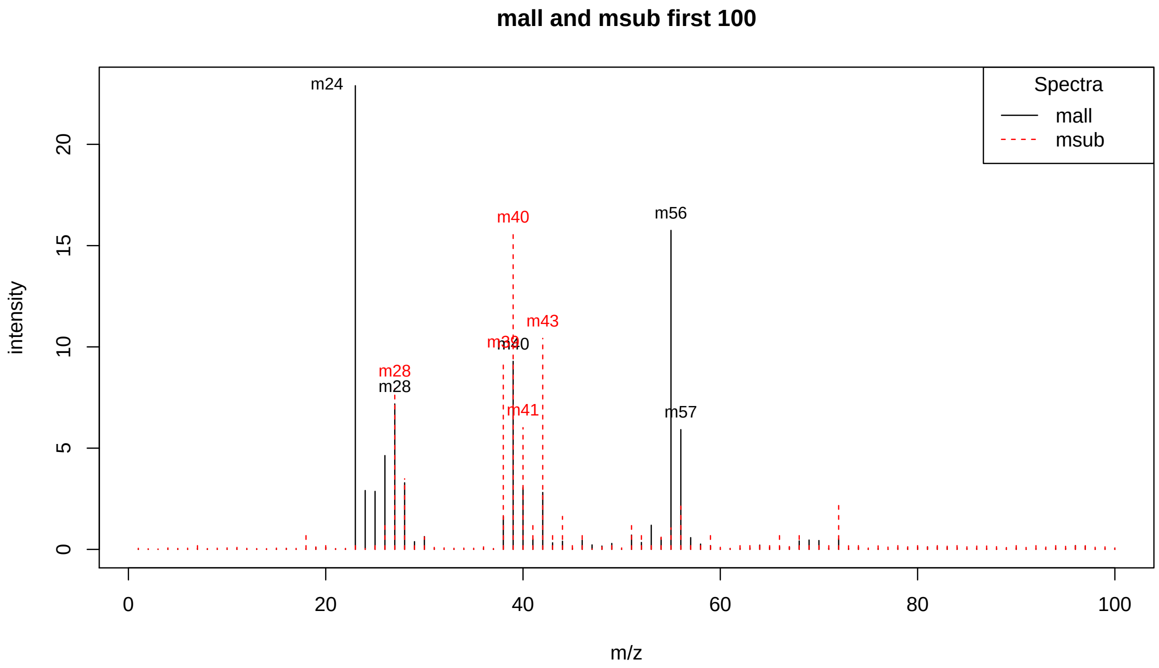

Figure 4.9: **Mass spectrum mall vs msub first** 100 **masses** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. For every spectrum the ions, which contribute the most to the spectrum are labelled.

In the upcoming sections these spectra will be used as reference to analyse the performance of the NMF algorithms.

4.3.1 Mix - without Noise

A positive combination of the spectra **msub** and **mall** without any noise was suggested to be well reconstructed by all the NMF algorithms. In this section it was analysed if this suggestion was correct. The following linear mixing structure was used to construct 50 spectra $S = (s_1, \dots, s_{50})$:

$$s_i = \alpha_i * mall + (1 - \alpha_i) * msub, \tag{4.1}$$

where the coefficient $\alpha_i = 0.01 + (i - 1) * 0.02$ reflects the weight of the **mall** spectrum used for the constructed spectra.

In Table 5 the results achieved by the NMF algorithms considering their optimal solutions (with respect to the underlying objective function) from 1000 random initialization are stated. The coefficients r_1 and r_2 are the correlation coefficients between the estimated NMF basis spectra and the true basis spectra **mall** and **msub** (respectively r_{s1} and r_{s2} for the scaled variants). All algorithms performed well with almost classifying all spectra correctly and estimating basis spectra that are (nearly) perfectly correlated to the true basis spectra. Nevertheless, only the GEO algorithm and SNMF/L classified all spectra correctly, but as the correlation of the scaled spectrum of the first SNMF/L basis to the scaled spectrum of **mall** is low compared to the other with 0.86, the GEO algorithm will be preferred. The good performance of the GEO algorithm in this case is no surprise as the underlying data matrix S is almost separable.

Table 5: Quality measures - NMF of S, results are based on 1000 random initializations

	Quality measures						
Algorithms	Evar (%)	r_1/r_2	r_{s1}/r_{s2}	purity	entropy	CPU time (seconds)	niter
LEE	1	0.99 / 0.99	0.99 / 1	0.98	0.12	0.25	440
BRUNET	1	0.99 / 0.99	0.99 / 0.99	0.96	0.21	0.24	430
ALS	1	0.99 / 0.99	0.99 / 0.99	0.94	0.27	1.05	600
AHCLS	0.99	0.99 / 0.99	0.99 / 0.99	0.98	0.12	0.92	600
SNMF/L	0.99	0.98 / 0.99	0.86 / 0.99	1	0	0.24	95
GEO	1	0.99 / 0.99	0.99 / 0.99	1	0	0.17	0

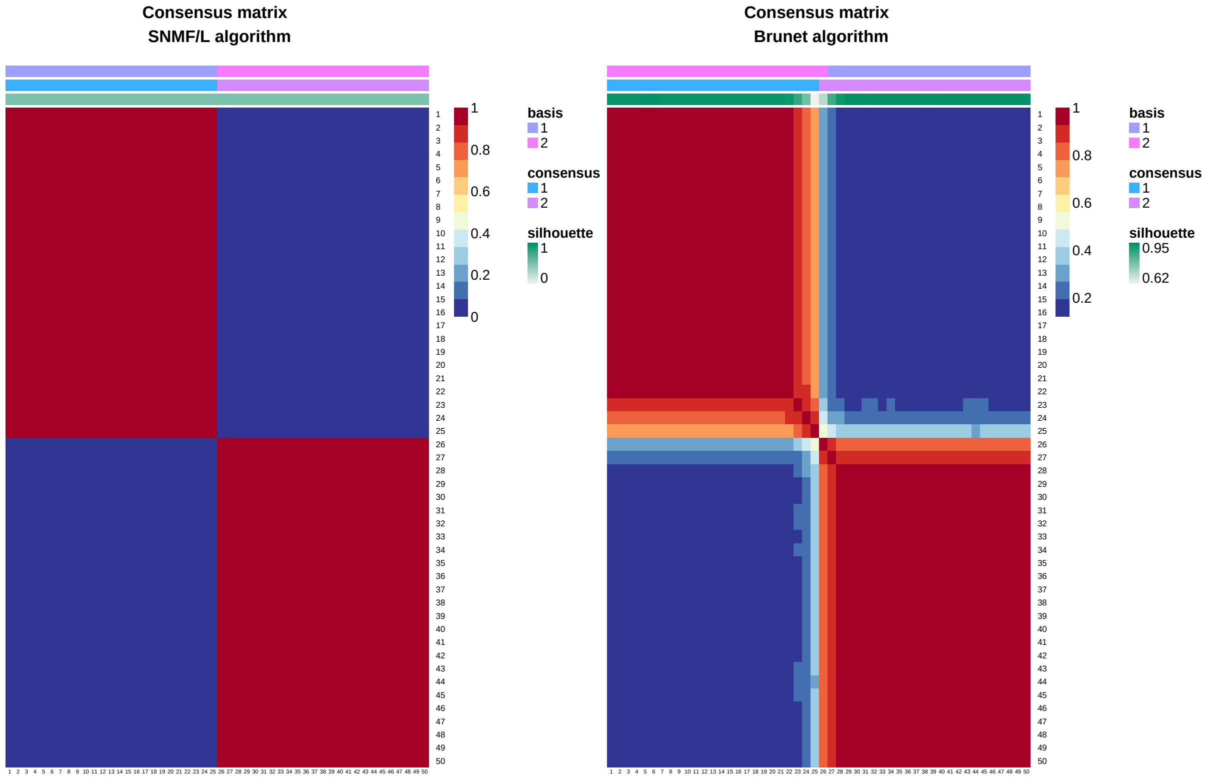

Figure 4.10: **Consensus matrix** - A heat map based on the consensus matrix obtained from the 1000 random initializations for the NMF algorithms BRUNET and SNMF/L. While SNMF/L generates stable clusters, the BRUNET algorithm varies greatly in the cluster assignments of the spectra, which are influenced almost equally by the two constitutive spectra.

The stability of the cluster assignments can depend on the chosen random initialization (since different stationary points potentially lead to different cluster assignments), despite for the GEO algorithm which does not need any initialization of the basis or weight matrix. This cluster stability can be analyzed by a so called *consensus map* (see Gaujoux and Seoighe, 2010), which creates a heatmap on the basis of the consensus matrix. As Figure 4.10 shows the BRUNET algorithm has issues to assign the correct clusters for the spectra, which are almost equally influenced by **mall** and **msub**, whereas SNMF/L reproduces under every random initialization (1000) the correct clusters. Furthermore, for the clusters assigned by LEE and AHCLS a similar instability as for BRUNET can be obtained (ALS even shows severe instability).

To complete this analysis, the structure of the optimal weight matrix H^* obtained by the GEO algorithm is depicted in Figure 4.11. It confirms the already stated arguments of achieving nearly perfect reconstruction and in addition the relative error of H^* ($\frac{\|H^*-H\|}{\|H\|}$ where H is the matrix containing the true coefficients α_i and $1-\alpha_i$ in the i-th column) is on a low level.

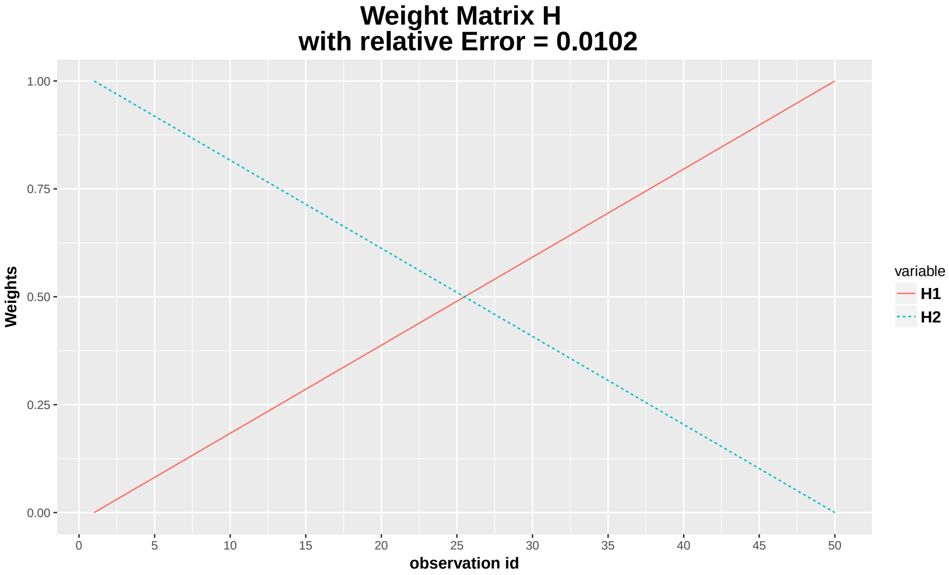

Figure 4.11: **Structure of the weight matrix** H^* - For every spectrum the two corresponding weights of the NMF basis spectra were pointed out. The linear structure can be almost perfectly reconstructed with a low relative error of H^*.

4.3.2 Mix - with Additive Noise

In this section the robustness to noise of the NMF methods is analyzed. For this reason the spectra S constructed by (4.1) are perturbed with noise. Every noise spectrum is formed by a random poisson vector (of length 297) with $\lambda = mean(mall)$ and normed to sum up to 70 (as mentioned in Section 4.1 the ordinary spectra are normed to 100). The normalization of the noise spectra to 70 should simulate the lesser importance of noise compared to the ordinary spectra. For every spectrum s_i a noise spectrum n_i is added. The perturbed spectra $\tilde{s_i}$ are represented in the following way:

$$\tilde{s_i} = s_i + n_i \quad \forall i = 1, \ldots, 50. \tag{4.2}$$

In Table 6 the obtained results by the NMF methods applied to these perturbed spectra $\tilde{S}$ are stated. All algorithms still perform well in terms of their cluster ability with a high purity of 0.98 and low entropy of 0.12, except for the SNMF/L method, which had a slightly lower purity of 0.96 and higher entropy of 0.24. The two constitutive spectra still can be reconstructed appropriately with correlation coefficients r_1 and r_2 between 0.96 and 0.99, however the correlation coefficients of the scaled spectra fell off, what indicates that the methods, especially GEO, have trouble to distinguish between the noise and the constitutive spectra in the higher masses. This effect occurs since the intensities for the higher masses are on a very low level for the spectra **mall** and **msub**. Considering these results and the good cluster stability (see **??**) achieved by AHCLS, it can be regarded as the best NMF method (W^*, H^*).

Table 6: Quality measures - NMF of $\tilde{S}$, results are based on 1000 random initializations

	Quality measures						
Algorithms	Evar (%)	r_1/r_2	r_{s1}/r_{s2}	purity	entropy	CPU time (seconds)	niter
LEE	0.93	0.99 / 0.99	0.78 / 0.87	0.98	0.12	0.35	840
BRUNET	0.93	0.99 / 0.99	0.75 / 0.87	0.98	0.12	0.41	460
ALS	0.93	0.99 / 0.99	0.78 / 0.86	0.98	0.12	0.97	600
AHCLS	0.93	0.99 / 0.99	0.79 / 0.87	0.98	0.12	1.06	600
SNMF/L	0.93	0.98 / 0.99	0.67 / 0.86	0.96	0.24	0.36	95
GEO	0.89	0.96 / 0.96	0.40 / 0.66	0.98	0.12	0.16	0

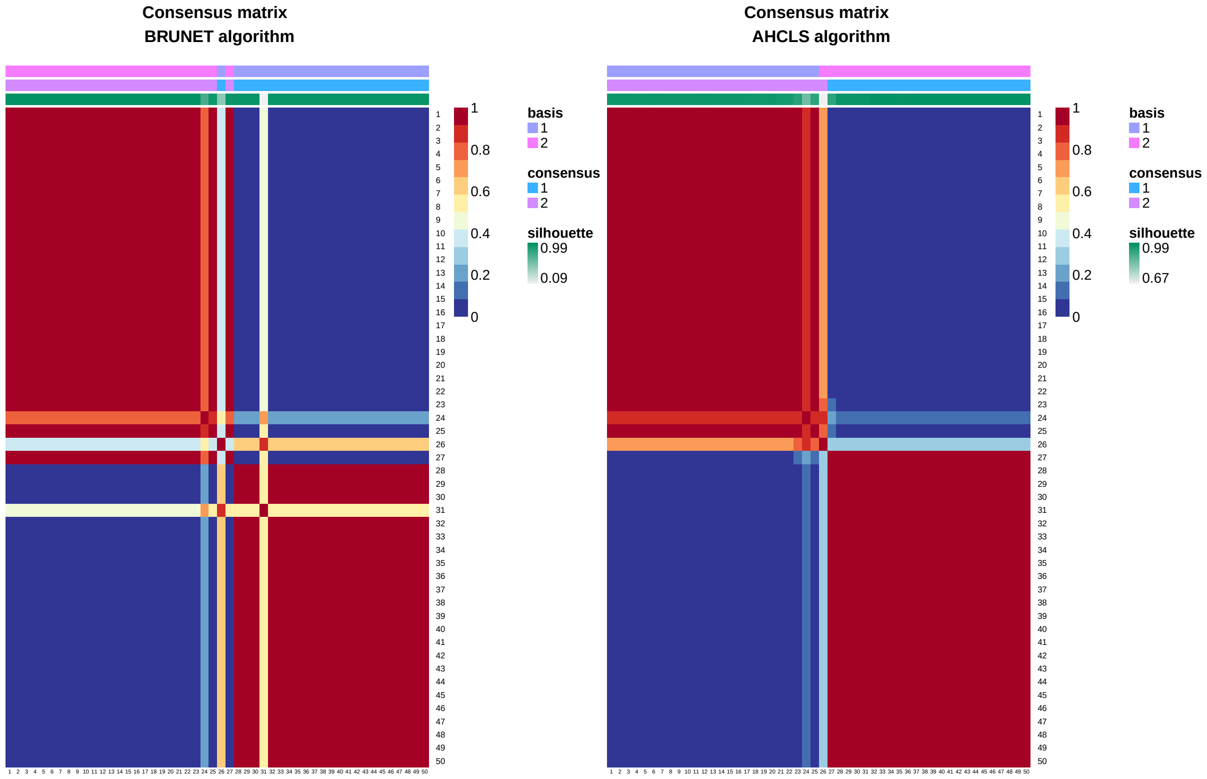

Figure 4.12: **Consensus matrix (with noise spectra)** - A heat map based on the consensus matrix obtained from the 1000 random initializations for the NMF algorithms BRUNET and AHCLS. AHCLS produces stable clusters (despite of two spectra which tend to be difficult to classify) compared to BRUNET, where some instability could be observed.

The estimated basis spectra of AHCLS and the spectra of **mall** and **msub** are depicted in Figure 4.13 to get a better impression of their similarities. a As this figure shows, adaptation to constitutive spectra **mall** and **msub** is good, as all peaks have been reconstructed by the basis spectra.

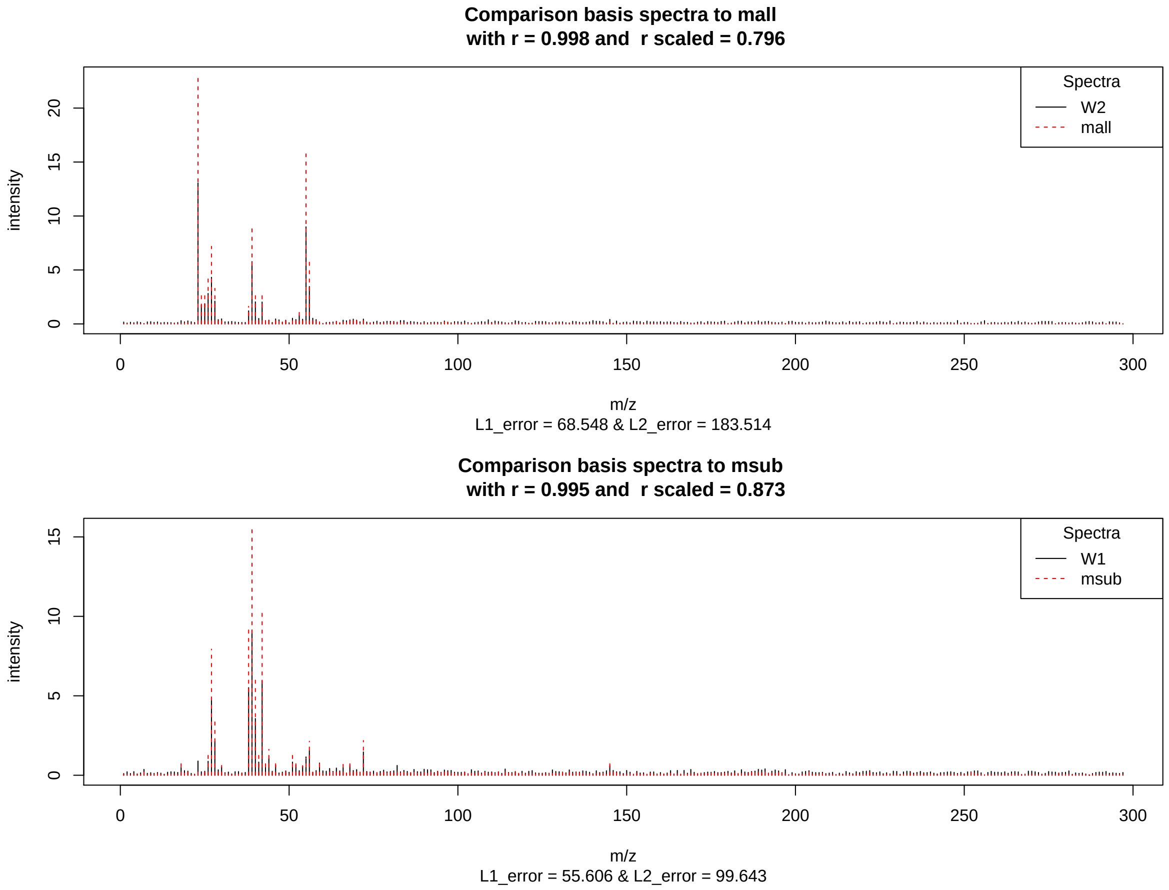

Figure 4.13: **Mass spectrum** W^* **vs mall and msub** - The typical form of a mass spectrum is used with ion mass to charge ratios (m/z) on the x-axis and their relative intensity on the y-axis. In addition to the correlation coefficient the $L1$-error and $L2$-error are stated.

It remains to examine the structure of H^* and to what extent the noise influenced the structure. As shown in Figure 4.14 noise can be treated adequately and the linear structure can be well preserved. The relative error of H^* increased compared to the noiseless case, but is still acceptable.

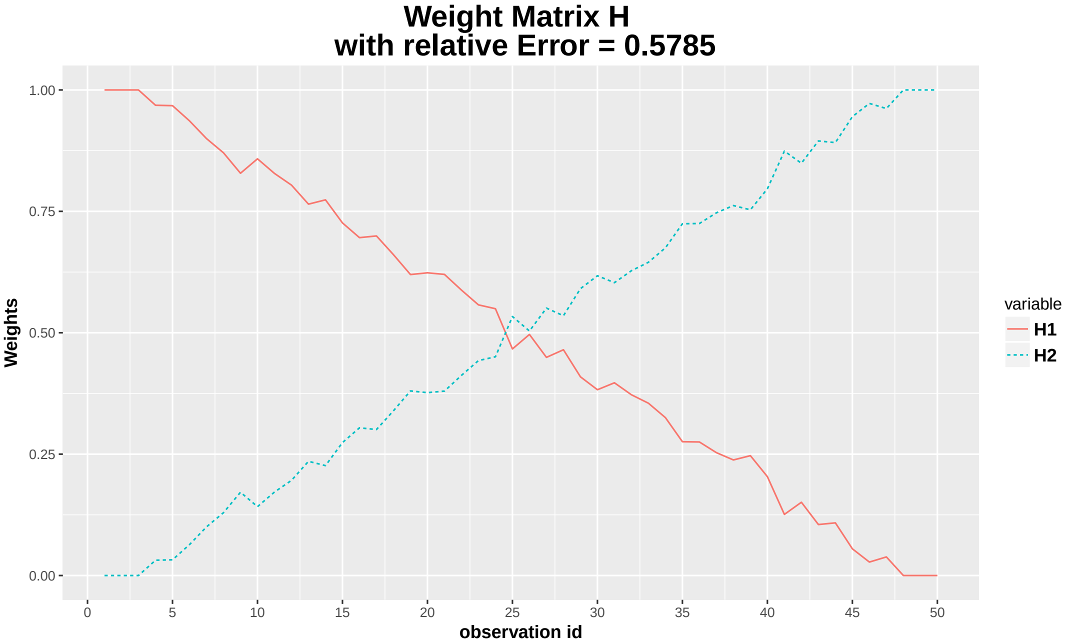

Figure 4.14: **Structure of the weight matrix** H^* - For every spectrum the two corresponding weights of the NMF basis spectra were pointed out. The linear structure is still observable for H^* (from the best AHCLS solution), but not as smooth as in the noiseless case.

Noise on the coefficients

The scenario (4.2) can be further adjusted by considering a perturbation of the linear coefficients α_i, hence the following imprecision for α_i (respectively $1 - \alpha_i$) is considered:

$$\tilde{\alpha}_i = \alpha_i * u_i, \tag{4.3}$$

where u_i is a random variable which is uniformly distributed in the interval $[0.5; 2]$.

The performance of the NMF method under this additional perturbation are summarized in Table 7. The reconstruction of the two constitutive spectra **mall** and **msub** nearly not changed as it would be expected. The cluster assignment is clearly influenced by this sort of perturbation, but still can be considered high with a purity of 0.92 and an entropy between 0.34 and 0.40. As could be expected the spectra, which are almost equally influenced by the spectra **mall** and **msub** tended to be misclassified (see Figure 4.15).

Table 7: Quality measures - NMF of $\tilde{S}$ with noisy $\tilde{\alpha}_i$, results are based on 1000 random initializations

	Quality measures						
Algorithms	Evar (%)	r_1/r_2	r_{s1}/r_{s2}	purity	entropy	CPU time (seconds)	niter
LEE	0.96	0.99 / 0.99	0.75 / 0.89	0.92	0.34	0.2	430
BRUNET	0.95	0.99 / 0.99	0.74 / 0.88	0.92	0.39	0.39	600
ALS	0.96	0.99 / 0.99	0.75 / 0.89	0.92	0.34	1.08	600
AHCLS	0.96	0.99 / 0.99	0.77 / 0.89	0.92	0.34	1.06	600
SNMF/L	0.96	0.98 / 0.99	0.62 / 0.89	0.92	0.34	0.19	80
GEO	0.94	0.99 / 0.98	0.45 / 0.67	0.92	0.40	0.16	0

Similar to the previous scenario the AHCLS showed the best performance considering these quality measures. The basis and weight matrix of the best AHCLS solution will be denoted as (W^*, H^*).

Since this perturbation strategy influences the cluster assignments it is of interest to analyze the weight matrix H^*. Therefore, the structure of H^* is plotted in Figure 4.16. In contrast to the previous perturbation scenario, the linear structure is not so clear, but a linear trend can still be identified. It is not surprising that the relative error of H increases further, but it is still quite small.

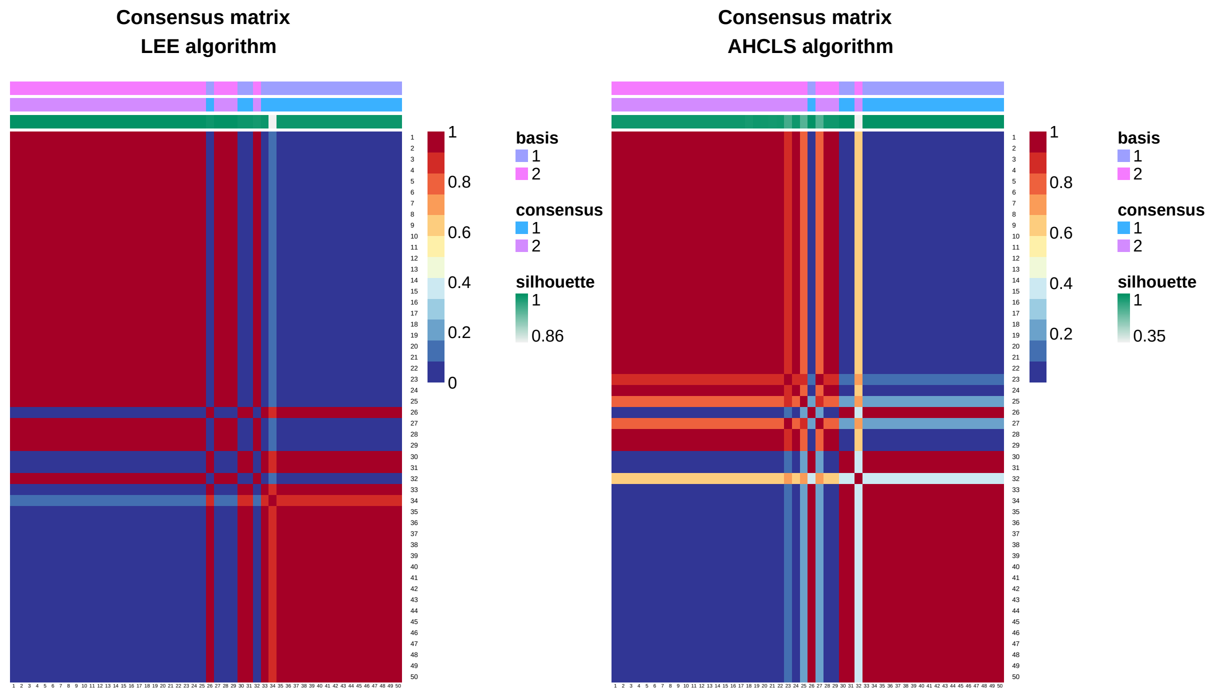

Figure 4.15: **Consensus matrix** - A heat map based on the consensus matrix obtained from the 1000 random initializations for the NMF algorithms LEE and AHCLS. Compared to the previous analysis the clustering is not that stable any more. The spectra influenced by both spectra almost equally tended to be wrong classified.

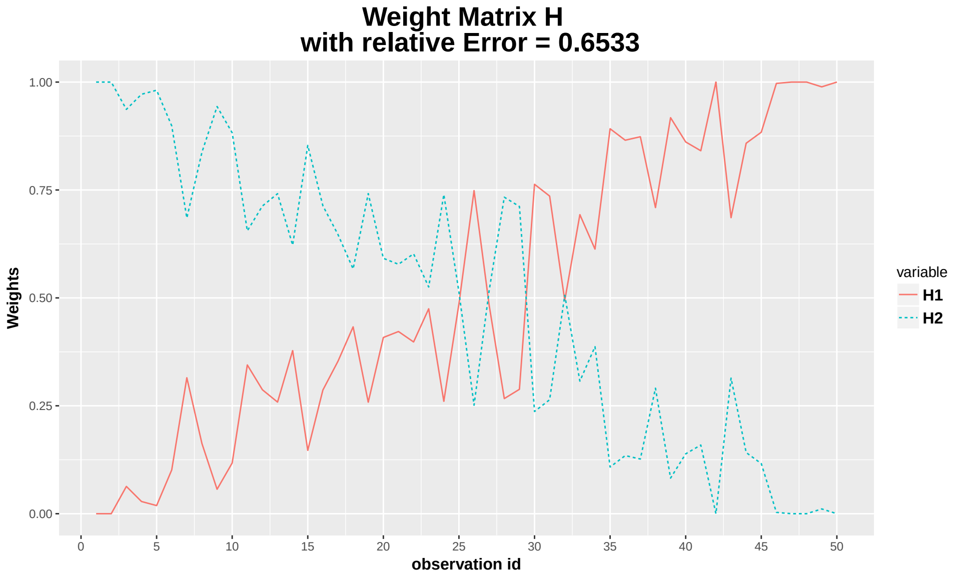

Figure 4.16: **Structure of the weight matrix** H^* - For every spectra the two corresponding weights of the NMF basis spectra were pointed out. The linear structure is still observable, but the perturbation is clearly visible.

4.4 Rank Estimation "allende"

In this section a brief example is given how a rank estimation for CoMeCS data could be performed. Consider for instance the task to identify the meteorite specific spectra of meteorite "allende", where 213 spectra were measured from 4 corns and on 2 different targets. Hence, it is not clear if the spectra are homogeneous or if they differ for every corn. Therefore, a factorization of only two basis spectra, one resembling the meteorite and one the substrate could deliver misleading results. In order to decide, which factorization rank r is needed, the proposed methods of Section 3.3 were applied to the data of meteorite "allende". The NMF method SNMF/L with 30 random initializations was used to estimate the NMF's for the ranks from 2 to 7. In the following, the suggested rank according to the respective method based on the results depicted in Figure 4.17 and Figure 4.18 is listed:

- **Brunets method**: The factorization rank of 3 was suggested as optimal since the cophenetic correlation decreased the first time from 1 to 0.99 for a rank 4 factorization.
- **Hutchins method**: The inflection point of the rSS curve is detected at rank 3 and therefore rank 3 considered to be optimal.
- **Frigyesi and Höglund method**: As shown in Figure 4.18 the NMF of the randomized data resulted in very high rSS, which makes the detection of a decrease in the rSS curve of the not randomized impossible by a simple glance on the plot.

The SVD-based rule uses only the singular value decomposition of the data and would suggest a factorization rank of 69, since the first 70 singular values were necessary to achieve more than 90% of the total sum of singular values. This high number for the factorization is rather impractical and as shown in Figure 4.17 the clusters start to be unstable after rank 3.

These methods only provide a guideline for the estimation, but the final decision has to be made by the user and is often application dependent. In the obtained example also rank 4 could be a useful factorization rank considering the high increase of evar, the still high cophenetic correlation of 0.995, but the rather low silhouette coefficients for the basis factors and the weight factors (coefficients) indicate instability of the cluster assignments. In conclusion, the rank 4 NMF factorization is less stable in context of clusters, but is significantly better approximating the spectra of meteorite "allende" than the rank 3 NMF factorization.

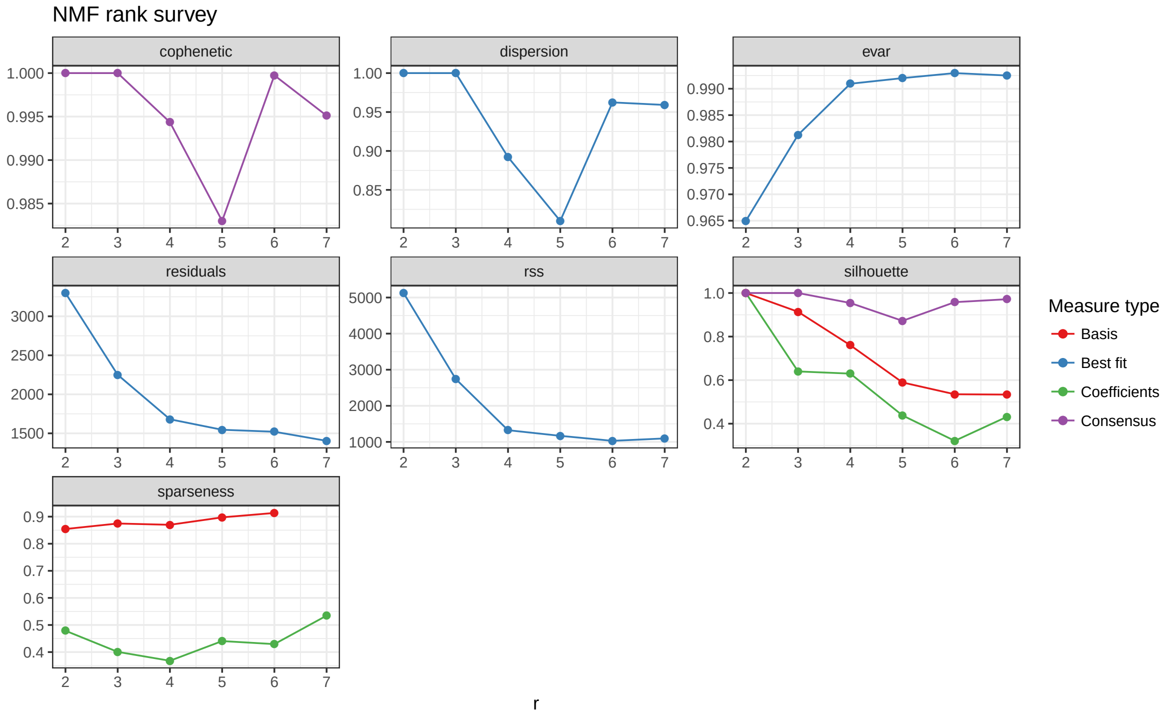

Figure 4.17: **Rank survey for "allende"** - The rank survey for the data based on meterorite "allende" spectra is illustrated. Each point was obtained from 30 runs of the SNMF/L algorithm with the respective rank *r* performed on "allende" spectra. The measure type "best fit" corresponds to the quality measures obtained by the best model obtained from the 30 runs. "Basis" respectively "Coefficients" are the corresponding basis factors respectively weight factors of the best model. The points of "Consensus" are the values of the consensus matrix obtained by the 30 runs of the SNMF/L algorithm.

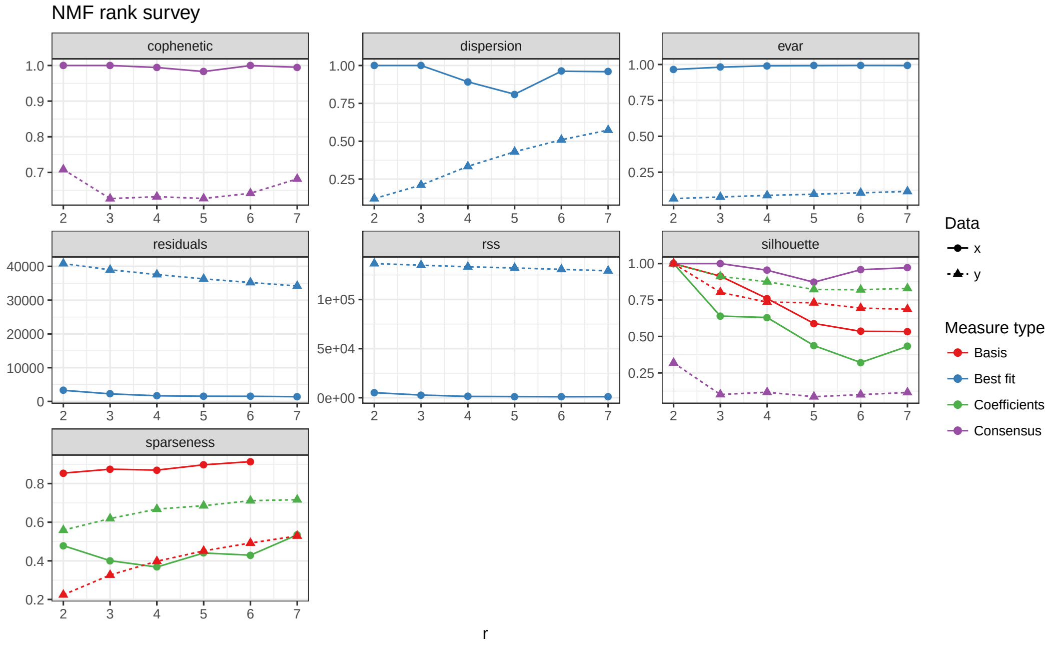

Figure 4.18: **Rank survey for "allende"** - The rank survey for the data based on meterorite "allende" spectra is illustrated. Each point (x-Data) was obtained from 30 runs of the SNMF/L algorithm performed on "allende" spectra (identical to Figure 4.17) and each triangle (y-Data) was obtained from 30 runs of the SNMF/L algorithm performed on the randomly permutated data matrix (consisting of "allende" spectra). The random permutation of entries destroys the structure of the spectra, hence NMF results in a bad approximation since no latent structure could be obtained.

To gain further insight of the cluster structure the consensus maps of the NMF factorizations with rank 3 and 4 are compared (see Figure 4.19). The clusters of the factorization with rank 4 are still pretty well separated, however since one cluster is very small the rank 3 factorization seems more appropriate. It should be mentioned that the obtained cluster structures of the rank 3 and 4 factorizations are not very similar. In conclusion, both factorizations could provide useful basis factors, but the rank 3 factorization seems to be more reliable for this case.

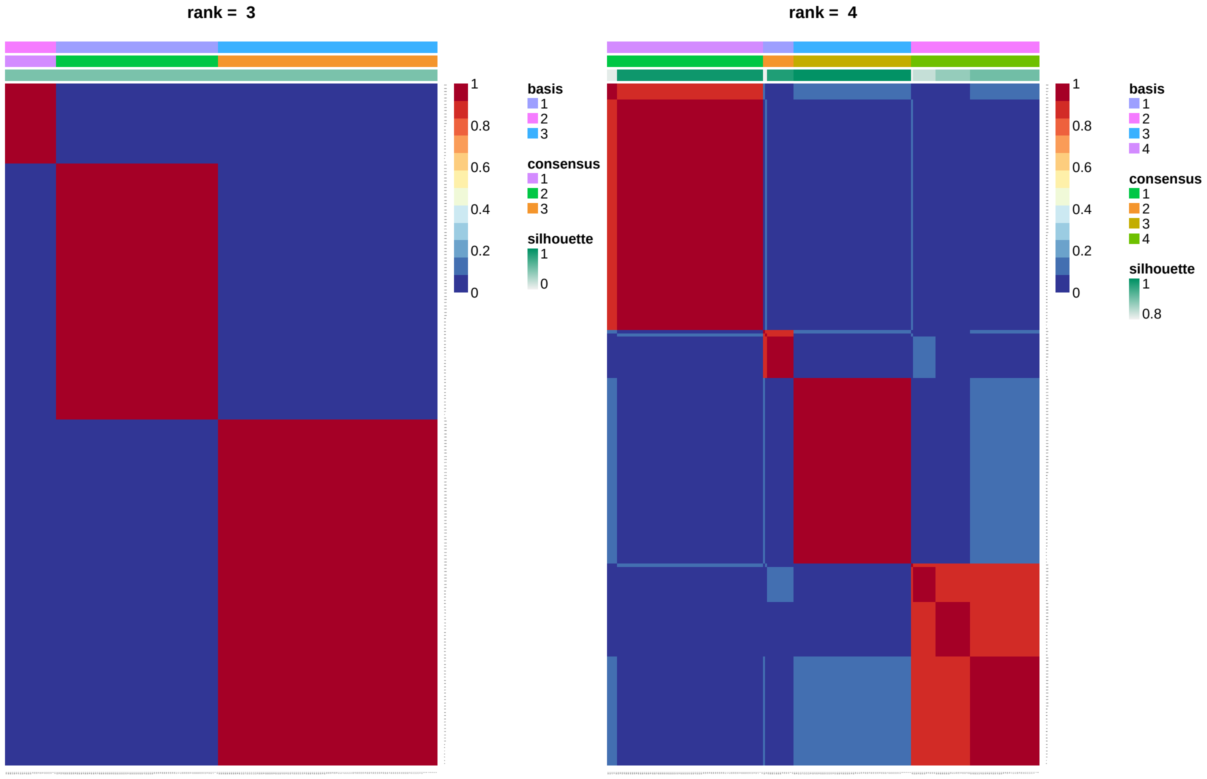

Figure 4.19: **Consensus matrix** - A heat map based on the consensus matrix obtained from the 30 random initializations for the NMF algorithm SNMF/L with factorization rank 3 and 4. It can be obtained that the clusters for rank 4 can still be clearly identified, however they are not very similar to the cluster obtained by rank 3.

4.5 Conclusion

In the first numerical experiment carried out in this section it was tried to separate the spectra of a meteorite from the spectra of substrate. All tested NMF methods achieved acceptable results. The GEO algorithm, which is based on the assumption of separability, performed better than most of the other methods. Only the SNMF/L method estimated basis spectra, which are more dissimilar than the basis spectra of GEO. These good results for the GEO algorithm were not expected, since the assumption of separability is violated. Moreover, the application of the various deterministic initialization methods in combination with the SNMF/L method resulted in solutions, which have been very close to the best solution obtained from 1000 random initializations. Therefore, the use of such initialization methods would be preferred, as it produces without the need of any replication reasonable results. One basis spectrum of the best NMF (application of SNMF/L algorithm) has shown to be very similar to the typical spectrum of substrate, while the other basis spectrum was considered as dissimilar to the spectrum of substrate. In addition, the obtained weight matrix was used to compare the clustering results of NMF more closely, in particular with the clusters based on kNN and PCA. This analysis showed that the clustering imposed by NMF performs better than the reference clustering for the task of locating the corn (see Figure 4.7).

The simulations, where two spectra known to be from a meteorite or respectively from a substrate, have been used to simulate the data by a positive combination of these two spectra. In the noiseless case all algorithms performed very well. Especially, the GEO and SNMF/L algorithms were able to perfectly reconstruct the classification. Then noise was added to test the robustness to noise of the NMF methods. The results in the case of noise were still very good, as the basis spectra were very similar to the two constitutive spectra and the linear weighting scheme was still observable.

In the last section, the challenge of estimating the optimal factorization rank was investigated and it was shown that in the case of spectra observed by multiple corns of the same meteorite, such rank estimation could be advantageous. It provides information that the spectra of the different meteorites are possibly not that homogeneous.

In conclusion, the simulations carried out showed that NMF algorithms could be useful for separating spectra in order to identify, for example, the spectra of a meteorite corn.

Appendix

R Code

At first the R Code of all implemented initialization methods are stated. These functions could be used as seeding methods for the **nmf**-function in the NMF package. They all have the following two parameters in common: (i) target, which denotes for the target matrix that is required to be factorized, (ii) model, which denotes for the NMF model and contains information such as the used factorization rank.

```
############### ACOL ##################
## p      .... the number of columns to be used for averaging,
   default= n/2 , n...number of columns in the target matrix

rand_ACOL <- function(model, target, p = 1) {

  p_a <- NULL
  m <- dim(target)[1]
  n <- dim(target)[2]
  r <- nbasis(model)
  j <- 1
  W <- matrix(0,m,r)

  if(p == 1)
  {
    p_a =  round(n/2)
  }
  else {
    p_a = p

    }
  cat("It was chosen to take p=",p_a,"as number of columns to average")

  while(j <= r)
  {
    rand_col <- sample(1:n,p_a)
    a_col <- apply(target[,rand_col],1,mean)
    W[,j] <- a_col
    j <- j + 1
  }

  # initialize basis matrix W  according to ACOL with every column
  constructed of p randomly chosen column of the target matrix
  basis(model) <- W
  # initialize weight matrix H  by fast NNLS algorithm from Van Benthem
  coef(model) <- fcnnls(W,target)$x
  # return updated object
  return(model)
}
# Register ACOL init
setNMFSeed('ACOL', rand_ACOL, overwrite=TRUE)

########### random C #################
# q .... number of longest columns to be considered

rand_C <- function(model, target, p = 1, q = 5) {

  p_a <- NULL
  q_a <- NULL
  m <- dim(target)[1]
  n <- dim(target)[2]
  r <- nbasis(model)
  j <- 1
  W <- matrix(0,m,r)

  if(p == 1)
  {
    p_a =  round(n/2)
  }
  else { p_a = p}
  cat("It was chosen to take p=",p_a,"as number of columns to average \n")
  if( q == 5)
  {
    q_a = 5
  }
  else{ q_a = q}

  norms_target <- apply(target,2,function(x){sum(x^2)})
  names(norms_target) <- 1:n
  sort_norm <- sort(norms_target,decreasing = TRUE)
  q_col <- as.numeric(names(sort_norm))[1:q_a]
  cat("longest columns:",q_col)

  while(j <= r)
  {
    rand_col <- sample(q_col,p_a,replace = TRUE)
    a_col <- apply(target[,rand_col],1,mean)
    W[,j] <- a_col
    j <- j + 1
  }

  # initialize basis matrix W  according to ACOL with every column
  constructed of p randomly chosen column of the target matrix
  basis(model) <- W
  # initialize weight matrix H  by fast NNLS algorithm from Van Benthem
  coef(model) <- fcnnls(W,target)$x


  # return updated object
  return(model)
}
# Register C init
setNMFSeed('Rand_C', rand_C, overwrite=TRUE)



######## SVD seeding ##################
SVD_seeding <- function(model, target) {

r <- nbasis(model)
svd_target <- svd(target)

# initialize basis matrix W with  absolute value
of first k columns of U matrix from SVD

basis(model) <- abs( svd_target$u[,1:r])

# initialize weight matrix H with absolute value
of first k rows of singualr values times right singular vectors

coef(model) <- abs( diag(svd_target$d)[1:r,] %*% t(svd_target$v))

# return updated object
return(model)
}
# Register SVD init
setNMFSeed('SVD', SVD_seeding, overwrite=TRUE)

######## Spherical k-means seeding ##################
Sp_kmean <- function(model, target) {

  if( !require.quiet('skmeans') )
    stop("Seeding method 'Spherical' requires package
    ‘skmeans‘ to be installed")

  r <- nbasis(model)
  sk_result <- skmeans(t(target),r)

  # initialize W with the centroids obtained from spherical K-Means
  basis(model) <- t(sk_result$prototypes)
  # initialize weight matrix H  by fast NNLS algorithm from Van Benthem
  coef(model) <- fcnnls(t(sk_result$prototypes),target)$x

  # return updated object
  return(model)
}
# Register SVD init
setNMFSeed('Spherical', Sp_kmean, overwrite=TRUE)



######## NICA seeding ##################
NICA_1 <- function(model, target, maxIter= 1000, eps_conv=1e-4,
 gamma = 0.1, negopt = 1, negdev = "E") {
  m <- dim(target)[1]
  n <- dim(target)[2]
  r <- nbasis(model)
  j <- 1

  #  k eigenvectors of M M^T are the rows in P matrix ,
  P_1 the first r eigenvectors

  e_M <- eigen(target %*% t(target))
  P <- e_M$vectors
  P_1 <- e_M$vectors[1:r,]
  P_1_M <- P_1 %*% target

  mu <- apply(P_1_M,1,mean)
  cp_1 <- P_1_M %*% t(P_1_M) - mu %*% t(mu)
  #  Z matrix calculated like it is suggested in theoretical part
  e_PM <- eigen(cp_1)
  V <- e_PM$vectors %*% diag( 1 / sqrt(e_PM$values)) %*% t(e_PM$vectors)
  Z <- V %*% P_1_M

  # initialize with a random orthonormal matrix
  if( !require.quiet('pracma') )
    stop("Seeding method 'NICA' requires package ‘pracma‘ to be installed")
```

```
T_0 <- randortho(r)
T_1 <- diag(rep(1,r))
lim <- rep(1000,maxIter)

while ( lim[j] > eps_conv  && j <= maxIter)
{
  if(j > 1)
  {
    T_0 <- T_1
  }
  Y <- T_0 %*% Z
  for(u in 1:r)
  {
    T_1[,u] <- T_0[,u] - 2 * gamma * sum(pmin(0,Y[u,])*Z[u,])
  }

  # symmetrical decorrelation step
  T_sW1 <- La.svd(T_1)
  T_1 <- T_sW1$u %*% diag(1 / T_sW1$d) %*% t(T_sW1$u) %*% T_1

  j <- j + 1
  lim[j] <- max(abs(diag(abs(T_1 %*% t(T_0))) - 1))
}

Y <- T_1 %*% Z
i_TV <- inv(T_1 %*% V)
nuler <- matrix(0,m-r,r)
i_TV_0 <- rbind(i_TV,nuler)
W <- inv(P) %*%  i_TV_0




# option nr 1
if (negopt == 1 )
{
  # initialize basis matrix W with  absolute
  # value of first k columns of U matrix from SVD

  basis(model) <- abs(W)

  # initialize weight matrix H with absolute
  # value of first k rows of singular values times
  # right singular vectors
```

```
    coef(model) <- abs(Y)
  }
  else if(negopt == 2)
  {
    basis(model) <- abs(W)

    c_m <- abs(W) %*% t(abs(W)) %*% target
    if( negdev == "E")
    {
      alpha_H <- sum(target * c_m)/sum(c_m * c_m)
    }
    else
    {
      alpha_H <- sum(target)/sum(b_m)
    }

    coef(model) <- alpha_H * t(abs(W)) %*% target
  }
  else
  {

    b_m <- target %*% t(abs(Y)) %*% abs(Y)
    if( negdev == "E")
    {
      alpha_W <- sum(target * b_m)/sum(b_m * b_m)
    }
    else
    {
      alpha_W <- sum(target)/sum(b_m)
    }

    basis(model) <- alpha_W * target %*% t(abs(Y))
    coef(model) <- abs(Y)

  }

  # return updated object
  return(model)
}
# Register NICA init
setNMFSeed(’NICA’, NICA_1, overwrite=TRUE)
```

In the following, the implemented NMF algorithms are stated, which can be used in the framework of the NMF package.

```
############ Basic ALS ########################
ALS <- function(M, seed, maxIter= 600,eps=.Machine$double.eps) {

  j <- 1
  stop_c <- 0
  W <- basis(seed)
  H <- coef(seed)
  m <- dim(M)[1]
  n <- dim(M)[2]

  # Solve alternatively the LS equations
  # condition to stop if the LS equation is
  # not solveable anymore due to instability of the matrix

  while( (stop_c == 0) && (j <= maxIter) &&
  (rcond(t(W)%*% W) > 2^(-10)) && (rcond(H %*% t(H)) > 2^(-10)) )
  {
    # if ( any(rowSums(H)==0) )
    # {
    #   break
    # }
    # if ( any(colSums(W)==0) )
    # {
    #   break
    # }
    H_1 <- H
    H <- solve(t(W)%*% W,t(W) %*% M)
    H <- pmax(H,eps)

    #norm(H_1-H) < 2^(-8)
    if ( (rcond(H %*% t(H)) < 2^(-10)) || (any(rowSums(H) < 2^(-10)) ))
    {
       H <- H_1
       stop_c <- 1
    }
    else
    {
      W_1 <- W
      W <- t(solve(H %*% t(H),H %*% t(M)))
      W <- pmax(W,eps)
      if ( any(colSums(W) < 2^(-10)) )
      {
        W <- W_1
        stop_c <- 1
      }
    }

    j <- j + 1
  }

  #W[W == eps] <- 0
  #H[H == eps] <- 0

  basis(seed) <- W

  coef(seed) <-  H

  seed@extra$iteration <- j-1
  # return updated data
  return(seed)
}

############# AHCLS #########################
#l_H = 0.2, l_W = 0.3, a_H = 0.5, a_W = 0.7
AHCLS <- function(M, seed, maxIter= 600,eps=.Machine$double.eps, l_H =
0.05, l_W = 0.01, a_H = 0.6, a_W = 0.8) {

  j <- 1
  stop_c <- 0
  m <- dim(M)[1]
  n <- dim(M)[2]
  r <- nbasis(seed)
  W <- basis(seed)
  H <- coef(seed)

  if( !require.quiet('pracma') )
    stop("Seeding method 'AHCLS' requires package `pracma` to be installed")

  b_H <- ((1-a_H)*sqrt(r) + a_H)^2
  b_W <- ((1-a_W)*sqrt(r) + a_W)^2
  E <- ones(r)
  I_1 <- diag(rep(1,r))
  temp_x_h <- t(W) %*% W + l_H * b_H * I_1 - l_H * E
  temp_y_h <- t(W) %*% M
  temp_x_W <- H %*% t(H) + l_W * b_W * I_1 - l_W * E
  temp_y_W <- H %*% t(M)
```

```
# W_0 <- W + 1
 # Solve alternatively the CLS equations
 # condition to stop if the CLS equation
 # is not solveable anymore due to instability of the matrix

 while( (stop_c == 0) && (j <= maxIter) &&
  (rcond(temp_x_h) > 2^(-10)) && (rcond(temp_x_W) > 2^(-10)) )
 {
  # W_0 <- W
   #fcnnls(W,target)$x

   # save H before the update
   H_1 <- H

   temp_x_h <- t(W) %*% W + l_H * b_H * I_1 - l_H * E
   temp_y_h <- t(W) %*% M

   H <- solve(temp_x_h ,temp_y_h)
   H <- pmax(H,eps)


   temp_x_W <- H %*% t(H) + l_W * b_W * I_1 - l_W * E
   temp_y_W <- H %*% t(M)

   if (rcond(temp_x_W) < 2^(-10) || any(rowSums(H) < 2^(-10)))
   {
     H <- H_1
     stop_c <- 1
   }
   else
   {
     W_1 <- W
     W <- t(solve(temp_x_W,temp_y_W))
     W <- pmax(W,eps)
     if ( any(colSums(W) < 2^(-10)) )
     {
       W <- W_1
       stop_c <- 1
     }
   }


   j <- j + 1
 }
```

```
  basis(seed) <- W

  coef(seed) <- H

  seed@extra$iteration <- j-1

  # return updated data
  return(seed)
}

######### recursive geometric algorithm ##########

FastSepNMF <- function(M,seed, norm_l = 2){

  m <- dim(M)[1]
  n <- dim(M)[2]
  r <- nbasis(seed)

  W <- basis(seed)
  H <- coef(seed)
  S <- numeric(r)

  j <- 1

  # Normalization of the columns of M so that the y sum to one
  if (norm_l == 1)
  {
    M_n <- t(as.matrix(apply(M,1, function(x) {x/sum(x)})))
  }
  else
  {
    M_n <- M
  }

  normM <- apply(M_n,2,function(x){sum(x^2)})
  nM <- max(normM)
  names(normM) <- 1:n

  # Perform r recursion-steps (unless the
  # relative approximation error is smaller than 10^-9)
  #  && (max(normM)/ nM > 10^(-9))
  while( j <= r)
  {
```

```
 # select the column of M with largest l2-norm
 max_R <- max(normM)

 if( j == 1){ norm_M1 <- normM}

 # Check for ties up to 1e-6 percision
 b <- names(normM)[(max_R - normM)/ max_R <= 10^(-6)]
 if ( length(b) > 1)
 {
   d <- which.max(norm_M1[b])
   b <- names(d)
 }
 # Update index set, and extracted column
 S[j] <- as.numeric(b)
 W[,j] <- M[,as.numeric(b)]

 #cat("S=",S)

# Compute (I-w_{j-1}w_{j-1}^T)...(I-w_1w_1^T) W[,j], that is,
 # R^(j)[,S[j]], where R^(j) is the jth residual (with R^(1) = M).
# if(j > 1)
 #{
 #  k <- 1:(j-1)
  # W[,j] <- W[,j] - W[,k] * as.numeric(t(W[,k]) %*% W[,j])

 #}

 if(j > 1)
 {
   for ( k in 1:(j-1))
   {
     temp <-  W[,k] * as.numeric(t(W[,k]) %*% W[,j])
    # cat("temp=",temp)
     W[,j] <- W[,j] - temp
   }

 }

 # Normalize W[,j]
 W [,j] <- W[,j] / sqrt(sum(W[,j]^2))

 # Update the norm of the columns of M after orhogonal projection using
 # the formula ||r^(j)_u||^2 = ||r^(j-1)_u||^2 - ( U(:,j)^T m_u )^2 for
 # all u.
```

```
    normM <- as.numeric(normM - (W[,j] %*% M_n)^2)
    names(normM) <- 1:n

    j <- j + 1
  }

  W <- pmax(M[,S],10^(-3))
  H <- fcnnls(W,M)$x

  basis(seed) <- W

  coef(seed) <- H

  seed@extra$iteration <- 1
  seed@extra$columns <- S

  # return updated data
  return(seed)

}
```

List of Figures

1.1 Comparison NMF vs PCA basis 6
1.2 NMF vs PCA for Facial image recognition 6

3.1 Illustration of different NMF bases 41
3.2 Rank survey - randomly generated non-negative data 52

4.1 Spectrum - Example for meteorite Renazzo 56
4.2 Image of Target 4E1 57
4.3 Spectrogram - basis spectra of meteorite "ochansk" 60
4.4 Spectrogram - basis spectra of meteorite "ochansk" restricted to the first 100 mass 61
4.5 Spectrogram - basis spectra of meteorite "ochansk" vs substrate 62
4.6 Optimal Weight Matrix H^* 63
4.7 Location of the corn "ochansk" 64
4.8 Spectrum - spectra of "mall" and "msub" 65
4.9 Spectrum - spectra of "mall" and "msub" restricted to the first 100 mass 66
4.10 Consensus map - SNMF/L and BRUNET for mix without noise 67
4.11 Weight Matrix H^* no noise 68
4.12 Consensus map - BRUNET and AHCLS for mix with additive noise . . 70
4.13 Spectrum - spectra of W in comparison to "mall" and "msub" 71
4.14 Weight Matrix H^* with additive noise 72
4.15 Consensus map - LEE and AHCLS for noise on the coefficients 74
4.16 Weight Matrix H^* with additive noise 74
4.17 Rank survey - "allende" 76
4.18 Rank survey - "allende" 77
4.19 Consensus map - Rank 3 vs Rank 4 78

Bibliography

Arora, S., R. Ge, R. Kannan, and A. Moitra (2012). “Computing a nonnegative matrix factorization–provably”. *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pp. 145–162.

Arora, S. et al. (2013). “A practical algorithm for topic modeling with provable guarantees”. *International Conference on Machine Learning*, pp. 280–288.

Berry, M. W. et al. (2007). “Algorithms and applications for approximate nonnegative matrix factorization”. *Computational Statistics & Data Analysis* 52(1), pp. 155–173.

Boumal, N., V. Voroninski, and A. Bandeira (2016). “The non-convex Burer-Monteiro approach works on smooth semidefinite programs”. *Advances in Neural Information Processing Systems*, pp. 2757–2765.

Boutsidis, C. and E. Gallopoulos (2008). “SVD based initialization: a head start for nonnegative matrix factorization”. *Pattern Recognition* 41(4), pp. 1350–1362.

Brunet, J.-P., P. Tamayo, T. R. Golub, and J. P. Mesirov (2004). “Metagenes and molecular pattern discovery using matrix factorization”. *Proceedings of the National Academy of Sciences* 101(12), pp. 4164–4169.

Candès, E. J., X. Li, Y. Ma, and J. Wright (2011). “Robust principal component analysis?” *J. ACM* 58(3), 11:1–11:37.

Catral, M., L. Han, M. Neumann, and R.J. Plemmons (2004). “On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices”. *Linear Algebra and its Applications* 393 (Supplement C), pp. 107–126.

CoMeCS-Project (2017). *Comet and Meteorite Materials - Studied by Chemometrics of Spectroscopic Data*. URL: `http://www.lcm.tuwien.ac.at/comecs/` (visited on 12/13/2017).

Das, G. and D. Joseph (1990). “The complexity of minimum convex nested polyhedra”. *Proc. 2nd Canad. Conf. Comput. Geom*, pp. 296–301.

d’Aspremont, A., L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet (2007). “A direct formulation for sparse PCA using semidefinite programming”. *SIAM Review* 49(3), pp. 434–448.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). “Maximum likelihood from incomplete data via the EM algorithm”. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38.

Dhillon, I. S. and D. S. Modha (2001). “Concept decompositions for large sparse text data using clustering”. *Machine Learning* 42(1), pp. 143–175.

Frigyesi, A. and M. Höglund (2008). “Non-negative matrix factorization for the analysis of complex gene expression data: identification of clinically relevant tumor subtypes”. *Cancer Informatics* 6, p. 275.

Gaujoux, R. and C. Seoighe (2010). "A flexible R package for nonnegative matrix factorization". *BMC Bioinformatics* 11(1), p. 367.

Gillis, N. (2011). "Nonnegative matrix factorization: complexity, algorithms and applications". *Unpublished Doctoral Dissertation, Université Catholique de Louvain. Louvain-La-Neuve: CORE.*

— (2017). "Introduction to nonnegative matrix factorization". *ArXiv Preprint* 1703.00663.

Gillis, N. and R. Luce (2018). "A fast gradient method for nonnegative sparse regression with self-dictionary". *IEEE Transactions on Image Processing* 27(1), pp. 24–37.

Gillis, N. and S. A. Vavasis (2014). "Fast and robust recursive algorithms for separable nonnegative matrix factorization". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(4), pp. 698–714.

Gregory, D. and N. Pullman (1983). "Semiring rank: boolean rank and nonnegative rank factorization". *J. Combin. Inform. System Sci.* 3, pp. 223–233.

Grippo, L. and M. Sciandrone (2000). "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints". *Operations Research Letters* 26(3), pp. 127–136.

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

Huang, K., N. D. Sidiropoulos, and A. Swami (2014). "Non-negative matrix factorization revisited: uniqueness and algorithm for symmetric decomposition". *IEEE Transactions on Signal Processing* 62(1), pp. 211–224.

Hutchins, L. N., S. M. Murphy, P. Singh, and J. H. Graber (2008). "Position-dependent motif characterization using non-negative matrix factorization". *Bioinformatics* 24(23), pp. 2684–2690.

Hyvärinen, A. and E. Oja (2000). "Independent component analysis: algorithms and applications". *Neural Networks* 13(4), pp. 411–430.

Janecek, A. and Y. Tan (2011). "Using Population Based Algorithms for Initializing Nonnegative Matrix Factorization". *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings, Part II*. Springer Berlin Heidelberg, pp. 307–316.

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

Kitamura, D. and N. Ono (2016). "Efficient initialization for nonnegative matrix factorization based on nonnegative independent component analysis". *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5.

Langville, A. N. et al. (2014). "Algorithms, initializations, and convergence for the nonnegative matrix factorization". *ArXiv preprint arXiv:1407.7299*.

Lee, D. D. and H. S. Seung (1999). "Learning the parts of objects by non-negative matrix factorization". *Nature* 401(6755), pp. 788–791.

— (2001). "Algorithms for non-negative matrix factorization". *Advances in Neural Information Processing Systems*, pp. 556–562.

Li, Q. and G. Tang (2016). "The nonconvex geometry of low-rank matrix optimizations with general objective functions". *ArXiv Preprint ArXiv:1611.03060*.

Lin, C.-J. (2007a). "On the convergence of multiplicative update algorithms for nonnegative matrix factorization". *IEEE Transactions on Neural Networks* 18(6), pp. 1589–1596.

— (2007b). "Projected gradient methods for nonnegative matrix factorization". *Neural Computation* 19(10), pp. 2756–2779.

Luce, R., P. Hildebrandt, U. Kuhlmann, and J. Liesen (2016). "Using separable nonnegative matrix factorization techniques for the analysis of time-resolved raman spectra". *Applied Spectroscopy* 70(9), pp. 1464–1475.

Naik, G. (2015). *Non-negative Matrix Factorization Techniques: Advances in Theory and Applications*. Springer Berlin Heidelberg.

Oja, E. and M. Plumbley (2004). "Blind separation of positive sources by globally convergent gradient search". *Neural Computation* 16(9), pp. 1811–1825.

Paatero, P. (1997). "Least squares formulation of robust non-negative factor analysis". *Chemometrics and Intelligent Laboratory Systems* 37(1), pp. 23–35.

Paatero, P. and U. Tapper (1994). "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values". *Environmetrics* 5(2), pp. 111–126.

Pascual-Montano, A. et al. (2006). "Nonsmooth nonnegative matrix factorization (nsNMF)". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(3), pp. 403–415.

Plumbley, M. (2002). "Conditions for nonnegative independent component analysis". *IEEE Signal Processing Letters* 9(6), pp. 177–180.

— (2003). "Algorithms for nonnegative independent component analysis". *IEEE Transactions on Neural Networks* 14(3), pp. 534–543.

Qiao, H. (2015). "New SVD based initialization strategy for non-negative matrix factorization". *Pattern Recognition Letters* 63 (Supplement C), pp. 71–77.

R-Core-Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `http://www.R-project.org/`.

Recht, B., C. Re, J. Tropp, and V. Bittorf (2012). "Factoring nonnegative matrices with linear programs". *Advances in Neural Information Processing Systems*, pp. 1214–1222.

Rezaei, M. and R. Boostani (2011). "An efficient initialization method for nonnegative matrix factorization". *Journal of Applied Sciences* 11.

Rousseeuw, P. J. (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". *Journal of Computational and Applied Mathematics* 20 (Supplement C), pp. 53–65.

Srebro, N. and T. Jaakkola (2003). "Weighted low-rank approximations". *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 720–727.

Stewart, G. W. and J. Sun (1990). *Matrix Perturbation Theory*. Boston: Academic Press.

Van Benthem, M. H. and M. R. Keenan (2004). "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems". *Journal of Chemometrics* 18(10), pp. 441–450.

Varmuza, K. (2010). "Computer methods in mass spectrometry for chemical structure assignment". *Encyclopedia of Spectroscopy and Spectrometry* 3. Elsevier.

Wild, S. (2004). “Seeding non-negative matrix factorizations with spherical k-means clustering”. MA thesis. University of Colorado.

Wild, S., J. Curry, and A. Dougherty (2004). “Improving non-negative matrix factorizations through structured initialization”. *Pattern Recognition* 37(11), pp. 2217–2232.

Yuan, Z. and E. Oja (2004). “A FastICA Algorithm for Non-negative Independent Component Analysis”. *Independent Component Analysis and Blind Signal Separation: Fifth International Conference, ICA 2004, Granada, Spain, September 22-24, 2004. Proceedings*. Springer Berlin Heidelberg, pp. 1–8.