Analysis of incomplete data sets in numerical chemometrics

Dissertation

zur

Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität Rostock

vorgelegt von

Martina Beese

Rostock, 11.11.2024

https://doi.org/10.18453/rosdok_id00005109

Gutachter: Prof. Dr. Klaus Neymeyr, Universität Rostock
Prof. Dr. Anna de Juan Capdevila, Universitat de Barcelona

Tag der Verteidigung: 7. März 2025

Contents

1. Introduction

1.1. Nonnegative matrix factorization of incomplete data sets

In a world where big data and sustainability are increasingly relevant, it is important to harness the information content of high-dimensional data in an efficient and resource-conserving way. This includes extracting the maximum amount of information from the data. In particular, if parts of the data are faulty or even missing, the goal is to work with the given data. No attempt is made to fill the data gaps, as such a step can be costly.

For example, spectroscopy processes large amounts of measured spectroscopic data that can be analyzed using chemometric methods. Malinowski [46] describes chemometrics as the use of "mathematical and statistical methods for handling, interpreting and predicting chemical data". A key tool for such analysis is nonnegative matrix factorization (NMF). Beyond spectroscopy, NMF has many applications in data analysis. NMF for nonnegative data can be interpreted as linear dimensionality reduction [26]. A well-known example of linear dimensionality reduction is the so-called cocktail party problem [13] where the goal is to separate the individual sources from a set of mixed signals. NMF can solve the source separation problem, but the solution is known to be NP-hard [26]. For the historical context of NMF see chapter 2.

This thesis focuses on numerical chemometrics, i.e. the use of a numerical approach to analyze spectroscopic data. In particular, the focus is on NMF of matrices with unknown or unreliable matrix entries, which are considered as incomplete. Incompleteness can occur when a data set is subject to measurement errors or is incomplete by design (see chapter 3 for more information and classification).

Instead of relying on additional data to eliminate the incompleteness or analyzing only the largest complete parts of the data, the goal is to use incomplete data as efficiently as possible. This thesis provides a new approach to solving the nonnegative matrix factorization problem in an incomplete setting by analyzing the underlying geometric relations. The analysis is performed using a divide-and-conquer approach on the incomplete data set. That is, parts of the data set that are not affected by the incompleteness are analyzed and their results are combined, see chapter 3.

This leads to the following roadmap. First, the theory of the nonnegative matrix factorization problem is presented in chapter 2. The NMF theory is then applied and further developed to allow an analysis of incomplete data sets, see chapter 3. The algorithmic solution for the analysis of incomplete data is presented in chapter 4. The analysis of incomplete data is closely related to the analysis of subsystems, which is discussed in chapter 5. The approaches of subsystem analysis, especially their detection, make it possible to detect areas suffering from incompleteness, which is a key step in the analysis of experimental data. An application of the developed methods to experimental data is presented in chapter 6.

Parts of the results of this thesis have been published in [8].

1.2. Data sets and notation

Two model data sets are used to illustrate the results of this thesis. These data sets, data set 1 and 2, are described in the appendix, see Sec. A.1.

An overview of the notation used is provided next. Additionally, for matrix and vector indexing the Matlab notation is used, meaning $D(:, j)$ denotes the j-th column of a matrix D, $D(i, :)$ the i-th row and $D(i : i + 10, :)$ the submatrix formed by the rows i to $i + 10$ of D.

$\mathbb{R}_+$	Set of nonnegative real numbers including 0.
$D \in \mathbb{R}^{k\times n}$	A matrix of rank s, mostly nonnegative, see problem 2.3.
$C \in \mathbb{R}^{k\times s}$	A matrix of rank s and left factor of the nonnegative matrix factorization problem 2.3.
$S^T \in \mathbb{R}^{s\times n}$	A matrix of rank s and right factor of the nonnegative matrix factorization problem 2.3.
$\mathcal{C}$	A finitely generated convex cone, see Eq. (2.2).
colcone(D)	Cone generated by the columns of D, see Eq. (2.3) and (2.7).
rowcone(D)	Cone generated by the rows of D, see Eq. (2.4) and (2.7).
$\mathcal{E}_D$	Index set of essential rows, see Eq. (2.5).
$\tilde{\mathcal{E}}_D$	Index set of essential columns, see Eq. (2.6).
esi	Set denoting the essential entries, given by the Cartesian product of $\mathcal{E}_D$ and $\tilde{\mathcal{E}}_D$.
D_{esi}	A matrix reduced to its essential rows and columns, see Sec. 2.2.1.
$U\Sigma V^T$	Truncated singular value decomposition of D with rank s, see Thm. 2.11.
σ_i	Singular values with $1 \leq i \leq s$.
$T \in \mathbb{R}^{s\times s}$	Invertible matrix to define the set of feasible solutions, see Def. 2.15.
$\mathcal{I}_S(D)$	Inner polytope in the V-space of D, see Eq. (2.11), it is spanned by data points a_i.
$\mathcal{F}_S(D)$	Outer polytope in the V-space of D, see Eq. (2.10).
$\mathcal{I}_C(D)$	Inner polytope in the U-space of D, see Eq. (2.13), spanned by data points b_i.
$\mathcal{F}_C(D)$	Outer polytope in the U-space of D, see Eq. (2.12).
$\mathcal{M}_{\mathcal{S}}$	Set of feasible solutions of the V-space, see Eq. (2.9).
$\mathcal{M}_{\mathcal{C}}$	Set of feasible solutions of the U-space, see Eq. (2.14).
$\mathcal{H}$	Affine subspace in the V-space spanned by selected data points, see Eq. (3.5).
$\mathbf{M_R}$	Restricted set of feasible solutions, see Lem. 3.25.
$\mathcal{D}$	Row space of D, see Sec. 5.2.1.

2. Nonnegative matrix factorization

In the last decades, nonnegative matrix factorization (NMF) has become increasingly popular. Not least by a publication in Nature in 1999 by Lee and Seung [39], who shifted the focus to determining a possible NMF rather than focusing on the underlying chemical or physical situation and the corresponding model behind the NMF [26]. Previously, NMF had been used in analytical chemistry, where it was described by Wallace in 1960 [85], with the goal of recovering the pure spectra of the chemical species present in a measurement, such as a chemical reaction. In 1994, Paatero and Tapper [54] introduced the problem of finding an NMF in its modern form, but until Lee and Seung it was limited to the field of chemometrics [26]. In addition to analytical chemistry and geoscience and remote sensing where NMF was first used, it is now applied in many other fields, such as audio signal processing, computational geometry, image processing, and document clustering [26].

In order to properly address the problem of nonnegative matrix factorization, it is formally introduced in Sec. 2.1. Then, two approaches to solve this problem are presented. The first is based on a cone representation of the given data matrix (see Sec. 2.2) and the second uses a singular value decomposition of the data matrix and corresponding polytopes (see Sec. 2.3). Both approaches cover not only the determination of a possible nonnegative factorization, but also certain properties of the data matrix are investigated, e.g., if it is possible to reduce a data matrix without losing information. Additionally, a connection between the two approaches is made in Sec. 2.3.6. Finally, the influence of noise in a data set and its influence on how to solve the nonnegative matrix factorization problem or how to find factors whose product approximate the given data matrix reasonably well is considered (see Sec. 2.4).

2.1. The nonnegative matrix factorization problem

In order to gain an understanding of what a nonnegative matrix factorization is, the NMF problem is introduced.

Problem 2.1 (General nonnegative matrix factorization problem, [26, 65])**.** *Let a nonnegative matrix $D \in \mathbb{R}^{k\times n}$ be given. A factorization $D = CS^T$ is desired, with $C \in \mathbb{R}_+^{k\times r}$ and $S^T \in \mathbb{R}_+^{r\times n}$, such that r is minimal.*

Closely related to the NMF problem 2.1 is the nonnegative rank.

Definition 2.2 (Nonnegative rank, [14, 26, 65])**.** *The* nonnegative rank *of $D \in \mathbb{R}_+^{k\times n}$ is the smallest $r \in \mathbb{N}$ for which Prob. 2.1 has a solution. It is denoted by* $\operatorname{rank}_+(D) = r$.

Based on this definition, $\operatorname{rank}(D) \leq \operatorname{rank}_+(D) \leq \min(k, n)$ (see [14]). This work focuses on the case where $\operatorname{rank}(D) = \operatorname{rank}_+(D)$, and thus on cases where both C and S^T have full rank. Therefore, Prob. 2.1 is formulated more precisely as follows (see [65]).

Problem 2.3 (NMF problem [65])**.** *Let a nonnegative matrix $D \in \mathbb{R}^{k\times n}$ be given, with $s = \operatorname{rank}(D) = \operatorname{rank}_+(D)$. A nonnegative factorization is sought, that means finding factors $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$ so that $D = CS^T$.*

In the following, when referring to the NMF problem, Prob. 2.3 is meant. When faced with such a problem, the question arises whether there exists a solution. For the case $\operatorname{rank}(D) = s \leq 2$ Thomas showed in 1974 that there exists always a nonnegative factorization, see [80]. But what about arbitrary $\operatorname{rank}(D) > 2$? And if there is a solution, is it unique? These questions will be examined next.

2.1.1. Characterizing the NMF problem

Looking at the NMF problem as in Prob. 2.3, it is not obvious that the factors $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$ exist and whether they are unique. Therefore, a short analysis of the problem is given.

The NMF problem can be categorized as an inverse problem, because it is inverse to the multiplication of two nonnegative matrices, see [20] for inverse problems. A further categorization is possible to specify whether a problem poses numerical difficulties or not [20]. Therefore, Hadamard's definition is used to check the "well-posedness" of a problem.

A problem is called *well-posed* if the following conditions are met (see [20, 65])

1. a solution exists,
2. the solution is unique,
3. the solution is continuously dependent on the data (stability condition).

If one of these conditions is not met, the problem is called *ill-posed*.

The first step to check if it is well-posed is to look at the existence of a solution. This step is closely related to the rank and the nonnegative rank of D, because if $s = \operatorname{rank}(D) < \operatorname{rank}_+(D)$ there is no NMF of D where C and S^T both have rank s. A well-known example of this is the following nonnegative matrix from [80]

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

For this matrix $\operatorname{rank}(D) = 3 < 4 = \operatorname{rank}_+(D)$ holds and thus there is no full rank factorization of D where $\operatorname{rank}(C) = \operatorname{rank}(S^T) = \operatorname{rank}(D)$. Thus, the first condition for a well-posed problem is not fulfilled. However, it is possible to relax the existence condition and implement additional specifications to ensure a solution. Thus, the previous restriction to the NMF problem 2.3 with $\operatorname{rank}(D) = \operatorname{rank}_+(D)$ is used, which ensures a full rank factorization.[1]

The next step is to verify that a solution is unique. Therefore, trivial ambiguities are considered first, see [65,75]. These are due to permutations and scaling of the factors C and S^T. If $P \in \mathbb{R}^{s\times s}$ is a permutation matrix and $\Delta \in \mathbb{R}^{s\times s}$ is a diagonal matrix with positive diagonal entries, then $CS^T = (C\Delta^{-1}P^T)(P\Delta S^T)$ are both nonnegative factorizations of D.

Unfortunately, even excluding trivial ambiguities from the consideration does not lead to a unique solution. This is illustrated by the following example from [65]

$$D = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}. \tag{2.1}$$

The NMF problem 2.3 is thus ill-posed. Nevertheless, an analysis of the NMF problem is of interest. The knowledge that there can be multiple solutions to the NMF problem motivates the analysis of this ambiguity.

[1] To check whether $\operatorname{rank}(D) = \operatorname{rank}_+(D)$ is NP-hard. This implies the NP-hardness of the NMF problem, see [26].

The notion of a solution in the context of the NMF problem is defined next.

Definition 2.4. *The pair of $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$ is called a* solution *or a* feasible pair*, if it is a factorization of a matrix $D \in \mathbb{R}_+^{k\times n}$ with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*, as in problem 2.3. Furthermore, a factor $S^T \in \mathbb{R}_+^{s\times n}$ is called* feasible *if there exists a corresponding $C \in \mathbb{R}_+^{k\times s}$ so that C and S^T form a feasible pair. Similarly, the feasibility for the factor $C \in \mathbb{R}_+^{k\times s}$ can be defined.*

There are several approaches to computing solutions to the NMF problem 2.3. Two of them are presented next, namely a cone-based approach and one based on a singular value decomposition of the data matrix. Finally, the ambiguity of the solution to the NMF problem and how to determine it is discussed. The choice of which approach to use depends on the application and preferences. In this thesis, both approaches are used for the considerations in chapter 3.

2.2. Solving the NMF problem using convex cones

The NMF problem 2.3 can be interpreted geometrically as a nested cone problem [26]. The first step is to take a closer look at how cones are defined.

The cones treated in this thesis are convex and finitely generated, i.e., finitely many vectors are sufficient to construct the cone. Thus, a convex and finitely generated cone $\mathcal{C}$ is given by

$$\mathcal{C} = \left\{ \sum_{i=1}^{l} \alpha_i v_i : \ \alpha_i \geq 0 \right\}, \tag{2.2}$$

with its generators $v_1, \ldots, v_l$, the cone is the conic hull of them [26]. For simplicity, this thesis uses the simplified term *cone* to describe $\mathcal{C}$. An example is shown in Fig. 2.1. It is possible to define cones generated by a matrix $D \in \mathbb{R}^{k\times n}$ using Eq. 2.2. For example, the columns of D become the generators and the conic hull of them leads to the corresponding cone. The cone generated by the columns of D is given by

$$\operatorname{colcone}(D) = \left\{ \sum_{j=1}^{n} \alpha_j D(:, j) : \ \alpha_j \geq 0 \right\}. \tag{2.3}$$

In the same way, it is also possible to define the cone generated by the rows of D

$$\operatorname{rowcone}(D) = \left\{ \sum_{i=1}^{k} \alpha_i D(i, :) : \ \alpha_i \geq 0 \right\}. \tag{2.4}$$

Both cones span a subspace of dimension s in the k- and n-dimensional space, if $\operatorname{rank}(D) = s$. This definition of the cones follows the definition of [8] as opposed to [26] where only the column space cone of D is defined, using D^T for the row space cone. The column and row space cones of the example matrix of Eq. (2.1) are shown in Fig. 2.1 with the generators marked by dashed lines.

To bridge the connection between cones and the NMF problem, a *nested cone problem* is considered. The idea is to find a cone $\mathcal{C}$, such that $\operatorname{colcone}(D) \subseteq \mathcal{C} \subseteq \mathbb{R}_+^k$. The matrix that generates $\mathcal{C}$ should have fewer columns than D. So that only a small number of vectors generate the edges of $\mathcal{C}$ that encloses $\operatorname{colcone}(D)$. Ideally, the number of edges reaches its minimum with $s = \operatorname{rank}(D) = \operatorname{rank}_+(D)$. Thus, a matrix $C \in \mathbb{R}_+^{k\times s}$ is sought where $\mathcal{C} = \operatorname{colcone}(C)$. With $C^+D = S^T$ the other factor is calculated, where C^+ is the pseudoinverse of C [28]. If $S^T \in \mathbb{R}_+^{s\times n}$

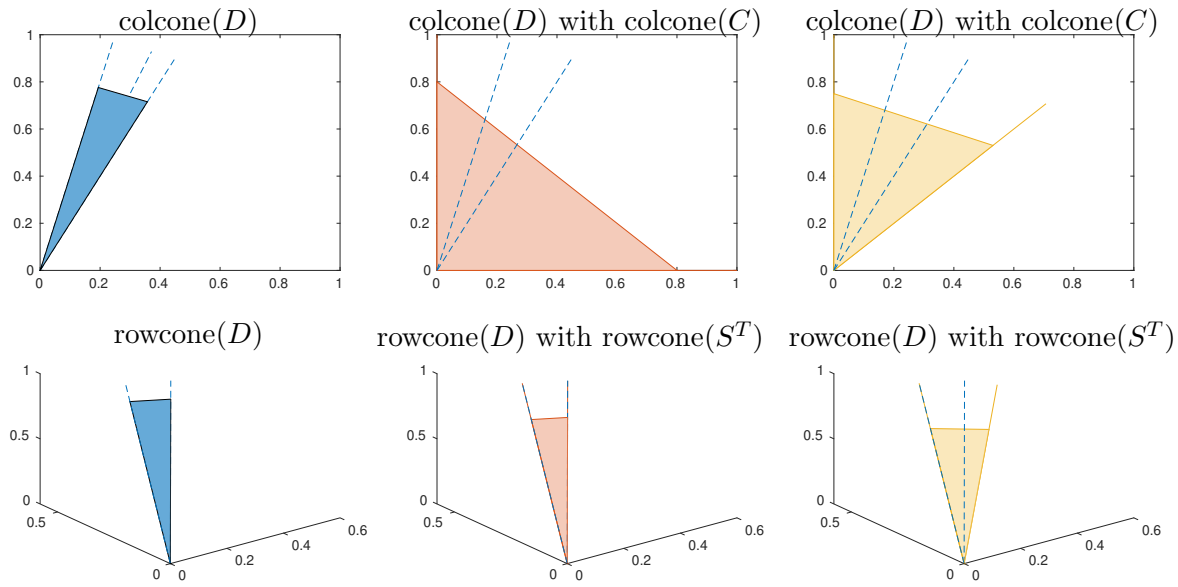

Figure 2.1.: Corresponding cones of the column space (upper row) and the row space (bottom row) to the matrix D of Eq. (2.1) in blue with the vectors corresponding to the columns or rows, marked by dashed lines. The cones corresponding to possible factorizations of Eq. (2.1) are colored in yellow and red, respectively. Thus, the second and third columns represent possible nonnegative factorizations of D. Since $D \in \mathbb{R}^{2\times 3}$, the cones of the column space are 2D and the cones of the row space are in 3D.

and rowcone$(D) \subseteq$ rowcone(S^T), then C and S^T form a feasible pair, as in Def. 2.4 and the nested cone problem becomes equivalent to problem 2.3. This is also shown in Fig. 2.1 with two possible feasible solutions marked in red and yellow. These colored cones correspond to the solutions given in Eq. (2.1).

This connection between the NMF problem and the nested cone problem can be simplified, since a feasible factor S^T already results when C is selected (in the way described). This is a well-known concept, see [26]. It is repeated in the following lemma in a slightly different form to fit the style of the following theory.

Lemma 2.5 (From [8]). *Let $C \in \mathbb{R}^{k\times s}$ and $D \in \mathbb{R}^{k\times n}$ be nonnegative matrices with* rank$(D) =$ rank$_+(D) = s$, *where* colcone$(D) \subseteq$ colcone(C). *This implies that there exists a nonnegative matrix $S^T \in \mathbb{R}^{s\times n}$ so that $D = CS^T$. So C and S^T are a feasible pair.*

Proof. The connection colcone$(D) \subseteq$ colcone(C) leads to $D(:, j) \in$ colcone(C) for $j = 1, \ldots, n$. Consequently, each column of D can be represented by a conical combination of the columns of C, i.e. $D(:, j) = Ca$ with $a \in \mathbb{R}^s_+$. These linear factors are stored in S, where $S(j, :) = a^T$, which implies that S is a nonnegative matrix. Thus, C and S^T form a nonnegative factorization of D and are a feasible pair. □

The same holds for rowcone(D) and rowcone(S^T) and the corresponding factor C. Lem. 2.5 allows to formulate the NMF problem 2.3 as a problem of finding $s =$ rank(D) vectors in the positive orthant such that either colcone(D) or rowcone(D) is nested, see [26].

It is again important to note that because of Lem. 2.5 it is not necessary to check whether S^T is feasible if C is feasible. This must be kept in mind for later arguments in chapter 3, since it is possible to check feasibility in both spaces (for both factors), but it is not necessary.

Instead of looking at just one feasible pair of C and S^T, it is possible to look at the entire set of solutions to the NMF problem and thus the entire set of feasible factors. These solutions can be computed by finding matrices $T \in \mathbb{R}^{s\times s}$ with $\operatorname{rank}(T) = s$ such that $\widehat{C} = CT^{-1}$ and $\widehat{S}^T = TS^T$ again form a feasible pair. This concept is known as ambiguity of the solution of the NMF problem. It is further investigated in Sec. 2.3.2, where a singular value decomposition is used as an approach to solve the NMF problem, which also allows an easier representation of the solutions.

Instead of dealing with convex cones, it is also possible to reduce the problem to convex hulls. This is done by normalizing the vectors spanning the cone, which are thus on the same plane and, for colcone(D), in a $(k-1)$-dimensional space. Now, instead of finding a cone, the goal is to find a polytope containing the determined convex hull with s vertices. How this approach is used to further analyze NMF problems can be found in [26]. For this work, it is usually sufficient to look at cones. The concept of convex hulls is rather used to connect the cone approach with the approach of using a singular value decompositions to solve NMF problems. The latter is done in Sec. 2.3.

Going back to convex cones and the initial definition in Eq. (2.3) and (2.4) as well as Fig. 2.1, it can be seen that not all columns or rows are necessary to define the cones. This is the topic of the next subsection.

2.2.1. Determining essential information

Looking at the definition of the cones of the column and row space of D (in Eq. (2.3) and (2.4)), it becomes clear that a certain amount of redundancy can occur. This means that neglecting certain vectors will result in the same cone.

The first step in reducing redundancy is to eliminate all repeated rows and columns, including those with different scaling. Without loss of generality, only the first occurrence of such rows or columns is considered. Therefore, a temporary index set is used, first for rowcone(D)

$$\mathcal{T}_D = \Big\{ i \in \{1,\dots,k\} :\ D(i,:) \neq \beta D(\ell,:) \text{ for all } \ell \in \{1,\dots,i-1\},\ \beta \in \mathbb{R}_+ \Big\}.$$

The same consideration can be done for colcone(D) and another index set is given by

$$\widetilde{\mathcal{T}}_D = \Big\{ j \in \{1,\dots,n\} :\ D(:,j) \neq \beta D(:,\ell) \text{ for all } \ell \in \{1,\dots,j-1\},\ \beta \in \mathbb{R}_+ \Big\}.$$

With these temporary index sets, the set of generators of the cones can be reduced, because using only rows or columns corresponding to the temporary index set means that each edge of the respective cone corresponds to only one index.

The redundancy can be reduced even further by using only the edges to define the cones. Looking at Eq. (2.3) and (2.4) this still leads to the same cones, because each inner point of the cone can be represented by a conic combination of the edges. This further reduces the temporary index sets.

The corresponding index set that identifies the edges of rowcone(D) is given by

$$\mathcal{E}_D = \Big\{ i \in \mathcal{T}_D :\ D(i,:) \neq \sum_{\ell\in\mathcal{T}_D\setminus\{i\}} \alpha_\ell D(\ell,:),\ \text{for all } \alpha_\ell \in \mathbb{R}_+ \Big\}. \tag{2.5}$$

The index set for colcone(D) is given analogously and is denoted by

$$\tilde{\mathcal{E}}_D = \Big\{ j \in \widetilde{\mathcal{T}}_D :\ D(:,j) \neq \sum_{\ell\in\widetilde{\mathcal{T}}_D\setminus\{j\}} \alpha_\ell D(:,\ell),\ \text{for all } \alpha_\ell \in \mathbb{R}_+ \Big\}. \tag{2.6}$$

These index sets allow to rephrase Eq. (2.3) and (2.4) using only the edges, cf. [8]. The resulting cone representations are given by

$$\begin{aligned} \operatorname{colcone}(D) &= \left\{ \sum_{j \in \tilde{\mathcal{E}}_D} \alpha_j D(:,j), \text{ for all } \alpha_j \in \mathbb{R}_+ \right\}, \\ \operatorname{rowcone}(D) &= \left\{ \sum_{i \in \mathcal{E}_D} \alpha_i D(i,:), \text{ for all } \alpha_i \in \mathbb{R}_+ \right\}. \end{aligned} \tag{2.7}$$

The rows and columns defining $\mathcal{E}_D$ and $\tilde{\mathcal{E}}_D$ are especially important for the cones. Therefore, the following definition is given.

Definition 2.6. *The i-th row of a data matrix D is called* essential *if it is an edge of the cone of the row space and the first occurrence of this row, including different scales, i.e. $i \in \mathcal{E}_D$. In the same way the j-th column is called essential if $j \in \tilde{\mathcal{E}}_D$.*

In the same style, the rows and columns that do not belong to an edge are denoted as *non-essential*. These terms are based on extremal vectors, see [68, 73], and will be discussed in detail in Sec. 2.3.7.

Since both cone representations of Eq. (2.3)-(2.4) and Eq. (2.7) are equivalent, a statement can be made about the essential rows and columns.

Remark 2.7. *There are at least $s = \operatorname{rank}(D) = \operatorname{rank}_+(D)$ essential rows or columns. This is because the cone representation of Eq.* (2.7) *spans the same space as when all rows or columns of D are considered, see Eq.* (2.3)-(2.4) *and* (2.7).

Even if this minimum is not reached in general, a drastic data reduction is often possible if only essential rows and columns are considered. This is of particular interest when considering large matrices. A reduction to the essential parts can reduce the subsequent computational cost by several orders of magnitude and speed up the analysis, see [25].

Determining possible edges of the cones for matrices with high dimensions can also be computationally expensive. To reduce this cost, it is possible to look at submatrices of the original data matrix D and determine their essential parts to obtain a superset of the essential rows and columns of D. This is shown in the following lemma.

Lemma 2.8 (From [8])**.** *For a given matrix $D = \begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix} \in \mathbb{R}_+^{k \times n}$ holds $\mathcal{E}_D \subseteq \mathcal{E}_{D_{11}} \cup \mathcal{E}_{D_{21}}$ for the index sets of essential rows, where the rows of D_{21} are numbered from $k_1 + 1$ to k. Here $D_{11} \in \mathbb{R}_+^{k_1 \times n}$, $D_{21} \in \mathbb{R}_+^{k_2 \times n}$ with $k = k_1 + k_2$ and $\mathcal{E}_D$ denotes the index set of essential rows of D, $\mathcal{E}_{D_{11}}$ belongs to D_{11} and $\mathcal{E}_{D_{21}}$ to D_{21}.*

Proof. Looking at the temporary index sets it is clear that $\mathcal{T}_D \subseteq \mathcal{T}_{D_{11}} \cup \mathcal{T}_{D_{21}}$ holds. Thus, the connection between the index sets of essential rows is as follows:

$$\begin{aligned} \mathcal{E}_D =& \left\{ i \in \mathcal{T}_D : \; D(i,:) \neq \sum_{\ell \in \mathcal{T}_D \setminus \{i\}} \alpha_\ell D(\ell,:), \text{ for all } \alpha_\ell \in \mathbb{R}_+ \right\} \\ \subseteq& \left\{ i \in \mathcal{T}_{D_{11}} : \; D(i,:) \neq \sum_{\ell \in \mathcal{T}_{D_{11}} \setminus \{i\}} \alpha_\ell D(\ell,:), \text{ for all } \alpha_\ell \in \mathbb{R}_+ \right\} \\ & \cup \left\{ i \in \mathcal{T}_{D_{21}} : \; D(i,:) \neq \sum_{\ell \in \mathcal{T}_{D_{21}} \setminus \{i\}} \alpha_\ell D(\ell,:), \text{ for all } \alpha_\ell \in \mathbb{R}_+ \right\} \\ =& \mathcal{E}_{D_{11}} \cup \mathcal{E}_{D_{21}} \end{aligned}$$

□

This lemma can also be formulated for the transposed case and essential columns. However, it is not possible to extend it directly for the essential rows and columns together in one step. A counterexample is given next.

Example 2.9. *Let the two matrices*

$$D_{11} = \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 1 \end{pmatrix}, \quad D_{21} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}$$

be given. For both matrices, all rows and the first two columns are essential. As in Lem. 2.8 the matrices are combined to $D = \begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$. *Now only the first 3 rows are essential and the essential columns of the submatrices are not a superset of the essential columns of* D*, since the third column can no longer be represented by a linear combination of the first two columns. This coincides with the fact that the rank of the individual submatrices is smaller than the rank of* D*. This relation is further investigated in chapter 3.*

The combination of the essential rows and columns by forming the Cartesian product $\mathcal{E}_D \times \tilde{\mathcal{E}}_D$ defines the term *essential information* of a matrix D. When D is reduced to its essential information, i.e., it contains only essential rows and columns, it is referred to as $D_{esi} \in \mathbb{R}_+^{k_{esi} \times n_{esi}}$, $k_{esi} \leq k$, $n_{esi} \leq n$.

When a data matrix is reduced to its essential parts, the question arises whether the structure is preserved. In particular, is a solution of the complete matrix is also feasible for the reduced matrix and vice versa? This may seem to be clear, because the essential rows and columns remain the same and thus seemingly the corresponding cones, but since also the dimensions and thus the spaces of the cones change, it is not so trivial. This relationship was already mentioned in [68], but is described in detail in the following lemma.

Lemma 2.10. *When determining the solution of a nonnegative matrix* $D \in \mathbb{R}^{k \times n}$, $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*, both* D *and its reduced version* $D_{esi} \in \mathbb{R}^{k_{esi} \times n_{esi}}$ *can be considered as equivalent. This means that any solution of the NMF problem of* D *can be transformed into a solution of the NMF problem of* D_{esi} *and vice versa.*

Proof. The reduced matrix D_{esi} is constructed from D by eliminating non-essential rows and columns. Thus, a solution that is feasible for D is also feasible for D_{esi} by performing the same reduction for both factors (eliminating the same rows of $C \in \mathbb{R}_+^{k \times s}$ and the same columns of $S^T \in \mathbb{R}_+^{s \times n}$ as in D).

The reduction of D to D_{esi} can also be done by looking at the entries of D_{esi} given by $D(p, q)$ with $p \in \mathcal{E}_D,\ q \in \tilde{\mathcal{E}}_D$. To prove that D_{esi} results in the same feasible factors as D, it is necessary to also look at the non-essential parts of D. For all rows and columns of D holds

$$D(i,:) = \sum_{p \in \mathcal{E}_D} \hat{\alpha}_{i,p} D(p,:) \text{ and } D(:,j) = \sum_{q \in \tilde{\mathcal{E}}_D} \hat{\beta}_{j,q} D(:,q), \text{ with } \hat{\alpha}_{i,p},\ \hat{\beta}_{j,q} \in \mathbb{R}_+.$$

Thus, each matrix entry can be constructed using D_{esi} by

$$D(i,j) = \sum_{p \in \mathcal{E}_D} \sum_{q \in \tilde{\mathcal{E}}_D} \hat{\alpha}_{i,p} \hat{\beta}_{j,q} D(p,q) = \sum_{r=1}^{k_{esi}} \sum_{t=1}^{n_{esi}} \alpha_{i,r} \beta_{j,t} D_{esi}(r,t),$$

with $\alpha_{i,r},\ \beta_{j,t} \in \mathbb{R}_+$ corresponding to $\hat{\alpha}$ and $\hat{\beta}$ respectively.

The same can be done for the factorization $D_{esi} = C_{esi} S_{esi}^T$, with the factors $C_{esi} \in \mathbb{R}^{k_{esi} \times s}$, $S_{esi}^T \in \mathbb{R}^{s \times n_{esi}}$ and using the r-th and t-th unit vector e_r and e_t

$$D(i,j) = \sum_{r=1}^{k_{esi}} \sum_{t=1}^{n_{esi}} \alpha_{i,r} \beta_{j,t} e_r^T D_{esi} e_t = \sum_{r=1}^{k_{esi}} \sum_{t=1}^{n_{esi}} \alpha_{i,r} \beta_{j,t} e_r^T C_{esi} S_{esi}^T e_t = \sum_{r=1}^{k_{esi}} \alpha_{i,r} e_r^T C_{esi} \cdot \sum_{t=1}^{n_{esi}} \beta_{j,t} S_{esi}^T e_t.$$

With this it is possible to define the factors $C \in \mathbb{R}_+^{k \times s}$ and $S^T \in \mathbb{R}^{s \times n}$ of D by

$$C(i,:) = \sum_{r=1}^{k_{esi}} \alpha_{i,r} e_r^T C_{esi}, \; i = 1, \dots k \quad \text{and} \quad S^T(:,j) = \sum_{t=1}^{n_{esi}} \beta_{j,t} S_{esi}^T e_t, \; j = 1, \dots, n.$$

Since $\alpha_{i,r} \geq 0$ and $\beta_{j,t} \geq 0$, as well as C_{esi} and S_{esi}^T are nonnegative, C and S^T are also nonnegative. So C and S^T form a feasible pair of D. Therefore, D and D_{esi} can be considered equivalent when determining the solutions of D. □

Thus, D_{esi} is sufficient to compute solutions of the NMF problem for D and consequently the cone representation of D_{esi} can be used to compute the solutions of D instead of the cone representation of D.

This also means that if for two matrices D_1 and D_2 the corresponding matrices $D_{1,esi}$ and $D_{2,esi}$ are identical (up to permutation and scaling), they can be considered equivalent and a solution only needs to be computed for one matrix to obtain a solution for the other. This results in a minimization of the computational effort, since the solutions for a matrix do not need to be computed if they are known for an "equivalent" matrix. This relation is of particular interest in the context of incomplete data sets and online analysis and is further investigated in chapter 3.

After introducing the cone-based approach and seeing how the NMF problem 2.3 can be solved, it is possible to list its advantages and disadvantages. On the one hand, the cone-based approach is a simple way to represent a matrix by using the row and column spaces. No additional computational steps are required to achieve this representation. However, the "quality" of the representation and how easy it is to find enclosing cones and thus a solution depends strongly on the dimensions of D. A large number of rows and/or columns leads to a high-dimensional row or column space, even if the rank of D is small. Thus, the representation of high-dimensional data is difficult.

This inevitably leads to the question of what other ways are possible to deal with high-dimensional data. An approach based on the singular value decomposition of the data matrix, which takes advantage of the low rank of D with respect to k and n, is presented next.

2.3. Solving the NMF problem using a singular value decomposition

In this section, the solutions of the NMF problem are determined with an approach based on singular value decomposition. The aim is not only to determine one possible solution of the NMF problem, but to extend the approach to determine all solutions. Furthermore, the connection to the cone approach of Sec. 2.2 is drawn together with the transfer of the concept of essential information. Finally, a short presentation of possible numerical approaches to determine all solutions of the NMF problem is given.

2.3.1. The singular value decomposition

Instead of using the rows and columns of a matrix directly to find solutions to the NMF problem 2.3, it is possible to use a low-rank approximation as a prior step and use this approximation to

represent D. For this purpose, the singular value decomposition (SVD) is introduced.

Theorem 2.11 (Singular value decomposition [28]). *If $D \in \mathbb{R}^{k\times n}$, then there exist orthogonal matrices $\hat{U} \in \mathbb{R}^{k\times k}$ and $\hat{V} \in \mathbb{R}^{n\times n}$ such that $\hat{U}^T D\hat{V} = \hat{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{k\times n}$ with $p = \min\{k, n\}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.*

Accordingly, D is represented by $\hat{U}\hat{\Sigma}\hat{V}^T$. Assuming $\operatorname{rank}(D) = s$, D can also be represented by $D = \sum_{i=1}^{s} \sigma_i u_i v_i^T$, where $u_i = \hat{U}(:, i)$ and $v_i = \hat{V}(:, i)$ are the singular vectors and σ_i the singular values. Thus, only the first s singular vectors and values are used to represent D, since the remaining singular values are equal to 0, see [28]. Therefore, the truncated matrices $U \in \mathbb{R}^{k\times s}$ and $V \in \mathbb{R}^{n\times s}$ are defined as $U = (u_1, \ldots, u_s)$ and $V = (v_1, \ldots, v_s)$ and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_s) \in \mathbb{R}^{s\times s}$. This means that $D = U\Sigma V^T$ is a truncated SVD and will be referred to as SVD of D.

Using the SVD, a factorization of D with proper dimension and rank is already obtained to find a solution to the NMF problem 2.3, using $C = U\Sigma$ and $S^T = V^T$. However, C and S^T contain negative entries for $s > 1$, see the following Thm. 2.13 and Lem. 2.14. The goal is to transform the factors so that they become nonnegative. Therefore, the factor $T \in \mathbb{R}^{s\times s}$ with $\operatorname{rank}(T) = s$ is introduced, so that $C = U\Sigma T^{-1}$ and $S^T = TV^T$ are nonnegative matrices and a feasible pair of D. This concept of transforming C and S^T so that they become nonnegative is known as target transformation [46].

To transform C and S^T into a feasible pair, multiple matrices T can be found. Thus, multiple feasible pairs are also possible. This ambiguity is known from Sec. 2.1.1, since the NMF problem is ill-posed. The ambiguity is investigated next to identify not only one solution of the NMF problem but all.

2.3.2. Ambiguity of the solutions of the NMF problem

The ambiguity of the solution to the NMF problem is known as *rotational ambiguity* in the context of chemometrics [1, 82]. However, it is not a rotation in the mathematical sense and is therefore simply referred to as ambiguity in this thesis. Ambiguity is documented in the early work of chemometrics in [38] and [10]. It is an ongoing research topic to determine the ambiguity and find ways to reduce it. Possible ways to reduce it are explored in the following chapters.

The heart of ambiguity analysis is the factor $T \in \mathbb{R}^{s\times s}$ with $\operatorname{rank}(T) = s$, so that $C = U\Sigma T^{-1}$ and $S^T = TV^T$ form a feasible pair of $D \in \mathbb{R}_+^{k\times s}$, $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, with its SVD $D = U\Sigma V^T$. Finding such a T can be done by solving optimization problems. There are s^2 degrees of freedom, corresponding to the number of entries of T. To allow faster analysis, the goal is to reduce the degrees of freedom. To achieve this goal, prior considerations must be made. Thus, the terms "reducible" and "irreducible" are needed and used in the following theorem. This allows to classify matrices.

Definition 2.12 (From [65], see also [48]). *A matrix $M \in \mathbb{R}^{n\times n}$ with $n \geq 2$ is called* reducible, *if there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$, so that*

$$PMP^T = \begin{pmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{pmatrix}$$

with $M_{11} \in \mathbb{R}^{m\times m}$ and $M_{12} \in \mathbb{R}^{m\times(n-m)}$, $1 \leq m < n$. Otherwise M is called irreducible.

The notion of irreducibility is used as a condition in the theorem of Perron and Frobenius [21,55], which concerns the spectrum of a nonnegative matrix. The theorem is reduced to two aspects needed in this work and is based in its form on [65].

Theorem 2.13 (Perron-Frobenius). *Let $M \in \mathbb{R}^{n\times n}$ be a nonnegative and irreducible matrix. Then*

(i) the spectral radius of M is a simple eigenvalue of M and

(ii) there exists a component-wise positive eigenvector corresponding to this eigenvalue.

Proof. See [81]. □

Applying this theorem to a matrix $D \in \mathbb{R}^{k\times n}$ with its corresponding SVD, where the eigenvectors of D^TD are stored in V (with D^TD an irreducible matrix), yields the following lemma. An application to the transposed case DD^T with the eigenvectors stored in U is analogous.

Lemma 2.14 (Thm 3.1 from [51]). *Suppose $D \in \mathbb{R}^{k\times n}$, with* $\operatorname{rank}(D) = s > 2$*, is a nonnegative matrix. If D^TD is an irreducible matrix, then "any nonzero linear combination of the columns $2, \ldots, s$ of V has at least one negative component" [51].*

Proof. See [51]. □

The key of this lemma is that if S^T is constructed from the columns of V, by $S^T = TV^T$, then there is always a contribution from the first column of V, since S^T must be nonnegative to be feasible. Thus, all entries of the first column of T are unequal to zero. Using the notation $T(i,j) = t_{ij}$ results in $t_{i1} \neq 0$ for $i = 1, \ldots, s$.

The next step is to reduce the degrees of freedom of T. So far, all s^2 entries of T had to be determined. To achieve this goal, a scaling matrix $\Delta = \operatorname{diag}(t_{11}, \ldots, t_{s1}) \in \mathbb{R}_+^{s\times s}$ is inserted so that $D = U\Sigma T^{-1}TV^T = U\Sigma T^{-1}\Delta\Delta^{-1}TV^T$, see [51]. This choice of Δ is possible because $t_{i1} \neq 0$ for $i = 1, \ldots, s$. This results in the first column of $\Delta^{-1}T$ containing only ones. This means that the scaling of the resulting factors C and S^T is fixed. This step eliminates the scaling ambiguity, see for example [51]. Consequently, the degrees of freedom of T are reduced to $s^2 - s$. The justification for this reduction is, that no qualitatively different solutions to the NMF problem are excluded during this step [65]. This means that a solution C and S^T is equivalent to a solution $\widehat{C} = a \cdot C$ and $\widehat{S}^T = a^{-1} \cdot S^T$, with $a > 0$, which is considered to be qualitatively the same.

It is possible to reduce the degrees of freedom of T even further. This is because two matrices $\widehat{C}$ and C are considered qualitatively the same if they differ only in the order of their columns. So a permutation matrix $P \in \mathbb{R}_+^{s\times s}$ is used. The permutation of the columns of C and S is done by $D = U\Sigma T^{-1}TV^T = CS^T = U\Sigma T^{-1}PP^TTV^T = \widehat{C}\widehat{S}^T$, see [51]. Thus, C and $\widehat{C}$ are considered as equivalent if $\widehat{C} = PC$. This step eliminates the permutation ambiguity.

Therefore, the focus can be shifted from finding a complete factor to finding the first column of C and S such that there are $s-1$ columns such that C is feasible. This step is transferred to T. Thus, the problem of finding T is reduced to finding the first row of T. Additionally, for each first row, $s-1$ remaining rows must be found so that T is still invertible and leads to a feasible pair. For this construction it is sufficient to find one set of possible $s-1$ rows and not all possible remaining rows.

Together with the scaling matrix Δ this leads to

$$P^T\Delta^{-1}T = \begin{pmatrix} 1 & x_1 & \ldots & x_{s-1} \\ 1 & w_{11} & \ldots & w_{1(s-1)} \\ \vdots & \vdots & & \vdots \\ 1 & w_{(s-1)1} & \ldots & w_{(s-1)(s-1)} \end{pmatrix}, \tag{2.8}$$

with $(w_{i1}, \ldots, w_{i(s-1)})^T$ the i-th remaining row.

For simplification, from now on the T to be sought has the structure of (2.8). Thus, the problem of finding a suitable T can be reduced to finding $x_1, \ldots, x_{s-1}$ where suitable w_{ij} exist. This structure of T allows to consider not only at one feasible solution for the NMF problem 2.3, but all possible feasible pairs, the so-called set of feasible solutions.

Definition 2.15 (Set of feasible solutions (SFS), see [51, 65])**.** *Let $D \in \mathbb{R}_+^{k\times n}$ with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s \geq 2$ and $D = U\Sigma V^T$ an SVD of D. Further, D^TD and DD^T are irreducible. A vector $x \in \mathbb{R}^{s-1}$ is called* feasible*, if a $W \in \mathbb{R}^{(s-1)\times(s-1)}$ exists so that*

$$T = \begin{pmatrix} 1 & x^T \\ \mathbf{1} & W \end{pmatrix}$$

is an invertible matrix, with $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^{s-1}$, and the factors $C = U\Sigma T^{-1}$ and $S^T = TV^T$ are nonnegative. The set of feasible solutions (SFS) *can thus be defined as*

$$\mathcal{M}_{\mathcal{S}} = \{x \in \mathbb{R}^{s-1} : \text{ exists } W \in \mathbb{R}^{(s-1)\times(s-1)}, \text{ with } T = \begin{pmatrix} 1 & x^T \\ \mathbf{1} & W \end{pmatrix}, \operatorname{rank}(T) = s \text{ and } C, S \geq 0\} \tag{2.9}$$

The set of feasible solutions $\mathcal{M}_{\mathcal{S}}$ can be represented in the space $\mathbb{R}^{s-1}$. With this representation, the question arises how to determine $\mathcal{M}_{\mathcal{S}}$.

2.3.3. Geometric construction of the SFS

Borgen and Kowalski [10] introduced a geometric construction in 1985 for $s = \operatorname{rank}(D) = \operatorname{rank}_+(D) = 3$. For arbitrary s it was later generalized in [61] and [34]. The following explanation and notation is based on [34] and [67].

Based on Eq. (2.9) it is possible to bound the SFS. Therefore, the relation $TV^T = S^T \geq 0$ is used, but only the first column of S and thus the first row of T is considered. This means that only the nonnegativity constraint of $\mathcal{M}_{\mathcal{S}}$ is taken into account. It follows that $\mathcal{M}_{\mathcal{S}}$ is a subset of

$$\mathcal{F}_S = \{x \in \mathbb{R}^{s-1} : (1, x^T)V^T \geq 0\}. \tag{2.10}$$

The convex set $\mathcal{F}_S$ is called the outer polytope or FIRPOL (based on the term "first polygon" [10, 61]).

To include the remaining constraints of Eq. (2.9) a second polytope is used. It is based on the convex hull of the rows of D. This is because all rows of D must be representable by a conic combination of the columns of S. This ensures the nonnegativity of C. The second polytope is given by

$$a_i = \frac{((DV)(i, 2:s))^T}{(DV)(i,1)} = \frac{((U\Sigma)(i, 2:s))^T}{(U\Sigma)(i,1)}, \quad \mathcal{I}_S = \operatorname{convhull}\{a_i : \ i = 1, \ldots, k\}. \tag{2.11}$$

The convex set $\mathcal{I}_S$ is called the inner polytope or INNPOL (based on the term "inner polygon" [10, 61]). Both polytopes are shown for data set 1 in Fig. 2.2.

With these two polytopes it is now possible to determine $\mathcal{M}_{\mathcal{S}}$ geometrically. This is based on [10, 61] and the following formulation is based on Thm. 4.6 from [51].

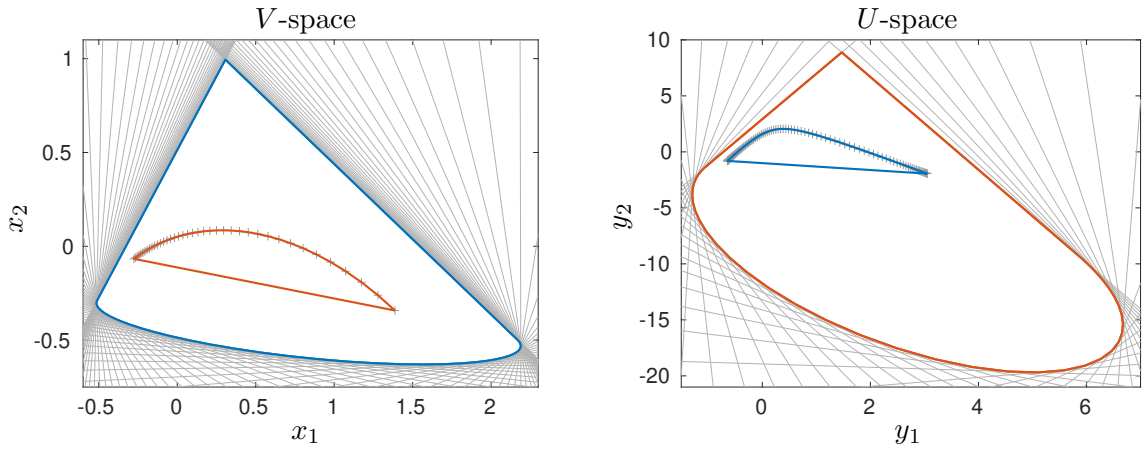

Figure 2.2.: Left: V-space of data set 1. The outer polygon from Eq. (2.10) is marked in blue and the inner polygon from Eq. (2.11) is marked in red with the data points a_i as gray crosses. The facets that bound $\mathcal{F}_S$ are marked with gray lines, see Sec. 2.3.5. Right: The corresponding U-space. The outer polygon from Eq. (2.12) is marked in red and the inner polygon from Eq. (2.13) is marked in blue, with the corresponding data points b_j marked by gray crosses.

Theorem 2.16. *Let $D \in \mathbb{R}_+^{k\times n}$ be given with DD^T and D^TD irreducible matrices and* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$. *Then x is an element of $\mathcal{M}_\mathcal{S}$, if and only if $x \in \mathcal{F}_S$ and the convex hull of x and $s-1$ other vectors in $\mathcal{F}_S$ form a simplex containing $\mathcal{I}_S$.*

Proof. See [51], Thm. 4.6. □

This theorem enables a geometric construction of $\mathcal{M}_S$ by finding all simplices that enclose $\mathcal{I}_S$ and lie within $\mathcal{F}_S$. A visualization for $\mathcal{M}_S$ of data set 1 is shown in Fig. 2.3 with a possible simplex marked by black dashed lines. This construction can be seen as a connection between the cone approach of Sec. 2.2, in particular the nested polygon problem described there and the SVD approach. In both cases, a polytope must be found that contains all points corresponding to the rows or columns of D and is bounded by a second polytope (for cones defined by the positive orthant).

The geometric construction of $\mathcal{M}_S$ from Thm. 2.16 leads to the following lemma.

Lemma 2.17 (Lem. 3.5 from [67])**.** *Let $D \in \mathbb{R}_+^{k\times n}$ be given with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ *and DD^T and D^TD irreducible. Then the origin ($x = 0 \in \mathbb{R}^{s-1}$) is an interior point of $\mathcal{I}_S$.*

Proof. See [67]. □

This implies that $\mathcal{M}_\mathcal{S}$ does not contain the origin. In addition, it is known that $\mathcal{M}_\mathcal{S}$ is bounded and also that $\mathcal{I}_S$ and $\mathcal{F}_S$ are convex polytopes [51].

Instead of looking only at T and thus at feasible factors S^T resulting in a corresponding C, it is possible to do it the other way around. That is, determine C directly based on a geometric construction. Therefore, the geometric construction is applied to D^T. Equivalent to Eq. (2.10) and (2.11) the outer and inner polytopes are defined by

$$\mathcal{F}_C = \{y \in \mathbb{R}^{s-1} : \; U\Sigma \begin{pmatrix} 1 \\ y \end{pmatrix} \geq 0\}. \tag{2.12}$$

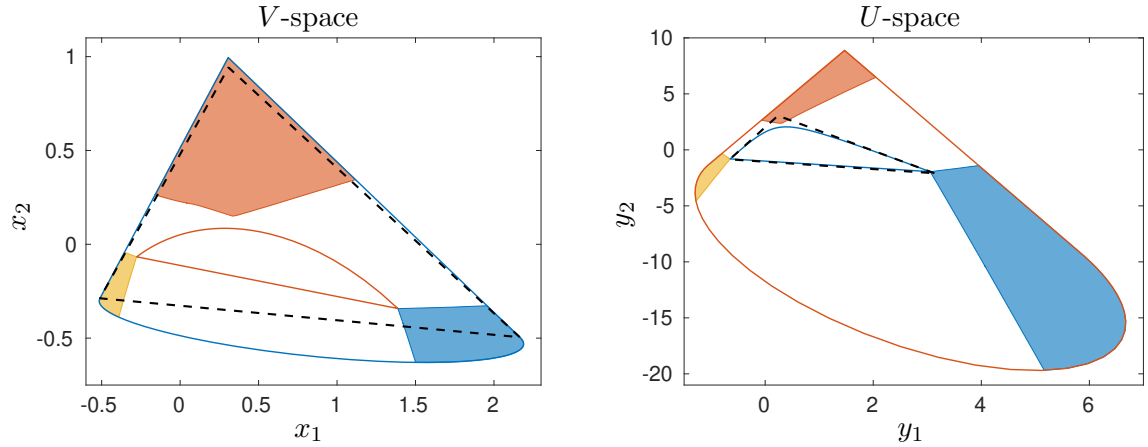

Figure 2.3.: The same low-dimensional representation of data set 1 as in Fig. 2.2, but with the SFS added and a solution marked by dashed triangles. This solution corresponds to the factors shown in Fig. A.1, where the color of the profiles corresponds to the color of the areas they are in. Left: V-space of data set 1 with marked $\mathcal{M}_\mathcal{S}$. Right: Corresponding U-space with marked $\mathcal{M}_\mathcal{C}$. The corresponding profiles with their band representation are shown in Fig. 2.4.

Thus, instead of the rows, the columns of D are considered with

$$b_j = \frac{((U\Sigma)^{-1}D)(2:s,j)}{((U\Sigma)^{-1}D)(1,j)} = \frac{V^T(2:s,j)}{V^T(1,j)}, \quad \mathcal{I}_C = \text{convhull}\{b_j : \ j = 1,\dots,n\}. \tag{2.13}$$

The two polytopes polytopes $\mathcal{I}_C$ and $\mathcal{F}_C$ are shown in Fig. 2.2 on the right. Additionally the b_j are marked with gray crosses. The b_j as well as the a_i from Eq. (2.11) play a special role in the further theory, which motivates the next definition.

Definition 2.18. *The a_i from Eq. (2.11) and b_j from Eq. (2.13) are called data representing points or just* data points. *This term is motivated by the fact that the i-th row of the data matrix is represented by a_i. The same holds for the j-th column and b_j.*

Analogous to Thm. 2.16, solutions can be found with the inner and outer polytopes $\mathcal{I}_C$ and $\mathcal{F}_C$. Thus, simplices are formed that contain $\mathcal{I}_C$ and are inside $\mathcal{F}_C$. The resulting set of solutions is

$$\begin{aligned}\mathcal{M}_\mathcal{C} = \Big\{ y \in \mathbb{R}^{s-1} : \text{exists } T \in \mathbb{R}^{s\times s}, \text{ with } \operatorname{rank}(T) = s, \\ (T^{-1})(:,1) = \begin{pmatrix} 1 \\ y \end{pmatrix}, \text{ and } U\Sigma T^{-1} \geq 0, \ TV^T \geq 0 \Big\}.\end{aligned} \tag{2.14}$$

However, $\mathcal{M}_\mathcal{C}$ is not directly equivalent to $\mathcal{M}_\mathcal{S}$ when looking at T. This is due to a different scaling since each set ensures that the first row of T or T^{-1} contains only ones. Thus, $\mathcal{M}_\mathcal{C}$ and $\mathcal{M}_\mathcal{S}$ are equivalent up to scaling. A set that is equivalent to $\mathcal{M}_\mathcal{C}$ except for scaling is shown together with $\mathcal{F}_C$ and $\mathcal{I}_C$ in Fig. 2.3. The spaces defined by the inner and outer polytopes are denoted by the V-space for $\mathcal{F}_S$ and $\mathcal{I}_S$ and the U-space when looking at $\mathcal{F}_C$ and $\mathcal{I}_C$, see [32]. The set of both spaces is called the low-dimensional representation.

The set of feasible solutions is thus visualized by subsets of the outer polytopes in the low-dimensional representation. It is also possible to represent the ambiguity by plotting the columns of feasible C and S, known as profiles. In addition, the envelope of all profiles, known as band boundaries, can be represented. This allows easier visual comparison of different ambiguities, e.g., when analyzing submatrices, and is used in chapter 3. It can also be used to visually verify solutions.

2.3.4. Band boundary representation of the SFS

The SFS representation using band boundaries is used to visualize all solutions of the NMF problem. It is based on [24] and [78]. The idea is to determine the minima and maxima that can be reached by the columns of all feasible C and S, respectively. The columns of C and S are also called profiles, which comes from chemometrics. The upper and lower bounds for S are determined element-wise by

$$u_{S,i} = \max_{x \in \mathcal{M}_S} (1, x^T)(V^T(:, i)),$$
$$l_{S,i} = \min_{x \in \mathcal{M}_S} (1, x^T)(V^T(:, i)).$$

Analogously the band boundaries can be defined for C.

When $\mathcal{M}_S$ consists of separate subsets, the boundaries u_S and l_S are usually determined for each of these subsets separately. The resulting corresponding band boundaries of data set 1, with the low-dimensional representation in Fig. 2.3, are shown in Fig. 2.4. The profiles corresponding to the simplices of Fig. 2.3 are marked with black dashed lines. These profiles coincide with the profiles of Fig. A.1, on which the data set is based.

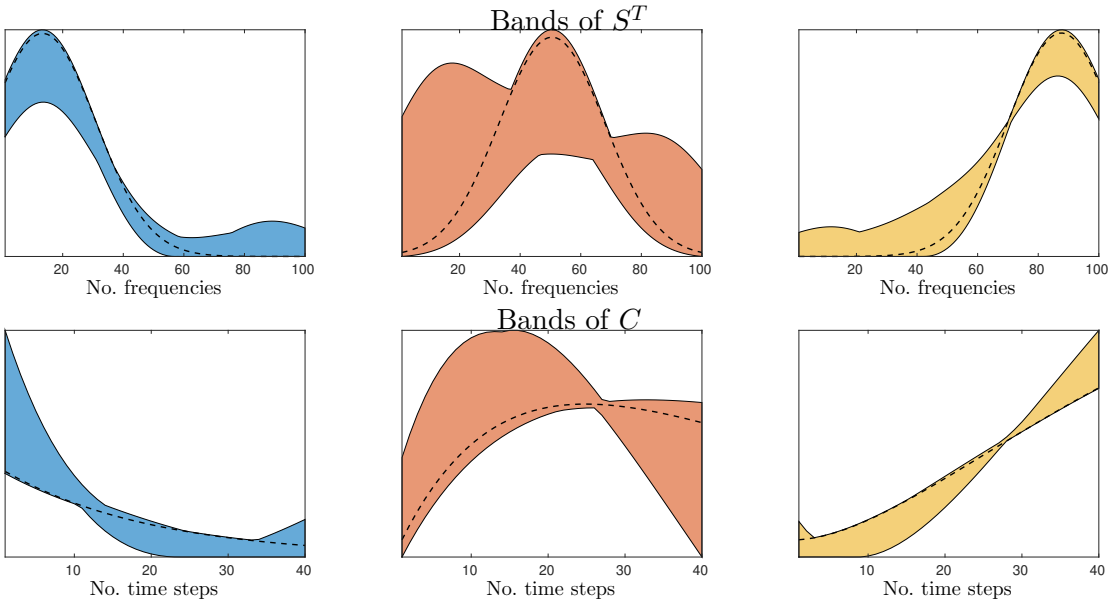

Figure 2.4.: Visualization of the SFS of Fig. 2.3 with the profiles corresponding to the selected simplex (triangle) as dashed lines. Upper row: Bands corresponding to the V-space and the factor S^T. Bottom row: Bands corresponding to the U-space and the factor C.

So far, the band boundaries, the SFS, and the U- and V-space have been considered separately for each factor. However, the factors C and S^T are directly related to each other, since a chosen feasible C already results in a feasible S^T. This connection is known as duality and describes how the U- and V-space are coupled. This is discussed next.

2.3.5. Duality

The concept of duality in the context of the U- and V-space was proposed by Henry [32]. As stated there, this concept "does not derive a duality relationship in the classical mathematical sense; it presents a duality relationship that is useful in multivariate receptor modeling", it

is a duality relation in the form of an isomorphism, according to [32]. It was conveyed to (multivariate) curve resolution by Rajkó [59], which is the basis of the duality used here.

To define and use duality again, the polytope $\mathcal{F}_S$ of Eq. (2.10) of the V-space is considered. The boundary of this polytope is determined by looking at where the inequality of Eq. (2.10), namely $V(:,1) + \sum_{i=1}^{s-1} x_i V(:,i) \geq 0$, becomes an equality, see [51, 67] and the basics in [32, 59]. Thus, the facets (or for $s = 3$ the edges) bounding $\mathcal{F}_S$ can be written as

$$E_j^S = \left\{ x \in \mathbb{R}^{s-1} : \frac{V(j, 2:s)x}{V(j,1)} = -1 \right\}$$

with $j = 1, \ldots, n$. These edges are shown as gray lines in Fig. 2.2 for data set 1.

Keeping in mind the data points b_j that define $\mathcal{I}_C$ from Eq. (2.13), the facets of $\mathcal{F}_S$ can be expressed by

$$E_j^S = \{ x \in \mathbb{R}^{s-1} : b_j^T x = -1 \}.$$

Thus, there is a relation between the data points in one space and the facets of the outer polytope in the other space. This relation can be formulated in a more general way and is called duality.

Definition 2.19 (From [67]). *An affine hyperplane* $E = \{x \in \mathbb{R}^{s-1} : \ z_E^T x = -1\}$ *and a vector* $z \in \mathbb{R}^{s-1}$ *are called* dual *if* $z_E = z$.

Correspondingly, points in the U-space are dual to affine hyperplanes in the V-space and vice versa.

The facets that bound $\mathcal{F}_C$ in the U-space are defined by

$$E_i^C = \left\{ y \in \mathbb{R}^{s-1} : \frac{(U\Sigma)(i, 2:s)y}{U(i,1)\sigma_1} = -1 \right\}.$$

Thus, all points on the boundary of $\mathcal{I}_S$ have a dual affine hyperplane which is a tangent plane to $\mathcal{F}_C$, see [67]. Points on $\mathcal{F}_C$ are also dual to tangent planes of $\mathcal{I}_S$. These relations also hold for $\mathcal{F}_S$ and $\mathcal{I}_C$, see [67]. More generally, an arbitrary point $y \in \mathbb{R}^{s-1}$ is dual to the hyperplane $E = \{x \in \mathbb{R}^{s-1} : \ y^T x = -1\}$, independent of the corresponding space.

Using this relation, the U- and V-space are called dual. This is visualized in Fig. 2.5, where dual objects are marked with the same color. This concept allows to compute the outer polytopes $\mathcal{F}_S$ and $\mathcal{F}_C$ in a fast way (see [67]) by using the given data points and duality instead of approximating the outer polytopes with half-spaces.

Duality also allows to describe the relation between the two factors C and S^T. When S^T, and thus T, is defined by s points in the V-space, the corresponding points of C in the U-space are automatically given. This is because the dual facets of the points defining S^T form a simplex whose vertices are the points belonging to C.

With the concept of duality and the SFS as a representation of the ambiguity at hand, it is now possible to look back to the cone approach of Sec. 2.2. Both approaches can be used to solve the NMF problem 2.3, but they are not yet comparable. Therefore, a connection will be drawn next.

2.3.6. Connection of the SVD approach to the cone representation

The low-dimensional representation and the cone representation from Sec. 2.2 are connected by the extremal points or vectors of each representation. First, the connection between both

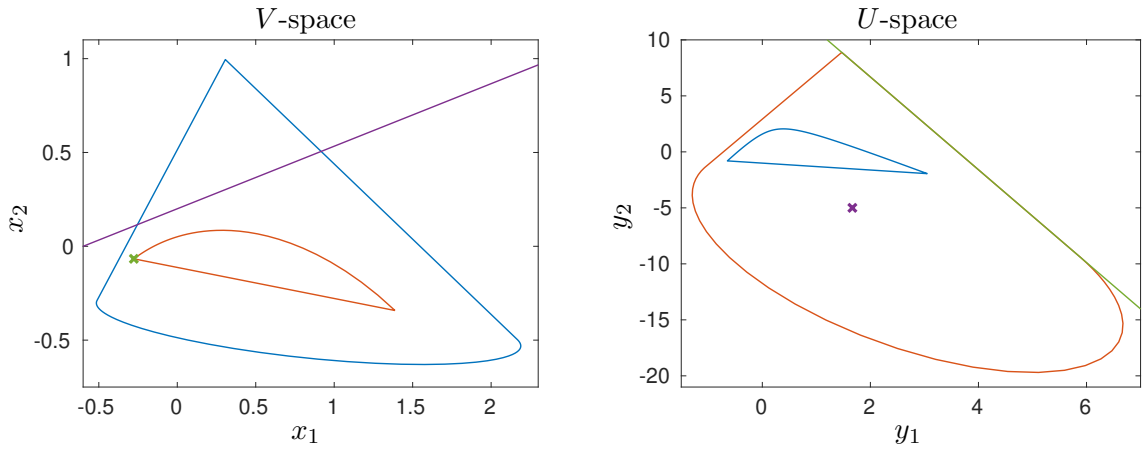

Figure 2.5.: Display of duality for data set 1. Dual objects are marked with the same color. This includes the inner and outer polytopes, as well as the green point on $\mathcal{I}_S$ selected in the V-space, which corresponds to an edge of the outer polytope in the U-space. Similarly, a point in U-space is selected with its corresponding dual line in the V-space, both in purple.

representations is made and a closer look at the extremal points and vectors is taken in Sec. 2.3.7.

Lemma 2.20 (From [8])**.** *Let $D \in \mathbb{R}_+^{k\times n}$ be given with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ *and an SVD $D = U\Sigma V^T$. Both DD^T and D^TD are irreducible. The row vector $D(j,:)$ is an edge of* $\operatorname{rowcone}(D)$ *if and only if the data point a_j, corresponding to $D(j,:)$, is a vertex of $\mathcal{I}_S$.*

Proof. The following equivalences hold:

$D(j,:)$ is an edge of $\operatorname{rowcone}(D)$

$\Leftrightarrow$ $D(j,:)$ is not representable by the other edges as $\sum_{i\in\mathcal{E},\, i\neq j} \alpha_i D(i,:)$ with nonnegative coefficients α_i

$\Leftrightarrow$ the V-space projection $e_j^T DV = e_j^T U\Sigma$ is not representable as $\sum_{i\in\mathcal{E},\, i\neq j} \alpha_i e_i^T DV = \sum_{i\in\mathcal{E},\, i\neq j} \alpha_i e_i^T U\Sigma$

$\Leftrightarrow$ the data point a_j is not representable as $\sum_{i\in\mathcal{E},\, i\neq j} \alpha_i a_i$

$\Leftrightarrow$ a_j is a vertex of $\mathcal{I}_S$.

□

It is also possible to strengthen the connection by looking at feasible solutions of the NMF problem. It must be shown that a solution in the cone representation is also a solution in the low-dimensional representation of the SVD.

Lemma 2.21. *Let $D \in \mathbb{R}_+^{k\times n}$, with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*, DD^T and D^TD irreducible, and $S^T \in \mathbb{R}_+^{s\times n}$ be given. If* $\operatorname{rowcone}(S^T) \supseteq \operatorname{rowcone}(D)$*, then the simplex in the low-dimensional representation of D corresponding to S^T encloses $\mathcal{I}_S(D)$ and vice versa.*

Proof. An analogous argumentation as was used in Lem. 2.21 is applied. The data representatives of S^T, meaning the vertices of the simplex defined by S^T, are denoted with dp_{S^T}.

$$
\begin{aligned}
& \text{rowcone}(S^T) \text{ encloses rowcone}(D) \\
\Leftrightarrow \quad & \text{for all rows of } D \text{ holds } e_j^T D = \sum_{i=1}^{s} \alpha_i e_i^T S^T \text{ with } \alpha_i \in \mathbb{R}_+,\ j = 1, \ldots, k \\
\Leftrightarrow \quad & \text{the } V\text{-space projection leads to } e_j^T DV = e_j^T U\Sigma = \sum_{i=1}^{s} \alpha_i e_i^T S^T V,\ \alpha_i \in \mathbb{R}_+,\ j = 1, \ldots, k \\
\Leftrightarrow \quad & \text{for all data points holds } a_j = \sum_{i=1}^{s} \alpha_i dp_{S^T},\ \alpha_i \in \mathbb{R}_+,\ j = 1, \ldots, k \\
\Leftrightarrow \quad & dp_{S^T} \text{ enclose } \mathcal{I}_S.
\end{aligned}
$$

□

The same reasoning can be done for the factor C and thus for all feasible solutions.

Remark 2.22. *Because of Lem. 2.20, the rows and columns corresponding to vertices and edges of the low-dimensional and the cone representation, respectively, are the same. With Lem. 2.21 the sets of all feasible pairs are also the same, up to scaling. Thus, both representations are equivalent.*

Lem. 2.10 has already shown that D and D_{esi} lead to the same solutions of the NMF problem. Especially when considering the SVD approach, this connection is not so obvious, since both matrices D and D_{esi} have different SVDs (if not all columns and rows of D are essential). This means that the low-dimensional representations are different. But because of Lem. 2.10, even with two different low-dimensional representations, both lead to the same solutions of the NMF problem.

Since these considerations are based on the vertices of the inner polytopes, the next step is to take a closer look at these extremal points.

2.3.7. Essential information in the low-dimensional representation

The concept of essential information appears not only in the cone representation, as in Sec. 2.2.1, but also when using the SVD.

SVD-based approaches to determine the essentiality of certain rows and/or columns of D were first published simultaneously in 2019 by Ghaffari et al. [25] and Sawall et al. [68]. However, methods to prioritize certain rows and columns and classifying them as more important than others were already known, especially in the context of linear unmixing [41]. The motivation behind these approaches was to obtain the purest rows and columns. In the chemical context, this means to obtain the spectra, channels, or pixel (depending on the application) where one chemical species appears ideally isolated, and thus pure, or with relatively small interferences from other chemical species. This is equivalent to obtaining extremal absorptivity relations, see [68].

In the context of essential information, the motivation shifts. The interest is no longer only to get the "purest" rows and columns to allow easier unmixing, i.e. to get a good guess at a possible nonnegative matrix factorization of the data set. It is also of interest to reduce the data set in a way that all information is preserved. This is especially desirable in the context of big data, where the analysis of large data sets leads to high computational cost.

When considering the low-dimensional representation, this data reduction can be formalized while preserving all information, see [68]. Information preservation for the low-dimensional representation means that the ambiguity remains the same. The ambiguity is defined by the inner and outer polytopes, in particular by their shape and size. This is defined by D, not least by the size of D since it limits the number of vertices of the inner polytopes. The question arises whether a certain row or column affects the polytopes and thus the ambiguity. That is, if a certain row and thus the corresponding data point is neglected, does $\mathcal{I}_S$ decrease? Or if a certain row is added, does $\mathcal{I}_S$ increase? This effect corresponds to the row belonging to a vertex of $\mathcal{I}_S$. The same can be phrased for columns and $\mathcal{I}_C$. Thus, it is now possible to consider essential rows and columns in the context of using an SVD, see [63, 73].

Definition 2.23. *A row of $D \in \mathbb{R}_+^{k\times n}$ is called* SVD-essential *if its data point is a vertex of $\mathcal{I}_S$. If multiple rows project to the same vertex, only the first of those rows is considered SVD-essential. The same holds for a column of D and $\mathcal{I}_C$.*

Thus, all rows and columns that affect the inner polytopes are called SVD-essential. Depending on the application, terms such as essential pixel [25] or essential spectra and essential frequencies [73] are defined, where, e.g., the pixel are assigned to the rows and the frequencies to the columns.

Initially, essentiality was defined in only one direction, e.g., selecting only essential rows. The combination of selection along both axes has already been shown in Lem. 2.10 and can allow a drastic reduction of the original data set D. But in order to apply Lem. 2.10 to the current definition of essentiality, it must first be shown that both approaches to essentiality coincide.

Remark 2.24. *Both definitions of essentiality using cones (Def. 2.6) and using the low-dimensional representation (Def. 2.23) are equivalent. This is because with Lem. 2.20 the index set of the edges of* rowcone(D) *coincides with the index set of the vertices of $\mathcal{I}_S$. Thus, the term essential can be used for both definitions, regardless of the underlying approach.*

Therefore, analogous to Sec. 2.2, essential information is defined as the combination of the index sets of essential rows and essential columns, as well as D_{esi}, which is the reduction of D to its essential rows and columns.

A visualization of essentiality is shown for data set 2 in Fig. 2.6 in the low-dimensional representation. In the data matrix, the corresponding essential rows and columns, as well as the entries belonging to both, are marked. This is shown in Fig. 2.7. For data set 1 it can be seen in Fig. 2.2 that all data points are needed to span the inner polytopes and thus all rows and columns are essential.

The main advantage of Rem. 2.24 is that it allows a simplification of known relations. For example, the proofs of Thm. 2.6 and Lem. 2.8 of [68], which describe the connection between linear combinations of rows/columns and essentiality, can be given by drawing the connection between the inner polytopes and the cones of the row and column spaces. Analogously, this approach is used in chapter 3 to provide "simpler" proofs (compared to using data points and the low-dimensional representation).

Instead of looking only at the vertices of $\mathcal{I}_S$ and $\mathcal{I}_C$ it is possible to use duality (see Subsec. 2.3.5) and to consider the facets of $\mathcal{F}_C$ and $\mathcal{F}_S$ instead. Thus, Def. 2.23 can be reformulated (if rows/columns appearing multiple times up to scaling are excluded):
A data point, and thus its corresponding row or column, is essential if its dual affine hyperplane is a facet of the corresponding outer polytope.
This enables an application to noisy data sets, which is further investigated in Sec. 2.4.3.

When determining essential information, the key is to do it in a space of the right dimension. Up to now, the low-dimensional representation in $s-1$ dimensions was always considered when

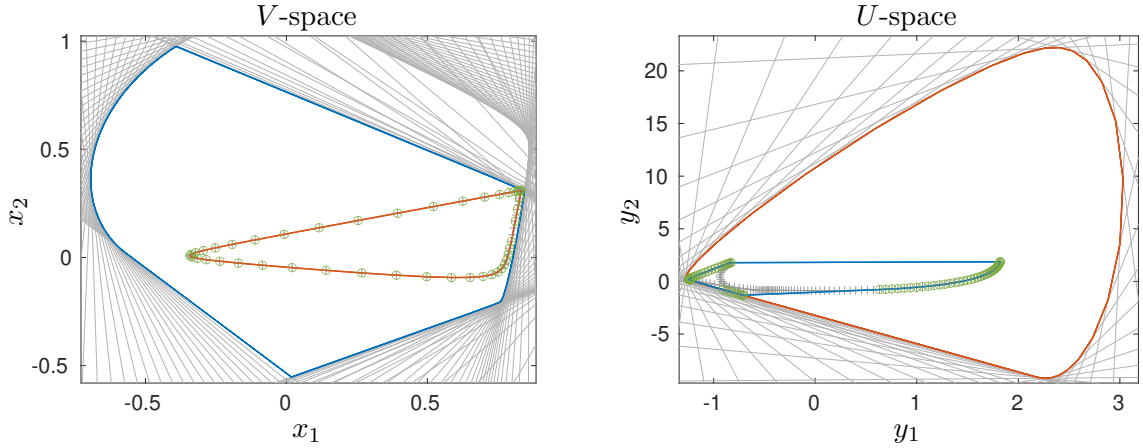

Figure 2.6.: Representation of data set 2 in the V-space (left) and U-space (right), with the data points marked by gray pluses and the data points corresponding to essential rows or columns are circled in green. It can be seen, that not all data points are needed to span the inner polytopes and thus not all data points are essential.

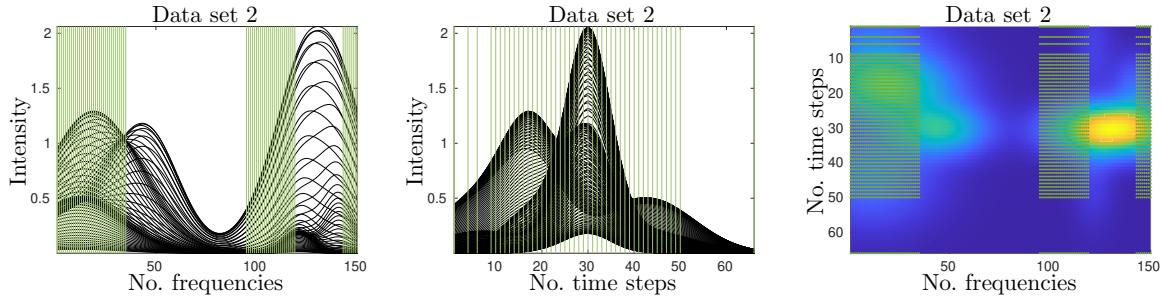

Figure 2.7.: Corresponding to Fig. 2.6 the essential rows and columns of the data matrix are shown in different representations of the data set 2. Left: The indices of the essential columns are highlighted in green. Center: Indices of essential rows are highlighted in green. Right: Only those entries of the data set that belong to both an essential row and an essential column are highlighted in green to visualize the total amount.

$\operatorname{rank}(D) = s$. But what if instead of $s-1$ dimensions the representation is done in $s-2$ dimensions? Or in an even smaller dimension, because the computational cost of determining the vertices is lower in that space?
The influence of such a choice of dimensions on the essential information is investigated next.

Theorem 2.25 (See Thm. 2.4 from [73])**.** *Let $\mathcal{I}_S$ be the inner and $\mathcal{F}_S$ the outer polytope in the $(s-1)$-dimensional low-dimensional representation of $D \in \mathbb{R}_+^{k\times n}$ with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$. The polytopes $\mathcal{I}'_S$ and $\mathcal{F}'_S$ denote the inner and outer polytopes belonging to the rank-m approximation of D with $m < s$. For an SVD-based low-rank approximation see Thm. 2.28. Suppose also that the linear subspace $\mathcal{P} = \{x \in \mathbb{R}^{s-1} : \ x_m = \ldots = x_{s-1} = 0\}$ is given.*
Then $\mathcal{I}'_S$ is the projection of $\mathcal{I}_S$ onto $\mathcal{P}$ and $\mathcal{F}'_S$ is the intersection of $\mathcal{F}_S$ with $\mathcal{P}$, where the points of both $\mathcal{I}'_S$ and $\mathcal{F}'_S$ are extended column-wise by $s-m$ zeros.

Proof. See [73], Thm. 2.4. □

This theorem can also be applied analogously to $\mathcal{I}_C$ and $\mathcal{F}_C$.

On the one hand, this theorem shows the effect of a wrongly assumed rank, namely that the

inner and outer polytopes can intersect. On the other hand, it shows how essential data points behave when D is approximated with different ranks m. This will be explained next.

Remark 2.26. *Looking only at the inner polytopes $\mathcal{I}_S$ and $\mathcal{I}'_S$ from Thm. 2.25 it becomes clear that only vertices of $\mathcal{I}_S$ can be vertices of $\mathcal{I}'_S$. For the essential index sets this means $\mathcal{E}_{D,m} \subseteq \mathcal{E}_{D,s}$, where the subscript s indicates the dimension $s-1$ used to determine $\mathcal{I}_S$ and the subscript m belongs to the $(m-1)$-dimensional $\mathcal{I}'_S$, with $m < s$. The same holds when looking at $\mathcal{I}_C$ and $\mathcal{I}'_C$, namely $\tilde{\mathcal{E}}_m \subseteq \tilde{\mathcal{E}}_s$. This can be extended to the essential information in general, i.e. $esi_m \subseteq esi_s$, where the subscripts s and m again indicate the dimension used.*

The basis of a proper selection of the essential information is the selection of the correct dimension in which the data set is represented, i.e. a properly selected rank s. This is of particular interest when considering data sets that are not noise-free, for example, when analyzing data sets based on chemical measurements. This case is considered in Sec. 2.4 for model data with simulated noise and in chapter 6 for measured experimental data.

Besides data reduction, the consideration of essential information has a wide range of applications. For example, it can be used for the design of experiments [68]. With a Fourier-based approach to determine the essential information, even accelerations during the measurements are possible [15]. But also in matrix augmentation essential information can be used to reduce ambiguity. Such a consideration is also a motivation for chapter 3, when several matrices are combined.

With this theory about the low-dimensional representation and its properties, it is clear that the essential information, and thus the vertices of $\mathcal{I}_S$ and $\mathcal{I}_C$, can be determined using the convex hull of the data points. But what about the set of feasible solutions? How can the SFS be efficiently determined without testing all possible points inside $\mathcal{F}_S$ and checking if they are feasible?

2.3.8. Numerical methods to approximate the SFS

In this section, possible approximations for the SFS $\mathcal{M}_\mathcal{S}$ in the V-space are shown. Approximations of $\mathcal{M}_\mathcal{C}$ in the U-space work analogously. Additionally, $\mathcal{M}_\mathcal{C}$ can be determined from $\mathcal{M}_\mathcal{S}$ using duality.

There are several ways to determine $\mathcal{M}_\mathcal{S}$, depending on the dimension of the problem, i.e., the rank of the factorization problem. For $s = 2$, Lawton and Sylvestre [38] described that the SFS covers the entire area between the inner and outer polytopes. The SFS can also be determined analytically for $s = 3$ using so-called boundary curves [6, 10, 61]. However, there is no analytical solution for $s > 3$. Therefore, alternative methods are needed to determine the SFS. Or rather, to approximate it, and this is where numerical methods come into play.

The following two methods are chosen because they are used later to compute the SFS and can be adapted for noisy data sets.

Polygon inflation algorithm

On the one hand, there is the polygon inflation algorithm (PIA) for $s = 3$ [69, 71]. The SFS can be split into several subsets, depending on the inner and outer polygons and thus on D. For example, Fig. 2.3 shows 3 separate subsets of the SFS. The PIA approximates the SFS subsets with inflating polygons. This is done based on an initial feasible factor S^T and the corresponding points in the SFS subsets, in the V-space. Using these points, polygons are spanned that are contained in the respective subsets of the SFS and these polygons are inflated

until they represent a sufficient approximation of the respective SFS subset. If there is only one connected SFS subset, the inner boundary of $\mathcal{M}_\mathcal{S}$ is approximated with such a polygon, as well as the outer polygon. The intersection of both polygons forms $\mathcal{M}_\mathcal{S}$. To achieve a given accuracy, tolerances regarding the quality of the approximation and the allowed negativity of the factors must be defined (and met).

Ray casting method

Another way to approximate the SFS is to use the ray casting method [72]. It is applicable for any s, but with exponentially increasing computational cost. In the ray casting method, equiangular rays are constructed starting from the origin discretizing the entire space. Then the intersection points of the rays with the subsets of the SFS ($\mathcal{M}_\mathcal{S}$) are determined. This is done using a bisection method. Starting with a point on the ray, first the boundary of $\mathcal{F}_S$ is approximated and it is checked if the corresponding point is feasible. In a next step, again using a bisection method, the inner boundary of $\mathcal{M}_\mathcal{S}$ is determined, i.e., the innermost feasible point on each ray. The result is an approximation of $\mathcal{M}_\mathcal{S}$.[2] Parameters such as the number of rays and nonnegativity tolerances are critical to the quality of the approximation. Comparing the ray casting method to PIA, it becomes clear that it has a "smaller precision-to-effort ratio" [72] when applied to $s = 3$. But the main advantage of the ray casting method is its simplicity and adaptability to arbitrary dimensions. It is described in more detail in chapter 4, where it is used in a modified version.

It is also possible to use PIA or the ray casting method if a data set contains noise and to approximate not only the SFS but also the restricting polytopes $\mathcal{F}_S$ and $\mathcal{I}_S$. This is discussed in the next section.

2.4. Numerical approaches to noisy data sets

The goal is to apply the presented methods to measured data sets, such as (optical) spectroscopic measurements of chemical reactions, which are subject to errors, either due to a limitation of the accuracy of the measuring device or due to other environmental influences. Therefore, a consideration of data sets affected by noise becomes necessary.

The first step is to look at model data, where the feasible solutions are known, and add noise so that the results can be compared. An analysis of experimental data sets is done in chapter 6.

The addition of noise can be realized by different types of perturbations. In this thesis, only homoscedastic noise is considered because it best describes the noise that occurs in the presented applications. Homoscedastic noise means that the noise is independent of the measured values and thus has a constant variance [46]. In contrast, there is heteroscedastic noise, which is related to the measured values and increases as the measured values increase. More details can be found in [11].

Thus, instead of considering a noise-free model data set $\widehat{D} \in \mathbb{R}_+^{k\times n}$, the noisy case considers $D = \widehat{D} + E$, where $E \in \mathbb{R}^{k\times n}$ is the residual matrix containing the noise. For modeled data E contains, e.g., normally distributed noise with mean zero.

This added noise can not only lead to small negative entries in D, but also to D becoming a full-rank matrix, i.e. $\operatorname{rank}(D) = \min(n, k) \gg s = \operatorname{rank}(\hat{D})$. Thus, the NMF problem of D cannot be solved directly with the introduced approaches. For this purpose, the NMF problem 2.3 is

[2] The properties and behaviour of the method are described in [72].

redefined to make the introduced approaches to solve the NMF problem applicable to noisy data sets.

Problem 2.27 (See [65], Prob. 2.5)**.** *Let $D \in \mathbb{R}^{k\times n}$ be given. The entries of D are nonnegative or negative, but small according to the amount, meaning*

$$D(i,j) \gg -\max_{l,\ell} |D(l,\ell)|, \text{ for all } i = 1,\dots,k,\ j = 1,\dots,n.$$

Furthermore, let $s \leq \operatorname{rank}(D)$, $s \in \mathbb{N}$, $\varepsilon_C, \varepsilon_S \geq 0$ be given. A factorization is sought after so that $\|D - CS^T\|_F$ is minimal with factors $C \in \mathbb{R}^{k\times s}$ and $S^T \in \mathbb{R}^{s\times n}$ with

$$\frac{C(i,h)}{\max|C(:,h)|} \geq -\varepsilon_C \quad \text{and} \quad \frac{S(j,h)}{\max|S(:,h)|} \geq -\varepsilon_S,$$

for $i = 1,\dots,k$, $j = 1,\dots,n$, for all $h = 1,\dots,s$.

There are slightly different approaches to the control parameter ε, depending on the method chosen to determine the factorization, see [70]. All approaches regulate the tolerance on how much negative entries in C and S^T are allowed to approximate D. These tolerances on the negative entries in C and S^T have several reasons. On the one hand, they are based on negative entries in D for example due to a baseline correction or background subtraction (see [70]). It is also possible that the best approximation of D, based on $\|D - CS^T\|_F$ from problem 2.27, contains negative entries, so there exists no corresponding feasible pair without negative entries.

How the product CS^T can be chosen to be a best possible approximation of D was described by Eckart and Young in 1936.

Theorem 2.28 (Eckart-Young Theorem [19], here from [28])**.** *Let $D \in \mathbb{R}^{k\times n}$ be given with its SVD $D = U\Sigma V^T$. The singular vectors are denoted by $u_i = U(:,i)$ and $v_i = V(:,i)$ and σ_i denotes the i-th singular value. If $r < \operatorname{rank}(D)$ and $D_r = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ with $p = \min\{k,n\}$ holds*

$$\min_{\operatorname{rank}(B)=r} \|D - B\|_2 = \|D - D_r\|_2 = \sigma_{r+1}$$

$$\min_{\operatorname{rank}(B)=r} \|D - B\|_F = \|D - D_r\|_F = \sqrt{\sigma_{r+1}^2 \cdots + \sigma_p^2}$$

Thus, the best rank-r approximation is obtained by using a truncated SVD. That is, $CS^T = \sum_{i=1}^{s} \sigma_i u_i v_i^T = D_s$ is the best rank-s approximation of D from problem 2.27. This consideration allows using the SVD-based approach of Sec. 2.3 and applying it to the matrix D_s to determine C and S^T with the negativity conditions of problem 2.27.

All of these considerations are based on a proper choice of s with which to approximate D. At the beginning of this chapter, s was defined as the rank of D (see Prob. 2.3). Since in the noisy case D is a full-rank matrix, a different definition of the rank is needed. Since the focus of the application is in chemistry, the definition provides a link to chemical systems.

Connections between the rank of a matrix and the underlying chemical system were made in 1960 by Wallace [85]. There, the rank of the data matrix corresponds to the number of chemical species in a system. The idea is to subtract (sufficiently good) rank-1 matrices until only values below a certain level of noise remain, where the number of rank-1 matrices defines the chemical rank.

In a more mathematical way it was defined in [52], where the number of relevant singular values above a certain tolerance ε_{chem} is assumed to be the chemical rank. This definition is justified by Thm. 2.28, since the approximation error is directly related to the singular values. The estimation of ε_{chem} and other approaches to the chemical rank are examined next.

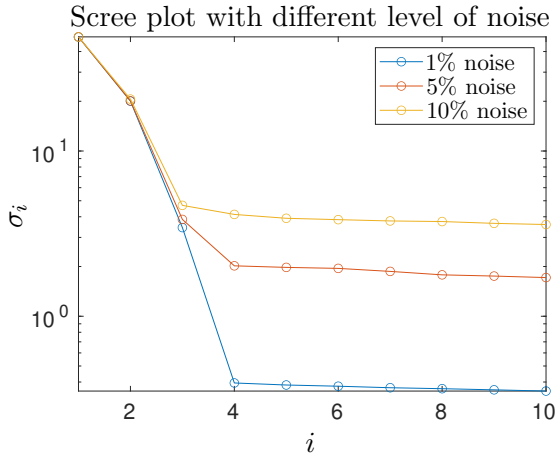

Figure 2.8.: Scree plot of data set 2 with different levels of added noise. For a small level of added noise there is a clear bend associated with a chemical rank of three, see blue curve. At higher noise levels the bend is less clear (red curve) or even a wrong rank selection is possible (yellow curve, the bend indicates a chemical rank of two).

2.4.1. How to determine the chemical rank for noisy data sets

In [46] Malinowski gives a good overview of possible approaches to determining the chemical rank. Some of them are presented in the following. In addition, further investigations are made in chapter 5 in the context of subsystem detection and analysis.

The goal of the first approach is to find a chemical rank that explains a certain amount of variance. Since the variance of a data set can be calculated based on the eigenvalues of the data matrix [46], a closer look is taken at the eigenvalues of DD^T and thus at the singular values of D. The eigenvalues of DD^T are the squared singular values of D. It is possible to determine the variance and thus the importance of each eigenvector or singular vector from the corresponding eigenvalue or singular value. Thus, only the first s singular vectors, which describe the majority of the variance of the data set, are considered relevant and are used to approximate the data set. The remaining singular values and vectors are assumed to belong to noise. However, this technique is subject to personal bias, depending on how much variance should be explained by the chosen approximation.

It is also possible to look at the logarithm of the singular values of the data set, plotted against the number of the respective singular value. This is shown in Fig. 2.8 for data set 2 with different amounts of noise. The corresponding curve should drop rapidly at a certain point and then level off. The number of singular values before the bend represents the number of species [75]. Thm. 2.28 supports this approach, because if the remaining singular values are all significantly small, the residuum of the approximation will be correspondingly small.

Both approaches also allow to estimate ε_{chem}, depending on the last selected singular value. This value can be selected as ε_{chem} with $\sigma_s \geq \varepsilon_{chem} \geq \sigma_{s+1}$. Assuming s, as the chemical rank is properly selected, the next step is to look back at the cone and the SVD-based approach and see how they are affected by noise.

2.4.2. The effect of noise on the approaches of Sec. 2.2 and 2.3

Although the factors $C \in \mathbb{R}^{k\times s}$ and $S^T \in \mathbb{R}^{s\times n}$ can be represented by the truncated SVD of a data set $D \in \mathbb{R}^{k\times n}$, the SVD-based approach of Sec. 2.3 is not yet directly applicable. The reason for this is that the rank-s approximation D_s of D can contain negative entries, which

means that the outer polytope no longer encloses the inner polytope, or in other words, that there are no nonnegative matrices C and S^T for which $CS^T = D_s$ holds.

The same is problematic for the cone-based approach of Sec. 2.2, i.e. if D_s contains negative entries, then the vectors spanning the row and column spaces are no longer within the positive orthant. A solution to this is to reformulate the problem 2.27 so that both factors C and S^T are necessarily nonnegative, resulting in a constrained best approximation of D. This ensures that there exists a nonnegative factorization and that the approximation of D is nonnegative (eliminating both constraints described above).

Nevertheless, it is still possible to use the SVD as a best approximation, taking into account the tolerances of problem 2.27, i.e. the nonnegativity constraints of problem 2.3 are relaxed. As a result, the outer polytope expands and the inner polytope is compressed. The associated SFS becomes larger or even starts to exist. This reasoning can be applied to both PIA and the ray casting method of Sec. 2.3.8 by adjusting the tolerances with respect to nonnegativity. The effect of different tolerances is shown in [69] Fig. 9 for PIA, but the results are identical for the ray casting method.

As the outer polytope $\mathcal{F}_S$ becomes larger, it is no longer dual to the convex hull of the data points in the dual space, i.e., the inner polytope $\mathcal{I}_C$ of the noise-free case, see Eq. (2.11). Correspondingly, the inner polytopes have to be re-determined based on the outer polytopes using duality. This also means that the essential rows and columns of D can no longer be determined from the data points, which are no longer the vertices of the inner polytopes. A way to determine essential information for noisy data sets is investigated next.

2.4.3. Determining essential information for noisy data sets

Since the inner polytopes are no longer the convex hull of the data points, the data points can no longer be used to identify the essential information, as was done in Def. 2.23. Thus, the relation to the facets of the outer polytopes is used to identify the essential information. A facet of $\mathcal{F}_S$ is dual to a vertex of $\mathcal{I}_C$. The same holds for $\mathcal{F}_C$ and $\mathcal{I}_S$. The following considerations are reduced to the use of $\mathcal{F}_S$, but can be applied analogously to $\mathcal{F}_C$. Thus, Def. 2.23 can be rewritten as:

The ℓ-th column of $D \in \mathbb{R}_+^{k\times n}$, with the SVD $D = U\Sigma V^T$, is essential if there exists a $y \in \mathbb{R}^{s-1}$, so that $V(\ell,:)(1,y^T)^T = 0$ and for all other rows $j = 1,\dots,n$, $j \neq \ell$, $V(j,:)(1,y^T)^T \geq 0$. If several indices ℓ satisfy the inequality for a y sharply, only the first of these ℓ is considered essential.

Thus, the ℓ-th column of D corresponds to a facet of $\mathcal{F}_S$. In the case of noise, small negative entries are allowed, as described in problem 2.27. These negative entries of the factor S^T must therefore satisfy

$$\frac{\min\limits_{j=1,\dots,n}(V(j,:)(1,y^T)^T)}{\max\limits_{j=1,\dots,n}|V(j,:)(1,y^T)^T|} \geq -\varepsilon_S. \tag{2.15}$$

This allows to find the boundary of the outer polytope depending on the tolerance ε_S. The approximation of the outer polytope with respect to the noise is now denoted by $\mathcal{F}_S$. The points that span the outer polytope are stored in the matrix $F \in \mathbb{R}^{N\times(s-1)}$, where N is the number of sampling boundary points. The number of boundary points depends on the method used. It is possible to specify this number, e.g., in the ray casting method by specifying the number of rays. These boundary points are used to determine the profiles that represent the boundary. Thus, the points are represented by $y_j = F(j,:)$, $j = 1,\dots,N$, which results in the profiles $p_j^T = (1, y_j^T)V^T$ for $j = 1,\dots,N$.

Eq. (2.15) can be rewritten for the profiles, more precisely for the relation between essential columns and the profiles

$$\frac{p_j(\ell)}{\max|p_j|} \geq -\varepsilon_S, \quad \text{for all } \ell = 1, \ldots, N.$$

Thus, the ℓ-th entry of the j-th profile is essential, and thus the ℓ-th column of D, if

$$\frac{p_j(\ell)}{\max|p_j|} = -\varepsilon_S. \tag{2.16}$$

This equality is satisfied for at least one $\ell = 1, \ldots, N$, because y_j is on the boundary of $\mathcal{F}_S$.

If the boundary is approximated numerically, an additional control parameter $\delta > 0$ is needed to determine the essentiality, since Eq. (2.16) holds only in an approximated manner. Thus, Eq. (2.16) is rewritten as

$$-\varepsilon_S - \delta \leq \frac{p_j(\ell)}{\max|p_j|} \leq -\varepsilon_S + \delta. \tag{2.17}$$

Since only the boundary points of $\mathcal{F}_S$ are used to determine p_j, only the right inequality needs to be considered. Smaller negative entries in the profiles are not possible, since this would mean that the data representing point of the profile is outside of $\mathcal{F}_S$, see Eq. (2.15).

The choice of the additional control parameter δ can be made in several ways. It is possible to select δ in dependence of ε_S and thus as a fixed constant. Another approach is to select the parameter depending on the data. In this way, not only the index of the minimum of each profile is essential, but also the indices belonging to entries that have values close to the minimum are classified as essential. This is of interest because the profiles are noisy and it is not that easy to distinguish between minima of profiles that are close to each other and to determine "the right profile".

Possible choices are either $\delta = 0.01|\frac{\min(p_j)}{\max(p_j)}|$ or a dependence on the difference between $-\varepsilon_S$ and the reached minimum $\frac{\min(p_j)}{\max(p_j)}$, i.e. $\delta = 1.01|-\varepsilon_S - \frac{\min(p_j)}{\max(p_j)}|$. It is advisable to choose a data-dependent approach, but as with most tolerances, the choice depends on the data given.[3]

In addition to all $\ell = 1, \ldots, N$ that satisfy Eq. (2.17), the entry where the maximum of p_j is reached is also considered as essential. This is because if it is not essential, this entry could be neglected. However, this results in a different denominator in Eq. (2.17) and thus can lead to changed essential information (e.g., an essential entry becomes non-essential). Even if it would not result in any changes, it is better to have a slight superset of the essential information than to have too few indices included. Especially when dealing with noisy data sets, it is not so black and white what is essential and what is not, especially when considering different tolerances. Therefore, it is better to have a superset of the essential information. Depending on the noise, it may even be beneficial to include non-essential parts of the data matrix, as this may stabilize the subsequent analysis by reducing the influence of noise [73].

A comparison of essentiality for data set 2 with and without noise is shown in Fig. 2.9.

The determination of the essential information can be realized by the ray casting method from Sec. 2.3.8. During the step where $\mathcal{F}_S$ is approximated, the essential information for the corresponding ray can also be determined. Thus, determining the essential information does not drastically increase the computational cost of using the ray casting method.

[3] The choice suggested in [73] is similar, but with a missing epsilon.

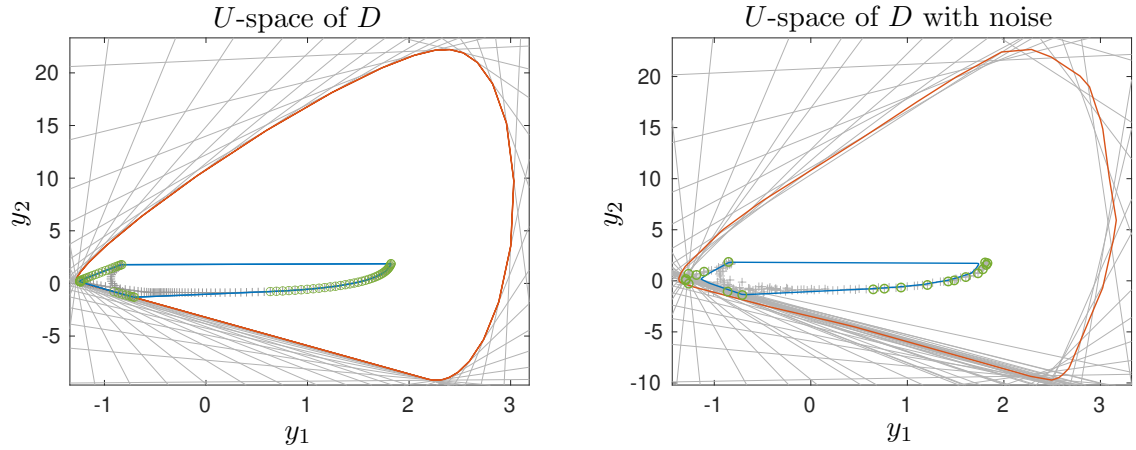

Figure 2.9.: Low-dimensional representation of data set 2 in the U-space with marked essential data points. Left: Model data set without noise. The data points of the essential columns (green) correspond to the vertices of $\mathcal{I}_C$ (blue). Right: Model data set with 1% homoscedastic noise added. The polytopes and essential information are computed using the ray casting method with standard tolerances. Both $\mathcal{I}_C$ and $\mathcal{F}_C$ are changed due to noise. The data points of the essential columns for the noisy case (green) no longer correspond to the vertices of $\mathcal{I}_C$. Also, $\mathcal{F}_C$ (red) is no longer bounded by the affine hyperplanes (gray lines), which are dual to the data points in the V-space.

2.5. Critical summary of the approaches to solve the NMF problem

The mathematical techniques presented in this chapter allow not only to find solutions to the NMF problem 2.3 of a given data matrix, but also to determine its underlying factor ambiguity. Both techniques presented are geometric approaches based on certain convex cones (directly based on the data matrix) and polytopes (based on an SVD of the data matrix). They allow a deeper understanding of crucial parts of the data, since only certain edges of the cones or vertices of the polytopes represent active constraints in the process of determining solutions to the NMF problem and the factor ambiguity, which is the SFS (Def. 2.15). These crucial parts led to the concept of essentiality (Def. 2.6 and 2.23), which allows data reduction.

The geometric approaches are strong for strictly nonnegative data matrices with a small rank number, but they fail when the data matrix contains small negative entries or when its rank number is increased due to noise, as in Sec. 2.4. Then numerical techniques based on SVD are very useful. For example, for small negative entries in the data matrix, the active constraints of the SVD-based approach can be relaxed to allow small negative entries in both factors. In the case of noisy data, low-rank approximations based on a truncated SVD (Thm. 2.28) can be used to allow a determination of the SFS.

All these tools are building blocks for the development of NMF solvers and factor ambiguity analysis methods for more complex incomplete data sets.

3. The shared low-dimensional representation

When faced with unknown or unreliable parts in a data matrix, as visualized in Fig. 3.2, the question arises how to analyze it. Is it possible to modify the approaches of chapter 2 and apply them to analyze such an incomplete data set? Or is it possible to reduce such an affected data matrix so that the cone or the SVD-based approach can be used directly?

These questions are the main topic of this chapter and are analyzed for both the noise-free and noisy cases. The questions can be reformulated to find a low-dimensional representation of an incomplete data matrix to analyze the data set. The goal is that this low-dimensional representation is shared by several complete submatrices of the incomplete data matrix and represents all given information. Hence, the name *shared low-dimensional representation* arises. A general flowchart of the approach proposed in this chapter for analyzing incomplete data sets is shown in Fig. 3.1.

The main content of this chapter has already been published in [8], but will be discussed in more detail here.

3.1. Motivation and background

Why might it be interesting to look at a data matrix that contains unknowns? Looking at the application of the NMF problem in chemometrics, data sets with unknown or missing parts, so-called incomplete data sets, appear in a variety of cases.

Such appearances can be divided into two classes. They are motivated by the application background. On the one hand, the data matrix may be affected by an instrument malfunction during the measurement, e.g., due to saturation [7, 31]. This case is simulated for data set 1 in Fig. 3.2. It may also be that it is simply not possible to measure certain things, e.g., when having trilinear data from excitation emission measurements [29]. On the other hand, it is possible that such incompleteness is created by design. This can be the case when the cost of measurement is too high for a complete measurement, thus creating incompleteness. Or when doing online process monitoring where future time steps are unknown [7]. Image fusion [56] or multiblock data fusion in general [76] are also use cases. Here, measured data matrices are combined into one to be analyzed, but depending on the axes, this fusion can create blockwise incompleteness in the data matrix.

With all these occurrences of incomplete data matrices, the question of how to handle them arises again. As stated in [57]: "The most radical – and common – solution is to delete as many variables and/or objects from the data as necessary to reach completeness". Thus, the incomplete data matrix is reduced until all unknown entries are eliminated. However, this typically results in a loss of information, especially if the major part of the matrix is unknown. Therefore, there are other approaches, such as NIPALS, an algorithm based on partial least squares that can be adapted to account for missing values [49, 87]. It is also possible to use so-called data imputation, a concept from statistics, where missing values are estimated, e.g., based on known neighboring values, thus creating a complete matrix that can be analyzed [7, 57, 83, 84].

However, these methods are limited when large blocks are missing or when special patterns of

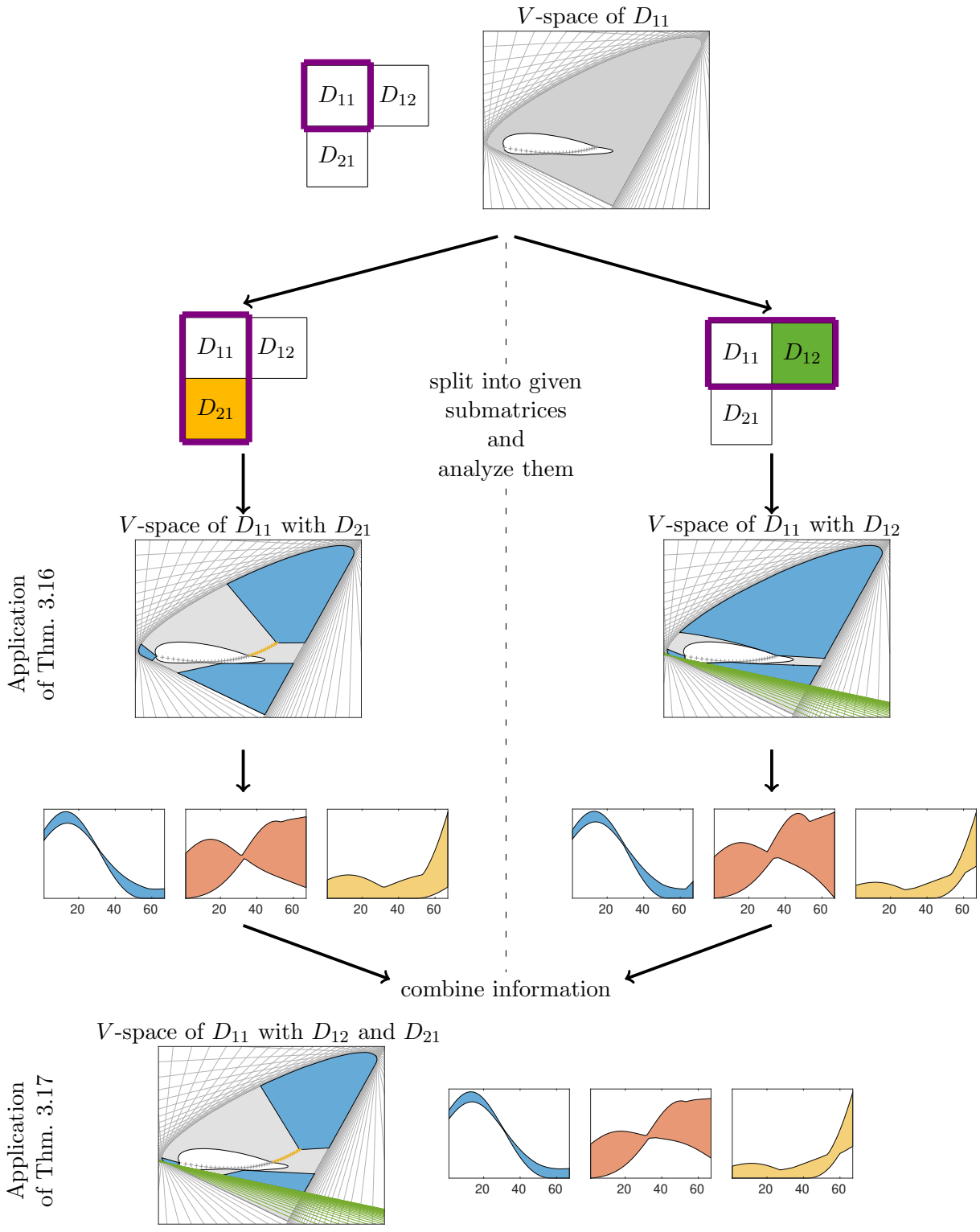

Figure 3.1.: Schematic overview of the approach for case 1 where $\operatorname{rank}(D_{11}) = \operatorname{rank}(D_{11}, D_{12}) = \operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ using the V-space representation of Sec. 3.4 for data set 1 from Fig. 3.2. The saturated block is shown as blank space. The goal is to represent all given information in the same V-space, which is achieved by splitting the incomplete data matrix into submatrices and adding their information to the representation of the shared block D_{11} (see Def. 3.1). These steps are considered in detail in Sec. 3.3 and 3.4, leading to Thm. 3.16 and 3.17, which make this construction possible.

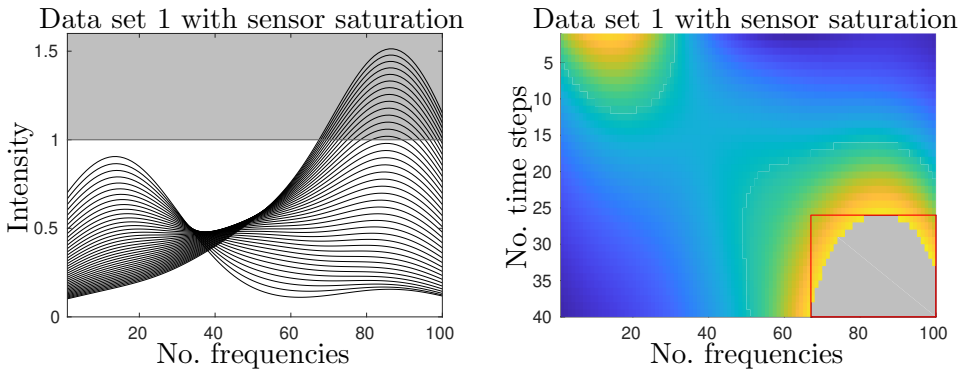

Figure 3.2.: Simulated sensor saturation for data set 1, where everything above an intensity of 1 is saturated and thus assumed to be missing. The saturated part is marked in gray (right). The submatrix marked by the red box (right) is the smallest submatrix containing all unknown values and thus corresponds to the block D_{22} in Eq. (3.1).

missingness occur. Such patterns can occur, e.g., when faced with sensor saturation where large blocks are missing instead of sparsely and randomly distributed missing values. Therefore, the missingness is not statistically based.

Even if data imputation in such a case would lead to a proper complete data matrix, it is not guaranteed that the result can be factorized as stated in the NMF problem 2.3 for a given s, even for model data. In the context of blockwise missing data, this has been addressed by Beyad and Maeder [9] in 2013. Their approach is based on linear regression using a given nonnegative factorization of a submatrix. If such knowledge of a solution to the NMF problem is not available, the approach of Alier and Tauler [3] can be used. Here the method MCR-ALS [77] is modified to find a single solution that represents the incomplete data set [3]. The result is obtained by an iteration in which both factors C and S^T are alternatingly optimized. The only restriction is that, in a chemical context, all underlying chemical species must appear in all given blocks of the matrix. This is summarized in the following as case 1. This restriction is explained in detail in Sec. 3.2. Apart from this limitation, the method is already in use, e.g., in image fusion, and has been tested with different noise levels, see [16, 56]. Recently, this approach has been further developed in [30] and applied in [58] to cover cases where the underlying chemical species do not appear in all blocks.

However, this approach only accesses one possible solution and does not capture the entire ambiguity. So is there a way to access the entire ambiguity of an incomplete data set? To do this, the specific problem must be formulated.

3.2. Problem description

To describe the general procedure of the approach used, only one block in the data matrix is assumed to be missing. This is shown in Fig. 3.2, where the submatrix in the red rectangle is assumed to be missing. If the missing block is not at the bottom right of the matrix, then a permutation of the rows and columns can be done to bring the matrix into this L-shaped structure. If more than one block is assumed to be missing (or if the missing entries are not blockwise missing as shown in Fig. 3.2), either one large block is used that contains all missing values, or the data set can be analyzed in an iterative way. The latter is described in Sec. 3.6.

Assuming $D \in \mathbb{R}^{k \times n}$ is a nonnegative matrix and can be represented as a block matrix consisting of 2×2 blocks as in Eq. (3.1), where the block D_{22} contains the unknowns (the missing entries).

If D has a nonnegative factorization $D = CS^T$, then the relations between the single blocks are as follows

$$D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} = CS^T = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}^T = \begin{pmatrix} C_1 S_1^T & C_1 S_2^T \\ C_2 S_1^T & C_2 S_2^T \end{pmatrix}. \tag{3.1}$$

For the individual blocks holds $D_{11} \in \mathbb{R}_+^{k_1 \times n_1}$, $D_{12} \in \mathbb{R}_+^{k_1 \times n_2}$, $D_{21} \in \mathbb{R}_+^{k_2 \times n_1}$ and $D_{22} \in \mathbb{R}_+^{k_2 \times n_2}$ with $k = k_1 + k_2$ and $n = n_1 + n_2$. These blocks are coupled by the factors $C \in \mathbb{R}_+^{k \times s}$ and $S^T \in \mathbb{R}_+^{s \times n}$, where $s = \operatorname{rank}(D) = \operatorname{rank}_+(D)$. In particular, D_{11} is connected to D_{12} through C_1 and to D_{21} through S_1^T.

The largest complete submatrix is (D_{11}, D_{12}) or $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ and they can be analyzed separately. However, the results of both submatrices are connected by C_1 and S_1^T which correspond to the block D_{11} which appears in both submatrices and is therefore shared. Because of this special role of D_{11} and its use in the following construction, it is defined next.

Definition 3.1. *The block D_{11} of Eq. (3.1) is called* shared block*, if there exist submatrices (D_{11}, D_{12}) and $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ of D that are complete and do not contain unknown entries.*

Alternatively, another structure of D is considered. It contains only 2×1 blocks. This is a special case of Eq. (3.1).

$$D = \begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} S^T \tag{3.2}$$

Let $D_{11} \in \mathbb{R}_+^{k_1 \times n}$ and $D_{21} \in \mathbb{R}_+^{k_2 \times n}$. Here no missing block appears, but a shared block can still be selected, because both matrices are connected by S^T. In this case, D_{11} is also selected as shared block. This selection is also motivated by the fact that Eq. (3.2) is used as an intermediate step to analyze Eq. (3.1).

This case has already been studied in [4] with the corresponding SFS and low-dimensional representation. In [4], it is considered as a complete data set and thus the SVD of D is used and analyzed as described in Sec. 2.3. In this thesis, however, the focus is shifted to the submatrix D_{11} with additional information contained in D_{12}. Thus, the analysis focuses on D_{11}.

This block or submatrix must have one important property. It must not be locally rank deficient. This term was introduced in [18] and [2] and is refined for the problem at hand.

Definition 3.2 (Local rank deficiency)**.** *Let $D \in \mathbb{R}_+^{k \times n}$ be given as in Eq. (3.1) with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$. Let $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) = s_1 < s$ hold for the submatrix $D_{11} \in \mathbb{R}^{k_1 \times n_1}$, thus $D_{11} = \hat{C}_1 \hat{S}_1^T$, $\hat{C}_1 \in \mathbb{R}_+^{k_1 \times s_1}$, $\hat{S}_1 \in \mathbb{R}_+^{n_1 \times s_1}$. Then D_{11} is* local rank-deficient *if there exists no C_1 belonging to D such that $\hat{C}_1 = C_1(:, 1 : s_1)$. Or alternatively, there exists no solution to the NMF problem of D such that either $C_1(:, s_1 + 1 : s) = 0$ or $S_1(:, s_1 + 1 : s) = 0$ is satisfied.*

In other words, D_{11} should have no rank deficiency or hidden rank deficiency [5] that can be resolved by adding rows and/or columns. This matrix augmentation results in D. An example of possible choices of D_{11} is shown in Fig. 3.3, which illustrates the difference between a proper choice (in purple) and one with a local rank deficiency (in yellow).

Given the matrix structure of Eq. (3.1) where D_{22} denotes the incomplete block, the NMF problem 2.3 can no longer be applied directly. Therefore, the problem is rephrased for incomplete data matrices.

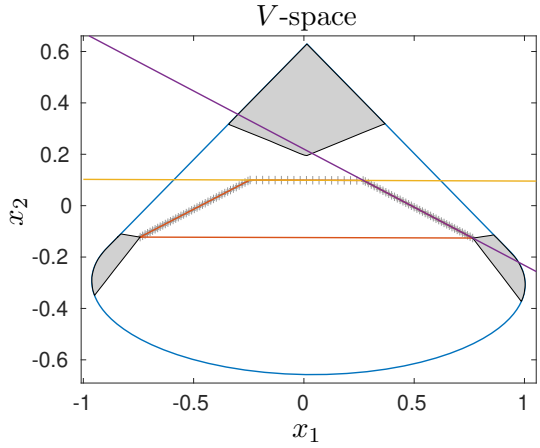

Figure 3.3.: Visualization of local rank deficiency. Two affine subspaces are shown in yellow and purple, spanned by the submatrices corresponding to the data points on the corresponding line. The SFS segments are shown in gray. Using Def. 3.2 for both submatrices and their corresponding data points, i.e., their affine subspaces, results in the yellow line corresponding to local rank deficiency. The purple line does not correspond to local rank deficiency because it intersects with two SFS segments.

Problem 3.3 (NMF problem for incomplete data)**.** *Let $D \in \mathbb{R}_+^{k\times n}$ be a matrix with the structure of Eq.* (3.1) *and* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*. Further, let D_{11} have no local rank deficiency. A nonnegative factorization is sought, where $D_{11} = C_1 S_1^T$, $D_{12} = C_1 S_2^T$ and $D_{21} = C_2 S_1^T$, with the factors $\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = C \in \mathbb{R}_+^{k\times s}$ and $(S_1^T, S_2^T) = S^T \in \mathbb{R}_+^{s\times n}$.*

Since the blocks of Eq. (3.1) and (3.2) play an important role in problem 3.3, they are examined in more detail. As can be seen in Eq. (3.1) and (3.2), the different blocks are connected by C and S^T. This connection can be expressed by using the row and column spaces of D.

Looking at Eq. (3.2). If $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) = s_1 \leq s = \operatorname{rank}(D) = \operatorname{rank}_+(D)$ and D_{11} has no local rank deficiency, then D_{11} can be represented by s_1 elements of a basis of the row space of D. The basis of D can be chosen such that the basis elements correspond to columns of a feasible S and thus to profiles.

The same connections can be made for Eq. (3.1), working with reduced spaces. This complicates the connection of which basis elements of D are used to represent which block. To make this concept more tangible, the term *(chemical) species* is used instead of basis elements of the corresponding spaces. A chemical species is represented by the i-th columns of C and S, which correspond to the i-th profiles, and thus by $C(:,i)(S(:,i))^T$. Therefore, the following representation of D is used to express the contribution of each species

$$D = \sum_{i=1}^{s} C(:,i)(S(:,i))^T.$$

From this equation it can be derived that the number of underlying chemical species of a data matrix is an upper bound of the rank and nonnegative rank.

The concept of species also allows to express the connection of the blocks of Eq. (3.1).

Definition 3.4 (Shared species)**.** *If a species appears in multiple blocks, it is called shared.*

For example, if $C(1:k_1,i) \neq 0$, $C(k_1+1:k_2,i) \neq 0$ and $S(1:n_1,i) \neq 0$, the i-th species is shared between D_{11} and D_{21}.

Looking at the particular block D_{22} from Eq. (3.1), it has a special role, especially with respect to the underlying species. This is because it is assumed that D_{22} contains missing entries and causes the incompleteness of D. Thus, the knowledge about which species appear in D_{22} is limited. In particular, if a species appears only in D_{22}, i.e., it exists a $j = 1, \ldots, s$ with $C(1 : k_1, j) = 0$ and $S(1 : n_1, j) = 0$, it cannot be analyzed (and not even detected) by looking only at the given blocks D_{11}, D_{12} and D_{21}. Thus, the general assumption of this chapter is that no species appears only in D_{22}.

Returning to the posed Prob. 3.3, the goal of this chapter is to find all nonnegative factorizations for each D with the structure of Eq. (3.1), as was done in Sec. 2.3 for the complete case, but under the assumption that D_{22} is unknown. Thus, a corresponding low-dimensional representation is sought in which all solutions of Prob. 3.3 are represented. To this end, cones, as introduced in Sec. 2.2, are used as a tool for this construction. The insights gained are then transferred to the SVD-based approach.

Prior to the analysis, a case differentiation must be made. Assuming that the given data set has the structure as in Eq. (3.1).

Case 1. *For a matrix with the structure in Eq.* (3.1) *holds* $\operatorname{rank}(D_{11}) = \operatorname{rank}(D_{11}, D_{12})$ *and* $\operatorname{rank}(D_{11}) = \operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ *or, if D is a complete matrix,* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = \operatorname{rank}(D_{11})$.

This means that all underlying species that are present in D also appear in D_{11}. The opposite, that not all species of D appear in D_{11}, is described next.

Case 2. *For a matrix with the structure in Eq.* (3.1) *holds* $\operatorname{rank}(D_{11}, D_{12}) > \operatorname{rank}(D_{11})$ *or* $\operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix} > \operatorname{rank}(D_{11})$ *or, if D is a complete matrix,* $\operatorname{rank}(D) = \operatorname{rank}_+(D) > \operatorname{rank}(D_{11})$.

This case differentiation can also be applied to Eq. (3.2) by considering only the submatrices D_{11} and D_{21}.

The general idea and procedure for handling case 1 is summarized in Fig. 3.1. The details are described in Sec. 3.3 and 3.4. The analysis of case 2 is done in Sec. 3.5, where the goal is to reduce case 2 to case 1. As in the previous chapter, the noisy case is also considered, see Sec. 3.8, as well as a critical evaluation of whether and when it makes sense to consider a data set as incomplete, see Sec. 3.7.

3.3. The shared representation using cones

In this section the representation of the cones is used as in Eq. (2.7), i.e., it is based only on the edges of the cones and thus on the essential information of the data matrix. The goal is to solve the problem 3.3 for case 1 and thus to find a representation that contains all information of the known blocks D_{11}, D_{12} and D_{21}. Therefore, a matrix, in this case D_{11}, is analyzed, which is row- and column-wise augmented by D_{21} and D_{12}. For this purpose, the first matrices to be considered are those that are augmented in one direction only, either by rows or columns. Parts of this section have already been published in [8].

3.3.1. Analyzing row- or column-wise augmented matrices

In order to look at augmented matrices, some considerations about essential information have to be made beforehand. Sec. 2.2 concluded with the introduction of essential information in

the context of the cone representation. It is possible to project the determination of essential information of D onto the factors C and S^T. For the following consideration it must be kept in mind that in order to verify that C is feasible for D, $\text{colcone}(D) \subseteq \text{colcone}(C)$ must hold.

Lemma 3.5 (From [8]). *Let $D = CS^T$ be a nonnegative matrix factorization of $D \in \mathbb{R}_+^{k\times n}$, with $\text{rank}(D) = \text{rank}_+(D) = s$ and $C \in \mathbb{R}^{k\times s}$ and $S^T \in \mathbb{R}^{s\times n}$. Then the ℓ-th column of D is an edge of $\text{colcone}(D)$ if the ℓ-th column of S^T is an edge of $\text{colcone}(S^T)$ and vice versa.*

Proof. The complementary case is shown where a certain column is not an edge. Thus, holds:

$$
\begin{aligned}
& D(:,\ell) = De_\ell \text{ is not an edge of colcone}(D), \text{ where } e_\ell \text{ is the } \ell\text{-th unit vector} \\
\Leftrightarrow \quad & D(:,\ell) = (CS^T)e_\ell = \sum_{i\in\tilde{\mathcal{E}}} \alpha_i CS^T e_i,\ \alpha_i \geq 0 \text{ (i.e., a conic combination exists)} \\
\Leftrightarrow \quad & C\left(S^T e_\ell - \sum_{i\in\tilde{\mathcal{E}}} \alpha_i S^T e_i\right) = 0,\ \alpha_i \geq 0 \\
\Leftrightarrow \quad & S^T e_\ell - \sum_{i\in\tilde{\mathcal{E}}} \alpha_i S^T e_i = 0,\ \alpha_i \geq 0 \text{ (because the columns of } C \text{ are linearly independent)} \\
\Leftrightarrow \quad & S^T(:,\ell) \text{ is a conic combination of the edges of colcone}(S^T) \\
\Leftrightarrow \quad & S^T(:,\ell) \text{ is not an edge of colcone}(S^T).
\end{aligned}
$$

□

The essence of this lemma is similar to Lem. 2.7 in [68], but with a different approach of the proof. Both imply that the essential information of D is also contained in the factors C and S^T. Thus, one solution is sufficient to determine the essential information, which allows a determination in a lower dimension than with D. Lem. 3.5 also allows to consider and compare the essential information of C and S^T instead of looking at D. This property is used in the following lemma when considering augmented matrices.

Lemma 3.6 (From [8]). *Let $D = CS^T$ be a nonnegative matrix factorization of $D \in \mathbb{R}_+^{k\times n}$ with $\text{rank}(D) = \text{rank}_+(D) = s$ and $C \in \mathbb{R}_+^{k\times s}$, $S^T \in \mathbb{R}_+^{s\times n}$. Adding a $(k+1)$-th row leads to a matrix $\tilde{D} \in \mathbb{R}_+^{(k+1)\times n}$. It is assumed that $\text{rank}(\tilde{D}) = \text{rank}_+(\tilde{D}) = s$, which is equivalent to the existence of a nonnegative matrix factorization $\tilde{D} = \tilde{C}S^T$ with $\tilde{C} \in \mathbb{R}^{(k+1)\times s}$. Then the index sets of the edges of $\text{colcone}(D)$ and $\text{colcone}(\tilde{D})$ are the same.*

Proof. This follows from Lem. 3.5, since $D = CS^T$ and $\tilde{D} = \tilde{C}S^T$ are based on the same S^T, and thus the index sets of the edges of $\text{colcone}(D)$ and $\text{colcone}(\tilde{D})$ are the same. □

Thus, the essential columns are the same when a new row is added to D under the rank condition. The case of an unfulfilled rank condition was shown in example 2.9. Lem. 3.6 can also be used when adding more than one row or whole blocks as in Eq. (3.2). However, the essential rows may change. This was addressed in Lem. 2.8. It is now possible to extend this lemma to essential information in general.

Remark 3.7. *If the conditions of Lem. 3.6 are fulfilled, then Lemma 2.8 holds for all essential information (esi) instead of $\mathcal{E}$, because the essential columns are the same due to Lem. 3.6.*

Instead of comparing only the index sets of the essential information and thus the set of edges of the cones, it is possible to look directly at the cones and compare what the effect is, e.g., on the cone of the column space when a row is added.

Lemma 3.8 (Based on [8]). *Let $D(1:\ell,:)$ be a submatrix of $D \in \mathbb{R}_+^{k\times n}$ with $\ell \leq k$, $\operatorname{rank}(D) = \operatorname{rank}_+(D) = \operatorname{rank}(D(1:\ell,:)) = s$. Then*

$$(\operatorname{colcone}(D))(1:\ell) = \operatorname{colcone}(D(1:\ell,:)),$$

where $(\operatorname{colcone}(D))(1:\ell)$ denotes the first ℓ dimensions of $\operatorname{colcone}(D)$.

Proof. Because of Lem. 3.6 the matrix D and the submatrix $D(1:\ell,:)$ have the same index set of edges $\tilde{\mathcal{E}}$ of the column cone. Thus, the same columns define the edges of the corresponding cone. Therefore, reducing the dimension of $\operatorname{colcone}(D)$ to $(\operatorname{colcone}(D))(1:\ell)$ leads to the same edges as for $\operatorname{colcone}(D(1:\ell,:))$. Since the edges of both cones are the same, both cones must coincide. □

These relations form the basis of the following section, where a closer look is taken at how the solutions are affected when D is augmented not only row- or column-wise, but in both directions simultaneously.

3.3.2. Construction of a shared representation

The goal is to include augmented rows and columns in the representation of the original row and column spaces of a data matrix. Therefore, some preliminary considerations are made.

If an additional constraint is added to the row space cone to be satisfied (e.g., an additional vector to be enclosed by $\operatorname{rowcone}(S^T)$), a new set of solutions is obtained. This set is a subset of the original set of solutions. However, it is only a constraint for the cone in the row space and not for the cone in the column space, since it is not projected into the column space. Thus, a C can be found whose cone encloses the column cone, but the cone of the dual S^T does not enclose the row cone and the additional constraint. Therefore, the set of feasible C is also influenced by the additional constraint in the row space. This is illustrated in the following example.

Example 3.9. *A rank 3 matrix D with a nonnegative rank 3 is factorized into $D = CS^T$, where C is the identity matrix*

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 2 & 1 \end{pmatrix}. \tag{3.3}$$

Assuming an additional constraint is added to the row cone. This constraint is represented by an additional row of D, resulting in $\tilde{D}$. Such a constraint or row is represented by the red row in $\tilde{D}$ in the following equation. The nonnegative rank is unchanged by the new row, so $\operatorname{rank}_+(D) = \operatorname{rank}_+(\tilde{D}) = 3$.

$$\tilde{D} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 2 & 1 \end{pmatrix} = \tilde{C}S^T \tag{3.4}$$

Because of Lem. 3.8 it is possible to consider $\operatorname{colcone}(D)$ instead of $\operatorname{colcone}(\tilde{D})$ (for the first 3 dimensions). Thus, when finding a solution based only on the column cone, it is possible to use the same C for D and $\tilde{D}$, where $C = \tilde{C}(:, 1:3)$.

However, the choice of C and thus $\tilde{C}$ from Eq. (3.4) results in a corresponding S^T which is unchanged compared to Eq. (3.3). Thus, the new row vector is not included in the row cone

spanned by S^T, resulting in negative entries in the last row of $\tilde{C}$. This means that the pair $\tilde{C}$ and S^T does not qualify as a feasible pair of $\tilde{D}$.

Therefore, when using the representation of the reduced column cone to compute solutions, it is important to verify that the dual factor S^T encloses the row cone. It is no longer sufficient to consider only one space.

The message of this example is summarized in the following theorem.

Theorem 3.10 (From [8])**.** *Assuming $D \in \mathbb{R}_+^{k\times n}$ has a nonnegative matrix factorization $D = CS^T$, with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$. Augmenting D by a nonnegative row results in $\tilde{D} \in \mathbb{R}_+^{(k+1)\times n}$. Assuming $\operatorname{rank}(\tilde{D}) = \operatorname{rank}_+(\tilde{D}) = s$ and thus $\tilde{D} = \tilde{C}S^T$ with $\tilde{C} \in \mathbb{R}_+^{(k+1)\times s}$ and $S \in \mathbb{R}_+^{n\times s}$.*

Then the solutions of $\tilde{D}$ can be determined with $\operatorname{rowcone}(\tilde{D})$ and $\operatorname{colcone}(D)$ instead of $\operatorname{rowcone}(\tilde{D})$ and $\operatorname{colcone}(\tilde{D})$.

Proof. From Lem. 3.8 follows that $(\operatorname{colcone}(\tilde{D}))(1:k) = \operatorname{colcone}(D)$. Furthermore, the cone of a feasible factor $\tilde{C}$ enclosing $\operatorname{colcone}(\tilde{D})$ also satisfies $(\operatorname{colcone}(\tilde{C}))(1:k) \supseteq (\operatorname{colcone}(\tilde{D}))(1:k)$. Thus, by Lem. 3.8, $\tilde{C}(1:k,:)$ is also a solution of the NMF problem of D. However, now not all factors satisfying $(\operatorname{colcone}(\tilde{C}))(1:k) \supseteq \operatorname{colcone}(D)$ have a feasible dual S^T (see example 3.9). Therefore an additional step is needed, where also the feasibility of S^T is checked. Alternatively, $\operatorname{rowcone}(\tilde{D})$ can be considered to obtain S^T. This is, because the rowcone has not been reduced. Therefore, no additional step is necessary when looking at $\operatorname{rowcone}(\tilde{D})$, since the dual factor is automatically feasible, see Lem. 2.5.
So it is possible to consider only $\operatorname{rowcone}(\tilde{D})$ and $\operatorname{colcone}(D)$ to get all solutions. □

If D is augmented by more than one row, Thm. 3.10 can be applied successively. Thus, Eq. (3.2), i.e., $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, can be analyzed with $\operatorname{colcone}(D_{11})$ and $\operatorname{rowcone}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ if they have the same rank. The same can be done for (D_{11}, D_{12}) using the transposed case. The natural next step is to look at the block matrix D of Eq. (3.1) and find reduced cones that are sufficient to compute all solutions. This requires a closer look at the reconstruction of the set of solutions based on submatrices.

Remark 3.11. *Let $D \in \mathbb{R}_+^{k\times n}$ be given, as in Eq. (3.1), with $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D) = s$. Then the feasible pair $C = \left(\begin{smallmatrix} C_1 \\ C_2 \end{smallmatrix}\right) \in \mathbb{R}_+^{k\times s}$ and $S^T = (S_1^T, S_2^T) \in \mathbb{R}_+^{s\times n}$ of D is uniquely reconstructed by C_1 and S_1 (except for contributions from the respective null space). This is because $S_2^T = C_1^+ D_{12}$ and $C_2 = D_{21}(S_1^T)^+$. Here C_1^+ denotes the pseudoinverse of C_1, see [28].*

With this remark it is now possible to construct reduced cones that still lead to the same set of solutions as when the cones of the whole matrix are considered.

Theorem 3.12 (From [8])**.** *Based on the premise that $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, where $D_{11} \in \mathbb{R}_+^{k_1\times n_1}$ being a submatrix of $D \in \mathbb{R}_+^{k\times n}$, see Eq. (3.1), and D has a nonnegative factorization with $C \in \mathbb{R}^{k\times s}$ and $S^T \in \mathbb{R}^{s\times n}$, the following holds*

$$(\operatorname{colcone}(D))(1:k_1) = \operatorname{colcone}(D_{11}, D_{12}),$$
$$(\operatorname{rowcone}(D))(1:n_1) = \operatorname{rowcone}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right).$$

The set of solutions for $D = CS^T$ is determined by these cones.

Proof. Applying Lem. 3.8 leads to $(\text{colcone}(D))(1:k_1) = \text{colcone}(D_{11}, D_{12})$ and thus both $\text{colcone}(D_{11}, D_{12})$ and $\text{rowcone}(D)$ are considered. This means that C_1 and S^T can be determined using these cones. However, it is sufficient to consider only C_1 and $S_1 = S(1:n_1,:)$, since they already determine C and S^T (see Rem. 3.11). Thus, only the first n_1 dimensions of $\text{rowcone}(D)$ are needed, which leads to the use of $(\text{rowcone}(D))(1:n_1) = \text{rowcone}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ (see Lem. 3.8). The next step is to consider the set of feasible solutions.
Possible solutions C_1 which enclose $\text{colcone}(D_{11}, D_{12})$, but whose dual S^T does not enclose $\text{rowcone}(D)$, are also not feasible when considering the reduced factor S_1^T and $\text{rowcone}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ and vice versa. This is because if the dual S^T is not feasible for $\text{rowcone}(D)$, then the number of edges of the cone spanned by $\text{rowcone}(D) \cup \text{rowcone}(S^T)$ is greater than s. However, a dimension reduction for D and S^T to $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ and S_1^T means that the cone spanned by $\text{rowcone}(D) \cup \text{rowcone}(S^T)$ is reduced. Due to Lem. 3.8 the number of edges remains the same (here greater than s) and thus the reduced dual factor S_1^T is also not feasible in the reduced space. Therefore, the set of feasible solutions of D is not affected by the reduction to $\text{colcone}(D_{11}, D_{12})$ and $\text{rowcone}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, where the solutions of D are constructed based on C_1 and S_1^T, see Rem. 3.11. □

With this theorem it is now possible to consider data matrices as presented in Eq. (3.1) without using the block D_{22} and hence to find solutions of problem 3.3. Thus, D_{22} can contain missing values or can be completely unknown, but all solutions of the NMF problem 2.3 of D can still be found. In the noise-free case, this also allows to reconstruct D_{22} using the factors C_2 and S_2^T.

This construction has the side effect, that it allows the block D_{22} to be very large.

Remark 3.13. *Based on the assumptions of Thm. 3.12, D_{11} must be a matrix of rank s. This means that D can be permuted so that $D_{11} \in \mathbb{R}^{s \times s}$ with $\text{rank}(D_{11}) = s$, $k_1 = s$ and $n_1 = s$. Thus, $D_{12} \in \mathbb{R}^{s \times (k-s)}$ and $D_{21} \in \mathbb{R}^{(n-s) \times s}$ result in $D_{22} \in \mathbb{R}^{(n-s) \times (k-s)}$. Therefore, up to $100 \cdot \frac{(k-s) \cdot (n-s)}{k \cdot n}\%$ of missing entries are possible and can still be handled with the proposed method (if a proper $s \times s$ shared block is selected).*

To allow an analysis of the SFS and the underlying ambiguity of problem 3.3, the insights of this section are transferred to the low-dimensional representation using an SVD, see Sec. 3.4. But first, an alternative approach is presented on how to analyze a data matrix with a missing block D_{22} by reconstructing it directly.

3.3.3. Alternative: Direct reconstruction of D_{22}

If the goal is a reconstruction of D_{22}, it is also possible to compute the missing block directly. However, this approach can only be used for noise-free data. The noisy case for this reconstruction is examined in Sec. 3.8.2.

Theorem 3.14 (From [8])**.** *Let $D \in \mathbb{R}_+^{k \times n}$ have the structure of Eq. (3.1) with $D_{11} = C_1 S_1^T$ and $\text{rank}(D_{11}) = \text{rank}(D) = \text{rank}_+(D) = s$, $C_1 \in \mathbb{R}_+^{k_1 \times s}$ and $S_1 \in \mathbb{R}_+^{n_1 \times s}$. Then the block D_{22} is determined by $D_{22} = D_{21} D_{11}^+ D_{12}$.*

Proof. Based on Eq. (3.1) holds $D_{12} = C_1 S_2^T$ and $D_{21} = C_2 S_1^T$. Because of $\text{rank}(D) = \text{rank}(D_{11})$, the column space of D_{11} already includes the column space of D_{12}. The same is true for the row space and D_{21}. This allows to solve the equations $D_{12} = C_1 S_2^T$ and $D_{21} = C_2 S_1^T$ with respect to S_2^T and C_2 from Eq. (3.1). Thus, S_2^T and C_2 are given by

$$S_2^T = C_1^+ D_{12}, \quad C_2^T = D_{21} (S_1^T)^+.$$

The use of the pseudoinverse C_1^+ and $(S_1^T)^+$ guarantees a solution of smallest Euclidean norm [28]. This allows the representation of D_{22} by

$$D_{22} = C_2 S_2^T = D_{21}(S_1^T)^+ C_1^+ D_{12} = D_{21}(C_1 S_1^T)^+ D_{12} = D_{21} D_{11}^+ D_{12}.$$

However, this representation of D_{22} is not necessarily unique, since the pseudoinverse is not necessarily unique due to contributions from the respective null spaces. □

If D_{11} is an $s \times s$ matrix, then D_{22} is unique, since D_{11} is then invertible and thus the pseudoinverse becomes the inverse.

This theorem holds only if $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D)$. For case 2 with $\operatorname{rank}(D_{11}) < \operatorname{rank}(D)$ no such equation can be formed, because either the column space of D_{12} is not included in D_{11} or the row space of D_{21}. The result is that it is not even possible to determine a unique approximation of D_{22} by considering only the species of D_{11} or the shared species. This is because it is possible to vary S_2 while keeping $C_1(:, 1 : s_1)$ and S_1 fixed. Since S_1 stays the same, C_2 does not necessarily change. Since $D_{22} = C_2 S_2^T$, different approximations can be obtained.

Case 2, where $\operatorname{rank}(D_{11}) < \operatorname{rank}(D)$, is discussed in detail in Sec. 3.5 using the low-dimensional representation. But first, the general approach for $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D)$ is transferred to the low-dimensional representation using an SVD.

3.4. The shared representation using an SVD

As described in Sec. 2.2 and 2.3, the drawback of using the cone representation to solve the NMF problem 2.3 is that the dimension of the representation depends on the rows and columns of the data matrix instead of the rank. Although this can be avoided by choosing D_{11} as a matrix of size $s \times s$, see Rem. 3.13, such a reduction is not always advisable, especially when transferring the proposed approach to noisy data sets. Therefore, the SVD representation is used in this section to solve problem 3.3. Therefore, all the results of Sec. 3.3 are transferred to the low-dimensional representation based on the SVD. This is done by taking Thm. 3.10 and 3.12, which contain the main results, and transferring them to the low-dimensional representation.

A flowchart of the procedure to analyze a data set $D \in \mathbb{R}_+^{k \times n}$ of Eq. (3.1) and to solve problem 3.3 using a low-dimensional representation has already been shown in Fig. 3.1. The explanations and proofs are given in this section.

Starting with the data points of a nonnegative matrix $D \in \mathbb{R}^{k \times n}$, see Eq. (2.11) and (2.13), they are computed based on the rows or columns of D using an SVD of $D = U\Sigma V^T$ with subsequent scaling.

The more general calculation of a data representing point belonging to an arbitrary vector $d_1 \in \mathbb{R}_+^n$ is done by a projection of the vector into the low-dimensional representation, in this case into the V-space by

$$\widehat{a} = \frac{((d_1^T V)(2 : s))^T}{(d_1^T V)(1)}.$$

For the U-space a projection of $d_2 \in \mathbb{R}_+^k$ is done by

$$\widehat{b} = \frac{(\Sigma^{-1} U^T d_2)(2 : s)}{(\Sigma^{-1} U^T d_2)(1)}.$$

This calculation can be used when D is extended by additional rows and/or columns and to represent them by their corresponding data points in the low-dimensional representation of D. However, considering these additional rows and/or columns is not as easy as it seems.

Remark 3.15. *Augmenting a matrix by rows or columns changes its SVD and thus its low-dimensional representation. Therefore, data points belonging to the augmented rows and columns cannot simply be added into the V- or U-space to compute the corresponding SFS.*

The construction of adding either a row or a column to D in a low-dimensional representation with the corresponding conditions is considered next. The idea follows the same structure as for cones in Thm. 3.10.

Theorem 3.16. *Assuming $D \in \mathbb{R}_+^{k\times n}$ with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ has a nonnegative matrix factorization $D = CS^T$, where $C \in \mathbb{R}^{k\times s}$, $S^T \in \mathbb{R}^{s\times n}$ and DD^T and D^TD are irreducible. Adding a $(k+1)$-th nonnegative row to D results in $\tilde{D} \in \mathbb{R}_+^{(k+1)\times n}$. Assuming $\operatorname{rank}(\tilde{D}) = \operatorname{rank}_+(\tilde{D}) = s$ and therefore $\tilde{D} = \tilde{C}S^T$ with $\tilde{C} \in \mathbb{R}_+^{(k+1)\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$.*

Then the solutions of the NMF problem for $\tilde{D}$ can be determined using $\mathcal{F}_S(D)$ and $\mathcal{I}_S(D)$ expanded by the data point of the new row instead of $\mathcal{F}_S(\tilde{D})$ and $\mathcal{I}_S(\tilde{D})$.

Proof. Because of Thm. 2.20 and Lem. 2.21 holds that the low-dimensional representation of D is equivalent to the cone representation of D. Adding a row results in an additional data point in the V-space. The convex hull of $\mathcal{I}_S(D)$ and the additional data point is equivalent to rowcone$(\tilde{D})$ because $\mathcal{E}$ is the same (based on conic combinations). To show that both representations are equivalent, it is also necessary to check that the solutions are the same. The same reasoning as for Lem. 2.21 is applied based on a feasible factor S^T for $\tilde{D}$ (or for rowcone$(\tilde{D})$). Thus, if S^T is feasible for rowcone$(\tilde{D})$, it is also feasible in the low-dimensional representation with the new data point added.
This means that even with an added data point, both representations are still equivalent, and thus Thm. 3.10 can be used to prove the statement. □

This theorem also holds for the transposed case when additional columns are considered.

As a consequence of Thm. 3.16, it is now possible to add data points to the low-dimensional representation and to make the SFS of, e.g., two matrices D_{11} and $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$, more comparable without changing the low-dimensional representation, as long as the rank is not changed by the new rows. This is illustrated in Fig. 3.4 for added rows and columns in the second and third rows, respectively. Using duality, the effect is also shown in the corresponding dual space, where the additional data points lead to affine hyperplanes that constrain the outer polytope.

This consideration also means that as long as the rank condition is satisfied, the modified inner polytope is enclosed by the original outer polytope $\mathcal{F}_S(D)$. But what happens if the outer polytope is also modified? Can an intersection between the two polytopes occur, resulting in an empty SFS? These questions are investigated in the next theorem.

Theorem 3.17. *Based on the premise that $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, with $D_{11} \in \mathbb{R}_+^{k_1\times n_1}$ and $D \in \mathbb{R}_+^{k\times n}$ from Eq. (3.1), with DD^T and D^TD being irreducible, and D has a nonnegative factorization $D = CS^T$ with $C \in \mathbb{R}^{k\times s}$ and $S^T \in \mathbb{R}^{s\times n}$, it holds that the set of feasible solutions for D is restricted by*

- *$\mathcal{I}_S(D_{11})$ expanded by the data points of D_{21} (in the V-space) and*
- *$\mathcal{I}_C(D_{11})$ expanded by the data points of D_{12} (in the U-space, thus a reduced $\mathcal{F}_S(D_{11})$).*

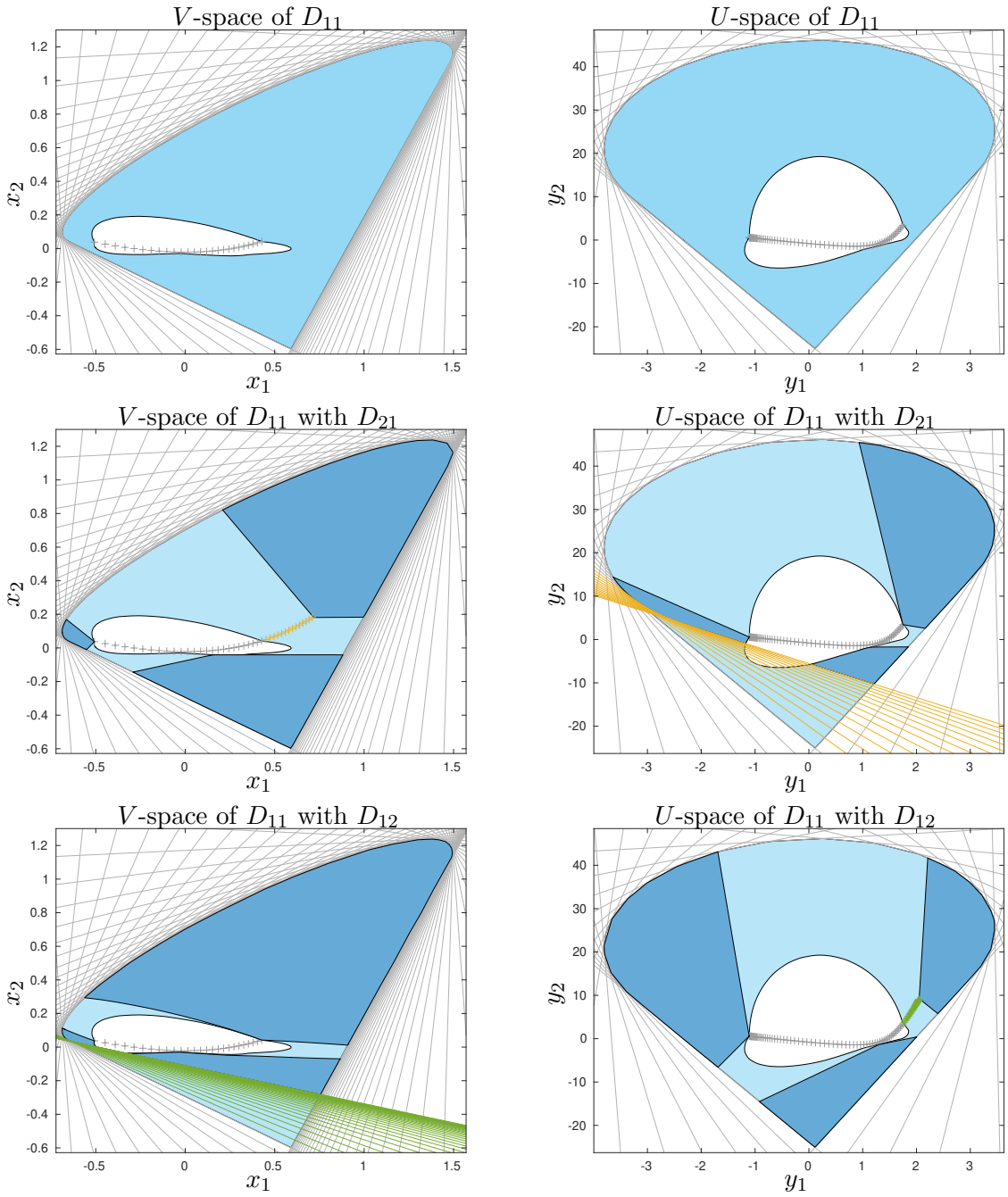

Figure 3.4.: Low-dimensional representation of the shared block D_{11} of the saturated data set 1 in the first row. The second and third rows use Thm. 3.16. Second row: The information of the submatrix D_{21} is added by adding data points in the V-space and thus adding the dual nonnegativity constraints in the U-space. Both are shown in yellow. Third row: The information of the submatrix D_{12} is added to the low-dimensional representation of D_{11} as green data points in the U-space and green lines in the V-space.

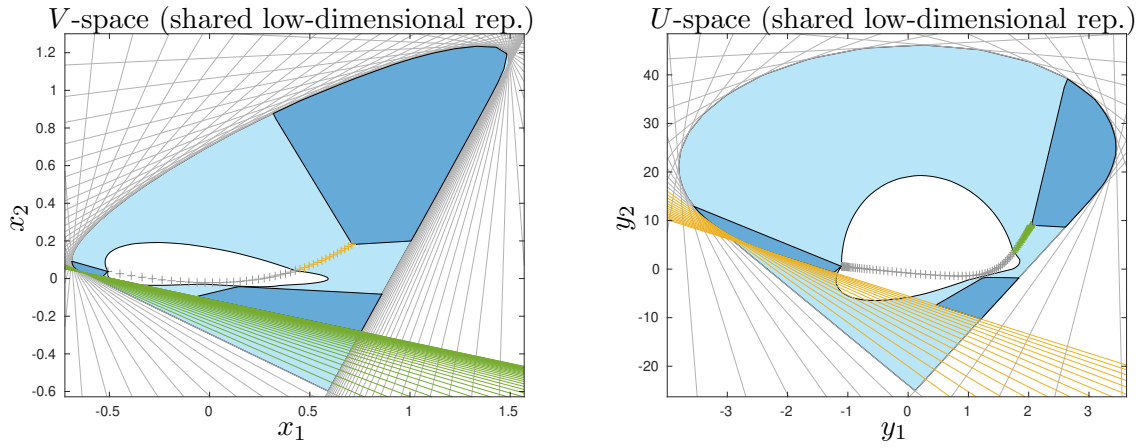

Figure 3.5.: Shared low-dimensional representation (Def. 3.18) of the saturated data set 1, where Thm. 3.17 is applied. The added information of D_{21} is marked in yellow and the information of D_{12} is marked in green in both the V- and U-space of D_{11}. This also visualizes the combination of the results of Fig. 3.4.

Proof. Thm. 3.16 leads to the statement that $\mathcal{I}_S(D_{11})$ expanded by the data points of D_{21} is equivalent to rowcone$\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and that $\mathcal{I}_C(D_{11})$ expanded by the data points of D_{12} is equivalent to colcone(D_{11}, D_{12}).
The goal is to reduce the problem to the one given in Thm. 3.12, which uses rowcone$\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and colcone(D_{11}, D_{12}) to determine the solutions of D. So the next step is to compare the solutions of both representations. To check if a solution of both representations is feasible for D, it has to be checked for both cones, the one in the row space and the one in the column space. The same can be done in the low-dimensional representation. Because of duality it is sufficient to check whether both expanded inner polytopes are enclosed by both factors instead of, e.g., looking at $\mathcal{I}_S$ and $\mathcal{F}_S$ in the V-space. Lem. 2.21, applied to the expanded $\mathcal{I}_S$ and rowcone$\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ as well as to $\mathcal{I}_C$ and colcone(D_{11}, D_{12}), shows that the solutions of both representations are the same. Therefore, Thm. 3.12 is used to show the assertion. □

This allows the SFS of D to be represented in the low-dimensional representation of the shared block D_{11} by including the data points of the blocks D_{12} and D_{21} without using the block D_{22}. Fig. 3.5 shows this for the saturated data set 1 from Fig. 3.2, with the corresponding band boundary representation in Fig. 3.6. The band boundaries for the resulting reconstructed factors C and S^T are shown in Fig. 3.7. The representation of Fig. 3.5 using the shared block is defined next.

Definition 3.18. *The representation of all constraints of $D \in \mathbb{R}_+^{k \times n}$ of Eq. (3.1) for case 1 in the low-dimensional representation of the shared block D_{11} is called* shared low-dimensional representation.

This shared low-dimensional representation allows not only to represent incomplete data sets for case 1 as in Eq. (3.1), but also to represent complete data sets that are augmented (for case 1) in one low-dimensional representation without the need to change the representation despite a changing SVD. Thus, an evolution of the SFS over time can be represented by the shared low-dimensional representation without knowing the entire data matrix a priori.

Looking at the shared low-dimensional representation, Thm. 3.17 also leads to the conclusion that if the inner and outer polytopes intersect, then the assumptions are not met, i.e., either D

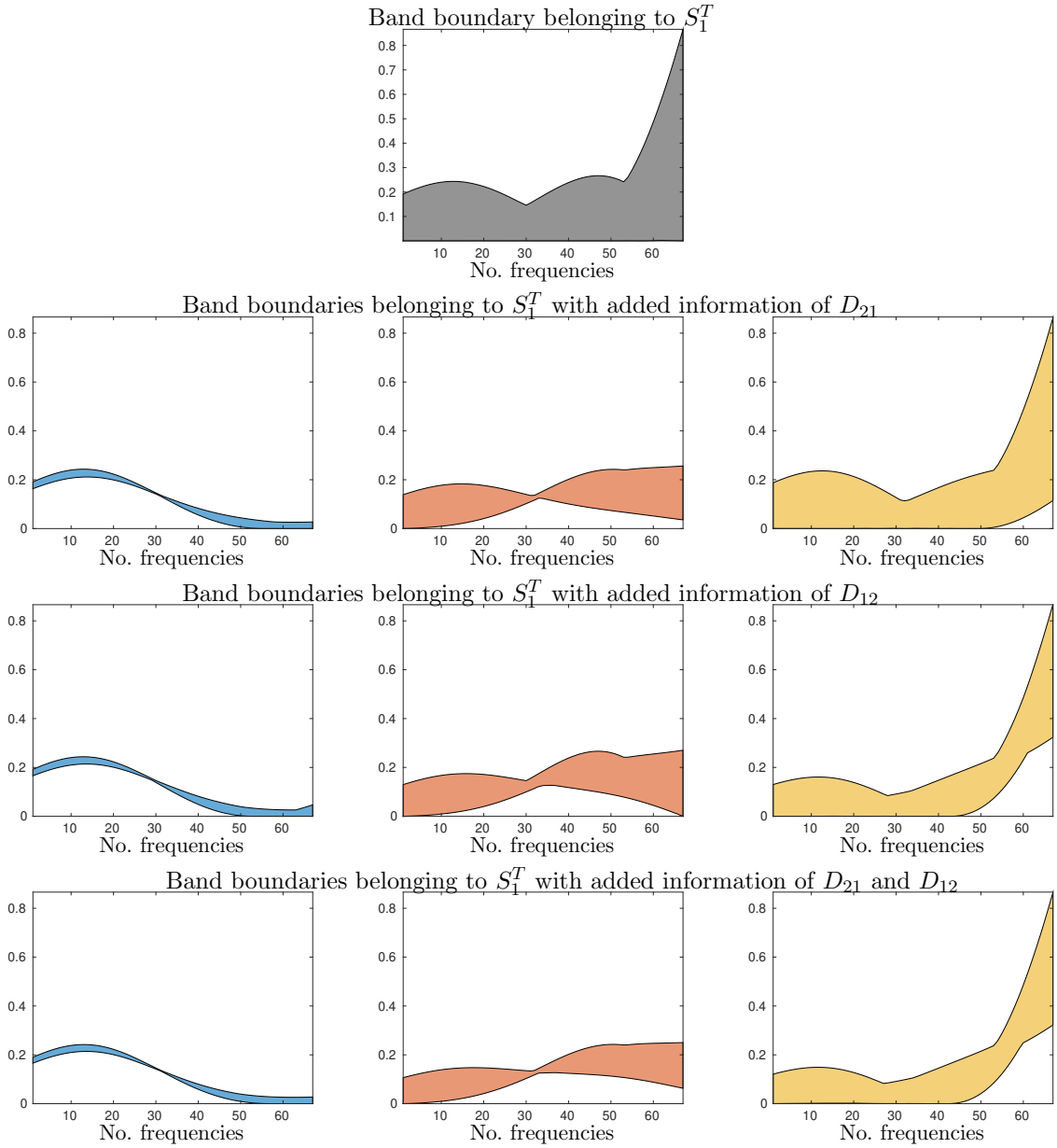

Figure 3.6.: Band boundary representation for factor S_1^T corresponding to the V-spaces of Fig. 3.4 and 3.5. First row: As shown in Fig. 3.4 in the first row, there is only one connected segment when considering the SFS of D_{11}. Therefore, the corresponding band representation is only one plot and it is shown in gray for all 3 underlying species. Second row: Corresponding to Fig. 3.4 (second row) the bands are shown when the information of D_{21} is added. Now the SFS consists of three separate segments and thus there are three band boundary representations. Third row: As in Fig. 3.4 the information of D_{12} is added to D_{11}, resulting in the band boundaries shown. Fourth row: Adding the information of both D_{21} and D_{12} results in the shared low-dimensional representation as shown in Fig. 3.5 with the corresponding band boundaries shown in this row. This is a subset of the intersection of the previously determined band boundaries.

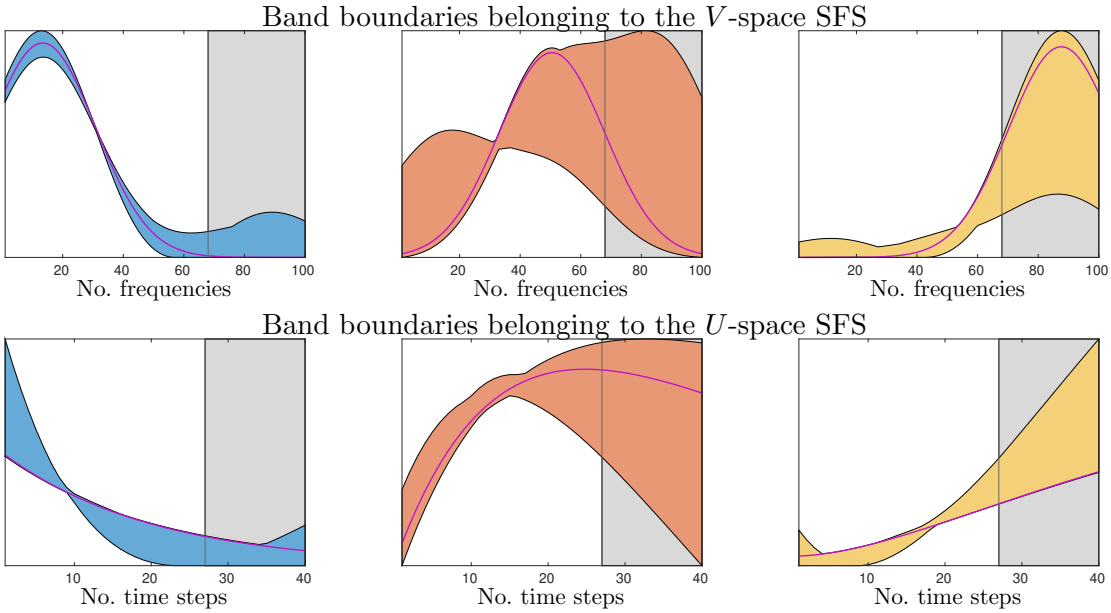

Figure 3.7.: Band boundary representation of the SFS corresponding to the shared low-dimensional representation of Fig. 3.5 and extended to show the entire C and S^T belonging to data set 1 underlaid in gray. Comparing this representation with Fig. 3.6, which shows only the band boundaries of S_1^T, the axes are chosen to give a better view of the ambiguity. Additionally, the profiles from A.1, which are also shown in Fig. 2.4, are marked in magenta.

contains negative entries or the rank condition is not satisfied, resulting in $\operatorname{rank}(D) > \operatorname{rank}(D_{11})$. How such intersections can occur due to a wrong rank assumption is shown in Thm. 2.25. Data sets that do not satisfy the rank condition and are therefore not classified into case 1 are discussed in Sec. 3.5. An example is data set 2.

But first a closer look is taken at the resulting band boundaries shown in Fig. 3.6 and the underlying SFS in Fig. 3.4 and 3.5.

3.4.1. The problem of artificially minimizing the SFS

When computing the SFS of (D_{11}, D_{12}) and $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$, there is a tendency to simply compare the sets of feasible solutions in the shared low-dimensional representation. This is because the standard methods for computing the SFS can simply be applied to the submatrices (D_{11}, D_{12}) and $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and the results need only be projected into the shared low-dimensional representation for comparison. However, computing the intersection of the SFS does not give the SFS of the complete data set or the SFS considering all given rows and columns.

Remark 3.19. *The intersection of the SFS of* (D_{11}, D_{12}) *and* $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ *in the shared low-dimensional representation leads to a superset of the SFS of the complete data set* D *from Eq.* (3.1).

This remark is visualized in Fig. 3.8 for the SFS and the band boundaries.

Considering the geometric construction of the SFS using a simplex rotation of Sec. 2.3.3, this fact is not surprising. That is, minimizing the SFS has the consequence that it is possible that simplices that were feasible are no longer feasible. Thus, to a point inside the minimized SFS, no feasible simplex can be found, since the possible corresponding vertices are now outside the

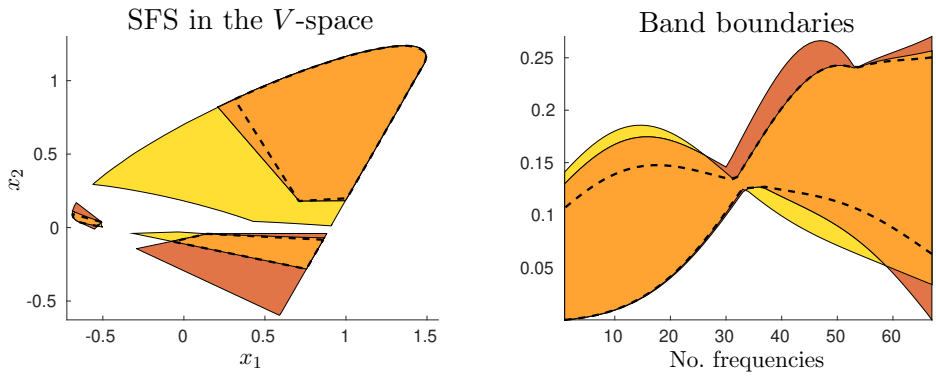

Figure 3.8.: Comparison of different SFS sets in the shared low-dimensional representation (left) and the corresponding band boundaries (right). In yellow is the SFS of (D_{11}, D_{12}), in red is the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and the orange region represents the intersection of both. The SFS with respect to the additional constraints of both D_{12} and D_{21} is enclosed by the dashed black lines, showing a subset of the orange segment. The same is shown with the same color coding for the band boundaries of one species (the red one in Fig. 3.6).

SFS. Therefore, the SFS has to be recomputed for every change, be it on the inner and outer polytopes or by artificially minimizing the SFS.

This reasoning can also be applied to comparing band boundaries. However, a different aspect comes into play here. When comparing band boundaries, the underlying scaling must be the same. This can be seen by comparing Fig. 2.4 and Fig. 3.7. Both figures show the same set of solutions, but scaled using the low-dimensional representation of D and the shared low-dimensional representation respectively. For better comparison, the same 3 selected profiles are shown in dashed black and magenta, respectively.

These relations are relevant not only for case 1, but also for the following case 2 and even more complex cases as in Sec. 3.6 and should be kept in mind when comparing different sets of feasible solutions.

3.5. How to handle the case of $\operatorname{rank}(D) > \operatorname{rank}(D_{11})$

After examining case 1 in detail, case 2 is analyzed in this section. Case 2 describes that not all species of the data set appear in the submatrix D_{11}, see Eq. (3.1) and (3.2) for assumed block structures. An example is shown in Fig. 3.9 for data set 2 with simulated incompleteness, which is classified as case 2. Therefore, a closer look at the submatrices is necessary.

As in Eq. (3.2), the selection of submatrices is done first along one dimension. This means that a submatrix $\hat{D} \in \mathbb{R}_+^{\hat{k} \times \hat{n}}$ of $D \in \mathbb{R}_+^{k \times n}$ has either the same number of rows or columns as the original matrix (either $\hat{k} = k$ or $\hat{n} = n$). This makes it possible to analyze submatrices that are a selection of rows. Considering submatrices with a column-wise selection is the transposed case. Cases where a submatrix is selected along both dimensions (with $\hat{k} < k$ and $\hat{n} < n$) are considered in Thm. 3.24.

Thus, looking first at Eq. (3.2), $\operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix} > \operatorname{rank}(D_{11})$ holds to classify as case 2. To access the SFS, a low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is used. However, in order to include all given information, the rank of each matrix must be taken into account. Therefore, affine subspaces are used. Let $\hat{k}$ data points d_i in the V-space be given. Then the affine subspace on which all

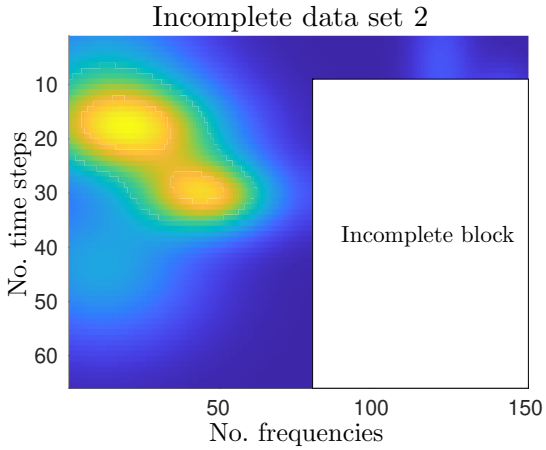

Figure 3.9.: Simulated incompleteness for data set 2, where only the first 9 rows and the first 80 columns are complete, the submatrix $D(10:66, 81:150)$ is assumed to be missing or unknown. This allows a classification as case 2 because the first 9 rows span a rank 2 submatrix.

data points lie is given by

$$\mathcal{H} = \{x \in \mathbb{R}^{s-1} : \ x = d_1 + \sum_{i=2}^{\hat{k}} \lambda_i (d_i - d_1), \ \lambda_i \in \mathbb{R}\}. \tag{3.5}$$

An affine subspace is visualized in Fig. 3.3 and in Fig. 3.10 for data set 2 by the yellow line. In the same way, affine subspaces can be defined in the U-space.

The next step is to look at the dimension of the affine subspace.

Lemma 3.20. *Let $\hat{D}$ be a submatrix of $D \in \mathbb{R}_+^{k\times n}$ with $\hat{D} \in \mathbb{R}_+^{\hat{k}\times n}$, $\hat{k} < k$, $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ and $\operatorname{rank}(\hat{D}) = \operatorname{rank}_+(\hat{D}) = \hat{s} < s$. Then the corresponding object spanned by the data points of $\hat{D}$ in the low-dimensional representation of D has the dimension $\hat{s} - 1$. Similarly, for an $(\hat{s}-1)$-dimensional affine subspace $\mathcal{H}$ spanned by data points of D, there exists a rank $\hat{s}$ submatrix that generates it.*

Proof. If there is a submatrix $\hat{D} \in \mathbb{R}^{\hat{k}\times n}$, its corresponding data points d_i in the V-space are calculated by $d_i = (\hat{D}V)(i, 2:s))^T/((\hat{D}V)(i,1))$, $i = 1, \ldots, \hat{k}$, where V belongs to the SVD of $D = U\Sigma V^T$. Therefore, the affine subspace spanned by the data points has a maximum dimension of $\hat{s}$, since $\operatorname{rank}(\hat{D}) = \hat{s}$ and thus also $\operatorname{rank}(\hat{D}V) = \hat{s}$.
However, the dimension reduction by scaling (here scaled by $(\hat{D}V)(i,1)$) maps the $\hat{s}$-dimensional subspace $\hat{D}V$ to an $(\hat{s}-1)$-dimensional affine subspace (which can be represented in the V-space of D). This is because the low-dimensional representation is based on a cone representation using rowcone$(U\Sigma)$ for the V-space and projecting it onto a hyperplane, resulting in the cone becoming a polytope and reducing the dimension by one. Thus, the data points span an $(\hat{s}-1)$-dimensional affine subspace denoted by $\mathcal{H}$ (see Eq. (3.5)).
Conversely, the reasoning is the same. Having an $(\hat{s}-1)$-dimensional affine subspace $\mathcal{H}$ in the V-space, it can be transferred to the space of the right singular vectors by considering $(1, x^T)^T$ for all $x \in \mathcal{H}$. Thus, a $\hat{s}$-dimensional subspace of the column space of V is spanned. The corresponding submatrix $\hat{D}$, on which the data points generating $\mathcal{H}$ are based, thus has $\operatorname{rank}(\hat{D}) = \hat{s}$. □

Thus, all spectra that lie in the row space of $\hat{D}$ have corresponding data points, that are elements of $\mathcal{H}$.

Using rank information of a submatrix of D is a well-known concept and is called "local rank" information [79]. Usually the goal of local rank information is to use it in a way that reduces the ambiguity. This reduction is achieved by fixing the rank of parts of the corresponding factors C and S^T. For example, if $\operatorname{rank}(D(1:\hat{k},:)) = \hat{s}$, this results in two factors where only the first $\hat{s}$ columns of C and S contain non-zero entries, i.e., $C(1:\hat{k}, 1:\hat{s}) = \hat{C}$ and $S(:, 1:\hat{s}) = \hat{S}$ with $D(1:\hat{k},:) = \hat{C}\hat{S}^T$. If such a reduction to $\hat{C}$ and $\hat{S}$ is not possible, but for all possible solutions $\operatorname{rank}(C(1:\hat{k}, 1:\hat{s})) = \operatorname{rank}(S(:, 1:\hat{s})) = \hat{s}$ holds, the submatrix has a local rank deficiency, see Def. 3.2. This means that the number of species that should be contained in this submatrix when analyzing the full matrix does not match the nonnegative rank of the submatrix. Thus, there is no nonnegative solution that factorizes both the submatrix, with $\hat{C}$ and $\hat{S}$, and the entire matrix. Therefore, the assumption for this section is that $\mathcal{H}$ is based on a properly chosen submatrix without local rank deficiency.

Considering now the matrix D_{11} as submatrix of $\begin{pmatrix} D_{11} \\ D_{12} \end{pmatrix}$ of Eq. (3.2), where $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) = s_1$ and D_{11} having no local rank deficiency. The data points of D_{11} span the $(s_1 - 1)$-dimensional subspace $\mathcal{H}$. Then holds the following lemma.

Lemma 3.21. *Assuming D_{11} is a properly selected submatrix of $D \in \mathbb{R}_+^{k \times n}$ from Eq. (3.2), with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ and DD^T, D^TD being irreducible. Let the affine subspace $\mathcal{H}$ of Eq. (3.5) be based on D_{11}. The intersection of $\mathcal{H}$ with $\mathcal{I}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is only part of the boundary of $\mathcal{I}_S\begin{pmatrix} D_{11} \\ D_{21.} \end{pmatrix}$.*

Proof. Assuming that $\mathcal{H}$ intersects with $\mathcal{I}_S$, this means that there exists no solution whose simplex has s_1 vertices lying on $\mathcal{H}$ (since $\mathcal{I}_S$ is not enclosed by the simplex). This also means that there is no solution to the NMF problem of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ that factorizes only D_{11}, which contradicts the choice of the submatrix. It remains to show that $\mathcal{H}$ actually touches $\mathcal{I}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$. Since $\mathcal{H}$ is spanned by the data points of D_{11}, which also define $\mathcal{I}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$, it must touch $\mathcal{I}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$. □

With this lemma it is possible to formulate statements about the position of the subsystem in the low-dimensional representation.

Remark 3.22. *The affine subspace $\mathcal{H}$ of Eq. (3.5), belonging to a properly chosen submatrix, does not intersect with the origin, since the origin must be an interior point of $\mathcal{I}_S$, see Lem. 2.17. Therefore, $\mathcal{H}$ cannot coincide with an axis.*

Having made the connection between $\mathcal{H}$ and $\mathcal{I}_S$, the natural next step is to take a closer look at the outer polytope.

Lemma 3.23. *Let $D_{11} \in \mathbb{R}_+^{k_1 \times n_1}$ be a submatrix of $D \in \mathbb{R}_+^{k \times n}$ with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) < s$, and DD^T and D^TD are irreducible. Assuming D_{11} has no local rank deficiency and spans $\mathcal{H}$ as in Eq. (3.5). Then $\mathcal{F}_S(D_{11})$ can be mapped to $\mathcal{H}$, which coincides with the intersection of $\mathcal{H}$ and $\mathcal{F}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$.*

Proof. The profiles corresponding to the points on the boundary of $\mathcal{F}_S(D_{11})$ contain at least one zero due to active constraints defining the boundary of $\mathcal{F}_S$, see Eq. (2.10). Thus, mapping these profiles to the low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ means that the corresponding data points lie on $\mathcal{H}$ and also on the boundary of $\mathcal{F}_S\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$, since the profiles contain zeros,

see Eq. (2.10). Since $\mathcal{F}_S(D_{11})$ is a convex set, there are no holes, especially at the boundary. Therefore, the transferred $\mathcal{F}_S(D_{11})$ is also a convex set, and since $\mathcal{F}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ is also a convex set, the transferred $\mathcal{F}_S(D_{11})$ must coincide with the intersection of $\mathcal{H}$ and $\mathcal{F}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$. □

The dual case where $\mathcal{I}_C$ and $\mathcal{F}_C$ are considered is equivalent, for example when having (D_{11}, D_{12}) from Eq. (3.1).

The next step is to determine the ambiguity using the rank condition, but not only for $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, but also for (D_{11}, D_{12}) of Eq. (3.1) and to combine the results. So the goal is to analyze case 2. For this, the concept of shared species is used, see Def. 3.4. This concept can also be expressed in terms of the corresponding ranks. For example, two blocks D_{11} and D_{21} have at least one shared species if

$$\operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) < \operatorname{rank}(D_{11}) + \operatorname{rank}(D_{21})$$

holds with a matrix structure of Eq. (3.1), or Eq. (3.2).

Theorem 3.24 (From [8]). *Let $D \in \mathbb{R}_+^{k \times n}$ with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ have a nonnegative matrix factorization $D = CS^T$ with $C \in \mathbb{R}^{k \times s}$ and $S^T \in \mathbb{R}^{s \times n}$, where DD^T and D^TD are irreducible. Also $\operatorname{rank}(D) > \operatorname{rank}(D_{11})$ holds, where D has the block structure from Eq. (3.1). Then the constraints of the species shared by either D_{11} and D_{21} or D_{11} and D_{12} can be represented by the low-dimensional representation of D_{11}, the shared block.*

Proof. The blocks D_{11} and D_{21} are considered first, with $\operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) > \operatorname{rank}(D_{11}) = s_1$. The case where D_{11} is extended in column direction by a block D_{12} can be treated analogously.
Starting with the low-dimensional representation of $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, whose V-space has the dimension $\operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) - 1$. The data points belonging to D_{11} lie on an $(s_1 - 1)$-dimensional affine subspace $\mathcal{H}$, see Eq. (3.5). For all data points lying on $\mathcal{H}$, the corresponding profiles are part of the row space of D_{11} because the data points do not change the dimension of $\mathcal{H}$ and thus the corresponding profiles do not change the dimension of the row space of D_{11}. Thus, the rows of D_{21} corresponding to a data point located on $\mathcal{H}$ lie in the row space of D_{11} (and thus correspond to shared species). All such rows of D_{21} are collected in a submatrix D'_{21}. It also holds $\operatorname{rank}(D_{11}) \geq \operatorname{rank}(D'_{21})$.
This allows to include the information of the shared species of D_{21} and D_{11} in the low-dimensional representation of D_{11}. This step already includes all constraints of D_{21} regarding the inner and outer polytopes in the low-dimensional representation of D_{11}. This is because in the V-space the intersection of the outer polytope and $\mathcal{H}$ coincides with a projection of $\mathcal{F}_S(D_{11})$ onto $\mathcal{H}$, see Lem. 3.23. So the outer polytope remains unchanged. The inner polytope is only affected by the data points. Because of Lem. 3.21 only data points lying on $\mathcal{H}$ can lead to further constraints of $\mathcal{I}_S(D_{11})$. These are already included by D'_{21}. Therefore, all constraints of D_{21} regarding shared species can be included by D'_{21} in the same way as for a shared low-dimensional representation in case 1, because $\operatorname{rank}(D_{11}) = \operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D'_{21} \end{smallmatrix}\right)$. □

There are several ways to determine the submatrix D'_{21} used in the proof of Thm. 3.24. The purpose of this matrix is to add the information of D_{21} to the low-dimensional representation of D_{11}. It can be determined by finding those rows of D_{21} whose data points lie on $\mathcal{H}$, the affine subspace formed by the data points of D_{11}. Alternatively, D'_{21} can be found by finding the rows that satisfy $\operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D_{21(i,:)} \end{smallmatrix}\right) = \operatorname{rank}(D_{11})$ for $i = 1, \ldots, k_2$. Thus, D'_{21} contains all rows that satisfy this rank condition.

This allows an analysis of the ambiguity of the species of D_{11} under the additional constraints of D'_{21}. But what about the ambiguity of the species that are not shared?
Looking at D_{21}, the approach is to project the SFS determined in the shared low-dimensional representation onto $\mathcal{H}$ in the V-space of $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$. This represents the SFS of the species of D_{11}. This limits the total SFS. The SFS of the species that are not shared can be recalculated. An example is shown in Fig. 3.10 (left), where the total SFS is constrained by the SFS projected to $\mathcal{H}$. The procedure for dealing with D_{12} is similar.

However, this approach has a major drawback. What if D_{21} does not contain any rows that are only mixtures of the species of D_{11}? Then D'_{21} would be empty and no additional ambiguity reduction could be done for D_{11} and thus for D_{21}. The solution is to look not only at the intersection of $\mathcal{H}$ and the inner polytope and determine D'_{21}, but also at the intersection of $\mathcal{H}$ and the SFS of $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$. That this approach leads to a superset of the desired restricted SFS of the species of D_{11} is shown next.

Lemma 3.25. *Let $D \in \mathbb{R}_+^{k\times n}$ be given with $\text{rank}(D) = \text{rank}_+(D) = s$, so there exists a factorization $D = CS^T$ with $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$. Furthermore, let DD^T and D^TD be irreducible matrices. As in Eq. (3.2) D is divided into two submatrices $D = \left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ with $D_{11} \in \mathbb{R}_+^{k_1\times n}$, $\text{rank}(D_{11}) = \text{rank}_+(D_{11}) = s_1 < s$ and $D_{21} \in \mathbb{R}_+^{k_2\times n}$, $k = k_1 + k_2$. The submatrix D_{11} has no local rank deficiency and spans the affine subspace $\mathcal{H}$ of Eq. (3.5). The corresponding $\mathcal{M}_\mathcal{S}(D_{11})$ is transferred to $\mathcal{H}$, since it is also part of the row space of D_{11}, and is denoted by $\mathbf{M}(D_{11})$. A restricted SFS for the species of D_{11} is defined as*

$$\mathbf{M_R}(D_{11}) = \Big\{x \in \mathbf{M}(D_{11}) : \exists y_1, \ldots, y_{s-1} \in \mathbb{R}^{s-1} \text{ with } y_1, \ldots, y_{s_1-1} \in \mathcal{H} \text{ and } \mathcal{I}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) \subseteq \text{convhull}\{x, y_1, \ldots, y_{s-1}\} \subseteq \mathcal{F}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)\Big\}.$$

Then $\mathcal{H} \cap \mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) \supseteq \mathbf{M_R}(D_{11})$ holds.

Proof. By definition, $\mathbf{M_R}(D_{11})$ is part of $\mathcal{H}$. Since only points corresponding to a feasible simplex of $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ are in the restricted SFS, it is also a subset of $\mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$. So $\mathcal{H} \cap \mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) \supseteq \mathbf{M_R}(D_{11})$. □

Next, a special case is examined, which allows an exact determination of the restricted SFS of Lem. 3.25.

Lemma 3.26. *Let the assumptions of Lem. 3.25 hold. If $\mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ consists of s separate segments and $s_1 = s - 1$, then $\mathcal{H} \cap \mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) = \mathbf{M_R}(D_{11})$.*

Proof. If $x \in \mathcal{H} \cap \mathcal{M}_\mathcal{S}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, then there exists a feasible simplex for $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ with a vertex on $\mathcal{H}$. Since there are s separate SFS segments and the simplex must enclose $\mathcal{I}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, the s_1 vertices of the s_1 segments that intersect $\mathcal{H}$ can be moved along the edges of the simplex until they are on $\mathcal{H}$. Since $\mathcal{H}$ is an (s_1-1)-dimensional affine hyperplane that lies on the boundary of $\mathcal{I}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, the facet of the simplex corresponding to the s_1 vertices is now also on the boundary of $\mathcal{I}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$. Since the affine hyperplanes on which the remaining facets lie remain the same, $\mathcal{I}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ is still contained in the new simplex. It is also still in $\mathcal{F}_S\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ because all vertices

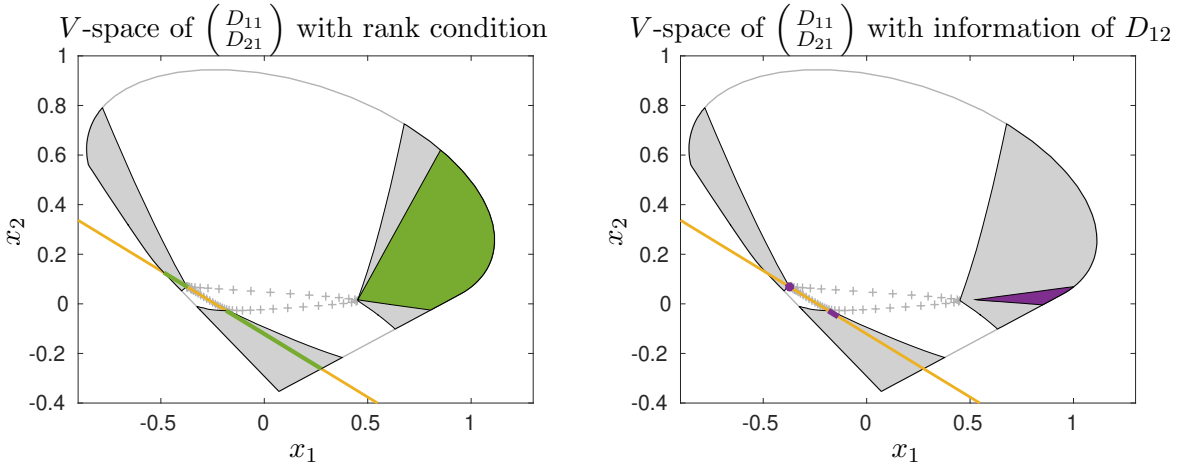

Figure 3.10.: Representation of the V-space of $\binom{D_{11}}{D_{21}}$ of the incomplete data set 2 with the corresponding SFS in gray and $\mathcal{H}$ in yellow. Left: Limited SFS due to rank conditions of D_{11} and thus due to the SFS of D_{11} shown in green. Right: Using the additional information of D_{12} to limit the ambiguity of D_{11} further reduces the SFS, shown in purple. The retrieval of this information is shown in Fig. 3.11.

are elements of the SFS segments. Thus for all $x \in \mathcal{H} \cap \mathcal{M}_{\mathcal{S}}\binom{D_{11}}{D_{21}}$ there exists a feasible simplex and thus all x are feasible and part of $\mathbf{M_R}(D_{11})$. □

Thus, it is possible to compute a superset of the reduced ambiguity with respect to the combined constraints of D_{11} and D_{21} by determining the intersection of $\mathcal{H}$ and the SFS of $\binom{D_{11}}{D_{21}}$. Fig. 3.10 (left) shows the restricted SFS for the special case of Lem. 3.26. The practical determination of $\mathbf{M_R}$ is examined in chapter 4. The profiles corresponding to the points of the reduced ambiguity can be represented in the shared low-dimensional representation based on D_{11}. They represent the SFS subsets of the shared species with respect to the constraints of D_{21}. This is shown in Fig. 3.11 in green.

The next step is to include the information of the submatrix D_{12}. For data set 2 D_{12} and D_{11} have the same rank. Therefore, their information can be included as for case 1. This is shown in Fig. 3.11 in light blue. The combination of both SFS by adding D_{12} and D_{22} is the intersection of the corresponding intervals and is shown in purple. This ambiguity reduction can also be used to reduce the ambiguity of species that are not shared. This is shown in Fig. 3.10 (right) for the segment that does not intersect the affine subspace.

The resulting band boundaries are shown in Fig. 3.12, highlighting the drastic reduction in ambiguity compared to considering only the SFS of $\binom{D_{11}}{D_{21}}$. In addition, a reconstruction of the profiles corresponding to the shared species is possible for high frequencies, since this region is described by D_{12}. However, the third species, which is not shared, cannot be reconstructed in this region because the information needed for this is in D_{22}, the missing block.

Another example of a case 2 analysis was given in [8], see the analysis of data set 2 there. For this data set, it is even possible to achieve uniqueness with the approach presented here.

The result of the analysis of case 2 is that now all possible cases that can occur regarding the rank can be analyzed. But what if there are several missing blocks and an L-shaped form of the accessible information as in Eq. (3.1) is not desired in the analysis, since this can lead to a loss of information. Such complex appearances of missing blocks are examined next.

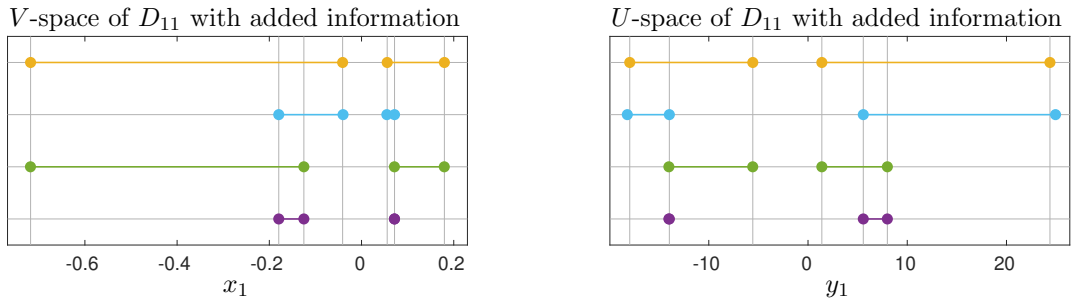

Figure 3.11.: Shared low-dimensional representation with the rank-2 shared block D_{11} of the incomplete data set 2. The SFS of D_{11} is shown in yellow, the reduced SFS due to D_{12} in blue ($\mathcal{I}_C$ grows and $\mathcal{F}_S$ shrinks). The reduced SFS due to D_{21} is shown in green and is the same as in Fig. 3.10 on $\mathcal{H}$. The combination of all these reductions is the intersection of all SFS segments and is shown in purple. It corresponds to the purple SFS in Fig. 3.10 (right).

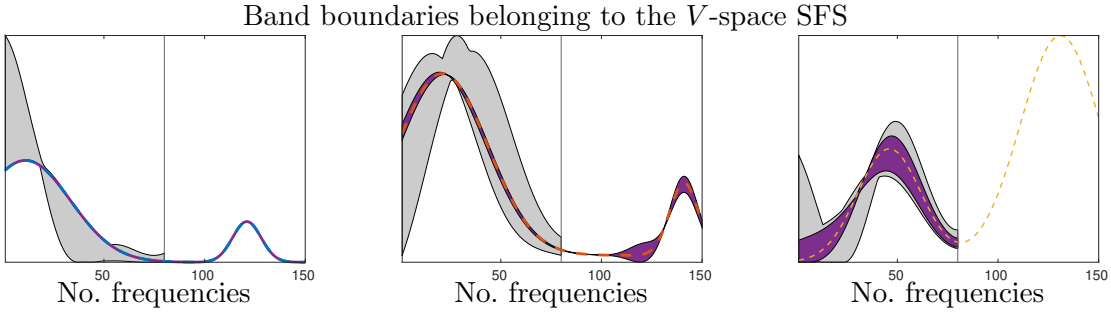

Figure 3.12.: Band boundaries corresponding to the V-space in Fig. 3.10 (right). The band boundaries corresponding to the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ are shown in gray and cover only the first 80 frequencies, due to the submatrix on which they are based. In purple are the band boundaries corresponding to the purple SFS in Fig. 3.10 (right). The original profiles used to generate data set 2, see Fig. A.2, are shown in blue, red and yellow. Only the shared species (left and center) are known for the frequencies 81-150, so only these can be reconstructed.

3.6. Complex block structures

After treating block structures as in Eq. (3.1) with one missing block, denoted by D_{22}, the question arises whether it is possible to analyze structures where more than one block is missing. Such structures have already been analyzed in [8], but are presented here in more detail.

For example, a matrix with a missing value pattern as in Eq. (3.6) can be considered as in Eq. (3.1), where all rows and columns with missing entries are permuted so that the missing entries appear only in D_{22}. However, entries that are not missing are also part of D_{22}. It can even happen that it is not possible to achieve such a structure, where all columns are included in D_{11} and D_{12} and all rows in D_{11} and D_{21} from Eq. (3.1). This would lead to ignoring whole columns or rows to get an L-shaped structure.

When having a matrix that classifies as case 1 an analysis with a complex block structure is not relevant if blocks D_{12} and D_{21} can be found so that all essential information of the matrix is included. This is because only essential rows and columns affect the polytopes. An example of such a data set is the saturated data set 1 as shown in Fig. 3.2. It is possible to consider more blocks to minimize the number of unused entries in the analysis. But as shown in Thm. 3.17, the selected blocks already lead to the same ambiguity as for the complete matrix.
However, if such blocks D_{12}, D_{21} cannot be found, then case 1 can also be analyzed by the following procedure. Even if the reconstruction of the missing parts is possible using Thm. 3.14, and thus the data set does not need to be analyzed as one with multiple missing blocks, the following procedure is of interest when applying the results to noisy data sets.

Looking at case 2, both blocks D_{12} and D_{21} do not contain all the information to reconstruct the ambiguity of the corresponding complete data set, see Sec. 3.5. Thus, the goal is to maximize the information used in the analysis. One solution is to allow multiple blocks with missing entries, thus minimizing the neglect of information. Another aspect why it is important to be able to handle more complex structures is that certain measurements may result in such a pattern, e.g., when multiple areas are saturated or when there are incomplete multisets.

A possible resulting block structure for a nonnegative matrix is shown next. In this section, the missing blocks are left blank for better understanding.

$$\tilde{D} = \left(\begin{array}{c|c|c} D_1 & D_3 & \\ \hline D_2 & & D_4 \end{array}\right) \tag{3.6}$$

Here, for example, the scenario could be that there are two saturated regions or it is a multiset. For the multiset, each column would be measured by a different technique, but for the second and third columns not everything is measured.

To make the connections between the blocks more visible, the blocks are reordered and a nonnegative factorization is shown to further emphasize the connections.

$$D = \left(\begin{array}{c|c|c} & D_1 & D_3 \\ \hline D_4 & D_2 & \end{array}\right) = \left(\begin{array}{c|c|c} & C_1S_2^T & C_1S_3^T \\ \hline C_2S_1^T & C_2S_2^T & \end{array}\right) = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix}^T \tag{3.7}$$

Even if blocks D_3 and D_4 are not directly connected at first sight (they do not share the same rows or columns), they have an effect on each other's ambiguity. Even if they do not contain any shared species.

The general notion of connection between blocks means that if a block is chosen at random, it must share its rows or columns with another block. If not all rows or columns are the same, then the block can be divided into submatrices so that one submatrix has the same rows or columns as another block.

To check if all blocks of a matrix are connected, the concept of irreducibility is used, see Def. 2.12. However, the data matrix is not directly checked for irreducibility, but a representative is used. This representative is constructed using the block structure of the matrix. Each block containing missing entries is represented by a zero and all others by a one. If the resulting matrix is reducible, it can be divided into irreducible systems that are analyzed separately. Thus, the data matrix contains different systems that are independent of each other.

Another aspect when considering the connection between blocks is whether the underlying species are shared between the connected blocks. If this is not the case for a block or a subset of blocks, they can be isolated and analyzed separately, since they contain different species and thus have no influence on the remaining parts.

Returning to the proposed example of Eq. (3.7), it can be seen that the representative of the block matrix is irreducible. Also, assuming that the species are shared between the adjacent blocks, the system does not need to be split. Thus, the next step is to go into detail on how to analyze this data matrix. The first step is to divide the matrix into submatrices that have an L-shape as in Eq. (3.1) (up to permutation), i.e.

$$\hat{D}_1 = \left(\begin{array}{c|c} & D_1 \\ \hline D_4 & D_2 \end{array}\right), \quad \hat{D}_2 = \left(\begin{array}{c|c} D_1 & D_3 \\ \hline D_2 & \end{array}\right).$$

Each of these matrices can be analyzed as described in Sec. 3.4 and 3.5 with the corresponding SFS as a result. However, this does not reflect the minimum possible ambiguity of the entire data set, so two additional steps must be taken.

First, the ambiguity of the species of D_1 and D_2 is compared, since both blocks appear in both submatrices. This can be done by using the low-dimensional representation of D_1 and D_2 and determining the intersection of the SFS segments with a subsequent recalculation, see Sec. 3.4.1.

Second, based on this reduced ambiguity of the shared species between D_1 and D_2, the ambiguity of the remaining species is recalculated. This is the same step as for case 2 in Sec. 3.5. Thus, the minimum ambiguity is achieved. This procedure is visualized in Fig. 3.13.

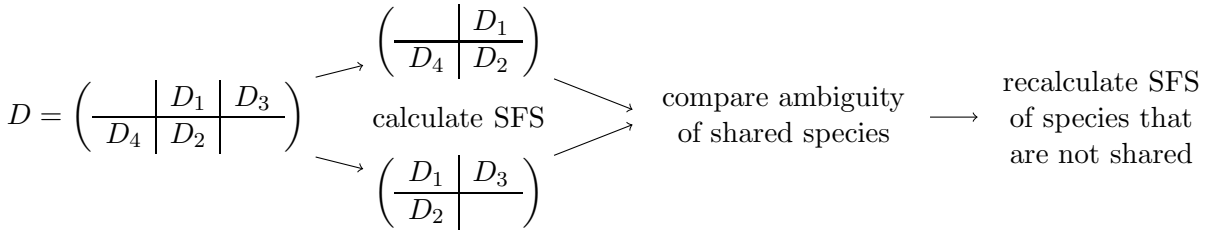

Figure 3.13.: Schematic representation of the handling of complex block structures, based on [8].

On this basis, it is possible to analyze even more complex structures, such as

$$D = \left(\begin{array}{c|c|c} D_1 & D_2 & D_5 \\ \hline & D_3 & \\ \hline & D_4 & D_6 \\ \hline & & D_7 \end{array}\right).$$

This example is only meant to illustrate what structures can be handled. However, a possible application where such a structure appears is in subsystem analysis, where only areas where certain species appear are analyzed, see chapter 5.

As before, the goal is to divide this matrix into submatrices that are again L-shaped as in Eq. (3.1). A possible decomposition is the following (where blocks are re-sorted, e.g., D_5 is now on the left to get the L-shaped structure).

$$\hat{D}_1 = \left(\begin{array}{c|c|c} D_5 & D_1 & D_2 \\ \hline & & D_3 \\ \hline & & D_4 \end{array}\right), \quad \hat{D}_2 = \left(\begin{array}{c|c} D_2 & \\ \hline D_3 & \\ \hline D_4 & D_6 \end{array}\right), \quad \hat{D}_3 = \left(\begin{array}{c|c} D_4 & D_6 \\ \hline & D_7 \end{array}\right).$$

The analysis of each block and the combination of the SFS is the same as before, with D_2, D_3 and D_4 as shared blocks. This procedure is also used in chapter 5 in the context of subsystem analysis.

3.7. When is it useful to apply the proposed approaches?

The analysis of incomplete data sets can be quite complex (especially for case 2). Thus, the question arises when considering an incomplete data set is superior to simply ignoring entire rows or columns if they contain missing entries. This is investigated in this section.

3.7.1. The role of essential information

As described in Sec. 2.3.7 only rows and columns that are essential affect the inner and outer polytopes and thus only these affect the SFS. This knowledge is applied to adding rows and columns to a matrix to study changes in the SFS.

The following considerations are done with a case differentiation regarding the rank of a given data set $D \in \mathbb{R}_+^{k\times n}$ as in Eq. (3.1). Again the cases 1 and 2 are considered.

Considering $\operatorname{rank}(D) = \operatorname{rank}(D_{11})$

Using Rem. 3.7 when extending D_{11} by D_{12}, only the essential columns can change. The same holds for an extension by D_{21} and the essential rows.

Thus, if the essential columns of D_{11} and (D_{11}, D_{12}) are the same, then the SFS of D_{11} and (D_{11}, D_{12}) is the same, since the polytopes of the low-dimensional representation do not change. The same holds for the transposed case when D_{21} is considered. Thus, if neither D_{12} nor D_{21} contributes new essential information, then the SFS of D and D_{11} is the same (using Thm. 3.17 and Rem. 3.11 to extend the factors of D_{11} to D).

But even if only one of the blocks D_{12} and D_{21} contributes essential information, it is not necessary to look at an incomplete matrix. It is sufficient to look at (D_{11}, D_{12}) or $\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$ and thus compute the SFS for a complete data set.

However, the case of $\operatorname{rank}(D) > \operatorname{rank}(D_{11})$ is more complex

Considering $\operatorname{rank}(D) > \operatorname{rank}(D_{11})$

Unlike the previous case, adding rows or columns that result in a higher rank has a direct effect on the essential information.

Lemma 3.27. *Expanding $D_{11} \in \mathbb{R}_+^{k_1\times n}$ row-wise by $D_{21} \in \mathbb{R}_+^{k_2\times n}$ with $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) < \operatorname{rank}\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right) = \operatorname{rank}_+\left(\begin{smallmatrix} D_{11} \\ D_{21} \end{smallmatrix}\right)$, inevitably results in new essential rows and columns.*

Proof. By Lem. 2.10 a matrix and its to essential information reduced version are considered as equivalent. This also means, that both have the same rank, since they are based on the same factors C and S^T. Since $\operatorname{rank}(D_{11}) < \operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ holds, the same relation must hold between both matrices when they are reduced to their essential information. So their essential information cannot be the same. Since increasing the rank means increasing the dimension of both the row space and of the column space, $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ contains both, new essential rows and columns. □

However, considering only D'_{21} from Thm. 3.24, where $\operatorname{rank}(D_{11}) = \operatorname{rank}\begin{pmatrix} D_{11} \\ D'_{21} \end{pmatrix}$ again leads to case 1 and thus affects the SFS only if D'_{21} contains additional essential information. But as in Sec. 3.5 it is also possible to include the knowledge of the SFS and thus the intersection of the SFS with $\mathcal{H}$ when reducing case 2 to case 1. This additional knowledge about the ambiguity does not necessarily lead to a smaller SFS for the shared low-dimensional representation and thus for the species of D_{11} or even for the species of D_{21}. However, unlike for case 1, this cannot be distinguished by determining the essential information, but depends solely on the shape of the SFS.

The result for case 2 is that there is no gain in trying to exclude a block. Even if there is no gain SFS-wise for the species of D_{11}, no general statements can be made. Also, because additional blocks that increase the rank always provide new essential information, excluding blocks is not advisable.

In addition to these considerations, it is possible to check how large the impact of the additional essential information is, and thus whether it is worthwhile to include it or not.

3.7.2. Determining the importance of essential information

Rather than simply determining whether certain rows and columns are essential, it is possible to take a closer look at how much impact each of these rows and columns has. The goal is to include only the most important essential ones to simplify the analysis.

A classification of essential information has been done by Zade et al. in [88]. There, the data point importance (DPI) is determined for each data point, where not only data points corresponding to essential rows or columns are evaluated, but all data points. The idea is that if a row or column is less important than others, it can be neglected without losing much information. This is desirable, for example, when dealing with incomplete data sets, to determine whether it is useful to include blocks such as D_{12} and D_{21} and thus analyze the data set as an incomplete data set or not. But also for data reduction, when reduction to essential information is not sufficient, while "keeping the structure of the data points in the abstract sample space" [88].

The DPI is calculated based on the hypervolumes of the inner polytopes. The hypervolume of the inner polytope $\mathcal{I}_S$ is denoted by V_0. Then one data point is removed and the hypervolume of the convex hull of the remaining data points is recomputed, denoted by V_1. Both hypervolumes are compared by computing the ratio

$$\mathrm{DPI} = \frac{\mathsf{V}_0 - \mathsf{V}_1}{\mathsf{V}_0}. \tag{3.8}$$

This determines the DPI for each row and column of the data matrix. Obviously, non-essential rows and columns have a DPI of zero (since they are inside the inner polytope), and only essential rows get a value greater than zero, indicating how important they are, based on the ratio between the hypervolumes. The idea is that rows with a small DPI, below a chosen threshold, can be neglected since they contribute only a small amount of information. This is

a fast way to determine the influence of rows and columns, since only the convex hull of the data points needs to be computed, and no SFS computation is required. However, if the data points are all close together, or if there are clusters of points, then they all have a small DPI, and neglecting a whole cluster of data points will result in structural loss, see [88]. During this reduction, it is also important that the rank of the data set is maintained.

However, when focusing on incomplete data sets, the focus is on how the solutions behave rather than on the inner polytope, since the goal is to retrieve minimal ambiguity. Thus, a change of $\mathcal{I}_S$ may not be sufficient to determine the importance of a row. This is due to the fact that two additional rows may have the same DPI, but a significantly different influence on the ambiguity of the feasible solutions. This is shown in Fig. 3.14, where only one of the additional data points leads to a change of the SFS.

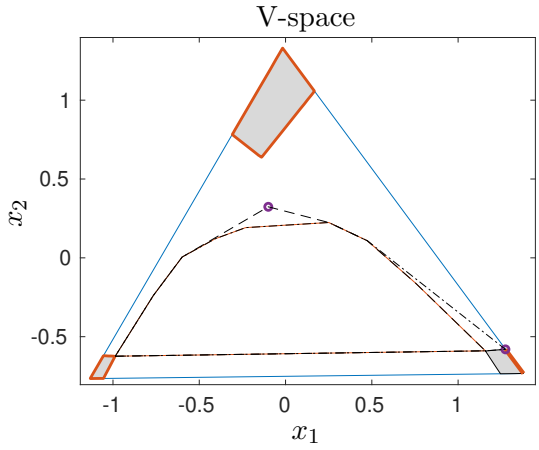

Figure 3.14.: V-space representation of a simulated data set. Two data points are added, marked by purple circles, and the inner polygon containing each is shown as a dashed line for one point and as a dot-and-dash line for the other. Both points have the same DPI (-0.0317, since the polygon is enlarged). However, the upper point does not change the SFS (gray areas), unlike the second point. The resulting reduced SFS segments for the inclusion of the second point are outlined in red, with one segment even being reduced to a line.

This leads to an alternative way to determine the importance of essential information. Namely, to compare the ambiguity of the solutions. This can be done by comparing the volume of the SFS or by comparing the corresponding bands. This has been done in [74], using both the volume of the SFS and the band integrals to measure the ambiguity. For data sets with $\operatorname{rank}(D) = s \geq 4$ this can be time consuming.

In the context of DPI and determining the importance of essential information, whether or not the SFS is considered, one important fact remains to be addressed. The DPI of data points depends on the selected SVD. This is shown in Fig. 3.15 with the DPI values in table 3.1. Here, the low-dimensional representation, and thus the SVD, is based on all data points, as opposed to only the essential ones. Both representations lead to very different DPI values and also to different volumes when comparing the SFS. However, the resulting solutions are the same because they are independent of the low-dimensional representation.

In conclusion, there is no approach without flaws. But a possible combination of several approaches, e.g., DPI and the volume of the SFS, could lead to decent results regarding the importance of essential information. Thus, additional tools are provided to identify when it makes sense to analyze data sets with missing entries as incomplete data sets or not.

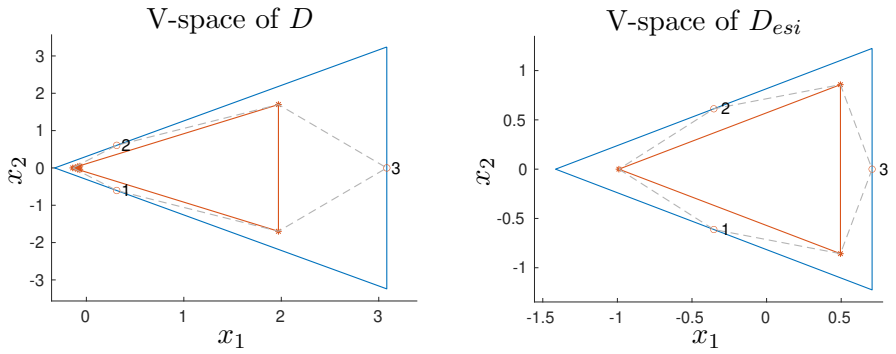

Figure 3.15.: Simulated data set D with its data points marked with red asterisks. Three added data points are marked with red circles and are numbered. The inner polygon containing them is dashed in gray. This also shows the effect of each added point and is thus a visualization of the DPI. Left: V-space of D with 3 data points at the boundary of the inner polygon which are essential and 5 data points inside near a vertex. Right: V-space of D_{esi} with the same data points, showing different DPI when using a different SVD. This is also shown in table 3.1.

	data point 1	data point 2	data point 3
reference is $\mathcal{I}_S(D)$	-0.0712	-0.0712	-0.5268
reference is $\mathcal{I}_S(D_{esi})$	-0.1429	-0.1429	-0.1429

Table 3.1.: DPI of the data points shown in Fig. 3.15 using D and D_{esi} as reference. Since all points are outside the original inner polytope, they have a negative DPI.

3.8. The influence of noise on incomplete data sets

To apply the methods presented in this chapter to chemical data, the influence of noise must be considered. Therefore, it is examined what difficulties arise in the presence of noise. Since the crucial part of the proposed method is the shared block D_{11}, it is first examined how noise affects this block and the resulting factors.

3.8.1. Choosing the shared block D_{11}

If there are several ways to choose the shared block D_{11} from Eq. (3.1) and (3.2), the question arises which is the best way. In the noise-free case, they are all equal. In the presence of noise, there may be "good" and "bad" choices. These are examined below.

When dealing with noisy data sets, the signal-to-noise ratio (SNR) of the data set, and thus of each submatrix, is crucial. When there are several D_{11} options to choose from, the one with the highest SNR is the preferred choice. For a data set with homoscedastic noise, this means that the submatrix with the highest signals should be chosen. Additionally, the rank of D_{11} has to be considered, which is denoted by $\operatorname{rank}(D_{11}) = \operatorname{rank}_+(D_{11}) = s_1$. As discussed in Sec. 2.4, the chemical rank can be determined using the singular values. Here, the distance of the s_1-th singular value from the (s_1+1)-th must be sufficiently large. The term "sufficiently large" depends on the data set. However, the larger the distance, the higher the SNR of all underlying s_1 species, especially those with a small amount in the submatrix.

In addition to SNR, the size of D_{11} is also important. As noted in Rem. 3.13, a shared block of size $s_1 \times s_1$ is sufficient when considering case 1. However, for noisy data sets, this means that the SVD has no noise filtering effect. This is because an approximation with s_1 singular

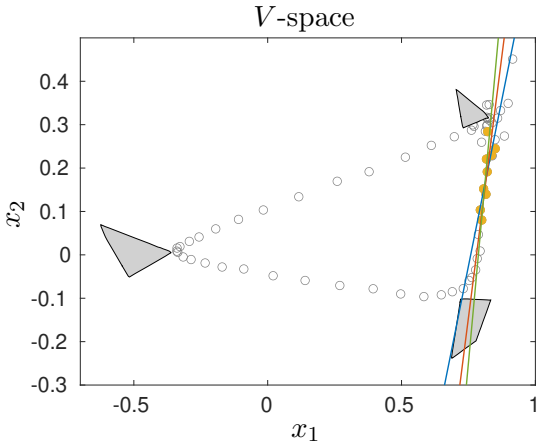

Figure 3.16.: The V-space of the first 80 columns of data set 2 with 1% standard distributed noise. The first 9 rows form the block D_{11} and have rank 2. The data points are marked by gray circles and the SFS by gray areas. The yellow circles indicate the 9 data points belonging to D_{11}. The affine subspace spanned by D_{11} is represented by lines based on different approximations. The blue line is obtained by linear curve fitting of the underlying data points. The red line is computed by approximating D_{11} with an SVD with $s = 2$ and transferring the results to the V-space. The green line is the transferred affine subspace of the original noise-free D_{11}. None of these lines intersect the upper SFS segment, and their intersection with the lower SFS segment is also different.

vectors and values represents D_{11} exactly. Thus, the noise is also represented in the singular vectors and values and distorts the shared low-dimensional representation, even with a high SNR. Consequently, additional rows and columns must be considered to limit the influence of noise in the SVD approximation. When in doubt, it is recommended to choose D_{11} as large as possible, but under the previous rank constraint. That is, maximize the distance of the smallest singular value belonging to a species from the noise level.

The procedures described in Sec. 3.4 and 3.5 generally work for a shared block. However, when considering case 2, the choice of D_{11} becomes even more crucial. This is examined next.

Choosing D_{11} for case 2

Besides the general notes on SNR and rank, there are additional effects to consider when considering case 2. The difficulties are due to the fact that not all species are present in D_{11}. Therefore, the focus is on Eq. (3.2), where $\operatorname{rank}(D_{11}) < \operatorname{rank}\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is assumed.

As described in Sec. 3.5, the intersection of the affine subspace spanned by D_{11} and the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ in the corresponding low-dimensional representation must be considered (same for the transposed case when adding D_{12}). If noise is present, the s_1 dimensional affine subspace spanned by D_{11} can be approximated. However, if the SNR is too small, the approximation of the affine subspace may differ from the true solution and the affine subspace spanned by it, see Fig. 3.16. Thus, if the intersection of the affine subspace with the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is computed, the intersection may be empty due to the distortion. This means that no solution to the factorization problem can be found under the constraint that D_{11} has s_1 species. This is in general a contradiction to the assumptions that require the existence of a factorization.

If it is known that a solution to the NMF problem exists for a given data set, then the rank

condition on the shared block can be relaxed by finding a solution that is close to the affine subspace. Thus, solutions for D_{11} with more underlying species are allowed, where the additional species are present with a small amount. Varying the affine subspace to allow intersection with the SFS subsets is also possible, but makes direct assumptions about the affine subspace of the species of D_{11}, which are unknown.

When considering case 2 in the context of subsystem analysis, there is another way to handle this problem. Here, the entire matrix is known, and only in a subsequent step is it reduced to an incomplete data set to allow subsystem analysis. One possible solution is a prior nonnegative factorization of the data matrix, with a local rank condition on the submatrices, e.g., the approximated D_{11} should have rank s_1. This ensures a nonnegative matrix factorization that takes into account the different submatrices. Thus, the approximation of the shared block and the corresponding affine subspace is less affected by noise, and the intersection of the subspace of D_{11} and the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is non-empty. An application of this approach can be found in [8].

The selection of the shared block is important not only when considering case 2, but also when the focus shifts to the reconstruction of the missing block D_{22}, see Thm. 3.14. Although the main result on the selection of D_{11} was discussed at the beginning of this section, the reason for it is different and is discussed next.

3.8.2. Reconstructing D_{22} with noise

As shown in Thm. 3.14, the missing block D_{22} can be reconstructed in the noise-free case for $\operatorname{rank}(D_{11}) = \operatorname{rank}(D) = \operatorname{rank}_+(D)$ (case 1). However, the question arises whether this still holds for noisy data sets. The following considerations assume that the entire data set is affected by the same level of noise. Since the reconstruction of D_{22} is based on the assumption that case 1 is considered, it holds in general that

$$D_{21}(i,:) = \sum_{j=1}^{k_1} \lambda_j D_{11}(j,:), \quad \text{with } \lambda_j \in \mathbb{R}, \text{for all } i = 1, \ldots, k_2. \tag{3.9}$$

So the missing block D_{22} is reconstructed with the same coefficients using the rows of D_{12}

$$D_{22}(i,:) = \sum_{j=1}^{k_1} \lambda_j D_{12}(j,:), \quad \text{for all } i = 1, \ldots, k_2. \tag{3.10}$$

The same construction can be done column-wise using D_{12} instead of D_{21} in Eq. (3.9), which results in the transposed case. For simplicity, only the row-wise construction described in Eq. (3.9) and (3.10) is considered.

When considering noise, Eq. (3.9) and (3.10) hold only approximately, since $D_{11} = C_1 S_1^T + E_{11}$ with the assumption that $E_{11} \in \mathbb{R}^{k_1 \times n_1}$ contains normal distributed noise. The same holds for $D_{12} = C_1 S_2^T + E_{12}$ with $E_{12} \in \mathbb{R}^{k_1 \times n_2}$. Thus Eq. (3.10) is reformulated to

$$D_{22}(i,:) = \sum_{j=1}^{k_1} \lambda_j C_1(j,:) S_2^T + \sum_{j=1}^{k_1} \lambda_j E_{12}(j,:), \quad \text{for all } i = 1, \ldots, k_2 \text{ with } \lambda_i \text{ from Eq. (3.9)}.$$

To identify the influence of noise, it is necessary to look closely at the coefficients λ_j. There are two cases regarding the coefficients λ_j:

(I) $\sum_{j=1}^{k_1} |\lambda_j| \leq 1$

(II) $\lambda_j \in \mathbb{R}$ arbitrary for $j = 1, \ldots, k_1$, so that case (I) is not fulfilled

For case (I), the influence of the noise on D_{22} is limited. Since E_{12} is assumed to contain normally distributed noise, negative coefficients are also allowed. Limiting the sum in case (I) means that the influence of E_{12} is limited. Thus, the noise level of D_{22} is limited by the noise level of D_{12}.

When considering case (II), this limitation no longer applies. Thus, the noise level of D_{22} is not limited and the noise is amplified compared to the remaining blocks.

An example is shown in Fig. 3.17. Here data set 2 is considered with 1% noise added. By selecting $D_{11} = D(1:17, 1:80)$ as shared block, it is ensured that D_{11} has 3 underlying species. This choice allows a classification as case 1. So D_{22} is reconstructed. The resulting reconstructed matrix is shown in Fig. 3.17 (left) together with the original noisy data set (right).

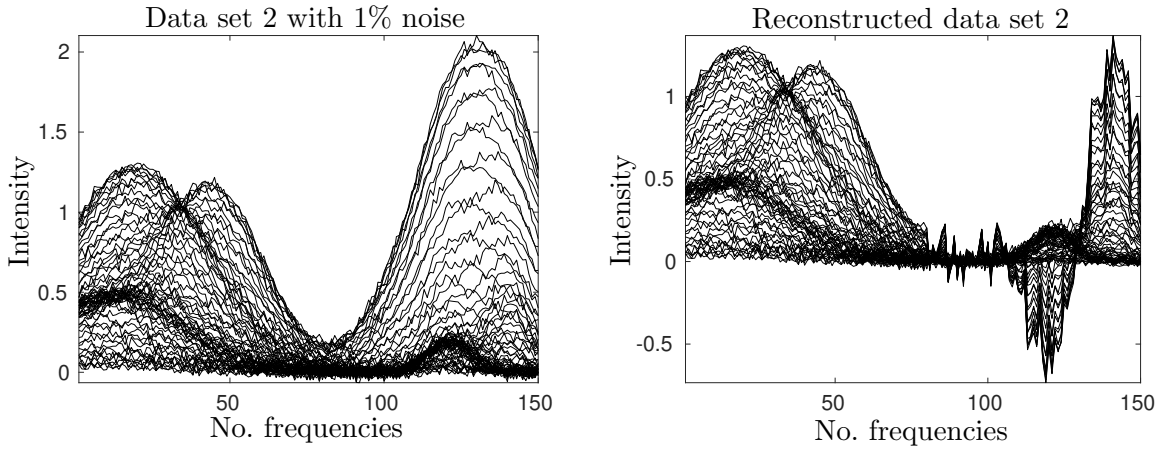

Figure 3.17.: Left: The data matrix of data set 2 with 1% added, homoscedastic noise drawn from a standard normal distribution. It is denoted by $D_{noise} \in \mathbb{R}^{66\times150}$. Right: Reconstructed data matrix based on blocks selected from the noisy data set $D_{11} = D_{noise}(1:17, 1:80)$, $D_{12} = D_{noise}(1:17, 81:150)$ and $D_{21} = D_{noise}(18:66, 1:80)$ with a reconstructed block $D_{22} \in \mathbb{R}^{49\times70}$ based on Thm. 3.14. The amplified noise can be seen from index 81 to 150.

In this construction it does not matter whether D_{22} contains essential information or not, because conic combinations of rows or columns also lead to case (II).

Since a reconstruction of D_{22} is not reliably possible without potentially amplified noise, the shared low-dimensional representation must be used to compute all solutions of the incomplete data set in the presence of noise.

3.8.3. How to handle noise for case 1

As described in Sec. 2.4, the analysis of a complete noisy data set is done by increasing the outer polytopes in the low-dimensional representation and decreasing the inner polytopes due to duality. Thus, small negative entries in the factors C and S^T are allowed, denoted by the tolerances $-\varepsilon_C$ and $-\varepsilon_S$, which are combined into a tolerance $-\varepsilon$ in this section, see problem 2.27.

When rows and columns are added to a noisy data matrix, the data points cannot be added to the V- or U-space as easily as described in Sec. 3.4. Thus, duality is used to compute the corresponding constraints on the outer polytopes in the shared low-dimensional representation. As in the complete case, the idea is to relax these constraints. However, the shared low-dimensional

representation represents only C_1 and S_1^T, while the additional constraints belong to the additional rows and columns and thus to C_2 and S_2^T. Therefore, these constraints cannot be directly relaxed in the shared low-dimensional representation.

To solve this problem, it is possible to look at the low-dimensional representation of (D_{11}, D_{12}) and compute the outer polytope $\mathcal{F}_S(D_{11}, D_{12})$ with respect to a chosen tolerance ε. The profiles representing the vertices of $\mathcal{F}_S(D_{11}, D_{12})$ are then projected to the shared low-dimensional representation by projecting only the first n_1 entries of the corresponding profiles. The same is done for $\binom{D_{11}}{D_{21}}$ and $\mathcal{F}_C\binom{D_{11}}{D_{21}}$, where the representing profiles, or rather the first k_1 entries, are used to determine $\mathcal{F}_C$ in the shared low-dimensional representation. The last step is to use duality to determine $\mathcal{I}_S$. It is even possible to use different tolerances for S_1^T and S_2^T with this approach. For example, if different measurement techniques are used that result in different levels of noise.

The computation of the outer polytopes such as $\mathcal{F}_S(D_{11}, D_{12})$ can be done using common methods such as PIA or ray casting, see Sec. 2.3.8. An alternative approach to determine the polytopes and the SFS of the shared low-dimensional representation is presented in the next chapter.

3.8.4. How to handle noise for case 2

The main problems of case 2 were already addressed at the beginning of this section on selecting the shared block. Otherwise, this case can be handled in the same way as case 1 above. For example, (D_{11}, D_{12}) is used to compute the low-dimensional representation with respect to the selected tolerances. Using the rank constraints imposed by D_{11}, the SFS of (D_{11}, D_{12}) is computed. Since the affine subspace spanned by D_{11} is disturbed by noise, the same procedure described at the beginning of this section must be applied. For example, finding a solution that is sufficiently close to the affine subspace, but whose representing points do not lie on the affine subspace.

When considering different noise levels between D_{11} and D_{12}, Lem. 3.23 no longer holds. So for $\mathcal{F}_C(D_{11}, D_{12})$ the tolerance of D_{12} is used and additionally the outer polytope of D_{11} is mapped to the subspace with the corresponding tolerance. For the selection of a solution, this means that the profiles corresponding to data points on or near the affine subspace of D_{11} should satisfy the softened constraints belonging to D_{11}.

How these considerations are implemented is shown in the next chapter.

3.9. Critical summary of the proposed approaches

The focus of this chapter is to further develop the methods for solving the NMF problem of chapter 2 and to make them applicable to incomplete data sets. The proposed shared low-dimensional representation (Def. 3.18) allows to determine the ambiguity of incomplete data sets and thus to analyze them. It is introduced for the special case that all species are present in the shared block (Def. 3.1), which is summarized as case 1. The main result and the proof why such a shared low-dimensional representation works are given in Thm. 3.12 and 3.17.

The introduced approach is generalized to be applicable when not all species are present in the shared block (this is case 2). The main difference to case 1 is that case 2 data sets may suffer from information loss compared to looking at the corresponding complete matrix. With these approaches, not only data sets with a single missing block can be analyzed, but also data sets with multiple missing blocks. The previous approaches are generalized for such structures, see Sec. 3.6.

Although it is useful to analyze incomplete data sets, it is not always recommended, as discussed in Sec. 3.7. If there is little or no information gain from adding blocks to a data set that result in an incomplete matrix structure, then it may be beneficial to not include these blocks and analyze the data set as a complete one. There are also problems when an incomplete data set is affected by noise, especially when the SNR is low. But even if the shared block is not ideally chosen, the analysis faces difficulties, as discussed in Sec. 3.8.

4. The modified ray casting method

In Sec. 2.3.8 the general ray casting method was briefly introduced. It is now used to access the SFS of the shared low-dimensional representation of an incomplete data set as described in Sec. 3.4. Therefore, the modified inner and outer polytopes are used. To account for these modified polytopes, the ray casting method must be modified. Therefore, the ray casting method is first introduced in more detail. Then, the modifications necessary to account for the modified inner and outer polytopes of the shared low-dimensional representation are described. This is divided into a description of the modifications for cases 1 and 2 as well as the effects of noise.

For simplicity, only the V-space and thus $\mathcal{F}_S$ and $\mathcal{I}_S$ are considered. The calculations in the U-space work analogously. As an alternative to the ray casting method, it is possible to use a geometric approach for $s = 3$, see [6], if the inner and outer polytopes are selected appropriately.

4.1. The general ray casting method

The general ray casting method [72] allows to approximate the boundary of $\mathcal{M}_S$ of Eq. (2.9). It is applicable for arbitrary s and thus in $s-1$ dimensions. As described in chapter 2, equiangular rays are used, starting at the origin and their intersection with $\mathcal{F}_S$ and $\mathcal{M}_S$ is determined.

For $s = 3$ these rays are given, using spherical coordinates, see [72], by

$$r_i = \begin{pmatrix} \cos\phi_i \\ \sin\phi_i \end{pmatrix}, \qquad \phi_i = 2\pi\frac{i-1}{m}, \qquad i = 1, \ldots, m.$$

The total number of rays is denoted by m and the angle of the current ray r_i is denoted by ϕ_i.

For each ray, the bisection method is used to determine the boundary of $\mathcal{F}_S$, starting with one point outside and one point inside $\mathcal{F}_S$ (e.g., the origin). In a subsequent step, the bisection method is used again to determine the part of the boundary of $\mathcal{M}_\mathcal{S}$ that does not intersect $\mathcal{F}_S$, the so-called inner boundary of $\mathcal{M}_\mathcal{S}$. The starting point is the intersection of the current ray with $\mathcal{F}_S$, if it marks a feasible point, and the origin. This step uses the gap-free intersection property described in Thm. 3.3 in [72], which means that if there is a feasible point on a ray, then the line segment formed by that point and the intersection with $\mathcal{F}_S$ (of that ray) is also feasible. The result is an approximation of the SFS $\mathcal{M}_S$.

The generalized version for arbitrary $s > 3$ works in the same way. The rays r_i are defined element-wise by

$$\begin{aligned} r_{i,1} &= \cos\phi_{i,1}, \\ r_{i,2} &= \sin\phi_{i,1}\cos\phi_{i,2}, \\ &\vdots \\ r_{i,s-2} &= \sin\phi_{i,1}\cdot\ldots\cdot\sin\phi_{i,s-3}\cos\phi_{i,s-2}, \\ r_{i,s-1} &= \sin\phi_{i,1}\cdot\ldots\cdot\sin\phi_{i,s-3}\sin\phi_{i,s-2}. \end{aligned}$$

The angles $\phi_{i,j}$ are given by $\phi_{i,j} = \pi\frac{i-1}{\ell_j}$, for $i = 1, \ldots, \ell_j$, $j = 1, \ldots, s-3$ and the last angle is given by $\phi_{i,s-2} = 2\pi\frac{i-1}{\ell_{s-2}}$, for $i = 1, \ldots, \ell_{s-2}$. The variable ℓ_j indicates how fine the discretization

is. The resulting total number of rays is given by $m = \ell_1 \cdot \ell_2 \cdot \ldots \cdot \ell_{s-2}$. For evenly distributed rays ℓ_j must be chosen as $\ell_{s-2} = \ell \cdot 2$ and $\ell_j = \ell$, $j = 1, \ldots, s-3$, with $\ell \in \mathbb{N}$.

Using these rays, the generalized method for arbitrary s can be summarized by the following steps for evenly distributed rays (analogous to $s = 3$). The following parameters are needed:

$$\begin{aligned} m &\ \ldots \text{ total number of rays} \\ \varepsilon_C \geq 0 &\ \ldots \text{ amount of permitted negativity of } C \text{ } (C \text{ is normalized}) \\ \varepsilon_S \geq 0 &\ \ldots \text{ amount of permitted negativity of } S^T \text{ } (S^T \text{ is normalized}) \\ \varepsilon_b > 0 &\ \ldots \text{ accuracy of the boundary approximations (numerical accuracy)} \end{aligned}$$

The procedure for each ray r_i, $i = 1, \ldots, m$, is as follows:

1. Find a point that is on r_i and outside of $\mathcal{F}_S$.
2. Bisection method to find the point on the boundary of $\mathcal{F}_S$:
   - Start with the interval between the origin and the found point outside $\mathcal{F}_S$ on the ray.
   - Choose a new interval by changing a boundary point so that one endpoint is inside and the other is outside $\mathcal{F}_S$.
   - Finish if the current interval is smaller than ε_b and select the inner point as the boundary point of $\mathcal{F}_S$.

If a point is outside $\mathcal{F}_S$, then the corresponding scaled profile has a minimum less than $-\varepsilon_S$.

The result of this procedure is the outer boundary of $\mathcal{F}_S$. To determine the inner boundary of $\mathcal{M}_S$ and thus the SFS, the following steps are performed for each ray.[1]

3. Determine if the current boundary point of $\mathcal{F}_S$ belonging to r_i is feasible.[2] If it is not feasible, continue with the next ray, otherwise continue with the next step.
4. Bisection algorithm to find the point of the inner boundary of the SFS on the current ray:
   - Start with the interval between the outer boundary point and the origin.
   - Choose a new interval by changing a boundary point so that one endpoint is feasible and the other is not.
   - Finish if the current interval is smaller than ε_b.

To determine whether a point is feasible or not, an optimization problem must be solved. The second to last rows of T from Def. 2.15 (stored in W) are searched for such that the normalized C and S^T are greater than $-\varepsilon_C$ and $-\varepsilon_S$, respectively, with a regular matrix T. In the case of noise-free data, these tolerances are set close to zero. This ensures that the corresponding simplex lies within $\mathcal{F}_S$ and encloses $\mathcal{I}_S$. If no suitable T can be found, the corresponding point on the ray is considered not feasible. By adjusting the tolerances ε_C and ε_S it is possible to apply this method to noisy data sets.

Since the SFS is determined by solving optimization problems that are highly dependent on a good initialization of T, points may be misclassified. To compensate for this, it is possible to use post-iterations, e.g., by initializing the optimization with feasible solutions of neighboring points. Possible approaches for this post-iteration are described in [72] and in detail for $s = 4$ in [33]. The advantages and disadvantages of the ray casting method (and how to deal with them) are examined in [72].

[1] It is advantageous to have the outer boundary of $\mathcal{F}_S$ to select initial values for the optimization and to find a solution, instead of determining $\mathcal{F}_S$ and $\mathcal{M}_S$ in one step.

[2] Feasibility of a point means that it belongs to a solution.

4.2. The ray casting method for the shared low-dimensional representation

Considering the shared low-dimensional representation of Sec. 3.4, additional constraints corresponding to additional blocks affect the low-dimensional representation of D_{11} from Eq. (3.1). These constraints are represented by the matrices D_{12} and D_{21} from Eq. (3.1). To take these constraints into account, the general ray casting method must be modified. This is done in two steps. First by looking at D_{12} and calculating the corresponding new $\mathcal{F}_S$, and second by taking D_{21} into account to determine the SFS. These two steps are described in detail next. As in the previous general ray casting method, only the V-space is considered. The dual case in the U-space is analogous.

4.2.1. Considering the additional block D_{12}

Adding D_{12} to the low-dimensional representation of the shared block means that $\mathcal{F}_S$ is modified by additional constraints, see Fig. 3.4. These additional constraints correspond to the nonnegativity of $S_2^T \in \mathbb{R}^{s\times n_2}$, with $S^T = (S_1^T, S_2^T) \in \mathbb{R}^{s\times n}$ of Eq. (3.1). If only a solution of the NMF problem of D_{11} is given, the submatrix S_2^T of the matrix is calculated using Rem. 3.11. This even holds row-wise for S^T when the profile corresponding to a point in the shared low-dimensional representation is considered.

Remark 4.1. *Let $x \in \mathbb{R}^{s-1}$ be an arbitrary point in the shared low-dimensional representation with $t_1 = (1, x^T)^T$. It is possible to determine the corresponding profile $\widehat{S}^T = (\widehat{S}_1^T, \widehat{S}_2^T)$ to t_1. This is done using the SVD of $D_{11} = U_{11}\Sigma_{11}V_{11}^T \in \mathbb{R}^{k_1\times n_1}$ of Eq. (3.1). Thus holds*

$$t_1^T V_{11}^T = \widehat{S}_1^T.$$

Using the SVD of $(D_{11}, D_{12}) = U_1\Sigma_1 V_1^T$ the following (overdetermined) system of equations is solved for t_2, which contains the data point of $\widehat{S}^T$ in the low-dimensional representation of (D_{11}, D_{12})

$$t_2^T V_1^T = (\widehat{S}_1^T, \widehat{S}_2^T) \;\Rightarrow\; t_2^T (V_1(1:n_1,:))^T = \widehat{S}_1^T.$$

The resulting $t_2 \in \mathbb{R}^s$ is unique, because $\operatorname{rank}(V_1) = \operatorname{rank}(V_{11}) = s$, since case 1 is considered. With t_2 the corresponding profile $\widehat{S}_2$ is calculated by $t_2^T(V_1(n_1+1:n,:))^T = \widehat{S}_2^T$. Thus, the profile $\widehat{S}^T = (\widehat{S}_1^T, \widehat{S}_2^T)$ corresponding to the data point x is determined.

With this remark it is now possible to check the nonnegativity of $\widehat{S}_2^T$ when a point is given in the shared low-dimensional representation. This also allows to check the nonnegativity of $\widehat{S}_2^T$ in the second step of the ray casting method. Thus, the outer polytope of the shared low-dimensional representation is computed. This approach allows to approximate the outer polytope also in the noisy case, see Sec. 4.4.1.

The next step is to determine the SFS of the data set considering the additional block D_{21}.

4.2.2. Considering the additional block D_{21}

If the outer polytope is already determined, the next step is to check which points along a ray are feasible to determine the SFS. Again, the general procedure is the same as in the general ray casting method in Sec. 4.1. The feasibility must be checked not only for C_1 and S_1^T, which result directly from the shared low-dimensional representation, but also for S_2^T and C_2. Both

S_2^T and C_2 are determined using Rem. 3.11 and can thus be checked for nonnegativity. With this step, the expanded inner polytope is taken into account. The result is the SFS of the shared low-dimensional representation.

4.2.3. The special case of $s = 2$

The steps described above can be simplified if $s = 2$ is considered. Not only is the number of rays fixed with $m = 2$ (because the low-dimensional representation is 1D), but also the calculation of the SFS becomes simple, since the SFS is defined by the intervals between $\mathcal{F}_S$ and $\mathcal{I}_S$. Thus, the outer polytope can be computed as before, and the inner polytope can be computed by adding the data points corresponding to D_{21}. In the presence of noise, instead of adding the data points to $\mathcal{I}_S$, the inner polytope is computed using duality (computing $\mathcal{F}_C$ instead of $\mathcal{I}_S$), as described in Sec. 2.4.

4.3. Modified ray casting method for case 2

If case 2 is considered, there is no direct shared low-dimensional representation, but it can be returned to case 1, see Thm. 3.24. The focus is on the V-space and thus on matrices like $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ from Eq. (3.2). An application to the U-space is analogous.

As introduced in Sec. 3.5, the submatrix D_{11} spans an $(s_1 - 1)$-dimensional affine subspace in the space of the low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$. This affine subspace is denoted by $\mathcal{H}$, see Eq. (3.5). To obtain the additional constraints on C_1 of D_{11}, the first step is to check which parts of D_{21} affect the polytopes of D_{11} Because of Lem. 3.23 only data points satisfy this condition. The profiles corresponding to these data points are stored in D'_{21} and allow an analysis as in case 1, see Sec. 3.5.

As noted in Sec. 3.5, even if D'_{21} is empty, the SFS of D_{11} can be affected by D_{21}. The solution is to compute the restricted SFS $\mathbf{M_R}(D_{11})$ as given in Thm. 3.25 and not only the intersection of $\mathcal{H}$ and the SFS of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$, which is a superset. A modified ray casting method is used for this. Therefore, two approaches are presented next.

4.3.1. Modified ray casting on $\mathcal{H}$ to determine the restricted SFS

Intuitively, the space of the low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ is used to apply the ray casting method. However, it is not necessary to do a ray casting of the entire space of the low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and check if the current point is on $\mathcal{H}$. Instead, it is sufficient to do the ray casting on the affine subspace $\mathcal{H}$ to determine $\mathbf{M_R}(D_{11})$. However, since the origin is an interior point of the inner polytope, see Lem. 3.5 of [67], the starting point of the ray casting method must be shifted to $\mathcal{H}$. Since the ray casting method is based on the gap-free intersection property, see Thm. 3.3 of [72], it must be verified that shifting the starting point to $\mathcal{H}$ does not affect the method and that the entire SFS can still be determined. This is proved in the following lemma.

Lemma 4.2 (Gap-free intersection property)**.** *Let $D \in \mathbb{R}_+^{k \times n}$ be given, where $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$, DD^T and D^TD are irreducible matrices. Furthermore, D has an SVD $D = U\Sigma V^T$ and $V(:,1) > 0$.*
If $d \in \mathcal{I}_S$ and $x_0 = d + r$ is an element of the inner boundary of $\mathcal{M}_S$ with $r \neq 0$, then there

exists an $x^* = d + \alpha^* r \in \partial\mathcal{F}_S$ *such that the line segment* $\{d + \alpha r : \ \alpha \in [1, \alpha^*]\}$ *is a subset of* $\mathcal{M}_\mathcal{S}$.

Proof. If $x_0 \in \mathcal{M}_\mathcal{S}$, then there exist points $y_1, \ldots, y_{s-1} \in \mathcal{M}_\mathcal{S}$ such that convhull$\{x_0, y_1, \ldots, y_{s-1}\}$ is a feasible simplex which represents a solution to the NMF problem of D (all these points are vertices of the simplex). Since $d \in \mathcal{I}_S$ holds, d is also an element of this simplex. The point $x_0 = d + r$ is a vertex of the simplex and it holds $x_0 \in [d, d + \alpha r]$ for $\alpha \geq 1$. So taking the simplex given by convhull$\{x_0, y_1, \ldots, y_{s-1}\}$ and replacing the vertex x_0 by $x = d + \alpha r$, with $\alpha \geq 1$ means that x_0 is also an element of this new simplex. This means it holds $\mathcal{I}_S \subseteq$ convhull$\{x_0, y_1, \ldots, y_{s-1}\} \subseteq$ convhull$\{x, y_1, \ldots, y_{s-1}\}$. Since $y_1, \ldots, y_{s-1} \in \mathcal{M}_\mathcal{S}$, these points are inside the outer polytope. Also, since $\alpha \in [1, \alpha^*]$, all $x = d + \alpha r \in [x_0, x^*]$ are inside the outer polytope, since $x^* \in \partial\mathcal{F}_S$, $x_0 \in \mathcal{F}_S$, and $\mathcal{F}_S$ is convex. This results in convhull$\{x, y_1, \ldots, y_{s-1}\} \subseteq \mathcal{F}_S$, i.e., the corresponding simplex represents a solution, see Thm. 2.16. Thus holds $\{d + \alpha r : \ \alpha \in [1, \alpha^*]\} \subseteq \mathcal{M}_\mathcal{S}$. □

To ensure the assumption $r \neq 0$, it is sufficient to choose d so that it does not coincide with a vertex of the inner polytope. This is because the SFS can only touch the inner polytope at its vertices. If it would intersect a point $\hat{x}$ that is not a vertex, the simplex construction would cause $\hat{x}$ to be on an edge or a higher dimensional element of the simplex to ensure that the inner polytope is enclosed. Thus, $\hat{x}$ is not a vertex of the simplex and thus not part of the SFS. This means that to do the ray casting method on $\mathcal{H}$, it is sufficient to choose a starting point d on $\mathcal{H}$ that is not a vertex. Since the inner polytope is finitely generated, there are edges, so d can be properly selected. This makes it possible to do the ray casting method on $\mathcal{H}$.

4.3.2. Modified ray casting in the low-dimensional representation of D_{11} to determine the restricted SFS

An alternative way to apply the ray casting method to $\mathcal{H}$ is to apply the method in the low-dimensional representation of D_{11} using additional constraints contained in D_{21}. This approach is similar to case 1 in Sec. 4.2. When using the ray casting method in the low-dimensional representation of D_{11}, it is necessary to not only check whether the current point on the ray leads to a solution for D_{11} but also whether additional columns for C can be found so that it also leads to a solution of the NMF problem of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$.

Again, the gap-free intersection property holds. This is because the restricted SFS $\mathbf{M_R}(D_{11})$, which is gap-free by Lem. 4.2, can be transferred to the low-dimensional representation of D_{11}. This is done by transferring the feasible line segment along each ray, using the underlying profiles. These profiles span a rank-two space and thus also represent a line segment in the low-dimensional representation of D_{11}. This line segment must be gap-free, otherwise profiles that are a convex combination of the start and end points of the line segment would not be feasible, which contradicts the gap-free intersection property of the restricted SFS.

Algorithmically, this results in a modified objective function for determining the SFS in the general ray casting method. Therefore, the current point is transferred to the space of the low-dimensional representation of $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ and checked for feasibility, with the additional constraint that s_1 points must lie on $\mathcal{H}$.

4.4. How to handle noise and other effects for the modified ray casting method

The idea behind using the ray casting method to compute the SFS of the shared low-dimensional representation is not only the simplicity of the method and its applicability to arbitrary dimensions, but also its easy modification to handle noisy data sets. For the general ray casting method, only the tolerances ε_C and ε_S of the nonnegativity need to be adjusted. This allows small negative entries in the factors C and S^T. The amount of allowed nonnegativity is chosen with respect to the normalized factors. The same can be done for the modifications of the ray casting method described in Sec. 4.2 and 4.3.

The effects of noise are not as different as described in Sec. 3.8 for incomplete data sets, so the focus is specifically on the modified ray casting method.

4.4.1. The effect of noise for case 1

Depending on whether the data set D is a fusion of several data sets (represented by the blocks D_{11}, D_{12} and D_{21} of Eq. (3.1)) or if it is an incomplete data set, different approaches are applied.

For an incomplete data set, the noise level is assumed to be homogeneous. The same tolerances are applied to all blocks and thus to C_1 and C_2 as well as S_1^T and S_2^T (from Eq. (3.1)). Since the tolerances are applied to the normalized factors, the profile $\widehat{S}$ from Rem. 4.1 is normalized and then the tolerances are applied to check if the corresponding point is in the outer polytope and if it is feasible. Thus, the ray casting method works as in the general case, but additionally the corresponding factors S_2^T and C_2 are checked for their smallest entries.

If the noise level of the different blocks is not homogeneous, e.g., if the data set is fused out of several ones, then different tolerances are needed for the factors, namely ε_{C_1}, ε_{C_2}, ε_{S_1} and ε_{S_2}. These tolerances can be applied in the same way to the factors S_2^T and C_2. This inevitably leads to an individual normalization for each profile $\widehat{S}_1^T$ and $\widehat{S}_2^T$ from Rem. 4.1, the same holds for C.

4.4.2. The effect of noise for case 2

For case 2, the same aspects as discussed in the previous section need to be considered, such as the same scaling of the submatrices and the noise level. These aspects are addressed in the same way by adjusting the objective function using tolerances.

The main difference to case 1 is the affine subspace $\mathcal{H}$ in which the ray casting method is applied. As discussed in Sec. 3.8, the main point is to properly approximate this affine subspace spanned by D_{11}. Thus, if $\mathcal{H}$ is properly approximated, the modified ray casting method can be applied as proposed with a modified objective function.

4.5. Critical summary of the modified ray casting method

The techniques presented in this chapter are a generalization of the ray casting method of [72] and allow an approximation of the SFS for incomplete data sets. This is done separately for the two cases 1 and 2 of chapter 3. For case 1, the constraints of the blocks D_{12} and D_{21} are added step by step to check the nonnegativity of the factors S_2^T and C_2. This leads to a stepwise modification of the ray casting method. For case 2, the gap-free intersection property, which ensures that the ray casting method works, is extended to an arbitrary starting point of the ray casting method (Lem. 4.2). Thus, the ray casting method is performed on the affine subspace

spanned by D_{11}. This allows the local rank constraint given by D_{11} to be directly taken into account when determining the SFS. Alternatively, the ray casting method can be performed in the space of D_{11}, taking into account the constraints of the additional blocks by modifying the objective function.

The advantage of this approach is its universal applicability to an arbitrary number of underlying species and its applicability to noisy data. The modifications to handle noisy data are reduced to modifications of the objective functions when approximating the SFS to account for the factors C_2 and S_2^T.

However, the computational complexity grows exponentially with higher dimensions. Furthermore, the approximation of the SFS is highly dependent on the optimization algorithm and its initialization (Sec. 4.1) and can thus be improved.

5. The detection and analysis of subsystems

The analysis of subsystems is especially desirable when a data matrix has a high chemical rank and is thus associated with many chemical species that make analysis difficult. This is the case, for example, for the methods of Sec. 2.3.8, which work best for $s \leq 4$. Therefore, it is desirable to divide the data matrix into smaller parts, called subsystems [66], and analyze them to examine the entire system.

But before considering subsystems, how to detect them, and how to analyze them, the question arises of what a subsystem is.

Definition 5.1 (Subsystem)**.** *A* subsystem *of a data matrix* $D \in \mathbb{R}_+^{k\times n}$*, with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*, is a union of several connected submatrices*[1] *of* D *having a chemical rank of* $s_1 < s$*.*

This term is considered separately from the term "submatrix". Submatrices are created by deleting certain rows and columns of a matrix and are therefore complete. The subsystems considered here can also be created by deleting only certain entries and not entire rows and columns, leading to incompleteness. This allows to include as much information about the selected species as possible in the subsequent subsystem analysis. Thus, the analysis of subsystems can be understood as the analysis of incomplete data sets, and the approaches of chapter 3 allow the analysis of such subsystems.

Thus, given a complete data set with the structure as in Eq. (3.1), instead of selecting (D_{11}, D_{12}) or $\begin{pmatrix} D_{11} \\ D_{21} \end{pmatrix}$ as a subsystem if the individual blocks all are associated with s_1 species, a subsystem consisting of D_{11}, D_{12} and D_{21} can now be selected.

Before subsystems can be analyzed, they must be detected. In the noise-free case, this can be done by determining the rank of the submatrices and checking whether these submatrices are connected. Since the goal is to apply the subsystem analysis to measured spectroscopic data, the case of noisy data sets is considered in this chapter. As an example, the noisy data set 2 is used as shown in Fig. 3.17 (left). However, the methods presented can also be applied to the noise-free case.

5.1. Subsystem detection

The goal is to identify subsystems associated with s_1 species. Thus, the submatrices forming the subsystem each have a maximum chemical rank of s_1, see Sec. 2.4.

Three steps must be taken to enable subsystem detection

1. determining the chemical rank,
2. identifying submatrices and their chemical rank,
3. combining the submatrices associated with the desired species into subsystems.

This section covers these steps in sequence.

[1] See Sec. 3.6 for the notion of connected submatrices or blocks.

5.1.1. Determining the chemical rank

Possible methods for determining chemical rank are presented in [46] and [43]. Only selected methods are presented here.

Scree test

The scree test, also known as the scree plot, was proposed by Cattell in 1966, see [12]. It was briefly introduced in Sec. 2.4.1. It is used to determine the size of the factor space [46]. The number of species is determined by plotting the singular values in descending order. The scree plot is usually done by a logarithmic plot of the magnitude of the singular values, see [43]. A scree plot for data set 2 is shown in Fig. 2.8. Ideally, the curve drops rapidly at a certain point and levels off. The singular values above this bend indicate the significant factors. Furthermore, this level can be defined as the noise level to identify the chemical rank of the data matrix. At high noise and thus low SNR, the smallest singular values corresponding to species are no longer distinguishable from the noise [43]. This is also visualized in Fig. 2.8, where different noise levels are used for the same data set.

Using singular vectors

Not only the singular values can be examined to determine the number of species, as in the scree test, but also the singular vectors. This is because the noise of a data set is reflected in particular by the singular vectors, which belong to singular values below the noise level and thus to the noise [43, 46]. The noise can be seen when the singular vectors oscillate wildly. This is shown in Fig. 5.1 for data set 2.

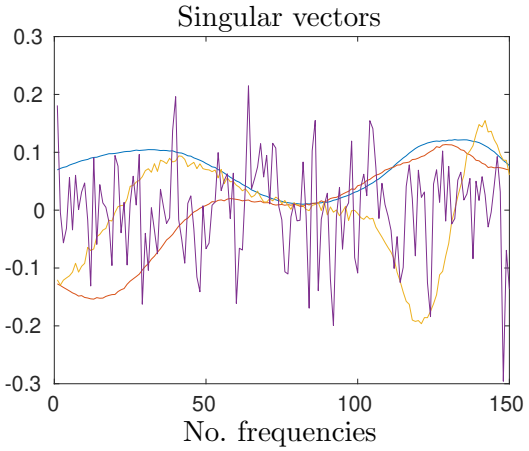

Figure 5.1.: The first 4 left singular vectors of data set 2 with 1% noise added. First in blue, second in red, third in yellow, fourth in purple. The first 3 contain signal, while the third is already a bit noisy. The fourth singular vector oscillates wildly and is therefore considered to belong to noise. Thus, the data set is associated with 3 species.

Looking at the residual matrix E

Another approach is to look at the residual matrix $E \in \mathbb{R}^{k \times n}$. It results from considering the noisy case and thus from $D = CS^T + E$ with $D \in \mathbb{R}^{k \times n}$, $C \in \mathbb{R}^{k \times s}$ and $S^T \in \mathbb{R}^{s \times n}$. The

data matrix D is approximated using different numbers of chemical species s. If the s chosen to approximate D is too small, then E will still contain structured residuals that do not correspond to noise [43]. The smallest s for which E contains only noisy residuals is taken as the chemical rank. An example is shown in Fig. 5.2.

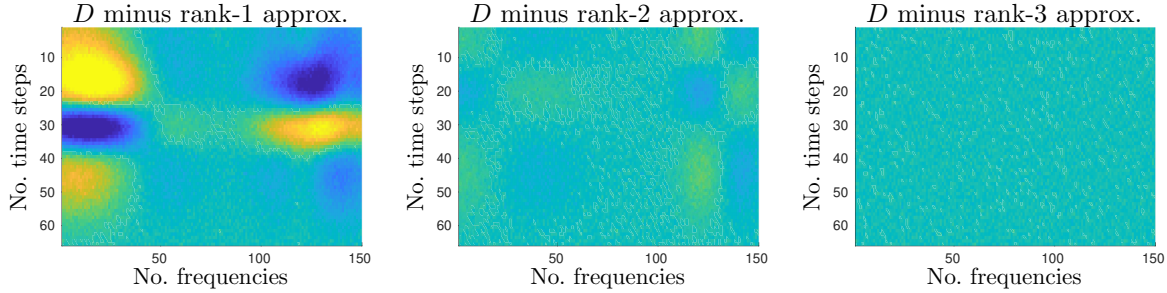

Figure 5.2.: Data set 2 with 1% noise, with different rank approximations. Shown are the differences between the original data set and a possible rank approximation. For comparability, the colormap limits are the same for all three plots, with cyan indicating values close to zero. Left: D minus a rank-1 approximation. The blue and yellow regions indicate contributions from the signals. Center: D minus a rank-2 approximation. It can be seen that parts of the signal are still contained by the blue tones. Right: D minus a rank-3 approximation. No clear signal contribution can be seen, so only noise is visible, which means that D is reasonably well approximated by a rank-3 approximation.

5.1.2. Identifying submatrices

So far, only the chemical rank of a matrix is determined. In the same way, the chemical rank of a submatrix can be determined. The goal is to identify submatrices such that their rank is smaller than that of the entire matrix. Instead of randomly selecting submatrices and determining their chemical rank, systematic approaches are presented next. The assumption of these systematic approaches is an inherent order of the data to obtain meaningful results [17, 86].

Evolving factor analysis

The method of evolving factor analysis (EFA) was first introduced by Gampp et al. in 1985 [22] and successively improved by Maeder and coworkers in [42, 44, 45]. EFA studies the evolution of the singular values of a sequence of submatrices of a given data matrix $D \in \mathbb{R}^{k \times n}$. This sequence of submatrices is given by

$$D_f[\ell] = D(1 : \ell, :) \in \mathbb{R}^{\ell \times n} \text{ for } 1 \leq \ell \leq k.$$

The corresponding singular values, in decreasing order, are denoted by $\sigma_i(D_f[\ell])$ with $i = 1, \ldots, \min\{\ell, n\}$. This notation is based on [53]. Since this sequence of submatrices starts from the first row, it is called forward EFA. A forward EFA plot is shown in Fig. 5.3 (left) for data set 2. When a new species appears at a certain index ℓ, a singular value evolves from the remaining singular values belonging to the noise. This singular value increases in value depending on its contribution to the current submatrix [46].

The same procedure can be done in reverse, starting with the last row of D, the so-called backward EFA. The submatrices are given by

$$D_b[\ell] = D(\ell : k, :) \in \mathbb{R}^{(k-\ell+1) \times n} \text{ for } 1 \leq \ell \leq k.$$

The corresponding singular values, in decreasing order, are again denoted by $\sigma_i(D_b[\ell])$, now with $i = 1, \ldots, \min\{k-\ell+1, n\}$, depending on the dimension of $D_b[\ell]$. The backward EFA plot shows the disappearance of a species. A backward EFA plot is shown in Fig. 5.3 (center).

A common practice is to combine both the forward and backward EFA into one plot to determine a rough estimate of the regions of existence or, for example, concentration windows for chemical species [35,45]. This approach assumes that the first substance to appear is the first to disappear, the second appears and disappears second, and so on. Thus, for the i-th species, regions of existence are spanned "from the point where the rank rises to i in the forward EFA calculation to the point where the rank rises to $N - i + 1$ in the backward calculation" [35], where N is the chemical rank of the entire matrix D. But even if this premise is not fulfilled, a combined visualization is advantageous. It is shown in Fig. 5.3 (right).

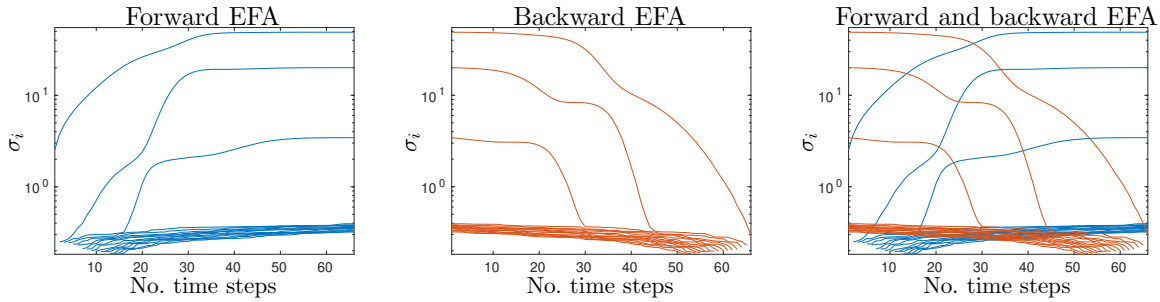

Figure 5.3.: Left: Forward EFA plot for the noisy data set 2. Center: Corresponding backward EFA. Right: Combination of both forward and backward EFA to show the appearance and disappearance of species.

This procedure can be applied not only in the row direction, but also in the column direction for D^T analogously [35]. Thus, regions of existence can also be found in the column direction. This makes it possible to find submatrices of a certain chemical rank in both row and column directions.

To interpret an EFA plot, it is necessary to take a closer look at its fundamentals and properties. Because of the interlacing property of singular values, see Cor. 8.6.3 of [28], holds

$$\sigma_i(D_f[\ell+1]) \geq \sigma_i(D_f[\ell]) \geq \sigma_{i+1}(D_f[\ell+1]), \quad \text{for } i = 1, \ldots, \min\{\ell, n\}.$$

The same can be formulated for backward EFA. This interlacing property describes the behavior of EFA curves with monotonous growth. In the special case of D consisting of only one row, which is repeated, this growth behavior can be specified. Such a construction of D describes, e.g., a chemically stationary system.

Remark 5.2. *If $D(1,:) = x^T$ for any non-zero $x \in \mathbb{R}^n$ and $D_f[\ell] = \begin{pmatrix} D_f[\ell-1] \\ x^T \end{pmatrix}$ for $\ell = 2, 3, \ldots, k$, then the growth behavior of the EFA curves is given by $\sigma_1(D_f[\ell]) = \|x\|_2\sqrt{\ell}$, see property 2.2 from [50].*

Thus, the EFA curve of the first singular value has a square root shape for the special case of such a data matrix D. The appearance of a second species can thus be detected by looking for a deviation of the square root profile of the first singular value.

If the underlying model is more complex than the described case, the EFA curves can no longer be described mathematically. This is, because "explicit mathematical representations of EFA curves require strong assumptions on the simplicity of a model problem" [50]. Nevertheless, a fine structure analysis of data sets can be performed if all rows have a fixed Euclidean norm and

are above the noise level. Such a fine structure analysis can, for example, provide information about the degree of linear independence of the underlying species, see [50].

This makes EFA a good technique for analyzing the structure and chemical rank of a data set. However, there is a drawback to this approach. This is that submatrices selected in this way always contain either the first or the last ℓ rows or columns. Another drawback is that it may be difficult to distinguish between a singular value corresponding to noise and one corresponding to a species that is only present in small amounts, see [35]. An approach that addresses these drawbacks is discussed next.

Fixed size moving window EFA

Keller and Massart proposed an approach called fixed size moving window evolving factor analysis (FSMW-EFA) in [36]. Instead of analyzing submatrices of increasing size, the size is fixed, resulting in the following submatrices being analyzed

$$D_{fsmw}[\ell, w] = D(\ell - w : \ell + w, :) \in \mathbb{R}^{(2w+1)\times n} \text{ for } w + 1 \leq \ell \leq k - w,$$

with a fixed window size parameter $w \in \{1, \ldots, \lfloor \frac{k-1}{2} \rfloor\}$, resulting in a window size of $2w+1$. The curves of the FSMW-EFA plot are thus given, similar to the original EFA plot, by $\sigma_i(D_{fsmw}[\ell, w])$ with $i = 1, \ldots \min\{2w + 1, n\}$. The window size is crucial for this analysis. The choice of this parameter depends on the data set and the local chemical rank. The window size is usually a small number, but should be at least equal to the expected number of chemical rank in the data set "to show all possible situations of rank overlap in the data set" [17]. A large window size means that the local rank information is spread out and the boundaries of the regions of existence are blurred [43].

However, if there are many underlying species with little overlap and widely spread, the window size can be oriented on the expected local rank. With a properly chosen window size, it is possible to determine the local rank of a data matrix in either row or column direction, see Fig. 5.4. Instead of a monotonous growth, the curves of the singular values increase and decrease depending on the local number of underlying species, i.e., the local rank. This makes it possible to identify regions of a certain local rank in a single plot without having to combine two methods, as was done in EFA with forward and backward analysis. However, it should be noted that FSMW-EFA is prone to a local rank deficiency [2], see Def. 3.2, which should be checked in a subsequent step, e.g., with the approach of Sec. 5.1.3.

The main advantage of FSMW-EFA over "normal" EFA is that with FSMW-EFA it is possible to detect small contributions of underlying species, even if they are not well separated from the remaining ones [36]. A combination of both methods is useful, not only to determine the local rank (using FSMW-EFA), but also to see if certain species occur multiple times, and thus in a larger region, than determined by the local rank (using EFA).

Other methods of identifying regions of interest are possible, and most are modifications of either EFA or FSMW-EFA [17]. One possible method is a modification of FSMW-EFA, as it is highly dependent on the window size chosen. Thus, an evolving size moving window EFA approach can be used, starting with a small number for the window size, e.g., 2, and increasing it until the maximum number of overlapping species is reached, see [40]. Also the exhaustive EFA (E-EFA) of [86] is a possible modification, where instead of analyzing the inside of a selected contiguous window, the outside is analyzed in each step.

Such modifications always depend on the goal and the application and are modified accordingly. For example, if the goal is to select subsystems not only along one direction, but both along rows and columns, a different method is proposed.

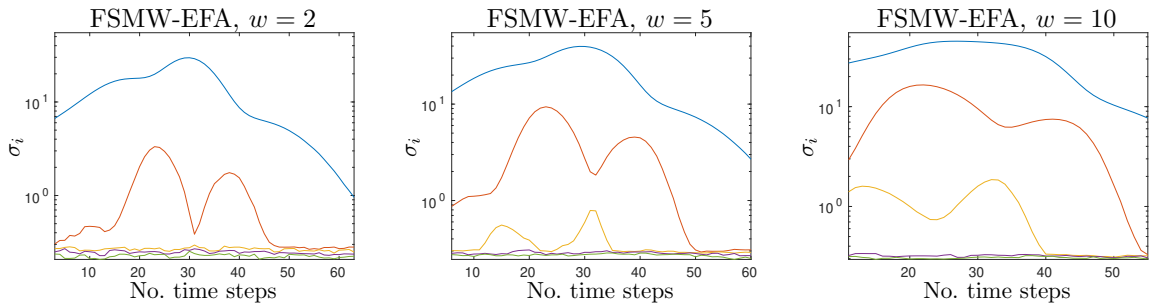

Figure 5.4.: FSMW-EFA with different window sizes w. The different window sizes result in different x-axes. Left: Only two singular values rise above the noise level, indicating a maximum local rank of 2. This contradicts the underlying noise-free profiles used to generate the data set, see Fig. A.2. Thus, the window size is too small and selected submatrices are likely to have a local rank deficiency. Center: The appearance of the third singular value can be observed, indicating submatrices of rank 3. But also submatrices of rank 1 and 2 can be observed. This is consistent with the local rank as designed by the noise-free simulated data set 2, see Fig. A.2. Right: The window size is too large, because all 3 singular values rise above the noise level for most time steps and thus no fine structure analysis is possible.

Combination of EFA methods

It is also possible to combine different methods, such as EFA and FSMW-EFA, to detect submatrices that do not span entire rows or columns. This also allows the identification of subsystems that are fused by multiple submatrices. The correct combination of submatrices is the topic of Sec. 5.1.3.

Similar approaches are already known, e.g., one proposed by Geladi and Wold in 1987 [23], before the introduction of EFA, where a moving window, e.g., of size 2-by-2, is used to scan the whole measurement. At the time, it was considered too time-consuming [40], and so modifications of EFA and FSMW-EFA were studied.

The idea here is to use FSMW-EFA, but not only to determine the singular values of each window, but to perform an EFA (either forward or backward). Thus, not only the number of species of each window is determined, but also their appearance or disappearance.

Such a combination of EFA methods can also be used to detect errors in a measurement. Especially if they cover a large area. In applications, this can be caused by sensor saturation, for example. Due to additional effects, the affected area may be larger than visible to the naked eye. Therefore, it is advantageous to select a reasonable area. An example of such a case is shown in Sec. 6.1. The explanation that such error detection works is that, under the assumption that errors describe nonlinear behavior, they lead to an increase in the rank and thus in the singular values (since SVD assumes linear relationships). This increase is detected by EFA. Another reason is that EFA methods only detect linear behavior [35].

This justification makes it clear that regular EFA can already detect affected areas, but these are entire rows or columns. The combined method of performing EFA on specific windows allows for the detection of precise areas that are affected.

Thus, to determine erroneous areas, a singular value belonging to the noise is selected and its behavior during the combination of FSMW-EFA and EFA is studied. Explicitly, the gradients of the EFA curves are examined, since large increases in the singular value and thus in the noise level are of interest. With this step it is possible to identify affected areas. An example is shown in Fig. 6.2 for an experimental data set.

Evolution of trace of the data points

A completely different approach is to look at the low-dimensional representation of a data matrix and the corresponding data points, see Eq. (2.11) and (2.13). Lem. 3.20 already described the relation between the rank of a submatrix and the dimension of the corresponding affine subspace spanned by these data points. Thus, if data points span an $(\hat{s}-1)$-dimensional affine subspace, the matrix containing the rows or columns that correspond to these data points has rank $\hat{s}$. Again, the evolutionary aspect comes into play, where data points are added step by step until they can no longer be represented by a $(\hat{s}-1)$-dimensional affine subspace. Thus, an appropriate submatrix can be found.

For noisy data sets, such a relationship is only approximate, i.e., the question is whether the data points can be approximated reasonably well by a $(\hat{s}-1)$-dimensional affine subspace. An indicator for the quality of the approximation can be the deviation of the data points from the affine subspace, or the volume of the convex hull of the data points (it must be relatively small), and also the underlying structure. If the data points oscillate around the selected affine subspace, this oscillation is most likely due to noise. This reasoning is justified by the approaches of Sec. 5.1.1. A small deviation of the data points from the affine subspace correlates with relatively small singular values $\sigma_{\hat{s}+1}$, $\sigma_{\hat{s}+2}$ (as used in the scree test). The same relation is used when looking at the volume (in at least $\hat{s}$ dimensions). Looking at the structure of the data points, oscillating data points are caused by an oscillation in the singular vectors, which belong to noise if there is an intrinsic order in the data set.

An example is shown in Fig. 5.5. As can be seen there, it can sometimes be advantageous to plot the data points of submatrices (and thus data points based on an SVD of the corresponding submatrix) instead of using the entire matrix. In this way, the influence of small amounts of an additional species can be represented, which would otherwise be lost due to smoothing based on the SVD.

However, determining submatrices of a certain rank with this procedure has a flaw. This approach is susceptible to local rank deficiency, in the same way as FSMW-EFA, see Fig. 3.3

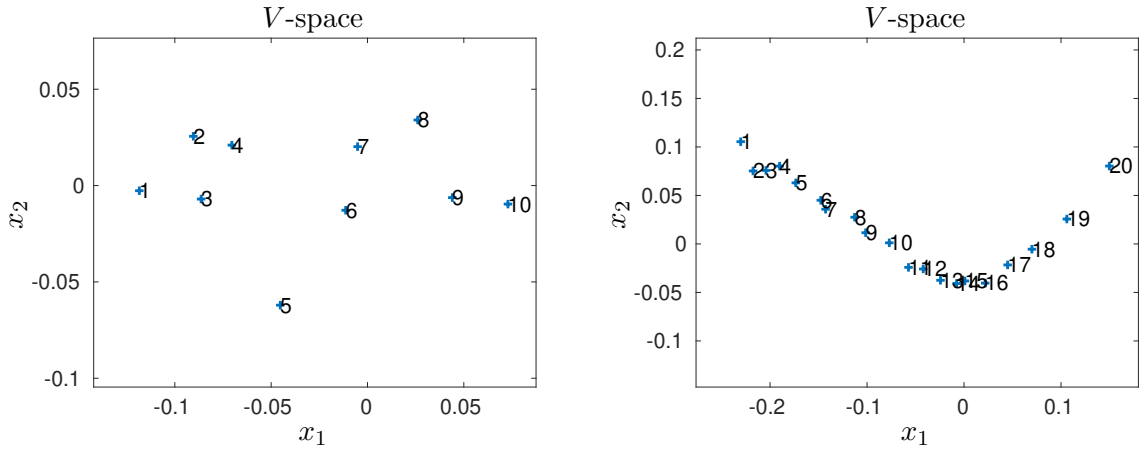

Figure 5.5.: Left: Data points in the V-space spanned by the first 10 rows of data set 2 with 1% noise added. Their oscillation in the direction of x_2 has no structure and is around the axis of $x_1 = 0$. Thus, the oscillation is assumed to be noise, resulting in the first 10 rows belonging to two species and a chemical rank of 2. Right: Data points in the V-space spanned by the first 20 rows. Their distribution with respect to x_2 is structured and thus not noise. It can also be seen that the first 10 data points lie approximately on a line and thus have a chemical rank of 2.

5.1.3. Combining submatrices into subsystems

When submatrices of a data set are identified, the next step is to combine them into subsystems. This approach is based on [66]. Therefore, columns or rows are identified that belong to at least two submatrices (these are, in a sense, shared blocks), and along these columns or rows, it is determined whether the submatrices have shared species. This is done by looking at the singular values of the individual matrices and their combination in the form of a scree plot. If the number of significant singular values does not change when looking at the combination of two matrices, then they share all species (at least in the selected window along the shared columns or rows). Also, if the number of significant singular values is less than the sum of the significant singular values of the individual submatrices, then there are shared species. This is shown in Fig. 5.6.

If the submatrices are of different sizes, an alternative way is to compare the singular values of a subset of columns or rows so that both are weighted equally. Alternatively, individual columns or rows can be selected to see if they affect the rank and thus are associated with an additional species.

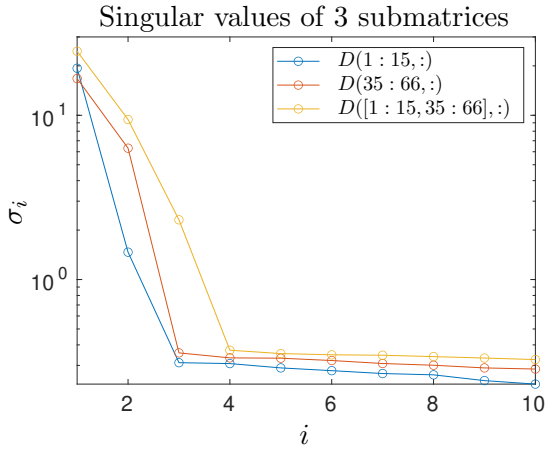

Figure 5.6.: Scree plot of the 10 largest singular values of selected submatrices of data set 2 with 1% added noise. Because 2 singular values are above the noise level for the submatrices $D(1:15,:)$ and $D(35:66,:)$, they both have 2 species. However, the combination of both has 3 significant singular values, so the individual submatrices share one species.

5.2. Subsystem analysis

Once the submatrices and subsystems are identified, the next step is to analyze them. For this purpose, the methods from chapters 2 and 3 are applied with the goal of obtaining a unique solution of the NMF problem 2.3 or, in the incomplete case, of problem 3.3 for a given data set. The analysis is divided into two steps.

The first step is to use the local rank information of the submatrices and to identify a (partially) unique solution based on the selected submatrices, if there exists one. A partially unique solution means that the profiles of a certain species are unique (except for scaling), but not all species have corresponding profiles that are unique [27]. In the context of a low-dimensional representation, this means that an SFS segment is point-like. This happens for example if the inner polytope touches the outer polytope [60] and is the intrinsic uniqueness of a data set. However, that such a situation occurs is a very strong assumption about the data. Therefore, the version used here relies on additional information, using local rank information of the selected submatrices

to obtain partially unique solutions, see Def. 5.4. In addition to partially unique solutions, there is full uniqueness, where all species have unique profiles.

Besides these two types of uniqueness there are others, see [27], which are not relevant in the following analysis. Thus, the focus is on either partial uniqueness or full uniqueness, which is abbreviated by the term uniqueness.

The second step is to analyze and possibly reduce the remaining ambiguity based on the selected subsystems as described in chapter 3. Thus, the minimal ambiguity of the data set is obtained by using local rank information.

The corresponding theory assumes that the data set is noise-free. It can be extended to the noisy case, as was done in chapter 2, by introducing tolerances regarding nonnegativity and local rank. To include information about selected submatrices, namely local rank information, the NMF problem 2.3 has to be extended.

Problem 5.3. *Let $D \in \mathbb{R}_+^{k\times n}$ be given with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$*, where DD^T and D^TD are irreducible. Furthermore, there exist $h \in \mathbb{N}$ submatrices of D denoted by $D_i \in \mathbb{R}_+^{k_i\times n}$ with $k_i < k$,* $\operatorname{rank}(D_i) = s_i < s$*, $i = 1, \ldots, h$. A nonnegative factorization is sought, i.e., finding factors $C \in \mathbb{R}_+^{k\times s}$ and $S^T \in \mathbb{R}_+^{s\times n}$ such that $D = CS^T$. In addition, the local rank constraints must be satisfied, i.e., for the factors C and S^T, $D_i = C(K_i, :)S^T$ must hold for each submatrix, where K_i is the index set belonging to the k_i selected rows and $C(K_i, :)$ has only s_i non-zero columns.*

The submatrix selection is done along one dimension only, but is analogous for the transposed case. Furthermore, the case can be made more complex by adding local rank constraints on S^T, which can be traced back to problem 5.3. The condition that $C(K_i, :)$ has only s_i non-zero columns ensures that no local rank deficiency occurs, see Def. 3.2.

Problem 5.3 allows to define partial unique solutions with respect to the local rank constraints. Thus the next definition follows, extending Def. 3 from [27].

Definition 5.4 (Partial uniqueness under local rank constraints)**.** *Let $D = \hat{C}\hat{S}^T \in \mathbb{R}_+^{k\times n}$ be a nonnegative matrix factorization under the local rank constraints of problem 5.3, with* $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ *and $\hat{C} \in \mathbb{R}_+^{k\times s}$, $\hat{S}^T \in \mathbb{R}_+^{s\times n}$. The i-th column of $\hat{C}$ is unique if for any other nonnegative matrix factorization of problem 5.3 of $D = CS^T$, with $C \in \mathbb{R}_+^{k\times s}$, $S^T \in \mathbb{R}_+^{s\times n}$, there exists an index j and a scalar $\alpha > 0$ such that $C(:, j) = \alpha\hat{C}(:, i)$.*

The definition can also be applied to the transposed case.

5.2.1. Determination of partially unique solutions

To determine partially unique solutions of a given data set, the first step is to look at the intrinsic uniqueness. This means looking at the low-dimensional representation of the data set and determining whether the inner and outer polytopes touch and thus limit the SFS. Ideally, such a case leads to a point-like SFS segment and thus partial uniqueness for a species. This concept is known as data-based uniqueness [27, 60]. Such an assumption is strong and not easy to fulfill, especially when dealing with noisy data sets. Thus, there are more general approaches to limit the underlying ambiguity of a data set by additional constraints. Such constraints can be selectivity or local rank assumptions. It is also possible to add information in the form of multisets or multiway data [79].

Taking a closer look at local rank assumptions, they were also investigated by Manne [47]. His analysis is based on finding submatrices that have only one species in common and thus uniquely

determine that shared species, see Thm. 2 in [47]. This idea is taken up and formalized in Thm. 5.5 and used for further analysis. Further work on how to limit the ambiguity of a data set were done in [64, 68].

The idea of using local rank constraints as in problem 5.3 is as follows. Given two submatrices $D_1 \in \mathbb{R}_+^{k_1 \times n}$ and $D_2 \in \mathbb{R}_+^{k_2 \times n}$ of $D \in \mathbb{R}_+^{k \times n}$ with $k_1, k_2 \leq k$, $\text{rank}(D) = \text{rank}_+(D) = s$, $\text{rank}(D_1) = \text{rank}_+(D_1) = s_1 \leq s$, $\text{rank}(D_2) = \text{rank}_+(D_2) = s_2 \leq s$. Both submatrices D_1 and D_2 have no underlying local rank deficiency, see Def. 3.2. If $\text{rank}\begin{pmatrix} D_1 \\ D_2 \end{pmatrix} = s_1 + s_2 - 1$, then D_1 and D_2 share one species which is uniquely determined.

This consideration can also be expressed in terms of the underlying row spaces. These are defined as $\mathcal{D}_i = \{y \in \mathbb{R}^n : \ y^T = x^T D_i, \ \forall x \in \mathbb{R}^{k_i}\}$ for a submatrix $D_i \in \mathbb{R}^{k_i \times n}$ of $D \in \mathbb{R}^{k \times n}$ with $k_i \leq k$. If $\text{rank}(D_i) = \text{rank}_+(D_i) = s_i$, then the row space $\mathcal{D}_i$ is spanned by s_i basis elements.

Thus, if there are 2 submatrices $D_1 \in \mathbb{R}_+^{k_1 \times n}$ and $D_2 \in \mathbb{R}_+^{k_2 \times n}$ of a matrix $D \in \mathbb{R}_+^{k \times n}$, their corresponding row spaces are given by $\mathcal{D}_1$ and $\mathcal{D}_2$. If $\dim(\mathcal{D}_1 \cap \mathcal{D}_2) = 1$, then they have one shared species which is uniquely determined. This is because if a profile $\hat{S} \in \mathbb{R}^n$ belongs to a solution (under local rank constraints) and corresponds to a species shared between D_1 and D_2 then it must be part of both row spaces, i.e., $\hat{S} \in \mathcal{D}_1$ and $\hat{S} \in \mathcal{D}_2$ resulting in $\hat{S} \in \mathcal{D}_1 \cap \mathcal{D}_2$. A one-dimensional intersection of the row spaces means that the profile of the shared species is uniquely determined (up to scaling).

If the intersection of the row spaces is of higher dimension, then D_1 and D_2 share more than one species. This is investigated in Sec. 5.2.3.

The concept of finding a partially unique solution based on submatrices is generalized in the next theorem, because in general two submatrices do not share only one species, but several. The essence of this theorem coincides with Thm. 2 of [47].

Theorem 5.5. *Let $D \in \mathbb{R}_+^{k \times n}$ be given, with $\text{rank}(D) = \text{rank}_+(D) = s$. Furthermore, there exist $h \in \mathbb{N}$ submatrices of D denoted by $D_i \in \mathbb{R}_+^{k_i \times n}$ with $i = 1, \ldots, h$, $k_i < k$, $\text{rank}(D_i) = \text{rank}_+(D_i) = s_i < s$, without local rank deficiency, see Def. 3.2, and $\mathcal{D}_i$ denote the corresponding row spaces. If $\dim\Big(\bigcap_{i=1,\ldots,h} \mathcal{D}_i\Big) = 1$, then the intersection consists of one basis element that represents a partially unique solution to the NMF problem of D and thus a profile of a species. Thus, the corresponding species is separated and its profile is uniquely determined.*

Proof. A nonnegative basis of $\mathcal{D}$ is used, so that each $\mathcal{D}_i$ is spanned by s_i basis elements of $\mathcal{D}$. This ensures that the submatrices D_i have no local rank deficiency. In the intersection of two row spaces are the basis elements and thus the profiles of the shared species. Since the intersection of all subspaces $\mathcal{D}_i$ is one-dimensional, only one basis element can and must be shared between them. This indicates partial uniqueness and can be shown by an indirect proof. Suppose there are two basis elements w and v with $w \neq \alpha v$, for all $\alpha \in \mathbb{R}_+$ shared by all D_i. However, since $w \neq \alpha v$, w and v span a two-dimensional space, which contradicts $\dim\Big(\bigcap_{i=1,\ldots,h} \mathcal{D}_i\Big) = 1$. Thus, the basis element is uniquely determined and corresponds to a partially unique solution of the NMF problem of D. □

Thus, partial unique solutions can be determined by looking at the intersections of subspaces. The intersection of two subspaces $\mathcal{D}_1 \cap \mathcal{D}_2 = \{z \in \mathbb{R}^n : \ z \in \mathcal{D}_1 \text{ and } z \in \mathcal{D}_2\}$ can be computed by $(D_1^T, -D_2^T)\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{0}$ where $x \in \mathbb{R}^{k_1}$ and $y \in \mathbb{R}^{k_2}$ are not part of the kernel of D_1^T and D_2^T respectively, $\mathbf{0}$ is the n-dimensional vector containing only zeros, and the intersection vectors are given by $z = x^T D_1 = y^T D_2$.

A more convenient approach would be to use the rank of the submatrices to determine whether basis elements are separable (for any number of submatrices). However, such a consideration "always" goes back to determining the dimension of the intersection of subspaces, and results in determining the dimension of the intersection of all subspaces as described in Thm. 5.5. Thus, there is no added value in using the rank to determine partial uniqueness.

Remark 5.6. *Thm. 5.5 can be transferred to the low-dimensional representation of D. The row spaces of the submatrices D_i can be represented by $(s_i - 1)$-dimensional affine subspaces in the low-dimensional representation of D. The intersection of the row spaces $\mathcal{D}_i$ can be transferred to the intersection of the affine subspaces. If their intersection is 0-dimensional, i.e., consists of a single point, this point corresponds to a partially unique solution.*

If all species of D can be separated as in Thm. 5.5, then they are all partially unique, which means that they form a fully unique solution and that the NMF problem under local rank constraints 5.3 has a unique solution. The question that arises in this context is how many of these submatrices are needed as a minimum to achieve full uniqueness. This is investigated next.

5.2.2. What is the minimum number of submatrices required for a unique solution?

If an arbitrary choice of submatrices along the rows of a given $D \in \mathbb{R}_+^{k\times n}$ is possible, the minimum number of submatrices needed to determine a unique solution of the NMF problem under local rank constraints 5.3 for a data set can be estimated. This estimation depends on the number of underlying species of each submatrix. It is described in the next theorem.

Theorem 5.7. *Let $D \in \mathbb{R}^{k\times n}$ be a nonnegative matrix with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$. Additionally, let $D_i \in \mathbb{R}_+^{k_i\times n}$, for $i = 1, \dots h$, $k_i < k$, be submatrices of D with $2 \leq \operatorname{rank}(D_i) = \operatorname{rank}_+(D_i) = m < s$, based on m basis elements of $\mathcal{D}$, so that the union of these submatrices is of rank s. The minimum h so that the basis elements of $\mathcal{D}$ are separable is $R(s,m)$. A lower bound is given by*

$$R(s,m) \geq \left\lceil \frac{2s}{m} \right\rceil.$$

An upper bound is given by

$$R(s,m) \leq s.$$

If $s \geq m(m-1)$, then

$$R(s,m) = \left\lceil \frac{2s}{m} \right\rceil.$$

Proof. The lower bound can be estimated by the following consideration. Each element must be contained in at least 2 row spaces of the submatrices in order to be separable. So there are at least $2s$ elements to divide into the row spaces of the submatrices. Since each submatrix must be associated with m species, thus each row space must be spanned by m basis elements, there must be at least $\frac{2s}{m}$ submatrices.
An upper bound for $R(s,m)$ is given by s, because for every s and m there is a construction as in example 5.8, so that every element is separable.
The exact estimation of $R(s,m) = \left\lceil \frac{2s}{m} \right\rceil$ is done by intersecting the bases and thus separating each basis element. The approach is based on completely separating systems of m-sets.[2] The proofs can be found in [62], Lem. 3, Eq. (2) and Thm. 6. □

[2] At this point I would like to thank Prof. Labahn and Prof. Kalinowski of the University of Rostock for a suggestion and helpful discussion on separating systems of m-sets.

Known values of $R(s,m)$ for $2 \leq s \leq 48$ and $1 \leq m \leq \min(20, s-1)$ are given in [62].

It is important to note that Thm. 5.7 does not mean that only $h = R(s,m)$ submatrices need to be chosen to achieve uniqueness, but rather that it is possible to need only $R(s,m)$ submatrices if they are chosen properly. It is also not guaranteed that such a selection exists for an arbitrary data set.

Assuming that arbitrary submatrices can be chosen with respect to the underlying basis elements. The upper bound of Thm. 5.7 is reached by the following construction. This partition can be obtained for arbitrary s and m.

Example 5.8. *Let $D \in \mathbb{R}_+^{k \times n}$, $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s$ and let the basis elements of the row space of D be represented by the numbers $1, \ldots, s$. If $1 < m < s$, then a possible, not necessarily smallest, selection of submatrices with $h = s$ is given by the bases represented by*

$$\{1, \ldots, m\},\ \{2, \ldots, m+1\}, \ldots,\ \{s-1, s, 1, \ldots, m-2\}, \{s, 1, \ldots, m-1\},$$

where the elements are computed modulo s. To separate a basis element, the intersection of m (consecutive) subspaces must be determined.

In contrast to this example, next is an example where $h = R(s,m)$ is reached.

Example 5.9. *Let $D \in \mathbb{R}_+^{k \times n}$ be given with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s = 7$. The basis elements of the row space of D are represented by the numbers $1, \ldots, 7$. When searching for submatrices, each based on $m = 3$ basis elements and thus 3 species, and spanning 3 dimensions, the minimum number of submatrices is given by $R(7,3) = 5$, based on Thm. 5.7. This minimum number is satisfied if the submatrices are chosen with the following bases*

$$\{1,2,3\},\ \{3,4,5\},\ \{5,6,7\},\ \{1,4,7\},\ \{2,4,6\}.$$

Intersecting these bases means that each basis element can be separated. Thus, the solution of the NMF problem of D is uniquely determined.

It is possible not to fix the number of underlying species, and thus the number of underlying basis elements of each submatrix, but to fix an upper limit. This leads to the following remark.

Remark 5.10. *Under the assumptions of Thm. 5.7, but changing the rank condition on D_i to $\operatorname{rank}(D_i) = \operatorname{rank}_+(D_i) \leq m$, the minimal h is denoted by $R(s,_{\leq} m)$. Then holds $R(s,_{\leq} m) \leq R(s,m)$. Lower and upper bounds for $R(s,_{\leq} m)$ are given in [37].*

The assumption of Thm. 5.7 regarding the submatrices is that there is no local rank deficiency, since every submatrix must be based on m basis elements of D. But what if this assumption is not true and some submatrices are rank deficient? Even in this case it can be possible to obtain a unique solution. This is shown in the next example.

Example 5.11. *Let $D \in \mathbb{R}_+^{k \times n}$ be given with $\operatorname{rank}(D) = \operatorname{rank}_+(D) = s = 5$. The basis elements of the row space of D are represented by the numbers $1, \ldots, 5$. Each selected submatrix spans an $m = 3$ dimensional space, but can be based on more than 3 basis elements. An example of possible bases is*

$$\{1,2,3\},\ \{1,4,5\},\ \{2,4,5+3\},\ \{3,4+2,5\},$$

where $5{+}3$ is a vector based on two basis elements. Intersecting these separates all basis elements, keeping in mind that $\{5\} \cap \{5+3\} = \emptyset$.

The goal is to identify submatrices so that the solution to the NMF problem is unique. But what if no unique solution is obtained? How can the results still be used to simplify the analysis?

5.2.3. What if there is no unique solution using local rank constraints?

If no unique solution to the NMF problem can be found, as described in the previous section, the idea is to use the given submatrices and maximize the information used to reduce the ambiguity as much as possible. The results are similar to those in Sec. 3.5, where the information of the submatrices is also used.

The idea is, as in the previous section, to compute the intersection of the subspaces defined by the submatrices. The difference is that these intersections do not only lead to one-dimensional subspaces. However, these intersections can be used to limit the ambiguity. This is because the basis elements or profiles corresponding to the shared species of D_1 and D_2 span the space given by $\mathcal{D}_1 \cap \mathcal{D}_2$. Thus, the selection of profiles for this shared species is limited compared to looking at the entire matrix given by $\begin{pmatrix} D_1 \\ D_2 \end{pmatrix}$. This can be phrased in a more general way.

Remark 5.12. *If for the intersection described in Thm. 5.5* $\dim\Big(\bigcap_{i=1,\dots,h} \mathcal{D}_i\Big) = \rho > 1$ *holds, then* ρ *basis elements are shared by all* $h \in \mathbb{N}$ *row spaces of the submatrices. That is, to determine these* ρ *basis elements, it is sufficient to look at the* ρ*-dimensional subspace spanned by the intersection instead of* $\mathcal{D}$*. Thus, to find a feasible factor for the shared species and their corresponding basis elements,* ρ *profiles and thus basis elements need only be determined in the subspace of the intersection of* $\mathcal{D}_i$*.*

To determine the solutions of the shared species and in general the solutions of the species that are not partially unique, their ambiguity must be determined. Since the ambiguity is determined in the low-dimensional representation, the ideas have to be conveyed.

5.2.4. Transfer to the low-dimensional representation

The general approach of using the intersection of row spaces to determine partially unique solutions can be transferred to the low-dimensional representation.

Considering the intersection of row spaces spanned by submatrices and thus affine subspaces spanned by submatrices in the low-dimensional representation, the uniqueness condition of Thm. 5.5 corresponds to the affine subspaces intersecting in a point. Similarly, if the intersection of the row spaces is not one-dimensional and thus the shared species are not partially unique, as in Rem. 5.12, the affine subspaces corresponding to the row spaces given by D_i, can also be intersected to limit the ambiguity. This is because in order to satisfy the local rank constraints, the representing points of the solutions must be on the corresponding affine subspaces spanned by the submatrices. This describes the same case as in Sec. 3.5 for case 2.

It is not as simple as described in Rem. 5.12 to just look at the subspace of the shared species and determine the ambiguity of them in that space or rather the corresponding low-dimensional representation. This is because there is not necessarily an underlying matrix of, e.g., the subspace $\mathcal{D}_1 \cap \mathcal{D}_2$ that is part of the NMF problem and leads to the same solutions (e.g., if there are no rows in the submatrices that are linear combinations of only the shared basis elements). Thus, there is no corresponding inner polytope, although the outer polytope can be computed by finding the intersection of the affine subspace of the shared species with the outer polytope of the entire data set. However, it is possible to resort to using the intersection of the affine subspace of the shared species with the SFS of the entire data matrix to obtain a superset of solutions, see Lem. 3.25. This goes back to the procedure used in Sec. 3.5 and the algorithm described in Sec. 4.3.

5.2.5. Analysis of subsystems based on submatrices

Finally, the previous considerations can be used to analyze not only submatrices, but also subsystems. As described in Sec. 5.1.3, the submatrices can be combined to subsystems. Also the knowledge of partially unique solutions can be transferred. The analysis is carried out as in chapter 3, depending on the case and thus on the distribution of the shared species. If the resulting subsystem is not as easy to represent as shown in Eq. (3.1), then the analysis can be done as described in Sec. 3.6 for more complex structures. Thus, reverting to the methods of chapter 3 allows the analysis of subsystems and thus of matrices with local rank constraints.

5.3. Critical summary of the detection and analysis of subsystems

The approach of using and analyzing subsystems is closely related to the analysis of incomplete data sets. Therefore, the main focus of this chapter is the detection of subsystems and, in a subsequent step, the analysis of the uniqueness of their solutions. For the detection of subsystems and their chemical rank only selected methods are presented, mostly SVD-based, which are sufficient for the use cases of this thesis. In general, other approaches, e.g., based on statistical methods, are possible, but the choice depends on the data.

The analysis of the subsystems and their underlying submatrices can be done as described in chapter 3. However, the focus is shifted from determining the SFS to identifying partially unique solutions using the subspaces spanned by selected submatrices. This is equivalent to using local rank constraints. This approach leads to general statements about the minimum number of submatrices needed to obtain a unique solution to the extended NMF problem 5.3. However, these estimates are only lower bounds and indirectly describe the dependence on the distribution of the species over the submatrices. Thus, there is a strong dependence on the given data matrix. Since an ideal choice of submatrices is not always possible, especially with measured data, the considerations regarding the minimum number of submatrices required are only theoretical.

The general concept of looking at the spaces spanned by the submatrices to identify partially unique solutions can be transferred to the low-dimensional representation. However, in the presence of local rank deficiency or when the subspace of interest has no representation in the data matrix, the transfer to the low-dimensional representation is not completely possible. Furthermore, considering the affine subspaces spanned by submatrices in the presence of noise to determine partially unique solutions or to give local rank constraints faces the same difficulties as described in Sec. 3.8 for case 2.

6. Application studies

To conclude, the methods presented in this thesis are applied to the following experimental data set.

Data set 3. *This experimental data set is a UV/Vis spectroelectrochemical (SEC) data set and has already been presented in [8]. It represents different oxidation states of a phenazine derivative and is measured over $k = 1518$ time steps and $n = 1000$ frequency channels. For this experimental setup, acetonitrile was used as the electrolyte with 0.1 M tetrabutylammonium hexaflourophosphate as the conducting salt. A gold mesh was used as the working electrode and a platinum wire as the counter electrode. Initially, only the phenazine derivative is present with 0.66 mM.*[1] *The potential is cycled during the measurement. The range is between -0.4 and 1.2 V, with a scan rate of 0.5 mV/s. There are two equilibrium potentials at 0.029 V and 0.266 V. The expected number of chemical species is three. This data set suffers from sensor saturation in several regions and is therefore ideal for applying the methods described in chapters 3 through 5.*

Data set 3 is shown in Fig. 6.1. For the analysis, first the submatrices and subsystems are identified as described in chapter 5 and analyzed with the methods presented in chapter 3 using the algorithms from chapter 4. The data set has already been analyzed in [8], but especially the identification of the subsystems and thus the identification of the saturated region is done in detail here.

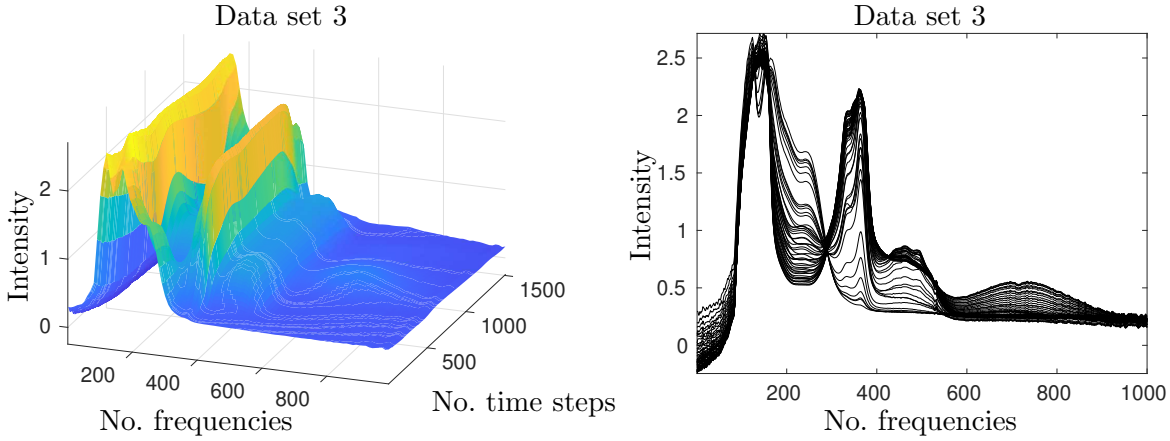

Figure 6.1.: Representation of data set 3. In both plots, affected areas are visible, e.g., the first frequency channels where the signal is clipped.

6.1. Analysis of data set 3

First, the methods of chapter 5 are applied to identify the regions suffering from sensor saturation. As a first step, backward EFA is used in frequency direction. This is shown in Fig. 6.2

[1] This additional chemical knowledge is used in the analysis.

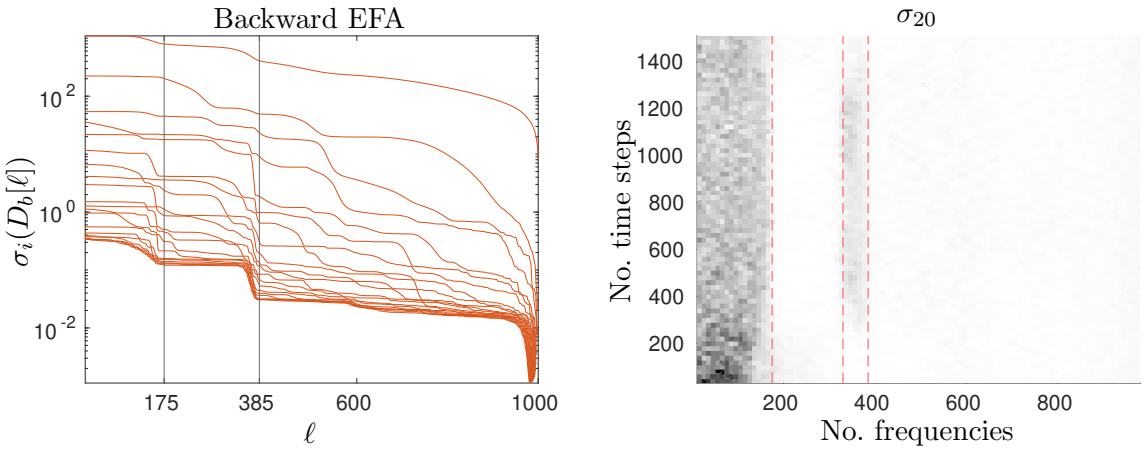

Figure 6.2.: Left: Backward EFA in the frequency direction of data set 3. Black lines mark the jumps in the noise level. Right: Combination of FSMW-EFA and backward EFA, showing the gradient of the curve corresponding to the 20th singular value. White indicates low values and black indicates high values. The red dashed lines mark the boundaries of the dark areas at 175, 330, and 385.

(left), where there are two jumps of the noise level at 175 and 385. This can already be seen in Fig. 6.1 where these jumps mark the right boundary of the sensor saturated areas along the frequency axis. Instead of excluding the entire affected part, i.e., the frequency channels up to 385, a detailed analysis is possible. This is shown in Fig. 6.2 on the right. A combination of the FSMW-EFA and backward EFA methods is used, as proposed in Sec. 5.1.2. To detect areas where the noise level grows rapidly, a singular value is used that is assumed to belong only to noise, in this case the 20th singular value is chosen. For each selected window of FSMW-EFA, backward EFA is performed to obtain curves of singular values. The numerically approximated gradients of the curves belonging to the 20th singular value are examined. As can be seen in Fig. 6.2 (right), there are areas where the gradient grows rapidly (marked in black). These are surrounded by red dashed lines. These areas are assumed to belong to the erroneous part of the data set.

However, not all time steps for a particular frequency channel are affected. For example, in the lower left corner of the right plot of Fig. 6.2 (frequency channel 142-175) there is a brighter part, indicating that this area is not or only sufficiently small affected. A similar relationship can be seen for frequency channel 330-385. Here only the middle part is affected by a high noise level. To determine the start and end of the erroneous part for these frequency channel, an EFA analysis in time direction is performed for this subwindow, see Fig. 6.3. A clear increase of the noise level can be seen at the beginning (time step 210). Not so clear is the increase at the end, at 1370 (it is oriented on the increase of the 5th singular value, which is assumed not to belong to an underlying species).

This identifies possible submatrices that are not affected by saturation. The entire data set with the marked erroneous parts is shown in Fig. 6.4 (left). The next step is to identify complete submatrices, possibly subsystems, and reorganize them to obtain an L-shaped structure at best (see Sec. 3.6). Such a division and re-sorting can be seen in Fig. 6.4 (center and right), with a corresponding labeling of the submatrices, which is used in the following. The submatrix D_2 is selected as shared block between D_1 and D_3. The submatrix D_4 leads to a more complex case and will be used later.

The next step is to determine the chemical rank of the submatrices. An example is shown for $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$ in Fig. 6.3 (right) for the evolution of the trace of the corresponding data points. It can

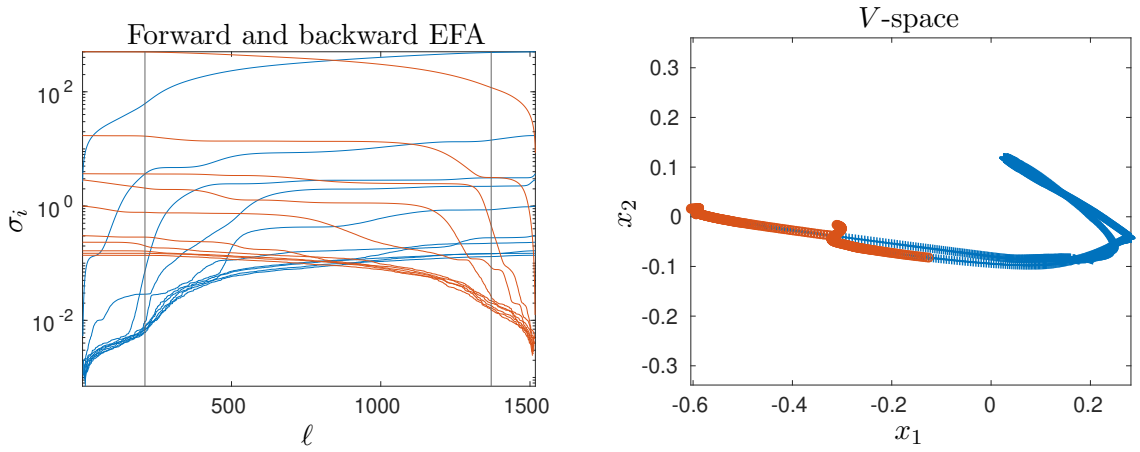

Figure 6.3.: Left: Forward and backward EFA in time direction, in blue and red, respectively, for the submatrix spanned by frequency channels 330 to 385 as marked in Fig. 6.2. Jumps in the noise level are marked by black lines. The jump in the backward EFA is based on the fifth singular value. Thus, the erroneous part corresponds only to time steps 210-1370. Right: Determination of the chemical rank for D_2 in the V-space of $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$. The data points of $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$ are marked in blue and the data points of D_2 are circled in red. The red marked data points lie approximately on a 1-dimensional affine subspace, so it is assumed that the chemical rank of D_2 is two.

be seen that D_2 has a chemical rank of 2 because it spans a 1-dimensional affine subspace. The chemical rank of $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$ can be determined in the 3-dimensional space, resulting in a chemical rank of 3.

The approach of looking at the data points is used because the singular values are not well separated, as shown in Fig. 6.2 (left) for a backward EFA in frequency direction (the part from 385 to 1000 is not affected by the erroneous part and can be used for the analysis). The same analysis is done for (D_2, D_1), indicating a chemical rank of 2.

The rank analysis shows that the current data set is classified as case 2 with the shared block D_2. Therefore, it is analyzed as described in Sec. 3.5.

First, (D_2, D_1) is analyzed. Since D_2 and (D_2, D_1) both have chemical rank 2, they can be analyzed as case 1. The impact of D_1 on the low-dimensional representation of D_2 is shown in Fig. 6.5 (right) in purple, the SFS of the low-dimensional representation of only D_2 is shown in green. A reduction of the outer polytope (i.e., the outer most interval points) is shown, but the impact is only on the left boundary point.

The reduced SFS can be used to analyze and reduce the ambiguity underlying $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$. This is shown in Fig. 6.5 (left), where the affine subspace spanned by D_2 is marked in yellow and the reduced SFS due to (D_2, D_1) of the species of D_2 is shown in purple. Opposed to Lem. 3.23, the right outer boundary of the purple SFS segment (this boundary point coincides with the boundary of the SFS segment using only D_2) does not coincide with the boundary of the SFS of $\mathcal{F}_S\left(\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}\right)$. This is due to noise. Both the right boundary point of the purple segment and the intersection of the affine subspace (yellow) and the right part of the SFS of $\mathcal{F}_S\left(\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}\right)$ belong to profiles that are close to zero for high frequency channel. Because the signal in this area is low (see Fig. 6.1) and has a low SNR, the two SVDs of D_2 and $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$ approximate this area slightly differently. So, even with the same tolerance for noise, the results can vary.

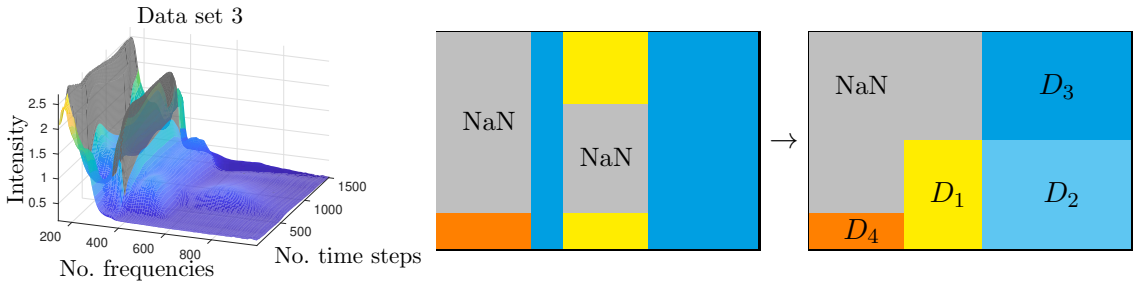

Figure 6.4.: Left: Data set 3 as in Fig. 6.1 but now with marked areas affected by sensor saturation (gray). The first 141 frequencies, which are completely affected by sensor saturation, are not shown. Center and right: Visualization of the structure of the data set. Again, gray areas indicate the incomplete blocks. For proper analysis, the data matrix (center) is rearranged into a structure with a connected incomplete block (right) so that D_1, D_2 and D_3 have an L-shaped structure. This representation is similar to Fig. 17 in [8].

Since the focus of the analysis is on the low-dimensional representation of the shared block D_2, the corresponding transferred SFS is used (Fig. 6.5 left in purple) instead of augmenting the SFS due to the boundary of the outer polytope of $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$.

The resulting SFS reduction is the intersection of the purple intervals with the light blue SFS segment and the dark blue segment. The impact on the shared species is also shown on the right in Fig. 6.5 in dark blue. This SFS is computed using the modified ray casting method from chapter 4 and the constraints given by the purple segment.

Additional knowledge can be used to further reduce the ambiguity. This can be chemical knowledge of the underlying reaction. For this data set, it is known that only one chemical species is present at the beginning. Thus, the spectrum of a shared species is known. It is marked by a single red circle in Fig. 6.5 (left). The resulting reduction in ambiguity is shown by red dashed lines (a line segment and an outlined area). The corresponding intervals belonging to the shared species in the shared low-dimensional representation are also shown in red in Fig. 6.5 (right).

This reduction based on external knowledge shows that it is advantageous to first use all available knowledge and subsequently perform an analysis regarding incompleteness or to apply only rank constraints and extrapolate the results to the remaining blocks. For the present data set, this means to use only the rank information regarding D_2 and to extrapolate the results from $\begin{pmatrix} D_2 \\ D_3 \end{pmatrix}$ to D_1, since no essential information is contained in this block when the first species is known. This means that D_1 does not reduce the SFS.

Finally, a look is taken at the submatrix D_4. This block is close to the saturated part and has a small SNR. It contains small amounts of the second species, but too small compared to the SNR. Thus, only the presence of the first species is assumed, which allows to approximate the spectrum of this species for frequency numbers 142-175, where this species shows a peak.

These application studies show that it is possible to apply the methods presented in this thesis to experimental data sets and to recover information from regions that would otherwise be neglected. Such regions are the submatrices D_1 and D_4 belonging to frequency channels partially affected by saturation. With traditional methods, these submatrices would not be included in the analysis.

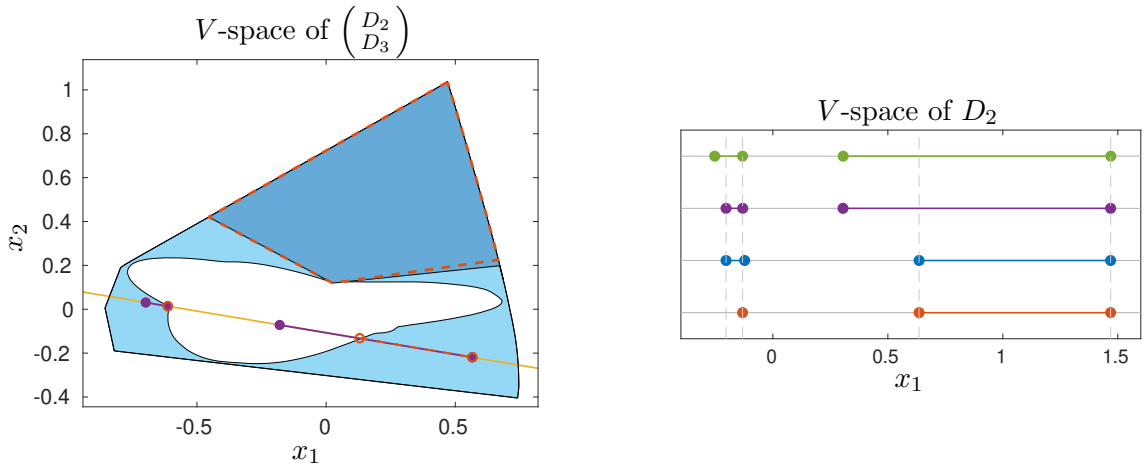

Figure 6.5.: Left: V-space representation of $\binom{D_2}{D_3}$. The corresponding SFS is marked in light blue. The affine subspace spanned by D_2 is marked in yellow. The reduced SFS due to the information of D_1 is marked by the dark blue area and the intersection of the light blue area with the purple intervals, shown on the right in dark blue. Using additional knowledge of a known spectrum, the SFS can be reduced to the areas whose boundaries are marked by the dashed red lines. Right: V-space of D_2 as a basis for the shared low-dimensional representation. The SFS of D_2 is shown in green. Adding the information from D_1 results in the purple intervals (only the outer boundaries are affected). Adding the constraints from D_3 limits the SFS to the blue intervals (only the inner boundaries are affected). The additional knowledge of a pure component spectrum results in a red point and a red interval. The latter contains the ambiguity of the second species. This representation is similar to Fig. 18 in [8], but differs slightly because the blocks are chosen slightly differently.

7. Summary and outlook

The question of finding a nonnegative matrix factorization (NMF) in an incomplete setting is addressed in this thesis, together with an analysis of the underlying ambiguity of solutions to the NMF problem. The main results are summarized in the following and their implications for future research are discussed.

Two general approaches are used to solve the NMF problem for incomplete data sets. One of them is the cone-based approach. It is used to directly represent a data matrix, namely its row and column spaces. This approach examines the effect on the solution of the NMF problem when information is added in the form of additional rows and columns. It provides easy access to the underlying geometric structures of the data matrix, but suffers in terms of representability because the representation depends on the dimensions of the matrix. The second method is based on an SVD. It is used to determine and represent the ambiguity of the nonnegative matrix factorization problem. The use of an SVD allows for a low-dimensional representation of the data matrix, depending on the underlying rank of the data matrix. It is possible to examine the effect of additional information on the ambiguity by transferring the results of the cone-based approach. This step allows information to be added to the low-dimensional representation without changing the underlying SVD. An important aspect of this construction is the underlying rank of the matrix. If the additional rows and columns increase the rank, the approach must be modified. This is done by considering (affine) subspaces spanned by submatrices in the representation of the SVD-based approach, which leads to constraints on the solutions.

Challenges such as the presence of noise are addressed, and approaches are presented for dealing with small amounts of noise when determining the ambiguity of solutions to the NMF problem of incomplete data sets. To allow the analysis of both noise-free and noisy data, an algorithmic approach to approximate the ambiguity is presented in chapter 4. The main advantage of this approach is its easy applicability to noisy data.

Chapter 5 shifts the focus to subsystems and their detection and analysis, which are a special case of incomplete data sets. This emphasizes the different applications of the methods proposed in chapter 3. Subsystem analysis is advantageous because data sets with a high chemical rank are more difficult to analyze. Subdividing these data sets into subsystems simplifies the analysis. Furthermore, subsystem analysis allows the use of local rank constraints in the form of submatrices underlying the subsystems to identify partially unique solutions to the NMF problem. An experimental spectroelectrochemical data set is used in chapter 6 to demonstrate the applicability of the presented methods, from the detection of incompleteness and subsystems to their analysis as incomplete data sets.

Thus, it is now possible to analyze the ambiguity of solutions to the NMF problem for incomplete data sets. The methods can be used, for example, in the context of subsystem analysis, online process monitoring, image fusion, or multiblock data fusion to determine the underlying ambiguities, as addressed in Sec. 3.1.

A further perspective is the application of the consideration of blockwise incomplete data to a kinetic hard-modeling approach. In SVD-based MCR analysis, kinetic hard-modeling can be used as an approach to obtain a factorization of a data set with concentration factors that satisfy

the underlying kinetic model [75]. If the data set suffers from incompleteness, there is no SVD of the entire data set, making it impossible to apply the kinetic hard-modeling approach directly. An example is when the intermediate of a chemical reaction suffers from sensor saturation, so the affected time steps would typically be neglected. To avoid the loss of information and still apply the kinetic model to all time steps, the idea is to use the approach of chapter 3. That is, either apply the model to the submatrices that do not suffer from saturation and combine the results, or find a shared block and analyze the data set by adding the constraints of the remaining blocks as well as the constraints of the kinetic model. This analysis can be done, for example, in the shared low-dimensional representation. In addition, this representation enables a visualization of the underlying parameter ambiguity of the kinetic model.

Beyond this possible application, there are other research questions that can be addressed. As considered in Sec. 5.1.2 and 6.1, determining the affected areas of a data set in a noisy environment is not easy. There are certain areas where it is not clear whether these areas are affected by incompleteness or whether the observed effects are due to noise. There are two aspects to keep in mind. On the one hand, the goal is to exclude the affected areas as thoroughly as possible. On the other hand, the excluded part of the data matrix should be as small as possible, as otherwise removing too large areas would increase the ambiguity of the solution. The goal is to find a balance between these two aspects. This can be achieved by gradually including parts of the potentially affected regions in the analysis. By comparing this inclusion with the case where such parts are not included, it is possible to evaluate how much the included part suffers from noise or other effects. This can be done, for example, by examining the different low-dimensional representations, the effect on the SFS, or the resulting profiles.

As described in chapter 3, this thesis focuses on blockwise missing data. However, what happens if a data set suffers from different, non-block-shaped structures of incompleteness? Such considerations lead to more general cases like those considered in Sec. 3.6 – still under avoidance of data imputation as described in Sec. 3.1. Such situations may arise, for example, with excitation emission data where the incomplete block may be triangular [29]. The question is not only how to choose the shared block, but also what area should and can be covered block by block if the approaches presented in this thesis are to be applied. Alternatively, the question can be asked, what other approaches exist to access the underlying ambiguity of the solutions to the NMF problem for such data sets that do not rely on a blockwise structure to resolve the data set.

In conclusion, there are still many open research questions that need to be addressed in the context of incomplete data, especially with respect to the underlying ambiguity of the solution to the corresponding NMF problem. However, the first step towards analyzing such ambiguity using the available information in a blockwise manner is taken by the methods presented in this thesis.

A. Appendix

A.1. Model data sets

The two model data sets that are used throughout this thesis are presented next.

Data set 1. *A simulated data set with a nonnegative matrix* $D \in \mathbb{R}^{40\times100}$*, based on three concentration and spectral profiles as shown in Fig. A.1. It can be considered as a spectroscopic data set where* C *satisfies the kinetic* $A \to B \to C$*, see [75].*

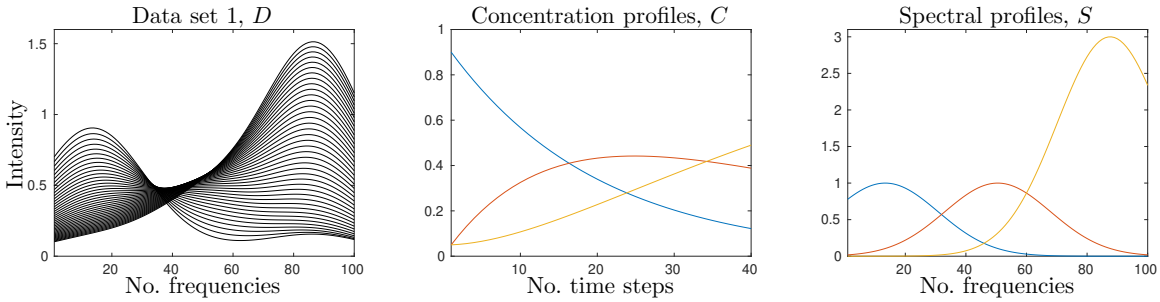

Figure A.1.: Representation of data set 1. Left: Representation of the complete data matrix D. Center: The modeled concentration profiles used for C and to generate D. Right: The modeled spectral profiles stored in S and used to generate D with $D = CS^T$. Each spectral profile corresponds to a concentration profile. They are marked with the same color.

Data set 2. *This data set is a model data set with* $D \in \mathbb{R}_+^{66\times150}$*. The factors* C *and* S^T *which are used to generate the data set are shown in Fig. A.2 and consist of three profiles each. The shape of the concentration profiles is based on elution profiles.*

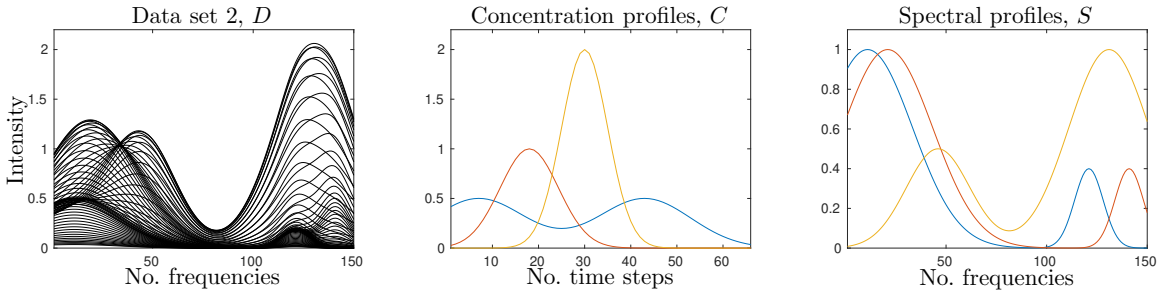

Figure A.2.: Representation of data set 2. Left: Representation of the data matrix D. Center: Modeled concentration profiles used to generate D. Right: Modeled spectral profiles used to generate D with $D = CS^T$. Each spectral profile corresponds to a concentration profile marked with the same color.

Bibliography

[1] H. Abdollahi and R. Tauler. Uniqueness and rotation ambiguities in multivariate curve resolution methods. *Chemom. Intell. Lab. Syst.*, 108(2):100–111, 2011.

[2] M. Akbari Lakeh, R. Rajkó, and H. Abdollahi. Local rank deficiency caused problems in analyzing chemical data. *Anal. Chem.*, 89(4):2259–2266, 2017. PMID: 28192909.

[3] M. Alier and R. Tauler. Multivariate curve resolution of incomplete data multisets. *Chemom. Intell. Lab. Syst.*, 127:17–28, 2013.

[4] M. Alinaghi, R. Rajkó, and H. Abdollahi. A systematic study on the effects of multi-set data analysis on the range of feasible solutions. *Chemom. Intell. Lab. Syst.*, 153:22–32, 2016.

[5] T. Andersons, M. Sawall, M. Beese, C. Kubis, and K. Neymeyr. On Hidden Rank Deficiency in MCR Problems. *J. Chemom.*, page e3608, 2024.

[6] T. Andersons, M. Sawall, and K. Neymeyr. Analytical enclosure of the set of solutions of the three-species multivariate curve resolution problem. *J. Math. Chem.*, 60:1750–1780, 2022.

[7] F. Arteaga, A. Folch-Fortuny, and A. Ferrer. Missing Data. In S. Brown, R. Tauler, and B. Walczak, editors, *In Comprehensive Chemometrics: Chemcial and Biochemical Data Analysis*, pages 615–639. Elsevier, 2020.

[8] M. Beese, T. Andersons, M. Sawall, C. Ruckebusch, A. Gómez-Sánchez, R. Francke, A. Prudlik, R. Franke, and K. Neymeyr. On the factor ambiguity of MCR problems for blockwise incomplete data sets. *Chemom. Intell. Lab. Syst.*, 249:105134, 2024.

[9] Y. Beyad and M. Maeder. Multivariate linear regression with missing values. *Anal. Chim. Acta*, 796:38–41, 2013.

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

[11] M. Bylesjö, O. Cloarec, and M. Rantalainen. Normalization and Closure. In S. Brown, R. Tauler, and B. Walczak, editors, *In Comprehensive Chemometrics: Chemcial and Biochemical Data Analysis*, pages 101–114. Elsevier, 2020.

[12] T. B. Cattell. The scree test for the number of factors. *Multivariate Behav. Res.*, 1(2):245–276, 1966.

[13] E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *J. Acoust. Soc. Am.*, 25(5):975–979, 1953.

[14] J. E. Cohen and U. G. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra Appl.*, 190:149–168, 1993.

[15] L. Coic, R. Vitale, M. Moreau, D. Rousseau, J. de Morais Goulart, N. Dobigeon, and C. Ruckebusch. Assessment of essential information in the Fourier domain to accelerate Raman hyperspectral microimaging. *Anal. Chem.*, 95(42):15497–15504, 2023. PMID: 37821082.

[16] A. de Juan. Chapter 2.5 - multivariate curve resolution for hyperspectral image analysis. In J. M. Amigo, editor, *Hyperspectral Imaging*, volume 32 of *Data Handling in Science and Technology*, pages 115–150. Elsevier, 2019.

[17] A. de Juan, M. Maeder, T. Hancewicz, and R. Tauler. Local rank analysis for exploratory spectroscopic image analysis. Fixed Size Image Window-Evolving Factor Analysis. *Chemom. Intell. Lab. Syst.*, 77(1):64–74, 2005.

[18] A. de Juan, S. Navea, J. Diewok, and R. Tauler. Local rank exploratory analysis of evolving rank-deficient systems. *Chemom. Intell. Lab. Syst.*, 70(1):11–21, 2004.

[19] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[20] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*. Kluwer Academic Publishers, 1996.

[21] G. F. Frobenius. Über Matrizen aus nicht negativen Elementen. *S.-B. Preuss. Akad. Wiss.*, 26:456–477, 1912.

[22] H. Gampp, M. Maeder, C. J. Meyer, and A. D. Zuberbühler. Calculation of equilibrium constants from multiwavelength spectroscopic data III: Model-free analysis of spectrophotometric and ESR titrations. *Talanta*, 32(12):1133–1139, 1985.

[23] P. Geladi and S. Wold. Local principal component models, rank maps and contextuality for curve resolution and multi-way calibration inference. *Chemom. Intell. Lab. Syst.*, 2(4):273–281, 1987.

[24] P. J. Gemperline. Computation of the range of feasible solutions in self-modeling curve resolution algorithms. *Anal. Chem.*, 71(23):5398–5404, 1999.

[25] M. Ghaffari, N. Omidikia, and C. Ruckebusch. Essential spectral pixels for multivariate curve resolution of chemical images. *Anal. Chem.*, 91(17):10943–10948, 2019. PMID: 31361465.

[26] N. Gillis. *Nonnegative Matrix Factorization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020.

[27] N. Gillis and R. Rajkó. Partial identifiability for nonnegative matrix factorization. *SIAM J. Matrix Anal. Appl.*, 44(1):27–52, 2023.

[28] G. H. Golub and C. F. van Loan. *Matrix computations, 4th edition*. The Johns Hopkins University Press, 2013.

[29] A. Gómez-Sánchez, I. Alburquerque, P. Loza-Álvarez, C. Ruckebusch, and A. de Juan. The trilinear constraint adapted to solve data with strong patterns of outlying observations or missing values. *Chemom. Intell. Lab. Syst.*, 231:104692, 2022.

[30] A. Gómez-Sánchez, C. Ruckebusch, R. Tauler, and A. de Juan. Dealing with missing data blocks in multivariate curve resolution. towards a general framework based on a single factorization model. *TrAC, Trends Anal. Chem.*, 179:117869, 2024.

[31] B. Grung and R. Manne. Missing values in principal component analysis. *Chemom. Intell. Lab. Syst.*, 42(1):125–139, 1998.

[32] R. C. Henry. Duality in multivariate receptor models. *Chemom. Intell. Lab. Syst.*, 77(1-2):59–63, 2005.

[33] A. Jürß. *Über nichtnegative Matrixfaktorisierungen und geometrische Algorithmen zur Approximation ihrer Lösungsmengen*. PhD thesis, Universität Rostock, 2017.

[34] A. Jürß, M. Sawall, and K. Neymeyr. On generalized Borgen plots. I: From convex to affine combinations and applications to spectral data. *J. Chemom.*, 29(7):420–433, 2015.

[35] H. R. Keller and D. L. Massart. Evolving factor analysis. *Chemom. Intell. Lab. Syst.*, 12(3):209–224, 1991.

[36] H. R. Keller and D. L. Massart. Peak purity control in liquid chromatography with photodiode-array detection by a fixed size moving window evolving factor analysis. *Anal. Chim. Acta*, 246(2):379–390, 1991.

[37] A. Kündgen, D. Mubayi, and P. Tetali. Minimal completely separating systems of k-sets. *J. Comb. Theory Ser. A.*, 93(1):192–198, 2001.

[38] W. H. Lawton and E. A. Sylvestre. Self modelling curve resolution. *Technometrics*, 13(3):617–633, 1971.

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

[40] Y.-Z. Liang, O. M. Kvalheim, A. Rahmani, and R. G. Brereton. Resolution of strongly overlapping two-way multicomponent data by means of heuristic evolving latent projections. *J. Chemom.*, 7(1):15–43, 1993.

[41] M. B. Lopes, J.-C. Wolff, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Near-infrared hyperspectral unmixing based on a minimum volume criterion for fast and accurate chemometric characterization of counterfeit tablets. *Anal. Chem.*, 82(4):1462–1469, 2010. PMID: 20095581.

[42] M. Maeder. Evolving factor analysis for the resolution of overlapping chromatographic peaks. *Anal. Chem.*, 59(3):527–530, 1987.

[43] M. Maeder and Y. M. Neuhold. *Practical data analysis in chemistry*, volume 26 of *Data Handling Sci. Technol.* Elsevier, Amsterdam, 2007.

[44] M. Maeder and A. Zilian. Evolving factor analysis, a new multivariate technique in chromatography. *Chemom. Intell. Lab. Syst.*, 3(3):205–213, 1988.

[45] M. Maeder and A. D. Zuberbuehler. The resolution of overlapping chromatographic peaks by evolving factor analysis. *Anal. Chim. Acta*, 181:287–291, 1986.

[46] E. R. Malinowski. *Factor analysis in chemistry*. Wiley, New York, 2002.

[47] R. Manne. On the resolution problem in hyphenated chromatography. *Chemom. Intell. Lab. Syst.*, 27(1):89–94, 1995.

[48] H. Minc. *Nonnegative matrices*. John Wiley & Sons, New York, 1988.

[49] P. R.C. Nelson, P. A. Taylor, and J. F. MacGregor. Missing data methods in pca and pls: Score calculations with incomplete observations. *Chemom. Intell. Lab. Syst.*, 35(1):45–65, 1996.

[50] K. Neymeyr, M. Beese, and M. Sawall. On properties of EFA plots. *J. Chemom.*, page e3381, 2021.

[51] K. Neymeyr and M. Sawall. On the set of solutions of the nonnegative matrix factorization problem. *SIAM J. Matrix Anal. Appl.*, 39:1049–1069, 2018.

[52] K. Neymeyr, M. Sawall, and T. Andersons. A note on rank deficiency and numerical modeling. *J. Chemom.*, page e3550, 2024.

[53] K. Neymeyr, M. Sawall, Z. Rasouli, and M. Maeder. On the avoidance of crossing of singular values in the evolving factor analysis. *J. Chemom.*, page e3217, 2020.

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

[55] O. Perron. Zur Theorie der Matrices. *Math. Ann.*, 64(2):248–263, 1907.

[56] S. Piqueras, C. Bedia, C. Beleites, C. Krafft, J. Popp, M. Maeder, R. Tauler, and A. de Juan. Handling different spatial resolutions in image fusion by multivariate curve resolution-alternating least squares for incomplete image multisets. *Anal. Chem.*, 90(11):6757–6765, 2018. PMID: 29697967.

[57] J. Podani, T. Kalapos, B. Barta, and D. Schmera. Principal component analysis of incomplete data – A simple solution to an old problem. *Ecol. Inform.*, 61:101235, 2021.

[58] A. Queral-Beltran, M. Marín-García, S. Lacorte, and R. Tauler. Multivariate curve resolution of incomplete and partly trilinear multiblock datasets. *Chemom. Intell. Lab. Syst.*, 247:105081, 2024.

[59] R. Rajkó. Natural duality in minimal constrained self modeling curve resolution. *J. Chemom.*, 20(3-4):164–169, 2006.

[60] R. Rajkó, H. Abdollahi, S. Beyramysoltan, and N. Omidikia. Definition and detection of data-based uniqueness in evaluating bilinear (two-way) chemical measurements. *Anal. Chim. Acta*, 855:21–33, 2015.

[61] R. Rajkó and K. István. Analytical solution for determining feasible regions of self-modeling curve resolution (SMCR) method based on computational geometry. *J. Chemom.*, 19(8):448–463, 2005.

[62] C. Ramsay, I. T. Roberts, and F. Ruskey. Completely separating systems of k-sets. *Discrete Math.*, 183(1):265–275, 1998.

[63] C. Ruckebusch, R. Vitale, M. Ghaffari, S. Hugelier, and N. Omidikia. Perspective on essential information in multivariate curve resolution. *TrAC, Trends Anal. Chem.*, 132:116044, 2020.

[64] M. Sawall. *Regularisierte nichtnegative Matrixfaktorisierungen und ihre Anwendungen in der Spektroskopie*. PhD thesis, Universität Rostock, 2011.

[65] M. Sawall. Analyse und Berechnung niedrigdimensionaler Darstellungen von Lösungsmengen zur nichtnegativen Matrixfaktorisierung. Habilitationsschrift, Universität Rostock, 2018.

[66] M. Sawall, C. Fischer, B. J. Elvers, S. Pätsch, and K. Neymeyr. A multi-method chemometric analysis in spectroelectrochemistry: Case study on molybdenum mono-dithiolene complexes. *Anal. Chim. Acta*, 1185:339065, 2021.

[67] M. Sawall, A. Jürß, H. Schröder, and K. Neymeyr. Simultaneous construction of dual Borgen plots. I: The case of noise-free data. *J. Chemom.*, 31(12):e2954, 2017.

[68] M. Sawall, C. Kubis, H. Schröder, D. Meinhardt, D. Selent, R. Franke, A. Brächer, A. Börner, and K. Neymeyr. Multivariate curve resolutions methods and the design of experiments. *J. Chemom.*, 34(2):e3159, 2020.

[69] M. Sawall, C. Kubis, D. Selent, A. Börner, and K. Neymeyr. A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. I: Concepts and applications. *J. Chemom.*, 27(5):106–116, 2013.

[70] M. Sawall, A. Moog, C. Kubis, H. Schröder, D. Selent, R. Franke, A. Brächer, A. Börner, and K. Neymeyr. Simultaneous construction of dual Borgen plots. II: Algorithmic enhancement for applications to noisy spectral data. *J. Chemom.*, 32(6):e3012, 2018.

[71] M. Sawall and K. Neymeyr. A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. II: Theoretical foundation, inverse polygon inflation, and FACPACK implementation. *J. Chemom.*, 28(5):633–644, 2014.

[72] M. Sawall and K. Neymeyr. A ray casting method for the computation of the area of feasible solutions for multicomponent systems: Theory, applications and FACPACK-implementation. *Anal. Chim. Acta*, 960:40–52, 2017.

[73] M. Sawall, C. Ruckebusch, M. Beese, R. Francke, A. Prudlik, and K. Neymeyr. An active constraint approach to identify essential spectral information in noisy data. *Anal. Chim. Acta*, 1233:340448, 2022.

[74] M. Sawall, S. Vali Zade, C. Kubis, H. Schröder, D. Meinhardt, A. Brächer, R. Franke, A. Börner, H. Abdollahi, and K. Neymeyr. On the restrictiveness of equality constraints in multivariate curve resolution. *Chemom. Intell. Lab. Syst.*, 199:103942, 2020.

[75] H. Schröder. Lösungsmengen von Modellparametrierungen zu nichtnegativen Matrixfaktorisierungsproblemen. Dissertation, Universität Rostock, 2019.

[76] A. K. Smilde, T. Næs, and K. H. Liland. *Multiblock data fusion in statistics and machine learning: Applications in the natural and life sciences*. John Wiley & Sons, 2022.

[77] R. Tauler. Multivariate curve resolution applied to second order data. *Chemom. Intell. Lab. Syst.*, 30(1):133 – 146, 1995.

[78] R. Tauler. Calculation of maximum and minimum band boundaries of feasible solutions for species profiles obtained by multivariate curve resolution. *J. Chemom.*, 15(8):627–646, 2001.

[79] R. Tauler, A. Smilde, and B. Kowalski. Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution. *J. Chemom.*, 9(1):31–58, 1995.

[80] L. B. Thomas. Rank factorization of nonnegative matrices (A. Berman). *SIAM Rev.*, 16(3):393–394, 1974.

[81] R. S. Varga. *Matrix iterative analysis*, volume 27 of *Springer Ser. Comput. Math.* Springer, Heidelberg, 2000.

[82] M. Vosough, C. Mason, R. Tauler, M. Jalali-Heravi, and M. Maeder. On rotational ambiguity in model-free analyses of multivariate data. *J. Chemom.*, 20(6-7):302–310, 2006.

[83] B. Walczak and D.L. Massart. Dealing with missing data: Part I. *Chemom. Intell. Lab. Syst.*, 58(1):15–27, 2001.

[84] B. Walczak and D.L. Massart. Dealing with missing data: Part II. *Chemom. Intell. Lab. Syst.*, 58(1):29–42, 2001.

[85] R. M. Wallace. Analysis of absorption spectra of multicomponent systems. *J. Phys. Chem.*, 64(7):899–901, 1960.

[86] A. C. Whitson and M. Maeder. Exhaustive evolving factor analysis (E-EFA). *J. Chemom.*, 15(5):475–484, 2001.

[87] S. Wold, C. Albano, W. J. Dunn, U. Edlund, K. Esbensen, P. Geladi, S. Hellberg, E. Johansson, W. Lindberg, and M. Sjöström. *Multivariate Data Analysis in Chemistry*, pages 17–95. Springer Netherlands, Dordrecht, 1984.

[88] S. V. Zade, K. Neymeyr, M. Sawall, C. Fischer, and H. Abdollahi. Data point importance: Information ranking in multivariate data. *J. Chemom.*, 37(1):e3453, 2023.

Danksagung

Meinen Dank möchte ich all denjenigen aussprechen, die mich während meiner Promotion unterstützt haben.

Besonders möchte ich Prof. Neymeyr danken für die Betreuung meiner Promotion, all die Anregungen und Ideen und auch die konstruktive Kritik, die unglaublich wertvoll war, auch wenn ich sie nicht immer hören wollte.

Ebenfalls einen besonderen Dank möchte ich der Arbeitsgruppe Numerische Mathematik aussprechen. Vor allem PD Dr. Mathias Sawall für die fachlichen Diskussionen, ebenso Tomass Andersons, aber auch Prof. Kurt Frischmuth für die manchmal überfachlichen Unterhaltungen.

Ebenso möchte ich den Kollegen des Leibniz Institut für Katalyse (LIKAT) und unseren Kooperationspartnern bei der Evonik Oxeno GmbH für die gute Zusammenarbeit danken. Besonders Adrian Prudlik und Prof. Francke vom LIKAT für ihre Zeit, mir geduldig die chemischen Hintergründe zu erläutern und für die Überlassung von Datensätzen.

Der Arbeitsgruppe von Prof. Ruckebusch in Lille möchte ich für die Gastfreundlichkeit und unglaublich inspirierenden Diskussionen danken. Merci, grazie e gracias!

Danke auch meinen Freunden und Familie und besonders Marvin fürs Unterstützen und Aushalten in all den Hochs und Tiefs, die eine Promotion so mit sich bringt. Danke!

"And there in the dark, he asks if it was really worth it.
Were the instants of joy worth the stretches of sorrow?
Were the moments of beauty worth the years?
And she turns her head, and looks at him, and says 'Always'."
– V. E. Schwab, The Invisible Life of Addie LaRue (slightly modified)