Non-Negative Dimensionality Reduction in Signal Separation

Dissertation zur Erlangung des Doktorgrades an der Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Mathematik der Universität Hamburg

vorgelegt von
Sara Krause-Solberg

Hamburg, 2015

Day of oral defense: April 27th, 2016

The following evaluators recommend the admission of the dissertation:

Prof. Dr. Armin Iske
Prof. Dr. Gerlind Plonka-Hoch

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, den 22. Dezember 2015

Contents

Introduction

Signals have a significant impact on our every day life. They are used for communication and entertainment, in engineering and medicine, for traffic control, space exploration and data compression. In all these applications, signals are used to transmit information. This is why there has been a growing interest in the field of signal processing. Due to the further development during the last decades, for example, in multimedia entertainment and information systems, signals have gained even more attention. Even though the foundations for (digital) signal processing were laid in the 17th century with the invention of integration and differentiation and in particular in the beginning of the 19th century with the introduction of the Fourier series [39], the systematic exploration began in the 1940s when Zuse presented the first programmable fully automatic digital computer. Since then, signal processing has become a fundamental and influential field of research. Due to the particular importance of technology in today's digital world and the extremely fast increasing performance of electronic devices, the efficient processing, analysis, organization and manipulation of digital data has become more important than ever before. Many of the latest advances facilitating our daily life do strongly depend on digital signal processing.

In many applications, plenty of signals are created and thus it might come to a superposition or mixture of signals. Moreover, the information contained in a signal might be encoded such that it is not readily available. Thus, the ability to extract information from a signal has become more and more essential for handling the huge amount of collected signals (see e.g. [30]). It is clear that this comes along with the need of tremendous computational power and the possibility to compress and efficiently store the data. In this context, the notorious *'curse of dimensionality'* [9] is a serious issue that concerns the development of advanced tools and forces balancing accuracy and storage capacity. However, the data can often be characterized by only few features and thus, it might be sufficient to store only these for retrieving the signal when needed. This is why research on the efficient extraction and reconstruction of information from data has been intensified.

Especially in signal processing, the extraction of meta data is used in several applications such as weather forecasts, where the relevant information needs to be selected from meteorological data and satellite images, or robot control, where a matching of visual, audio and other stimulations is demanded. Many applications, however, refer to audio data, as for example, acoustic echo cancellation and denoising, automatic transcription of music, application of audio effects to single instruments in a mixed recording, speaker separation in video conferences, emotion recognition from speech signals or hearing aids, which are able to accentuate different sources. In all these situations, an efficient method to analyze the auditory scene in order to extract the essential information is needed.

It is not surprising that the exceptional capability of humans to focus on a certain source within a mixture of multiple sound sources has aroused the interest of many researchers. The ability to suppress ambient and background noise and disturbing sources and to concentrate on a particular sound source has gone down as *'cocktail party effect'* [26]. This selective hearing is based on spatial distances between the sources, differences in pitch and quality or visual indicators such as lip reading [117]. Nevertheless, the current state of scientific and technical knowledge is far from attaining results similar to those of the human auditory system.

In the last decades, some relatively successful separation algorithms appeared, and thus, investigation on this topic has been intensified (see e.g. [5, 22, 67, 99, 108, 117, 122, 128]). One approach to technically solve the problem of extracting single sources from a mixed signal is blind signal separation. It relies on no assumptions concerning the position of sensors or sources in contrast to geometrical source separation by means of beamforming (e.g. [6]) or similar methods.

Blind signal separation (BSS) recovers a set of unknown source signals from a set of mixed signals or other observations. The set of observations is usually given as a set of recordings, each a different combination of the source signals, depending on the position of the sensor. In this context, 'blind' stands for the fact that the sources themselves are not individually observed and that there is no information available about the mixing process, i.e., the estimation is performed with hardly any knowledge about the sources, as for example location or activity time. This 'blindness' is not a negative property, in contrast, it is precisely the strength of BSS models making them flexible and useful in a wide range of applications [20].

A joint feature of many BSS methods is the assumption that the observations are a weighted sum of the unknown sources (for non-linear mixing models see for instance [74, 110]). This assumption involves the restriction that there are at least as many observations as latent sources in order to guarantee the solvability of the linear system describing the mixing process.

However, as in many applications there is only one sensor recording the mixed signal (e.g. monophonic music recordings), there is a strong demand for methods that can handle the highly under-determined situation of such a *single-channel problem*. To circumvent the problem of having fewer observations than latent source signals, the classical BSS methods are usually combined with a preprocessing step involving time-frequency analysis in order to construct a larger set of observations. In the time-frequency space each frequency evolution in time can be viewed as one observation. Thus, a monophonic recording becomes a large data matrix.

As one of the main difficulties of the BSS problem is its under-determination, there are several approaches how to further restrict the set of possible solutions. The most popular ones are *independent component analysis (ICA)* and *non-negative matrix factorization (NNMF)* or modifications of those. ICA-like methods aim for uncorrelated or stochastically independent source signals whereas NNMF-based methods focus on structural properties of the sources such as sparsity. Both methods are frequently used in single-channel separation, see e.g. [10, 20, 59, 60, 76] for ICA and [25, 91, 108, 121] for NNMF.

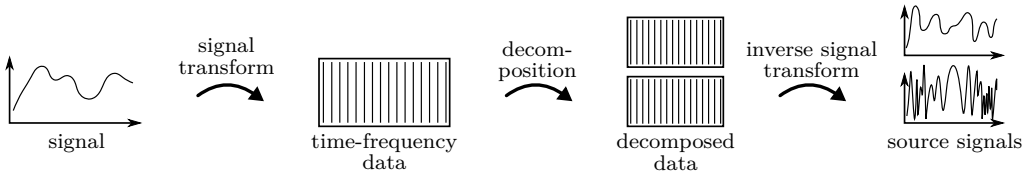


Figure 1: General proceeding of signal separation in the time-frequency domain.

A broadly used enhancement of ICA is *independent subspace analysis (ISA)* introduced by Hyvärinen and Hoyer in [58], popularized by Casey and Westner in [22] and used in many applications e.g. [48, 111, 128]. This technique combines the classical ICA method with a grouping of the extracted features. The source signals are found in so called independent subspaces spanned by these features. This approach can also be combined with NNMF (see e.g. [54]). Other decomposition methods that should not be overlooked are, for example, *azimuth discrimination and resynthesis (ADRess)* [6], which tries to locate the different sources in space by analyzing the phase-shifts of recordings made at different positions, and *computational auditory scene analysis (CASA)* [16], which tries to mimic the human ear by the consecutive application of different filters to the time-frequency data of the recorded signal.
As a consequence, BSS methods cannot only be classified into those operating in the time-amplitude or time-frequency domain, but also into those using ICA, NNMF or another decomposition method.
The general proceeding in signal separation in the time-frequency domain can be summarized in three steps (compare Figure 1). First, the input signal is transformed to the time-frequency space. This generates a data set or data matrix whose columns contain information on the frequencies of the signal at a certain time instant. This data matrix is typically high-dimensional and its size depends on the width of the signal's frequency band and its temporal duration. The actual separation or decomposition is performed on this data set by extracting features corresponding to different source signals. In this way, the data matrix is split into different matrices which are a linear decomposition of the original matrix. In the reconstruction step, the sources are computed from these data sets by applying an inverse signal transform. The first and the last step of this framework are quite well understood whereas the middle step is the one which causes trouble. Not only the development of decomposition techniques but also the dimension of the involved data sets represent a challenge, especially when it comes to almost-real-time computation which is desirable for many audio-related applications.
To reduce costs and speed up computation, dimensionality reduction can be included as a preprocessing step. The observation that often only few features (frequencies) are needed to sufficiently characterize a signal supports the idea to first drastically reduce the data's dimensionality before decomposing. The decomposition has to be followed by a lifting of the data to the original time-frequency domain before the inverse transform is applied.
The concept of dimensionality reduction is also known from other applications and there

are plenty of different techniques available (for an overview see [119]). Not all of them are well suited for the application in signal separation as the used decomposition technique might require some extra properties of the data. Even though the high-dimensional data possesses these properties, they are not necessarily conserved beyond the reduction step. The entry-wise non-negativity of the data is such a property since NNMF requires non-negative input data. The high-dimensional time-frequency data is usually non-negative, but the application of an intermediate dimensionality reduction step might cause negative entries in the low-dimensional representation. Thus, there is a need of sophisticated reduction methods which are able to preserve non-negativity.
This is the core motivation for this work. We want to improve the audio signal separation process for single-channel recordings by the use of non-negative dimensionality reduction methods. To this end we proceed as depicted in Figure 2. An input signal

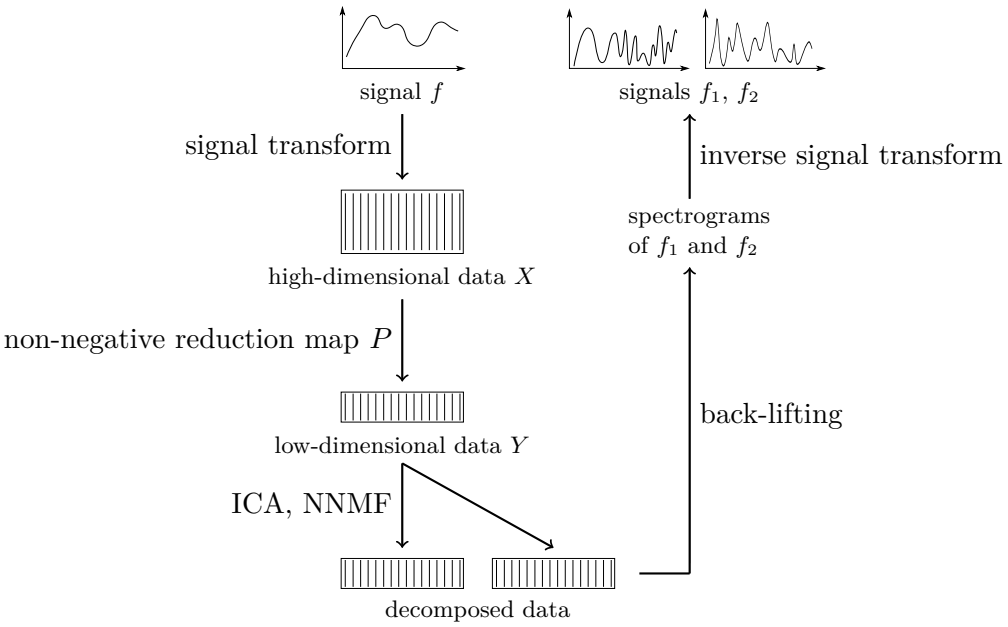


Figure 2: Signal separation with non-negative dimensionality reduction. Before a decomposition technique is applied, the dimensionality of the non-negative data X is reduced by a non-negativity preserving dimensionality reduction method P. This allows a decomposition of the low-dimensional data set Y by methods such as NNMF which require non-negativity input data.

f is converted to a high-dimensional non-negative data matrix X in the time-frequency domain. The application of a reduction map P which preserves the non-negativity of the data leads to a low-dimensional representation Y which can be decomposed by the above mentioned techniques. A suitable lifting transfers the decomposed data back to the high-dimensional space where an inverse signal transform is used to finally obtain the separated source signals f_i.

This procedure is especially well suited for high energy signals as percussion recordings or other transient signals. Usually, this kind of signals is particularly difficult to separate since the different sources have a similar and very wide frequency range and the frequencies within one source occur mostly independent from each other. Although there are successful BSS methods for speech recognition [100] and polyphonic music [22, 24, 79, 123], the separation of this particular class of signals is still a challenge. In [5, 22, 35, 54] the separation of drum tracks has also been studied but the methods are computationally expensive. A sub-band approach for transient signal separation is proposed in [134].
The interaction of dimensionality reduction and signal separation was discussed in a few publications, among them [36, 48, 49, 68, 117]. However, to the best of our knowledge none of these references have discussed or even commented on the need of non-negativity preserving dimensionality reduction for the application in this framework. This aspect was only considered in our works [47, 69].
In resent years, there has been put some effort into the investigation of non-negative dimensionality reduction methods (an overview can be found in [132]) for the application in different contexts. Nonetheless, we are interested in developing new non-negative preserving dimensionality reduction methods since the existing ones are not that suitable for the use in signal separation. Especially non-negative principal component analysis (NNPCA), which has been studied in different forms, is usually formulated with very restrictive constraints involving e.g. sparsity. In the literature, several approaches to NNPCA have been proposed. For example, in [133] and [4] algorithms to compute a local optimal solution of the NNPCA problem can be found and in [92] an extension for the multi-linear case of the latter is discussed. Another idea which uses a non-linear PCA is presented in [89]. There are also some non-linear non-negative dimensionality reduction methods available but they are based on similar sparsity assumptions which do not hold for our application [78, 127, 132].
All in all, this justifies the request for new non-negativity preserving dimensionality reduction methods. A common approach for creating new methods is the modification and improvement of well-established ones. This has the advantage that the analysis of those methods can be partly adopted and known facts can be recycled. Thus, one objective of this work is to provide a general framework how classical dimensionality reduction methods can be reformulated to extend their field of application to situations where the non-negativity of data sets needs to be preserved. If a dimensionality reduction problem is formulated as an optimization on the set of reduction maps, a non-negativity constraint requiring the image of the data set under the reduction map to be non-negative can be added. This makes the optimization even more complex and demands novel solution procedures.
To this end, we propose a *splitting approach* which permits to first solve the well-studied classical dimensionality reduction problem before applying a rotation in order to enforce non-negativity of the low-dimensional data set. Our approach uses a similar idea as Plumbley in [97] where a non-negative ICA algorithm is developed. We will extend and apply this idea to dimensionality reduction settings which can be written as optimization problems with rotationally invariant cost functionals. In this way, we create

non-negativity preserving dimensionality reduction methods. Furthermore, the reduction also needs to fulfill a certain condition to guarantee that angles between data points are not increasing under the reduction. If this is the case, the problem can be solved by our approach and the remaining task is the computation of a solution. For this class of reduction methods, the splitting approach is an elegant way of solving this particular constraint optimization problem.
For the second step of the splitting approach, a suitable rotation map can be constructed for the purpose of transforming the reduced data to the positive orthant of the Cartesian coordinate system. The sought rotation is given by the solution of an auxiliary constraint optimization problem on the group of orthogonal matrices. Due to the rotational invariance, the value of the cost functional is not changed by the rotation and the solution of the remaining optimization problem can be computed as in the non-constrained case. In comparison to other approaches this ansatz is able to compute a global (not necessarily unique) solution of the problem.
For the computation of the rotation we use the special structure of the admissible set of the auxiliary optimization problem. It relies on the theory of Lie groups and associated Lie algebras in order to transfer the optimization problem on the manifold $SO(d)$ of special orthogonal matrices to an optimization in the vector space $\mathfrak{so}(d)$ of skew symmetric matrices. We rigorously derive a steepest descent method on Lie groups which iterates along curves on the manifold starting in the direction of a tangent vector. Usually, it is quite difficult to determine such curves explicitly but the structure of a Lie group offers a simple and efficient way to do so. Similar results can be found in [97] in an application based informal formulation and for Newton's method in [1, 81]. Due to this technique we are able to construct a multiplicative update algorithm on the set of special orthogonal matrices which results in a suitable rotation.
This theory enables us to use non-negative dimensionality reduction as a preprocessing step for NNMF in blind signal separation. We will see that this combination leads to a quite good separation and comes close to the results obtained by PCA and ICA. The coupling of NNPCA and ICA yields similar results as PCA and ICA.

This work is organized as follows.

Chapter 1 of this thesis is concerned with the optimization on Lie groups. In Section 1.1, we discuss some general facts from differential geometry in order to rigorously derive a steepest descent method on Lie groups. In particular, we review differentiable manifolds (Section 1.1.1), Lie groups (Section 1.1.2), Lie algebras (Section 1.1.3) and the exponential map (Section 1.1.4) since the proposed steepest descent algorithm will benefit from the Lie group structure of the admissible set. In Section 1.2, we first recall briefly a steepest descent method in $\mathbb{R}^n$ (Section 1.2.1) before we extend this in Section 1.2.2 to Lie groups (Theorem 1.62). Generalizing an optimization algorithm on an abstract manifold is only the first step. The second step is developing efficient numerical methods which we discuss in Section 1.3. We transfer the concept of line search to Lie groups by searching along descent curves on the manifold instead of straight lines in $\mathbb{R}^n$ (Theorem 1.66). This leads to a multiplicative update algorithm (Algorithm 1.73) which can be efficiently implemented (Section 1.3.1). In Section 1.3.2, we apply the before-developed

theory to an optimization on the Lie group of special orthogonal matrices. This example will play a key role in our approach to non-negative dimensionality reduction (compare Section 2.3.2).

In Chapter 2 we discuss non-negative dimensionality reduction. We start with a gentle introduction to dimensionality reduction in Section 2.1 before we formulate the general dimensionality reduction task as an optimization problem in Section 2.2. In the subsequent subsections we briefly review some linear (principal component analysis in Section 2.2.1 and multidimensional scaling in Section 2.2.2) and non-linear (Isomap in Section 2.2.3 and others in Section 2.2.4) dimensionality reduction methods that fit into this formulation. Later, we will extend some of these methods to non-negativity preserving ones using the approach proposed in this chapter. The non-negative dimensionality reduction problem itself is formulated in Section 2.3, where we start with an example to motivate the need of non-negativity preserving methods (in Section 2.3.1). In Section 2.3.2 we introduce our splitting approach to non-negative dimensionality reduction methods. We state a sufficient condition (Theorem 2.33) which allows to successfully apply this approach to non-negative dimensionality reduction problems. Furthermore, we provide an alternative condition (Theorem 2.38) which relaxes the previous one. To end this chapter, we show in Section 2.4 how this framework applies to different dimensionality reduction methods introduced in Section 2.2. For non-negative principal component analysis (Section 2.4.1), we additionally determine a bound for the lowest dimension to which we can reduce such that the splitting approach can still be used (Theorem 2.41). We also discuss the splitting approach for non-negative multidimensional scaling (Section 2.4.2) and prove that this ansatz is appropriate if the data is lying in a linear subspace (Theorem 2.44).

The last chapter is concerned with applications. In Section 3.1, we explain the signal separation procedure and we briefly review the involved methods. Among them are short-time Fourier transform in Section 3.1.1, dimensionality reduction for signal separation in Section 3.1.2 and decomposition techniques, namely ICA and NNMF, in Section 3.1.3. In the second part of this chapter (Section 3.2), we will discuss some numerical examples. We start with an introduction of the considered examples (Section 3.2.1) before we show and analyze the results.

An alphabetical index of relevant terms and a short summary in English and German can be found at the very end of this work behind the bibliography. All figures in the present work are created by the author.

Acknowledgment

First and foremost, I would like to thank Prof. Dr. Armin Iske, not only for his support and effort as my supervisor but also for introducing me to the international research community. He encouraged me to start this PhD project, to find my own way in mathematics and to attend many conferences. Without him, this thesis would not have been possible.
I am deeply grateful that Prof. Dr. Gerlind Plonka-Hoch agreed on reviewing this thesis and I acknowledge the support of the DFG Priority Program SPP 1324 on *mathematical methods for extracting quantifiable informations from complex systems*.
Furthermore, I gratefully thank Prof. Bruno Torrésani for drawing my attention to the interesting field of signal processing.
Many thanks to my fellow PhD students and colleagues. I enjoyed working and spending time in and outside Geomatikum with you. The discussions with my working group, in particular during extensively long coffee breaks, are unforgettable. I am also most grateful to Benedikt, Matthias and Sebastian for proofreading this thesis. Furthermore, Matthias recorded some audio tracks for testing the algorithm.
Special thanks goes to my family and friends for all their support, especially during the last weeks. I know that it was not always easy! Kinka and Timme, your care and catering was awesome. Last, but by no means least, I want to thank my wonderful partner Arne for his continued and unfailing understanding, support and love. You have shown so much patience with me, you are amazing!

1 Optimization on Lie groups

In the last decades there has been growing interest in optimization methods on sets missing a vector space structure. In particular, the optimization on Lie groups is of major importance as it has various applications, for example in numerical linear algebra [80]. The wide field of examples concerns optimization with matrix constraints, as e.g. orthogonality conditions or conditions concerning the determinant.
For these reasons, numerous optimization algorithms on manifolds have been proposed, for an overview see e.g. [83]. In contrast to the optimization on $\mathbb{R}^n$, the optimization on manifolds encounters more difficulties as these are in general not convex. In particular, straight lines are often not contained in the manifold, which makes line search algorithms and other descent based techniques not directly applicable.
Basically, optimization algorithms on manifolds can be classified into projection and retraction based iterative methods. Projection methods perform updates without taking care about staying in the manifold and project the current iterate after each iteration step back to the manifold as e.g. in [40] or in [103] and for the Grassmannian and Stiefel manifold in [82]. The projection can be done either orthogonal to the update direction (i.e., to the tangent space at the previous step) or orthogonal to the manifold (i.e., to the tangent space at the current step). The former is computationally cheap but difficult to study analytically and the latter is computationally expensive but in general not as costly as retraction methods (see e.g. [77]). Retraction methods (e.g. geodesic flow) however, are the more natural approach as they try to generalize the optimization in vector spaces (see [109] and [102] and the references therein). Here, the basic idea is to optimize by following curves (as e.g. geodesics) on the manifold starting in the direction of a tangent vector in analogy to lines in vector spaces. This analogy permits generalizing standard methods such as steepest descent, Newton, conjugate gradient and others as also possible for the projection approach.
A bottleneck of many retraction methods is the computation of the geodesics themselves because it increases the complexity of standard methods considerably. Considering manifolds that are Lie groups, a certain type of retraction methods - so called Lie group methods - unfold their full potential. In fact, Lie group methods take advantage of the group structure of the manifold which gives them a head start compared to optimization on arbitrary manifolds.
Lie groups possess a certain structure which allows generalizing some nice properties of the optimization on vector spaces. More precisely, the tangent space of a Lie group at the identity can be endowed with an algebraic structure which allows inducing a special Riemannian metric on the Lie group. The structures on both sets are linked in a natural way by the exponential map which makes computation - and thus, optimization - feasible. Furthermore, due to this link there are particular curves on the Lie group

that can be used as retractions and that are computable at reasonable costs. This is why the optimization on Lie groups has gained more attention recently.
A general overview on optimization on matrix manifolds using the retraction approach can be found in Absil et al., see [2]. The same group developed 'Manopt' [13], an open source matlab toolbox for optimization on manifolds which does not use the particular structure of Lie groups. Newton's and conjugate gradient methods on Grassmannian and Stiefel manifolds are discussed in [32] (and the references therein) whereas Newton's method on Lie groups has been studied in [81]. A survey on Lie group methods and their applications to ODEs can be found in [61] and the application of similar methods in control theory is discussed in [15]. A very recent result concerning extremum seeking algorithms on manifolds can be found in [112]. This list does not claim to be complete, further references can be found e.g. in [83].
In this work, we wish to address the optimization on Lie groups using the method of steepest descent. We focus on the rigorous derivation of the algorithm, what in this way has not been done before, to the best of our knowledge. In comparison to [40], we exploit the Lie group structure of the manifold in order to reduce the computational cost. Our approach is inspired by a paper of Plumbley [97], where the optimization on the Lie group of special orthogonal matrices $SO(n)$ is described in an informal way, but a fundamental derivation of the underlying mathematics is missing. A similar but different approach can be found in [113] where an optimal rotation on $SO(3)$ is computed by a trust-region method using the matrix exponential and a description of $SO(3)$ based on quaternions.
In contrast, the method we propose is not restricted to a particular class of Lie groups and it is derived from scratch. In a certain sense it can be seen as a generalization of [96] and [98]. It is designed for optimization problems with arbitrary but smooth cost functional and a Lie group as constraint set and it is primarily based on the link between a Lie group and its associated Lie algebra. This is the core idea of all Lie group methods since both sets are linked in a canonical way by the exponential map. This allows outsourcing some steps of the optimization procedure to the Lie algebra, where computation is more comfortable due to the vector space structure.
As descent direction we have chosen the negative gradient. Certainly, other descent directions can be used and might lead to more sophisticated algorithms but this was not the objective of this work. For other descent directions compare e.g. [1] and [80].

This chapter concerns the optimization on Lie groups. In Section 1.1, we give an overview on some basic concepts of differential geometry, in particular Lie groups, in order to introduce the subject and to fix the terminology. The content of this section can be found in many text books but for the sake of completeness we recall among others the definition and main properties of differentiable manifolds in 1.1.1, Lie groups in 1.1.2 and Lie algebras in 1.1.3 and in 1.1.4 we introduce the exponential map. A short summary of this theoretical introduction is given in 1.1.5. In between we discuss some examples to make the theory more easily accessible and in view of the applications in this work. This theoretical part may seem a bit lengthy but we think that the effort of understanding the theory pays off in the following sections. Section 1.2 concerns a key part of this

work, namely the generalization of gradient descent methods to Lie groups. We first briefly recall the gradient descent in 1.2.1 before we come to its formulation on Lie groups in 1.2.2. In this subsection, we also introduce the algorithm further discussed and developed in the next and last section of this chapter. In Section 1.3.1, we modify the formerly derived algorithm to make computation feasible and efficient. This is followed by an example from non-negative dimensionality reduction in 1.3.2. Here we consider an optimization on the set of rotation matrices which will be a core part of our approach to non-negativity preserving dimensionality reduction methods introduced in Section 2.3. At the end of Section 1.3, we will summarize our achievements in 1.3.3.

1.1 Manifolds, Lie groups and Lie algebras

This section aims to give an overview of some basic ideas concerning differential geometry and in particular Lie groups.
We will start with a gentle introduction to differentiable manifolds with a focus on the tangent space, a linear approximation of a manifold at a point. The tangent space plays an important role, since it permits a generalization of the differential to manifolds and defines the tangent bundle as the union of all tangent spaces. We will describe a tangent vector's action on smooth functions by generalizing the concept of directional derivatives to smooth curves on the manifold. Furthermore, we introduce the Riemannian metric to endow the tangent spaces with inner products.
Next, we will consider differentiable manifolds which have additionally a group structure. These so called Lie groups have some very nice properties. Introducing vector fields, i.e., maps between a Lie group and its tangent bundle, allows us to define the associated Lie algebra of a Lie group as the set of its left-invariant vector fields.
Furthermore, we will see that the Lie algebra is isomorphic to the tangent space at the identity and naturally linked to the Lie group by the exponential map. This map defines curves on the Lie group, so called 1-parameter subgroups, which can be used for optimization.
At the end of this section, we point out the consequences of the presented theory for matrix groups. In particular, we will see that the exponential map is basically given by the matrix exponential.
All facts of this section can be found in the books [14] by Bredon, [126] by Warner, [50] by Hall and [120] by Varadarajan. For a deeper insight we refer to the same sources.

1.1.1 Manifolds

A basic structure in topology is the *topological space* which is a pair $(X, \mathcal{T})$ consisting of a set X and a topology $\mathcal{T}$ on X, where the topology $\mathcal{T}$ is a family of subsets of X (called open sets) fulfilling the following three axioms. 1. The empty set and the set X itself are open sets. 2. The intersection of a finite number of open sets is open. 3. The union of (finitely or infinitely many) open sets is open. A basis of the topological space $(X, \mathcal{T})$ is a subset of $\mathcal{T}$ such that every open set in $\mathcal{T}$ can be written as a union of elements of the basis. Recall that a topological space is second-countable if its topology has a

countable basis. Furthermore, a topological space is said to be *Hausdorff* if distinct points have disjoint neighborhoods, i.e., for all $x, y \in X$ with $x \neq y$ there exist disjoint open neighborhoods.

Definition 1.1. A *topological manifold* $\mathcal{M}$ of dimension n is a second-countable, Hausdorff topological space that is locally homeomorphic to $\mathbb{R}^n$.

Remark 1.2. From the definition it follows that a topological space looks locally like a piece of $\mathbb{R}^n$. Locally homeomorphic to $\mathbb{R}^n$ or *locally Euclidean* means that for all $p \in \mathcal{M}$ there exists an open neighborhood $U_p \subset \mathcal{M}$ and an injective, continuous map $\phi\colon U_p \to \mathbb{R}^n$ such that the inverse map $\phi^{-1}\colon \phi(U_p) \to U_p$ is also continuous.
The homeomorphism ϕ called *coordinate map* or *chart* induces a local coordinate system on U_p through the coordinate functions x_i given by $x_i = \pi_i \circ \phi$, $i = 1, \dots n$. Here, π_i is the projection on the ith component.
Consider a second chart ψ on a neighborhood V_p of $p \in \mathcal{M}$ with coordinate functions $y_i = \pi_i \circ \psi$. Then, two different coordinate systems (U_p, ϕ) and (V_p, ψ) are induced on the neighborhood $U_p \cap V_p$ of p and any point in the intersection has two coordinate descriptions. The *change of coordinates* from one system to the other is then defined by the map $\psi \circ \phi^{-1}$ since we have $(y_1, \dots y_n) = \psi = \psi \circ \phi^{-1} \circ \phi = \psi \circ \phi^{-1}(x_1, \dots x_n)$. The change of coordinates is continuous since ψ and ϕ^{-1} are so.

Remark 1.3. Locally Euclidean spaces do not need to be Hausdorff. To see this, consider the *line with two origins*, which is created by replacing the origin of the real line by two points. Then, any open neighborhood of either points consist of all nonzero numbers of an interval around zero. This space is not Hausdorff, because we cannot find disjoint neighborhoods for the two origins, but it is locally homeomorphic to $\mathbb{R}$.

Remark 1.4. In general, it is not necessary to require second-countability. However, this property guarantees that the manifold can be embedded in a finite dimensional Euclidean space. In fact, it gives us a partition of unity which is useful to pass from the local coordinate maps to global properties (e.g. Theorem 1.11).

Remark 1.5. A topological manifold is not necessarily connected, i.e., it might be the disjoint union of two non-empty open subsets. The *connected components* of a topological manifold $\mathcal{M}$ are its maximal connected subsets and a topological manifold is called *simply connected* if it is connected and every loop on $\mathcal{M}$ is null-homotopic (i.e., it can be contracted to a point).

Using the induced coordinate systems and the change of coordinates one can further classify manifolds.

Definition 1.6. An *n-dimensional differentiable manifold of class C^k* $(1 \leq k \leq \infty)$ is a topological manifold $\mathcal{M}$ of dimension n together with a collection of local coordinate systems $\{(U_\alpha, \phi_\alpha)\colon \alpha \in A\}$ with the following properties:

(i) every point in $\mathcal{M}$ is contained in at least one U_α, i.e., $\bigcup_{\alpha \in A} U_\alpha = \mathcal{M}$,

(ii) the change of coordinates $\phi_\alpha \circ \phi_\beta^{-1}$ from the set $\phi_\beta(U_\alpha \cap U_\beta) \subset \mathbb{R}^n$ onto the set $\phi_\alpha(U_\alpha \cap U_\beta) \subset \mathbb{R}^n$ for all $\alpha, \beta \in A$ is C^k,

(iii) the collection of coordinate systems is maximal with respect to (ii): if (U, ϕ) is a coordinate system such that $\phi \circ \phi_\alpha^{-1}$ and $\phi_\alpha \circ \phi^{-1}$ are C^k for all $\alpha \in A$, then (U, ϕ) belongs to the collection.

Remark 1.7. A manifold is called *smooth* if it is of class C^∞. The collection of coordinate systems is called *differentiable structure of class* C^k or *atlas*.

Definition 1.8. The continuous map $\varphi\colon \mathcal{M} \to \mathcal{N}$ is *k-differentiable* (or *smooth* if $k = \infty$) if and only if $\phi \circ \varphi \circ \psi^{-1}$ is k-differentiable for each coordinate map ψ of $\mathcal{M}$ and ϕ of $\mathcal{N}$. Then, we write $\varphi \in C^k(\mathcal{M}, \mathcal{N})$ or just $\varphi \in C^k$. For $C^k(\mathcal{M}, \mathbb{R})$ we usually write $C^k(\mathcal{M})$.

Remark 1.9. For a k-differentiable function $f\colon \mathcal{M} \to \mathbb{R}$ we write $\frac{\partial}{\partial x_i} f(p)$ to denote the partial derivative of $f \circ \phi^{-1}$ with respect to the ith argument evaluated at $\phi(p)$.

Now that we have introduced smooth mappings between manifolds, we are able do define some other important objects. Together with the linearization concept of the differential (which we introduce in the next subsection, see (1.1)) we can define submanifolds and embeddings. Let $\varphi\colon \mathcal{M} \to \mathcal{N}$ be a smooth mapping between manifolds. If the differential $(\mathrm{d}\varphi)_p\colon T_p\mathcal{M} \to T_{\varphi(p)}\mathcal{N}$ is injective for each $p \in \mathcal{M}$, it is called an *immersion*. If furthermore φ is injective, the pair $(\mathcal{M}, \varphi)$ is a *submanifold*. Finally, φ is an *embedding* if it is an injective immersion which is also a homeomorphism onto its image. That is, φ is open as a map into $\varphi(\mathcal{M})$ with the relative topology. Moreover, φ is a *diffeomorhism* if φ is bijective and φ^{-1} is C^∞. Last but not least, φ is a *submersion* if $(\mathrm{d}\varphi)_p$ is surjective for each $p \in \mathcal{M}$.

Remark 1.10. For φ being an embedding it is essential that it is an injective immersion. If φ would just be a homeomorphism on its image, only the topological and not the differentiable structure would be inherited.

For dimensionality reduction, high-dimensional data sets are considered as points on a manifold of dimension n. But, since it is easier to handle these sets in an Euclidean space, the following embedding theorem is very useful. It allows finding a Euclidean space (namely $\mathbb{R}^{2n}$) in which the manifold (and thus the data) can be embedded.

Theorem 1.11 (Whitney Embedding Theorem). *Every smooth manifold $\mathcal{M}$ of dimension n can be smoothly embedded in $\mathbb{R}^{2n}$, i.e., there exists a smooth embedding $g\colon \mathcal{M} \to \mathbb{R}^{2n}$.*

Proof. See [14]. □

Remark 1.12. This bound is sharp as for example the real projective plane, a 2-dimensional manifold, cannot be embedded in $\mathbb{R}^3$ without intersecting itself. The real projective plane can be thought of as the object we obtain by gluing a disk to the edge of the Möbius stripe (see [124]).

One way to construct an n-dimensional manifold $\mathcal{M}$ is to consider the special case of embedded submanifolds of $\mathbb{R}^N$. If, for example, $\mathcal{M}$ is a surface (i.e., $N = n + 1$), the tangent space at a point $p \in \mathcal{M}$ is the collection of all vectors starting in p and being

tangential to $\mathcal{M}$. In this case, the tangential space can be thought of as a copy of $\mathbb{R}^n$ attached to p. This graphic description of the tangent space relies on the fact that we have $\mathcal{M}$ embedded in $\mathbb{R}^N$. Nevertheless, there are also descriptions of the tangent space not depending on having the manifold a priori embedded in some Euclidean space.
The above heuristics are in fact nothing else but defining a tangential vector to a point p as the derivative of a curve on the manifold at the point where it passes through p. If the manifold is not embedded in an Euclidean space, the derivative of such a curve can be defined using the coordinate maps to locally transfer the problem.
In the further course of this chapter we wish to apply a tangent vector's action on smooth functions $f\colon U \subseteq \mathcal{M} \to \mathbb{R}$. Therefore, we will elaborate a slightly different heuristic which leads to an equivalent definition. The idea is to generalize the concept of directional derivatives (well-known from real analysis) to curves on manifolds.
Let $\gamma\colon\ ]-\epsilon, \epsilon[\to \mathcal{M}$ be a smooth curve with $\gamma(0) = p$ and let $\mathcal{F}_p(\mathcal{M})$ be the set of germs of smooth real-valued functions defined on a neighborhood of p. The *germ at p* is the equivalence class of smooth functions defined by the relation $f_1 \sim f_2$ if there exists an open neighborhood U of p with $f_1|_U = f_2|_U$. In the following, we will not distinguish between f and its equivalence class. The *tangent vector to the curve γ at $t = 0$* is defined as the mapping

$$\dot\gamma(0)\colon \mathcal{F}_p(\mathcal{M}) \to \mathbb{R}, \text{ with } \dot\gamma(0)f = \frac{\mathrm{d}}{\mathrm{d}t} f \circ \gamma(t)\Big|_{t=0}.$$

This definition of a tangent vector to a curve allows to formally define tangent vectors to a manifold at a point $p \in \mathcal{M}$.

Definition 1.13. A *tangent vector ξ_p to $p \in \mathcal{M}$* is a mapping

$$\xi_p\colon \mathcal{F}_p(\mathcal{M}) \to \mathbb{R}, \text{ with } \xi_p f = \dot\gamma(0)f,$$

where $\gamma\colon\ ]-\epsilon, \epsilon[\to \mathcal{M}$ is any curve with $\gamma(0) = p$.
The set of all tangent vectors at p is denoted by $T_p\mathcal{M}$ and it is called the *tangent space of $\mathcal{M}$ at p*.

Remark 1.14. The tangent vector to γ is defined as a mapping and not as time derivative $\lim\limits_{\tau\to 0} \frac{\gamma(\tau)-\gamma(0)}{\tau}$ as perhaps expected. However, if the manifold is embedded in an Euclidean space, this expression is well-defined and known as

$$\gamma'(0) = \frac{\mathrm{d}}{\mathrm{d}t}\gamma(t)\Big|_{t=0}.$$

The link between both, $\dot\gamma(0)$ and $\gamma'(0)$ is given by

$$\dot\gamma(0)f = \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma)\Big|_{t=0} = \mathrm{d}f(\gamma(0)) \cdot \gamma'(0).$$

This shows that $\{\gamma'\colon \gamma \text{ curve in } \mathcal{M}, \gamma(0) = p\}$ is isomorphic to $T_p\mathcal{M}$. Therefore, in our application we can identify both sets. For general manifolds, however, we need to stick to the abstract definition.

Clearly, there are different curves $\gamma_1 \neq \gamma_2$ defining the same tangent vector at p. Therefore, it is appropriate to consider equivalence classes of such curves instead of the curves themselves. For $p \in \mathcal{M}$ we define the equivalence relation on the set of curves γ like the one in Definition 1.13: Two curves γ_1 and γ_2 are equivalent if and only if there is a coordinate system (U_α, ϕ_α) so that $(\phi_\alpha \circ \gamma_1)'(0) = (\phi_\alpha \circ \gamma_2)'(0)$.

Remark 1.15. The tangent space is an n-dimensional vector space. This can be seen by considering for a chart ϕ the linear map $(\mathrm{d}\phi)_p \colon T_p\mathcal{M} \to \mathbb{R}^n$ defined as $(\mathrm{d}\phi)_p [\xi_p] = (\phi \circ \gamma)'(0)$, where γ is a curve defining ξ_p. This map is bijective (injective by construction of the equivalence relation and surjective since for a given vector $v \in \mathbb{R}^n$ the curve $\gamma = \phi^{-1} \circ g$ is in the preimage of v, where $g \colon]-\epsilon, \epsilon[\to \mathbb{R}^n$ with $g(t) = \phi(p) + tv$) and thus, $(\mathrm{d}\phi)_p$ induces the structure of an n-dimensional vector space on the tangent space $T_p\mathcal{M}$. This construction does not depend on the choice of (U_α, ϕ_α).

The linear map $(\mathrm{d}\phi)_p$ in Remark 1.15 is called the *differential of ϕ at p*. This concept can be generalized to C^k-maps between differentiable manifolds: For a differentiable map $\varphi \colon \mathcal{M} \to \mathcal{N}$ we define the differential

$$\begin{aligned} (\mathrm{d}\varphi)_p \colon T_p\mathcal{M} &\to T_{\varphi(p)}\mathcal{N} \\ \xi_p &\mapsto (d\varphi)_p [\xi_p] , \end{aligned} \tag{1.1}$$

where $(\mathrm{d}\varphi)_p [\xi_p] f = \xi_p(f \circ \varphi)$ for $f \in \mathcal{F}_{\varphi(p)}(\mathcal{N})$. The application of $(\mathrm{d}\varphi)_p$ to a tangent vector ξ_p is also called *pushforward of ξ_p along φ*.

Remark 1.16. Since the tangent vectors are operating on smooth functions $f \colon \mathcal{M} \to \mathbb{R}$, it makes sense to verify if the product rule known from real analysis also holds for the directional derivative on manifolds. For $f, g \in \mathcal{F}_p(\mathcal{M})$ it holds

$$\begin{aligned} \xi_p(f \cdot g) = \dot{\gamma}(0)(f \cdot g) &= \frac{\mathrm{d}}{\mathrm{d}t} f \cdot g \circ \gamma \Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t} \left((f \circ \gamma) \cdot (g \circ \gamma) \right) \Big|_{t=0} \\ &= \left[\frac{\mathrm{d}}{\mathrm{d}t} (f \circ \gamma) \cdot (g \circ \gamma) + (f \circ \gamma) \cdot \frac{\mathrm{d}}{\mathrm{d}t} (g \circ \gamma) \right] \Big|_{t=0} \\ &= g(p) \cdot (\xi_p f) + f(p) \cdot (\xi_p g). \end{aligned}$$

Remark 1.17. Furthermore, we verify the *chain rule* for smooth functions. Let $\varphi \colon \mathcal{P} \to \mathcal{N}$ and $\psi \colon \mathcal{M} \to \mathcal{P}$, $p \in \mathcal{M}$, $f \in \mathcal{F}_{\varphi(\psi(p))}(\mathcal{N})$ and $\xi_p \in T_p\mathcal{M}$, then it holds

$$\begin{aligned} (\mathrm{d}\varphi \circ \psi)_p [\xi_p] f &= \xi_p(f \circ \varphi \circ \psi) \\ &= (\mathrm{d}\psi)_p [\xi_p] f \circ \varphi \\ &= (\mathrm{d}\varphi)_{\psi(p)} [(\mathrm{d}\psi)_p [\xi_p]] f \\ &= (\mathrm{d}\varphi)_{\psi(p)} \circ (\mathrm{d}\psi)_p [\xi_p] f \end{aligned}$$

and thus

$$(\mathrm{d}\varphi \circ \psi)_p = (\mathrm{d}\varphi)_{\psi(p)} \circ (\mathrm{d}\psi)_p.$$

We will show that the collection of all tangent vectors ξ_p to a differentiable manifold can be endowed with a differentiable structure and thus builds itself a differentiable manifold. For the manifold $\mathcal{M}$ we define $T(\mathcal{M}) = \bigsqcup_{p\in\mathcal{M}} T_p\mathcal{M}$, the set of all pairs (p, ξ_p) with $\xi_p \in T_p\mathcal{M}$. Then, there is a canonical projection on the manifold $\pi\colon T(\mathcal{M}) \to \mathcal{M}$ with $\pi(p, \xi_p) = p$. Consider a coordinate system (U_α, ϕ_α) in $\mathcal{M}$, then we can define a map $\tilde{\phi}_\alpha$ on $\pi^{-1}(U_\alpha) \subset T(\mathcal{M})$

$$\begin{aligned}\tilde{\phi}_\alpha\colon \pi^{-1}(U_\alpha) &\to \phi_\alpha(U_\alpha) \times \mathbb{R}^n \subset \mathbb{R}^{2n}\\ (p, \xi_p) &\mapsto (\phi_\alpha(p), (\mathrm{d}\phi_\alpha)_p\,[\xi_p])\,.\end{aligned}$$

With the maps $\tilde{\phi}_\alpha$ we can define a canonical basis $\{\tilde{\phi}_\alpha^{-1}(W)\colon W \text{ open in } \mathbb{R}^{2n} \text{ and } \alpha \in A\}$ for a topology on $T(\mathcal{M})$ and thus, restricting its range to its image $\tilde{\phi}_\alpha$ is a homeomorphism. This shows that $T(\mathcal{M})$ is indeed a $2n$-dimensional, second-countable, locally Euclidean space. Furthermore, the definition of the coordinate maps yields a smooth change of coordinates $\tilde{\phi}_\alpha \circ \tilde{\phi}_\beta^{-1}$. Hence, the maximal collection containing $\{(\pi^{-1}(U_\alpha), \tilde{\phi}_\alpha)\colon \alpha \in A\}$ forms a differentiable structure on $T(\mathcal{M})$ (compare Definition 1.6).

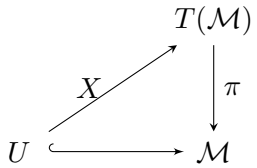
Definition 1.18. The smooth manifold $T(\mathcal{M})$ is called *tangent bundle of* $\mathcal{M}$.

The concept of a differential $(\mathrm{d}\varphi)_p$ of a mapping $\varphi\colon \mathcal{M} \to \mathcal{N}$ at a point p can be used to define a mapping on the corresponding tangent bundles

$$\begin{aligned}\mathrm{d}\varphi\colon T(\mathcal{M}) &\to T(\mathcal{N})\\ \mathrm{d}\varphi\,[p, \xi_p] &= (\varphi(p), (\mathrm{d}\varphi)_p\,[\xi_p])\,.\end{aligned} \tag{1.2}$$

The *differential* $\mathrm{d}\varphi$ inherits the properties of $(\mathrm{d}\varphi)_p$ as for example the chain rule ($\mathrm{d}\varphi\circ\psi = \mathrm{d}\varphi \circ \mathrm{d}\psi$).
Now, we can study mappings from a manifold to its tangent bundle and we have an idea what smoothness is in this setting. Such mappings will play an important role in the definition of a Lie algebra.

Definition 1.19. A *vector field* X *on an open set* $U \subset \mathcal{M}$ is a map $X\colon U \to T(\mathcal{M})$ such that $\pi \circ X = \mathbf{id}|_U$, called a lifting of U into $T(\mathcal{M})$, i.e., the following diagram commutes



Remark 1.20. For a point $p \in U$, the image is denoted $X(p) = (p, X_p)$, where X_p is an element of $T_p\mathcal{M}$. The set of smooth vector fields on an open set U forms a vector space over $\mathbb{R}$. Here, the vector space operations act only on the second part of the tuple (p, X_p) and therefore, we might sometimes write X_p instead of (p, X_p). The action of a vector field X on $C^\infty(\mathcal{M})$ is defined as

$$(Xf)\,(p) = X_p(f).$$

We have seen in Remark 1.15 that each tangent space $T_p\mathcal{M}$ of $\mathcal{M}$ is a vector space. Thus, it is natural to endow them with inner products $\langle \cdot, \cdot \rangle_{T_p\mathcal{M}}$ which clearly will depend on p. Furthermore, if this dependence on p is smooth, i.e., for any two smooth vector fields X and Y the mapping $p \mapsto \langle X_p, Y_p \rangle_{T_p\mathcal{M}}$ is smooth, we call the family $\left(\langle \cdot, \cdot \rangle_{T_p\mathcal{M}}\right)_{p\in\mathcal{M}}$ a *Riemannian metric on* $\mathcal{M}$ and $\mathcal{M}$ a *Riemannian manifold*.

Remark 1.21. Even though the family is called 'metric', it is not a metric in the classical sense. Nevertheless, it induces a metric on $\mathcal{M}$ similarly to the Euclidean inner product on $\mathbb{R}^n$. Roughly speaking, the distance of two points in $\mathcal{M}$ is defined as the length of the shortest curve $\gamma\colon \mathbb{R} \to \mathcal{M}$ connecting both points. This distance measure induces the same topology on $\mathcal{M}$ as used for the definition of the manifold.

In the following, we would like to define a product-like operation on the set of smooth vector fields in order to later on endow a subset of this vector space with the structure of an algebra. To this end, we introduce derivations δ on $C^\infty(\mathcal{M})$ as linear maps $\delta\colon C^\infty(\mathcal{M}) \to C^\infty(\mathcal{M})$ which fulfill the product rule

$$\delta(f \cdot g) = \delta(f) \cdot g + f \cdot \delta(g).$$

The vector space of all derivations on $C^\infty(\mathcal{M})$ is denoted by $\mathcal{D}(\mathcal{M})$. We observe that a vector field X defines in a natural way a derivation

$$\begin{aligned} L_X\colon C^\infty(\mathcal{M}) &\to C^\infty(\mathcal{M}) \\ f &\mapsto L_X(f), \end{aligned}$$

where $L_X(f)(p) = (\mathrm{d}f)_p\,[X_p]$. The mapping $X \mapsto L_X$ is an isomorphism of vector spaces between the space of smooth vector fields and $\mathcal{D}(\mathcal{M})$ (see [94]). Note that in general the composition of derivations is not a derivation itself since the product rule does not hold:

$$\delta_1 \circ \delta_2(f \cdot g) = \delta_1 \circ \delta_2(f) \cdot g + \delta_2(f) \cdot \delta_1(g) + \delta_1(f) \cdot \delta_2(g) + f \cdot \delta_1 \circ \delta_2(g).$$

In contrast, it can easily be seen from this that $\delta_1 \circ \delta_2 - \delta_2 \circ \delta_1$ is a derivation. This is an important observation. Now we can deduce that for two smooth vector fields X and Y there exists a smooth vector field $[X, Y]$ with

$$L_{[X,Y]} = L_X \circ L_Y - L_Y \circ L_X, \tag{1.3}$$

due to the isomorphy of the space of smooth vector fields and $\mathcal{D}(\mathcal{M})$.

Definition 1.22. If X and Y are smooth vector fields on $\mathcal{M}$, the vector field $[X, Y]$ is called *Lie bracket* of X and Y or *commutator*.

Remark 1.23. The Lie bracket of X and Y is anti-commutative ($[X, Y] = -[Y, X]$) and the Jacoby identity holds: $[[X, Y], Z] + [[Y, Z], X] + [[Z, X], Y] = 0$, for all smooth vector fields X, Y, Z on $\mathcal{M}$. A vector space with a bilinear operation which is anti-commutative and satisfies the Jacobi identity is called a *Lie algebra*. We will give more information on this in the next subsections.

1.1.2 Lie groups

Lie groups are a very important class of differentiable manifolds. They are closely related to Lie algebras since there is a natural link between a Lie group and its Lie algebra of left-invariant vector fields. Due to this, optimization on Lie groups will turn out to be practicable in an elegant way.

Definition 1.24. A *Lie group* G is a differentiable manifold which additionally has a group structure such that the group product $(g_1, g_2) \mapsto g_1 g_2$ and the inverse map $g \mapsto g^{-1}$ are smooth.

Remark 1.25. Group structure means that there is a map $G \times G \to G$ (also called group operation or group product) which is associative and which admits an identity element e and inverse elements.

Remark 1.26. Instead of requiring the group product and the inverse map to be smooth, it is sufficient to require the map $G \times G \to G$ defined by $(g_1, g_2) \mapsto g_1 g_2^{-1}$ to be smooth. It is easy to see that both definitions are equivalent.

Example 1.27 (General linear group). The general linear group $GL(n, \mathbb{R})$, the set of all $n \times n$ non-singular matrices (i.e., with non-zero determinant), is a Lie group. To see this, first of all notice that the set of all $n \times n$ matrices $M(n, \mathbb{R})$ is diffeomorphic to $\mathbb{R}^{n^2}$ and that the restriction of this diffeomorphism φ to $GL(n, \mathbb{R})$ is an injective immersion since $GL(n, \mathbb{R})$ is an open subset of $M(n, \mathbb{R})$. Thus $GL(n, \mathbb{R})$ is a (differentiable) submanifold of $\mathbb{R}^{n^2}$. Its dimension is n^2 since the diffeomorphism $\varphi|_{GL(n,\mathbb{R})}$ is also a coordinate map on the open set $GL(n, \mathbb{R})$. Furthermore, the matrix multiplication defines a smooth group product on $GL(n, \mathbb{R})$. We also observe, that $GL(n, \mathbb{R})$ has two connected components: the two sets of matrices with determinant less than zero and greater than zero. These two sets are open and connected since the determinant is continuous from $GL(n, \mathbb{R})$ to $\mathbb{R}$.

Definition 1.28. (H, φ) is a *Lie subgroup* of the Lie group G if

(i) H is a Lie group,

(ii) (H, φ) is a submanifold of G, i.e., $\varphi\colon H \to G$ is an injective immersion,

(iii) $\varphi\colon H \to G$ is a group homomorphism.

Example 1.29 (Orthogonal group). The *orthogonal group* $O(n, \mathbb{R}) \subset GL(n, \mathbb{R})$ or just $O(n)$ is the set of orthogonal matrices (i.e., $A^{-1} = A^T$ for $A \in O(n)$). The orthogonal group is a Lie subgroup of $GL(n, \mathbb{R})$: as before we take φ to be the inclusion map to show (ii), i.e., that $O(n)$ is a submanifold of $GL(n, \mathbb{R})$. For (i) we use that the product of two orthogonal matrices is indeed orthogonal and (iii) is obvious. In Remark 1.31 we will see that the dimension of $O(n)$ is $\frac{n(n-1)}{2}$.

Example 1.30 (Special orthogonal group). Furthermore, $O(n)$ has two connected components: the sets of orthogonal matrices with determinant -1 and 1. The latter is the set of *special orthogonal matrices* $SO(n)$, also called the set of rotation matrices, which

is a connected Lie subgroup of $O(n)$. In contrast, the set of orthogonal matrices with determinant -1 is not a Lie subgroup since it is not closed under multiplication.

Remark 1.31. Using the submersion Theorem (see e.g. [2] p. 26 or [105] p. 53) which basically reads

> the preimage $\varphi^{-1}(q)$ of a smooth mapping $\varphi\colon \mathcal{M} \to \mathcal{N}$, with $(\mathrm{d}\varphi)_p$ surjective for all $p \in \varphi^{-1}(q)$, is either empty or a differentiable manifold of dimension $\dim(\mathcal{M}) - \dim(\mathcal{N})$

we can show that the dimension of $SO(n)$ is $\frac{n(n-1)}{2}$.
Define $GL^+(n) \coloneqq \{A \in GL(n,\mathbb{R})\colon \det(A) > 0\}$ and $\mathrm{Sym}(n) \coloneqq \{B \in M(n,\mathbb{R})\colon B^T = B\}$ and consider $\varphi\colon GL^+(n) \to \mathrm{Sym}(n)$ given by $\varphi(A) = A^TA - \mathbf{Id}_n$. We observe that φ is differentiable with $(\mathrm{d}\varphi)_A\colon T_AGL^+(n) \to T_{\varphi(A)}\,\mathrm{Sym}(n)$ and

$$(\mathrm{d}\varphi)_A\,[\xi_A]\,f = \xi_A(f\circ\varphi) = \mathrm{d}f(A^TA - \mathbf{Id}_n)\cdot(\gamma'(0)^TA + A^T\gamma'(0)),$$

where $f \in \mathcal{F}_{\varphi(A)}\,\mathrm{Sym}(n)$ and γ a curve defining ξ_A. Now, we prove that $(\mathrm{d}\varphi)_{\tilde{A}}$ is surjective for $\tilde{A} \in \varphi^{-1}(0)$, i.e., for $\tilde{A}$ orthogonal with $\det(\tilde{A}) = 1$. To this end, let $\xi_0 \in T_0\,\mathrm{Sym}(n)$ be a tangent vector with defining curve β. We construct a preimage $\xi_{\tilde{A}}$ of ξ_0 using $\gamma(t) = \frac{1}{2}\tilde{A}\beta(t) + \tilde{A} \in GL^+(n)$ as defining curve. We compute

$$\begin{aligned}
(\mathrm{d}\varphi)_{\tilde{A}}\,[\xi_{\tilde{A}}]\,f &= \mathrm{d}f(\tilde{A}^T\tilde{A} - \mathbf{Id}_n)\cdot\frac{1}{2}\left(\left(\tilde{A}\beta'(0)\right)^T\tilde{A} + \tilde{A}^T\tilde{A}\beta'(0)\right)\\
&= \mathrm{d}f(0_n)\cdot\frac{1}{2}\left(\beta'(0)^T + \beta'(0)\right)\\
&= \mathrm{d}f(0_n)\cdot\beta'(0)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}(f\circ\beta)\Big|_{t=0}\\
&= \xi_0 f.
\end{aligned}$$

This shows that $(\mathrm{d}\varphi)_{\tilde{A}}$ is surjective and thus, we can apply the above mentioned theorem. This yields

$$\begin{aligned}
\dim(SO(n)) = \dim(\varphi^{-1}(0)) &= \dim(GL^+(n)) - \dim(\mathrm{Sym}(n))\\
&= n^2 - \frac{n(n+1)}{2}\\
&= \frac{n(n-1)}{2}.
\end{aligned}$$

Here, we used the facts that $GL^+(n) \subset \mathbb{R}^{n^2}$ is an open subset and $\mathrm{Sym}(n)$ an $\frac{n(n+1)}{2}$-dimensional vector space.

The special orthogonal group plays a very important role in this work. In order to compute a low-dimensional representation of high-dimensional time-frequency data which preserves the non-negativity of the input data, we will have to solve an optimization problem on the set $SO(n)$, see Section 1.3.2 and 2.3. To carry out this optimization, we will use the relation between $SO(n)$ and its Lie algebra. We will now introduce

left-invariant vector fields which are essential for the definition of the Lie algebra of a Lie group.

Definition 1.32. For $g \in G$, the *left-translation by g* is the diffeomorphism $\ell_g \colon G \to G$ defined by $\ell_g(h) = gh$. A vector field X on G is called *left-invariant* if for each $g \in G$ we have

$$\mathrm{d}\ell_g \circ X = X \circ \ell_g. \tag{1.4}$$

This means that the following diagram commutes

$$\begin{array}{ccc} G & \xrightarrow{\ell_g} & G \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X} \\ T(G) & \xrightarrow[\mathrm{d}\ell_g]{} & T(G) \end{array}$$

Note that a left-invariant vector field X is uniquely defined by its value at the identity $e \in G$ since

$$X(g) = X \circ \ell_g(e) \overset{(1.4)}{=} \mathrm{d}\ell_g \circ X(e), \qquad \text{for all } g \in G.$$

Remark 1.33. The vector field X in the above definition is not assumed to be smooth. However, it can be shown that left-invariant vector fields are smooth.
The set of all left-invariant vector fields on a Lie group G will be denoted by $\mathfrak{g}$.

Note that analogously right-invariant vector fields can be defined and all further considerations of this work can also be done for right-invariant vector fields.

Example 1.34. In Example 1.30, we introduced the Lie group $SO(n)$ of special orthogonal matrices. Let us now identify the left-invariant vector fields of $SO(2)$ in order to illustrate the above introduced theory. Here, we stick to $n = 2$ since there is a simple description of $SO(2) = \{ \left(\begin{smallmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{smallmatrix}\right) \colon \alpha \in \mathbb{R}\}$.

As a first step, we compute the tangent space $T_pSO(2)$ for $p = \left(\begin{smallmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{smallmatrix}\right) \in SO(2)$. Therefore, let $\beta \colon]-\epsilon, \epsilon[\to \mathbb{R}$ with $\beta(0) = \rho$ and $\beta'(0) = b$ be a smooth curve such that $\gamma \colon]-\epsilon, \epsilon[\to SO(2)$ with $\gamma(t) = \left(\begin{smallmatrix} \cos\beta(t) & -\sin\beta(t) \\ \sin\beta(t) & \cos\beta(t) \end{smallmatrix}\right)$ defines also a smooth curve. Then,

$$\gamma'(t) = \beta'(t) \left(\begin{smallmatrix} -\sin\beta(t) & -\cos\beta(t) \\ \cos\beta(t) & -\sin\beta(t) \end{smallmatrix}\right)$$

and

$$\gamma'(0) = b \left(\begin{smallmatrix} -\sin\rho & -\cos\rho \\ \cos\rho & -\sin\rho \end{smallmatrix}\right) = \left(\begin{smallmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{smallmatrix}\right) \left(\begin{smallmatrix} 0 & -b \\ b & 0 \end{smallmatrix}\right) = pB.$$

With the identification of Remark 1.14, we observe that the tangent space at the identity (i.e., $\rho = 0$) is given by the skew-symmetric matrices $T_{\mathbf{Id}_2}SO(2) = \{ \left(\begin{smallmatrix} 0 & -b \\ b & 0 \end{smallmatrix}\right), b \in \mathbb{R}\} = \mathrm{Skew}(2)$.

The second step is now to describe smooth vector fields X and the differential of the left-translation $\mathrm{d}\ell_g$. Any vector field X on $SO(2)$ can be described by a smooth function of the form $F\colon SO(2) \to \mathrm{Skew}(2)$ with $F(p) = B_p$ through

$$\begin{aligned} X\colon SO(2) &\to T(SO(2)) \\ p &\mapsto (p, pB_p). \end{aligned}$$

Furthermore, for $g = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ the left-translation $\ell_g\colon SO(2) \to SO(2)$ with $\ell_g(p) = gp$ has the differential

$$\begin{aligned} \mathrm{d}\ell_g\colon T(SO(2)) &\to T(SO(2)) \\ (p, pB) &\mapsto (gp, gpB). \end{aligned}$$

This can be seen using once more the identification of Remark 1.14 which yields

$$(\mathrm{d}\ell_g)_p\left[\gamma'(0)\right] = (\ell_g \circ \gamma)'(0) = g\gamma'(0).$$

Now, we use the definition of left-invariant (1.4) vector fields to determine which choices of B_p yield a left-invariant vector field X:

$$\begin{aligned} \mathrm{d}\ell_g \circ X(p) &= (gp, gpB_p) \\ X \circ \ell_g(p) &= (gp, gpB_{gp}). \end{aligned}$$

Hence, equation (1.4) yields the condition $B_p = B_{gp}$ for any p and g. Especially, for $p = \mathbf{Id}_2$, we have $B_{\mathbf{Id}_2} = B_g$ for any g and thus, the set of left-invariant vector fields of $SO(2)$ can be described as $\mathfrak{so}(2) = \{p \mapsto (p, pB)\colon B \in \mathrm{Skew}(2)\}$.

As another example, we compute the left-invariant vector fields of the additive group $\mathbb{R}$. This consideration will be useful in Section 1.1.4.

Example 1.35. As before, we first characterize the tangent space $T_r\mathbb{R}$. Let $\gamma\colon\,]-\epsilon, \epsilon[\to \mathbb{R}$ be a smooth curve with $\gamma(0) = r$ and let $f \in \mathcal{F}_r(\mathbb{R})$, then $\dot\gamma(0)f = \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))\Big|_{t=0} = f'(r)\cdot\gamma'(0)$. Thus, the tangent space is given by

$$\begin{aligned} T_r\mathbb{R} &= \{\xi_r\colon \mathcal{F}_r(\mathbb{R}) \to \mathbb{R} \text{ such that } \exists c \in \mathbb{R}\colon \xi_r f = cf'(r)\} \\ &= \left\{ c\,\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=r} : c \in \mathbb{R} \right\}. \end{aligned}$$

Next, we describe vector fields X by their action on $f \in C^\infty(\mathbb{R})$

$$(Xf)(r) = X_r(f) = c_r\,\frac{\mathrm{d}}{\mathrm{d}t} f(t)\Big|_{t=r}.$$

We need to choose the dependence of c_r on r such that X is left-invariant. Therefore, consider$(\mathrm{d}\ell_s)_r\colon T_r\mathbb{R} \to T_{s+r}\mathbb{R}$,

$$\begin{aligned} (\mathrm{d}\ell_s)_r\left[c\,\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=r} \right] f &= c\,\frac{\mathrm{d}}{\mathrm{d}t} f(\ell_s(t))\Big|_{t=r} \\ &= c\,\frac{\mathrm{d}}{\mathrm{d}t} f(t)\Big|_{t=r+s} \end{aligned}$$

and compute

$$((\mathrm{d}\ell_s \circ X)f)(r) = c_r \left.\frac{\mathrm{d}}{\mathrm{d}t} f(t)\right|_{t=r+s},$$
$$((X \circ \ell_s)f)(r) = (X \circ \ell_s)_r(f) = (X_{s+r})(f) = c_{r+s} \left.\frac{\mathrm{d}}{\mathrm{d}t} f(t)\right|_{t=r+s}.$$

Similarly, to the previous example we deduce $c_r = c_{r+s}$ for any r and s. In particular, $c_s = c_0$ and hence, the left-invariant vector fields on $\mathbb{R}$ are characterized by $\mathfrak{r} = \left\{ r \mapsto c_0 \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=r} : c_0 \in \mathbb{R} \right\}$. Moreover, this shows that $r \mapsto \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=r}$, shortly written as $\frac{\mathrm{d}}{\mathrm{d}t}$, is a basis of the vector space $\mathfrak{r}$.

1.1.3 Lie algebras

For each Lie group there is a special Lie algebra which is closely related to it. Due to this link, the properties of a Lie group can be reflected as properties of its associated Lie algebra. Therefore, we have special interest in Lie algebras because this relation can be used to solve optimization problems whose constrained sets are Lie groups.
The following definition seizes Remark 1.23 concerning the Lie bracket of a vector field.

Definition 1.36. A *Lie algebra* $\mathfrak{a}$ over $\mathbb{R}$ is a real vector space $\mathfrak{a}$ together with a bilinear map $[\ ,\ ]\colon \mathfrak{a} \times \mathfrak{a} \to \mathfrak{a}$ with the following properties:

(i) $[X, Y] = -[Y, X]$, for $X, Y \in \mathfrak{a}$ (anti-commutativity)

(ii) $[[X, Y], Z] + [[Y, Z], X] + [[Z, X], Y] = 0$, for $X, Y, Z \in \mathfrak{a}$ (Jacobi identity)

For $X, Y \in \mathfrak{a}$, $[X, Y]$ is called the *Lie bracket* of X and Y.

Example 1.37. The vector space $M(n, \mathbb{R})$ of all $n \times n$ matrices forms a Lie algebra if we set $[A, B] = AB - BA$.

Example 1.38. The vector space $\mathbb{R}^n$ endowed with the trivial Lie bracket $[x, y] = 0$ is a Lie algebra.

Theorem 1.39. *Let G be a Lie group and $\mathfrak{g}$ its set of left-invariant vector fields. Then,*

(i) *$\mathfrak{g}$ is a real vector space.*

(ii) *The map $F\colon \mathfrak{g} \to T_eG$ defined by $F(X) = X_e$ is an isomorphism from $\mathfrak{g}$ to the tangent space of G at the identity e. In particular,* $\dim \mathfrak{g} = \dim T_eG = \dim G$.

(iii) *Left-invariant vector fields are smooth.*

(iv) *The Lie bracket of two left-invariant vector fields is itself a left-invariant vector field (see equation* (1.3)*).*

(v) *$\mathfrak{g}$ is a Lie algebra under the Lie bracket operation on vector fields.*

Proof. See [126] p. 85. □

Basically, the last aspect of Theorem 1.39 summarizes the other ones. Moreover, it follows from (ii) that $\mathfrak{g}$ can be identified with the tangent space at the identity which will be useful in many applications.
The theorem motivates the following definition.

Definition 1.40. The *Lie algebra of the Lie group* G is the Lie algebra $\mathfrak{g}$ of left-invariant vector fields on G.

Remark 1.41. We also say that $\mathfrak{g}$ is the *associated* Lie algebra of G.
Equivalently, we could define the Lie algebra of a Lie group G as the tangent space T_eG at the identity. Then, we would have to require the vector space isomorphism F in 1.39(ii) to be an isomorphism of Lie algebras, i.e., a vector space isomorphism which preserves the Lie bracket, in order to induce a Lie algebra structure on T_eG.

Example 1.42. We reconsider Example 1.34 and compute the left-invariant vector fields of $SO(n)$. We start again with the computation of the tangent space $T_pSO(n)$ for $p \in SO(n)$. Therefore, let $\gamma\colon]-\epsilon, \epsilon[\to SO(n)$ be a smooth curve with $\gamma(0) = p$. Since $\gamma(t) \in SO(n)$ for all t we have

$$\gamma(t)^T\gamma(t) = \mathbf{Id}_n$$

and differentiation with respect to t yields

$$\gamma'(t)^T\gamma(t) + \gamma(t)^T\gamma'(t) = 0.$$

In particular, for $t = 0$ we get

$$\gamma'(0)^T p + p^T\gamma'(0) = 0,$$

which implies that $p^T\gamma'(0)$ is skew-symmetric. Thus, for the tangent space we know $T_pSO(n) \subseteq \{pB\colon B \in \mathrm{Skew}(n)\}$ by identifying $T_pSO(n)$ as in Remark 1.14.
To see that the two spaces coincide, we observe that the dimension of both are the same (compare Remark 1.31). Analogously to Example 1.34, we get the condition

$$(gp, gpB_p) = \mathrm{d}\ell_g \circ X(p) = X \circ \ell_g(p) = (gp, gpB_{gp})$$

and thus, the set of left-invariant vector fields of $SO(n)$ is $\mathfrak{so}(n) = \{p \mapsto (p, pB)\colon B \in \mathrm{Skew}(n)\} \simeq \mathrm{Skew}(n)$. This proves that the set of skew-symmetric matrices is isomorphic to the associated Lie algebra of $SO(n)$.
Furthermore, the skew-symmetric matrices can be endowed with an inner product $\langle B, B'\rangle_{\mathrm{Skew}(n)} = \langle B, B'\rangle_F = \sum_{i=1}^n \sum_{j=1}^n b_{ij}b'_{ij} = \mathrm{tr}(B^TB')$, where $B = (b_{ij})_{ij=1,\dots n}$ and $B' = (b'_{ij=1,\dots n}) \in \mathrm{Skew}(n)$. Equipped with this so called *Frobenius inner product* the vector space of skew-symmetric matrices becomes a Hilbert space. Thus, the isomorphism between $\mathrm{Skew}(n)$ and $\mathfrak{so}(n)$ induces an inner product on $\mathfrak{so}(n)$.

1.1.4 The exponential map

As already mentioned, there is a natural relation between a Lie group G and its associated Lie algebra $\mathfrak{g}$. This relation is given by a C^∞ map which defines a diffeomorphism on a neighborhood of $0 \in \mathfrak{g}$ onto a neighborhood of $e \in G$ and which is called the *exponential map*. This subsection is dedicated to the derivation of this map and its properties which we will then use in the next section.

Definition 1.43. A group homomorphism $\varphi\colon G \to H$ between Lie groups G and H is a *Lie group homomorphism* if it is smooth.
A vector space homomorphism $\psi\colon \mathfrak{g} \to \mathfrak{h}$ between Lie algebras is a *Lie algebra homomorphism* if it preserves the Lie brackets, i.e., $\psi([X,Y]) = [\psi(X), \psi(Y)]$.

As we have seen in the last section, we can identify the associated Lie algebra $\mathfrak{g}$ of G with the tangent space T_eG at the identity $e \in G$. Hence, for a smooth map $\varphi\colon G \to H$ we can construct $\hat{\mathrm{d}}\varphi\colon \mathfrak{g} \to \mathfrak{h}$ via the differential $\mathrm{d}\varphi$. Explicitly, $\hat{\mathrm{d}}\varphi[X]$ is defined as the unique left-invariant vector field of H with

$$\hat{\mathrm{d}}\varphi\,[X]\,(e_H) = \mathrm{d}\varphi\,[X(e_G)]\,. \tag{1.5}$$

Due to the left-invariance of this vector field, we can determine for $h \in H$

$$\hat{\mathrm{d}}\varphi\,[X]\,(h) = \hat{\mathrm{d}}\varphi\,[X]\,(\ell_h(e_H)) = \mathrm{d}\ell_h\left[\hat{\mathrm{d}}\varphi\,[X]\,(e_h)\right] = \mathrm{d}\ell_h\,[\mathrm{d}\varphi\,[X(e_G)]]\,. \tag{1.6}$$

Theorem 1.44. *Let G and H be Lie groups with associated Lie algebras $\mathfrak{g}$ and $\mathfrak{h}$, respectively. Let G be simply connected and $\psi\colon \mathfrak{g} \to \mathfrak{h}$ be a Lie algebra homomorphism. Then, there exists a unique Lie group homomorphism $\varphi\colon G \to H$ such that $\hat{\mathrm{d}}\varphi = \psi$.*

Proof. See [126] p. 101. □

Definition 1.45. A Lie group homomorphism $\varphi\colon \mathbb{R} \to G$ is called a *1-parameter subgroup* of G.

This somehow generalizes the concept of lines to groups. It is an important observation that $\{\varphi(t)\colon t \in \mathbb{R}\}$ is a subgroup of G. Especially, it is closed under multiplication and moreover, this subgroup is commutative: $\varphi(t_1)\varphi(t_2) = \varphi(t_1 + t_2) = \varphi(t_2 + t_1) = \varphi(t_2)\varphi(t_1)$. Thus, optimization in a 1-parameter subgroup can be interpreted as a line search in $\mathbb{R}$. The commutativity is of special importance since it guarantees that the solution of a line search does not depend on the order of steps. Furthermore, it will sometimes be convenient to interpret a 1-parameter subgroup as a curve in G.
In the following, we consider one particular 1-parameter subgroup, which is defined by a Lie algebra homomorphism in the sense of Theorem 1.44. Therefore, let G be a Lie group and $\mathfrak{g}$ its Lie algebra. Recall that the associated Lie algebra of $\mathbb{R}$ (which is isomorphic to $\mathbb{R}$ itself) has the basis $\frac{\mathrm{d}}{\mathrm{d}r}$ (compare Example 1.35). For $X \in \mathfrak{g}$ the map $\psi\colon \mathbb{R} \to \mathfrak{g}$ defined by $\lambda\frac{\mathrm{d}}{\mathrm{d}r} \mapsto \lambda X$ is a Lie algebra homomorphism of the Lie algebra of the additive

Lie group $\mathbb{R}$ into $\mathfrak{g}$. Since $\mathbb{R}$ is simply connected, Theorem 1.44 yields the existence of a unique Lie group homomorphism $\exp_X \colon \mathbb{R} \to G$ such that

$$\hat{\mathrm{d}} \exp_X \left[\lambda \frac{\mathrm{d}}{\mathrm{d}r} \right] = \lambda X. \tag{1.7}$$

This homomorphism $t \mapsto \exp_X(t)$ is the unique 1-parameter subgroup of G whose tangent vector at 0 is $X(e)$ (see equation (1.5) and the following lemma).

Lemma 1.46. *A 1-parameter subgroup φ is uniquely defined by its tangent vector at 0.*

Proof. Let $\varphi \colon \mathbb{R} \to G$ be a 1-parameter subgroup with tangent vector $\dot{\varphi}(0) = \xi_e$ at 0. Define the left-invariant vector field $X \colon G \to T(G)$, with $X(e) = (e, \xi_e)$. Recall that for $g \in G$ we have then $X(g) = \mathrm{d}\ell_g [\xi_e]$.
Now, we will show that φ is the solution of a differential equation with fixed initial condition and smooth right-hand side for which it is known (generalization of the Picard-Lindelöf Theorem to differentiable manifolds, see e.g. [42] p. 208) that it has at most one solution. This yields uniqueness. Using that φ is a homomorphism, the chain rule and the left-invariance of X we compute

$$\begin{aligned}
\dot{\varphi}(t) f &= \mathrm{d}\varphi \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_{r=t} \right] f = \frac{\mathrm{d}}{\mathrm{d}r} f\left(\varphi(r)\right)\Big|_{r=t} \\
&= \frac{\mathrm{d}}{\mathrm{d}r} f\left(\varphi(r+t)\right)\Big|_{r=0} = \frac{\mathrm{d}}{\mathrm{d}r} f(\varphi(t)\varphi(r))\Big|_{r=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}r} f\left(\ell_{\varphi(t)}\varphi(r)\right)\Big|_{r=0} = \mathrm{d}\left(\ell_{\varphi(t)} \circ \varphi\right) \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_{r=0} \right] f \\
&= \mathrm{d}\ell_{\varphi(t)} \circ \mathrm{d}\varphi \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_{r=0} \right] f = \mathrm{d}\ell_{\varphi(t)} \left[\dot{\varphi}(0)\right] f \\
&= \mathrm{d}\ell_{\varphi(t)} \left[\xi_e\right] f = X(\varphi(t)) f.
\end{aligned}$$

Hence, φ solves

$$\dot{\varphi}(t) = X(\varphi(t))$$

with

$$\varphi(0) = e.$$

□

Definition 1.47. The map $\exp \colon \mathfrak{g} \to G$ defined by $\exp(X) \coloneqq \exp_X(1)$ is called the *exponential map.*

Remark 1.48. Here, it is not yet clear, whether there is a connection between the exponential map as defined above and the exponential function. But we will see later that the exponential map for the general linear group (and its subgroups) is indeed given by the matrix exponential (compare Examples 1.50 and 1.52).
Let us first discuss some properties of the exponential map.

1. The image of lines tX in $\mathfrak{g}$ for $t \in \mathbb{R}$ can be described as $\exp(tX) = \exp_X(t)$. To see this, consider the maps $\varphi_1, \varphi_2 \colon \mathbb{R} \to G$ with $\varphi_1(t) = \exp_{sX}(t)$ and $\varphi_2(t) = \exp_X(st)$ for $s \in \mathbb{R}$ and conclude that $\varphi_1(t) = \varphi_2(t)$ for all $t \in \mathbb{R}$ by the following steps. First, we note that

$$\begin{aligned} \mathrm{d}\exp_X \left[\lambda \left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] &= \hat{\mathrm{d}}\exp_X \left[\lambda \frac{\mathrm{d}}{\mathrm{d}r}\right](e_G) \\ &= \lambda X\,(e_G)\,. \end{aligned} \tag{1.8}$$

Here we used the Definition of $\hat{\mathrm{d}}$ (see (1.6)) and (1.7).
Observe that equation (1.8) for the map φ_1 with $\lambda = 1$ and X replaced by sX reads

$$\dot{\varphi_1}(0) = \dot{\exp}_{sX}(0) = \mathrm{d}\exp_{sX} \left[\left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] = sX(e_G).$$

Furthermore, define $\gamma(t) = st$ and compute

$$\begin{aligned} \mathrm{d}\left(\exp_X \circ\gamma\right) \left[\left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] f &= \left.\frac{\mathrm{d}}{\mathrm{d}r} f \circ \exp_X \circ\gamma(r)\right|_0 \\ &= s \left.\frac{\mathrm{d}}{\mathrm{d}r} f \circ \exp_X(r)\right|_0 \\ &= \mathrm{d}\exp_X \left[s \left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] f \end{aligned}$$

and thus, using again equation (1.8) we observe that for φ_2 it holds

$$\dot{\varphi_2}(0) = \mathrm{d}(\exp_X \circ\gamma) \left[\left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] = \mathrm{d}\exp_X \left[s \left.\frac{\mathrm{d}}{\mathrm{d}r}\right|_0\right] = sX\,(e_G)\,.$$

Hence, as both φ_1 and φ_2 are Lie group homomorphisms, Lemma 1.46 yields

$$\varphi_1 = \varphi_2$$

and thus, we have the desired

$$\exp(sX) = \exp_{sX}(1) = \exp_X(s). \tag{1.9}$$

2. Now, it is straightforward to conclude the functional equation

$$\exp((t_1 + t_2)X) = \exp_X(t_1 + t_2) = \exp_X(t_1)\exp_X(t_2) = \exp(t_1X)\exp(t_2X)$$

by exploiting the above property (1.9) and that $\exp_X$ is a homomorphism.
3. Similarly, it follows $e = \exp(0) = \exp((t - t)X) = \exp(tX)\exp(-tX)$ which yields

$$\exp(tX)^{-1} = \exp(-tX).$$

4. The exponential map $\exp \colon \mathfrak{g} \to G$ is a diffeomorphism of a neighborhood of $0 \in \mathfrak{g}$ onto a neighborhood of $e \in G$. In particular, there is an inverse mapping $\log \colon U_e \subset G \to \mathfrak{g}$ of exp in a neighborhood U_e of e. For a proof see [126] p. 103.

5. In general, for $X, Y \in \mathfrak{g}$ it is

$$\exp(X+Y) \neq \exp(X)\exp(Y),$$

compare (1.12).

Theorem 1.49. *Let (H, φ) be a Lie subgroup of G and let $X \in \mathfrak{g}$. If $X \in \hat{\mathrm{d}}\varphi(\mathfrak{h})$, then $\exp(tX) \in \varphi(H)$ for all t. Conversely, if $\exp(tX) \in \varphi(H)$ for t in some open interval, then $X \in \hat{\mathrm{d}}\varphi(\mathfrak{h})$.*

Proof. To prove the first statement, we show that the following diagram commutes:

$$\begin{array}{ccc} H & \xrightarrow{\varphi} & G \\ {\scriptstyle\exp}\uparrow & & \uparrow{\scriptstyle\exp} \\ \mathfrak{h} & \xrightarrow[\hat{\mathrm{d}}\varphi]{} & \mathfrak{g} \end{array}$$

Therefore, let $Y \in \mathfrak{h}$ be a left-invariant vector field and define the smooth curve $\gamma\colon \mathbb{R} \to G$, $\gamma(t) = \varphi(\exp(tY))$. We compute the tangent vector of γ at 0 using the chain rule, equation (1.8) and equation (1.5)

$$\begin{aligned} \dot{\gamma}(0) &= \mathrm{d}\gamma \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_0 \right] \\ &= \mathrm{d}(\varphi \circ \exp_Y) \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_0 \right] \\ &= \mathrm{d}\varphi \circ \mathrm{d}\exp_Y \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_0 \right] \\ &= \mathrm{d}\varphi \left[Y(\exp_Y(0)) \right] \\ &= \hat{\mathrm{d}}\varphi \left[Y \right](e). \end{aligned}$$

This means that γ is a 1-parameter subgroup whose tangent vector at 0 is $\hat{\mathrm{d}}\varphi\left[Y\right](e)$. The same is true for $\exp_{\hat{\mathrm{d}}\varphi[Y]}\colon t \mapsto \exp(t\hat{\mathrm{d}}\varphi\left[Y\right])$ (compare equation (1.7)). By Lemma 1.46, we know that a 1-parameter subgroup is uniquely defined by its tangent vector at 0 and thus, $\varphi(\exp(tY)) = \exp(t\hat{\mathrm{d}}\varphi\left[Y\right])$. Setting $t = 1$ yields

$$\varphi(\exp(Y)) = \exp(\hat{\mathrm{d}}\varphi\left[Y\right]) \tag{1.10}$$

and the above diagram commutes.
To show the statement itself, let $X \in \hat{\mathrm{d}}\varphi\left[\mathfrak{h}\right]$. We construct $h \in H$ such that $\varphi(h) = \exp(tX)$. Since $X \in \hat{\mathrm{d}}\varphi\left[\mathfrak{h}\right]$, we can find $Y \in \mathfrak{h}$ with $X = \hat{\mathrm{d}}\varphi\left[Y\right]$. We set $h = \exp(tY)$ and compute with equation (1.10)

$$\varphi(h) = \varphi(\exp(tY)) = \exp(\hat{\mathrm{d}}\varphi\left[tY\right]) = \exp(t\hat{\mathrm{d}}\varphi\left[Y\right]) = \exp(tX).$$

For the second statement, let $I =]a, b[$ be an open interval with $\exp(tX) \in \varphi(H)$ for all $t \in I$. We observe that without loss of generality we may assume that $0 \in I$. If $0 \notin I$, we can construct an open interval I_0, with $0 \in I_0$ and $\exp(tX) \in \varphi(H)$ for all $t \in I_0$: For $t_1, t_2 \in I$ it holds $\exp((t_1 - t_2)X) = \exp(t_1 X)\exp(t_2 X)^{-1} \in \varphi(H)$ since $\varphi(H)$ is a subgroup and hence, closed under multiplication. Consider $0 < \epsilon < \frac{|b-a|}{2}$, then we have for all $t \in I_0 =]-\epsilon, \epsilon[$

$$t = \underbrace{\frac{a+b}{2}}_{\in I} - \underbrace{\left(\frac{a+b}{2} - t\right)}_{\in I}$$

and hence

$$\exp(tX) = \exp\left(\frac{a+b}{2}X\right)\exp\left(\left(\frac{a+b}{2} - t\right)X\right)^{-1} \in \varphi(H).$$

Thus, let $0 \in I$ and construct a preimage $Y \in \mathfrak{h}$ of X under $\hat{\mathrm{d}}\varphi$ with $X = \hat{\mathrm{d}}\varphi\,[Y]$. Note that the map $\exp_X \colon t \mapsto \exp(tX)$ can be decomposed in $\exp_X = \varphi \circ \alpha$, where $\varphi \colon H \to G$ is the Lie group homomorphism and $\alpha \colon I \to H$. It can be shown that α can be chosen to be smooth (see [126] p. 47). Now, let $Y \in \mathfrak{h}$ be the left-invariant vector field defined by $\dot{\alpha}(0) = Y(e_H)$ and show that for this choice of Y we get indeed $\hat{\mathrm{d}}\varphi\,[Y] = X$. For $g \in G$ and $f \in C^\infty(G)$ we compute

$$\begin{aligned}
\hat{\mathrm{d}}\varphi\,[Y]\,(g)f &\overset{(1.6)}{=} \mathrm{d}\ell_g\,[\mathrm{d}\varphi\,[Y(e_H)]]\,f \\
&= \mathrm{d}(\ell_g \circ \varphi)\,[Y(e_H)]\,f \\
&\overset{(1.2)}{=} Y(e_H)(f \circ \ell_g \circ \varphi) \\
&= \dot{\alpha}(0)(f \circ \ell_g \circ \varphi) \\
&= \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \ell_g \circ \varphi \circ \alpha)\Big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \ell_g \circ \exp_X)\Big|_{t=0} \\
&= \mathrm{d}(\ell_g \circ \exp_X)\left[\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\right] f \\
&\overset{(1.6)}{=} \hat{\mathrm{d}}\exp_X\left[\frac{\mathrm{d}}{\mathrm{d}t}\right](g)f \\
&\overset{(1.7)}{=} X(g)f
\end{aligned}$$

which completes the proof. □

Example 1.50. Reconsider Example 1.27 and Example 1.37 concerning the general linear group $GL(n, \mathbb{R})$. In Example 1.27, we have seen that the tangent space of $GL(n, \mathbb{R})$ at 0 is the set $M(n, \mathbb{R})$ of all $n \times n$ matrices. From Theorem 1.39(ii) it follows that $M(n, \mathbb{R})$ is isomorphic to the associated Lie algebra of $GL(n, \mathbb{R})$. One can show that $M(n, \mathbb{R})$ as associated Lie algebra of $GL(n, \mathbb{R})$ is endowed with the in Example 1.37 introduced Lie bracket (see [126] p.87) and thus $\mathfrak{gl}(n) \simeq (M(n, \mathbb{R}), [\ ,\ ])$.

Using this isomorphism, we will explore in the following that the exponential map in the setting of Example 1.50 is closely related to the classical matrix exponential:

$$e^A = \sum_{i=0}^{\infty} \frac{A^i}{i!}, \qquad \text{for } A \in M(n, \mathbb{R}).$$

We will see that this relation is explicitly given by

$$\exp(X) = e^{X(\mathbf{Id}_n)} \qquad \text{for } X \in \mathfrak{gl}(n).$$

To this end, we have to show that $e^{tX(\mathbf{Id}_n)}$ defines a 1-parameter subgroup whose tangent vector at 0 is equal to $X(\mathbf{Id}_n) = A$. Due to the uniqueness of such a 1-parameter subgroup (compare Lemma 1.46) it must be the same as the exponential map defined in 1.47. Remark that in this context e does not refer to the identity element of a group. The identity element of a matrix group will be denoted by $\mathbf{Id}_n \in M(n, \mathbb{R})$.
Indeed, $E\colon t \mapsto e^{tA}$ is a 1-parameter subgroup with tangent vector $\dot{E}(0) = A$ as we will show by the following steps.
1. We observe that E is a map from $\mathbb{R}$ into $GL(n, \mathbb{R})$. It is well-known that the sum $\sum_{i=0}^{\infty} \frac{A^i}{i!}$ converges absolutely for $A \in M(n, \mathbb{R})$ and thus, e^{tA} is well-defined. Moreover, for the partial sums

$$S_j(A) = \sum_{i=0}^{j} \frac{A^i}{i!}$$

we have

$$B \lim_{j\to\infty} S_j(A) = \lim_{j\to\infty} BS_j(A) \qquad \text{for any } B \in M(n, \mathbb{R}),$$

since the map $C \mapsto BC$ is continuous from $M(n, \mathbb{R})$ into itself. If now B is invertible (i.e., $B \in GL(n, \mathbb{R})$), it is

$$Be^AB^{-1} = B \lim_{j\to\infty} S_j(A)B^{-1} = \lim_{j\to\infty} BS_j(A)B^{-1} = \lim_{j\to\infty} S_j(BAB^{-1}) = e^{BAB^{-1}}. \quad (1.11)$$

In particular, B can be chosen such that BAB^{-1} is upper triangular (Jordan normal form) and thus, all partial sums of $e^{BAB^{-1}}$ are upper triangular too. Let $\lambda_1, \ldots, \lambda_n$ be the diagonal entries of BAB^{-1}, then the diagonal entries of $e^{BAB^{-1}}$ are $e^{\lambda_1}, \ldots, e^{\lambda_n}$.
For the determinant, we deduce

$$\det(e^A) = \det(Be^AB^{-1}) = \det(e^{BAB^{-1}}) = \prod_{i=1}^{n} e^{\lambda_i} = e^{\mathrm{tr}(A)} \neq 0,$$

which proves that e^{tA} is invertible and thus, $e^{tA} \in GL(n, \mathbb{R})$.
2. The map E is a group homomorphism. Let us prove that

$$e^Ae^B = e^{A+B} \quad \text{if } AB = BA. \quad (1.12)$$

Therefore, compute

$$\begin{aligned} e^A e^B &= \left(\sum_{k=0}^{\infty} \frac{A^k}{k!}\right)\left(\sum_{l=0}^{\infty} \frac{B^l}{l!}\right) \\ &= \sum_{k=0}^{\infty}\sum_{l=0}^{k} \frac{A^{k-l}}{(k-l)!}\frac{B^l}{l!} \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}\sum_{l=0}^{k} \binom{k}{l} A^{k-l}B^l \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}(A+B)^k = e^{A+B}, \end{aligned}$$

where we used the Cauchy product for absolutely convergent series for the second equality. The fourth equality is true if $AB = BA$. From this it follows

$$E(t_1 + t_2) = e^{(t_1+t_2)A} = e^{t_1 A}e^{t_2 A} = E(t_1)E(t_2)$$

and thus, E is a group homomorphism. Note that (1.12) is not an equivalence. As a counterexample consider e.g. $A = \begin{pmatrix} 0 & 0 \\ 0 & 2\pi\mathbf{i} \end{pmatrix}$ and $B = \begin{pmatrix} 0 & 1 \\ 0 & 2\pi\mathbf{i} \end{pmatrix}$. Then, it is $e^A = e^B = e^{A+B} = \mathbf{Id}_n$ but $AB = \begin{pmatrix} 0 & 0 \\ 0 & -4\pi^2 \end{pmatrix} \neq \begin{pmatrix} 0 & 2\pi\mathbf{i} \\ 0 & -4\pi^2 \end{pmatrix} = BA$. Compare Horn and Johnson [57] page 436 where also real examples can be found.
3. Moreover, E is smooth and hence a Lie group homomorphism. This follows from the absolute convergence of the series $\sum_{k=0}^{\infty} \frac{B^k}{k!}$. For $B = tA$ every entry

$$\left(e^{tA}\right)_{i,j} = \sum_{k=0}^{\infty} t^k \frac{(A^k)_{i,j}}{k!}$$

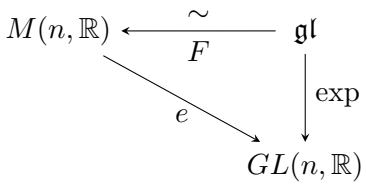
of e^{tA} is a power series in t and thus, the convergence radius of this series is

$$\sup\left\{|t|\colon \sum_{k=0}^{\infty} t^k \frac{(A^k)_{i,j}}{k!} \text{ converges}\right\} = \infty.$$

This shows that every entry of e^{tA} and hence e^{tA} itself are smooth.
4. Finally, we compute the tangent vector of E at 0:

$$\begin{aligned} \dot{E}(0) = E'(0) &= \frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=0}^{\infty} \frac{(tA)^k}{k!}\bigg|_{t=0} \\ &= \sum_{k=1}^{\infty} \frac{t^{k-1}A^k}{(k-1)!}\bigg|_{t=0} \\ &= A\sum_{k=0}^{\infty} \frac{(tA)^k}{k!}\bigg|_{t=0} \\ &= Ae^{tA}\Big|_{t=0} = A. \end{aligned}$$

Altogether, we have shown in four steps that E is a 1-parameter subgroup with tangent vector A at 0 and thus, the exponential map is equal to the matrix exponential on $M(n,\mathbb{R}) \simeq \mathfrak{gl}(n)$ in the sense of the following diagram.



Here, F is the isomorphism of Theorem 1.39(ii).
As mentioned before, the exponential map $\exp\colon \mathfrak{g} \to G$ is locally diffeomorphic at $0 \in \mathfrak{g}$. Let us now introduce the logarithm on a neighborhood of the identity in $GL(n,\mathbb{R})$ as inverse of the matrix exponential. Analogously to the power series of the logarithm known from real analysis, we define for any $A \in M(n,\mathbb{R})$

$$\log(A) = \sum_{m=1}^{\infty} (-1)^{m+1} \frac{(A - \mathbf{Id}_n)^m}{m},$$

whenever the series converges. In the following, we will restrict to sub-multiplicative matrix norms, since calculations will be based on $\|(A - \mathbf{Id}_n)^m\| \leq \|A - \mathbf{Id}_n\|^m$. The Frobenius norm $\|A\|_F = (\mathrm{tr}(A^T A))^{\frac{1}{2}}$ (see also Example 1.42) is an example for a sub-multiplicative norm.
According to the majorant criterion for matrix-valued series, $\sum_{m=1}^{\infty}(-1)^{m+1}\frac{(A-\mathbf{Id}_n)^m}{m}$ converges absolutely for $\|A - \mathbf{Id}_n\| < 1$. A majorant is given by $\sum_{m=1}^{\infty}\frac{\|A-\mathbf{Id}_n\|^m}{m}$ since

$$\left\|(-1)^{m+1}\frac{(A-\mathbf{Id}_n)^m}{m}\right\| \leq \frac{\|A-\mathbf{Id}_n\|^m}{m}$$

holds. The majorant $\sum_{m=1}^{\infty}\frac{\|A-\mathbf{Id}_n\|^m}{m}$ converges for $\|A - \mathbf{Id}_n\| < 1$ due to the quotient criterion.
Now, we can state the following theorem.

Theorem 1.51. *The function* $\log\colon \mathcal{A}(n) = \{A \in M(n,\mathbb{R})\colon \|A - \mathbf{Id}_n\| < 1\} \to M(n,\mathbb{R})$ *is smooth and it holds*

$$e^{\log(A)} = A$$

for all $A \in \mathcal{A}(n)$. *Conversely, for all* X *with* $\|X\| < \ln(2)$ *it holds*

$$e^X \in \mathcal{A}(n) \qquad \textit{and} \qquad \log(e^X) = X.$$

Proof. First, we observe that the set $\mathcal{A}(n) \subset GL(n)$ is an open neighborhood of $\mathbf{Id}_n \in GL(n)$. For the proof one uses the Neumann series to construct A^{-1} explicitly.
The function log is smooth on $\mathcal{A}(n)$ since a power series is smooth inside its radius of convergence. Next, we show that $e^{\log(A)} = A$ for all A with $\|A - \mathbf{Id}_n\| < 1$. To this end,

we will first consider diagonalizable matrices A and then we will generalize the result. Let A be diagonalizable with $A = CDC^{-1}$, where $D = \operatorname{diag}(d_1, \dots, d_n)$ is a diagonal matrix and d_i are the eigenvalues of A. Then,

$$(A - \mathbf{Id}_n)^m = (CDC^{-1} - \mathbf{Id}_n)^m = \left(C(D - \mathbf{Id}_n)C^{-1}\right)^m = C(D - \mathbf{Id}_n)^m C^{-1}.$$

Since $\|A - \mathbf{Id}_n\| < 1$, for each eigenvalue d_i of A we have $|d_i - 1| < 1$ and thus, by the continuity of the matrix multiplication it follows

$$\begin{aligned}\log(A) &= \sum_{m=1}^{\infty} (-1)^{m+1} \frac{(A - \mathbf{Id}_n)^m}{m} \\ &= C \left(\sum_{m=1}^{\infty} (-1)^{m+1} \frac{(D - \mathbf{Id}_n)^m}{m} \right) C^{-1} = C \operatorname{diag}\left(\ln(d_1), \dots, \ln(d_n)\right) C^{-1}.\end{aligned}$$

Applying equation (1.11) leads to

$$e^{\log(A)} = C e^{\operatorname{diag}(\ln(d_1),\dots,\ln(d_n))} C^{-1} = C \operatorname{diag}\left(e^{\ln(d_1)}, \dots, e^{\ln(d_n)}\right) C^{-1} = CDC^{-1} = A.$$

If, in contrast, A is not diagonalizable, note that it is the limit of a sequence of diagonalizable matrices $(A_k)_{k\in\mathbb{N}}$. This results from the fact that every matrix is similar to an upper triangular matrix and that a matrix is diagonalizable if all its eigenvalues are distinct. Therefore, small changes in the diagonal entries of the upper triangular matrix will give us such a sequence $(A_k)_{k\in\mathbb{N}}$. If $\|A - \mathbf{Id}_n\| < 1$ then $\|A_k - \mathbf{Id}_n\| < 1$ for sufficiently large k. From the previous considerations we know that $e^{\log(A_k)} = A_k$ and thus

$$e^{\log(A)} = A$$

due to the continuity of the matrix exponential and the logarithm.
Analogously, we show that $\log(e^X) = X$ for all X with $\left\|e^X - \mathbf{Id}_n\right\| < 1$. Thus, it remains to show that $\|X\| < \ln(2)$ implies $\left\|e^X - \mathbf{Id}_n\right\| < 1$:

$$\left\|e^X - \mathbf{Id}_n\right\| \leq \sum_{m=1}^{\infty} \frac{\|X^m\|}{m!} \leq \sum_{m=1}^{\infty} \frac{\|X\|^m}{m!} = e^{\|X\|} - 1 < 1.$$

□

Example 1.52. For the special orthogonal matrices (compare Example 1.42), we can show similarly as before that the exponential map is given by the matrix exponential. Additionally, we have to verify that $e^{tA} \in SO(n)$ for $A \in \operatorname{Skew}(n) \simeq \mathfrak{so}(n)$. This is a direct consequence of equation (1.12) since

$$e^{tA} \left(e^{tA}\right)^T = e^{tA} e^{tA^T} = e^{tA} e^{-tA} = e^0 = \mathbf{Id}_n.$$

Alternatively, this can be shown using Theorem 1.49.
Clearly, the restriction of the logarithm to $SO(n) \cap \mathcal{A}(n)$ is locally the inverse mapping of the matrix exponential on $\operatorname{Skew}(n)$.

1.1.5 Summing up the theoretical part

For our application, the main issue is that the associated Lie algebra $\mathfrak{so}(n)$ of the Lie group of special orthogonal matrices $SO(n)$ is isomorphic to the set of the skew-symmetric matrices. This enables us to carry over some features of the linear structure of the skew-symmetric matrices to the non-linear manifold $SO(n)$. The local diffeomorphism on page 18 relating both sets is not just a theoretical construction but identifying $\mathfrak{so}(n)$ and $\mathrm{Skew}(n)$ explicitly given by the matrix exponential (see Figure 1.1) which makes computation feasible. Due to this diffeomorphism we will be able to transfer optimization algorithms designed for vector spaces to Lie groups. This is our core ingredient for solving non-negative dimensionality reduction problems like ours (compare Section 2.3).
However, the generality of this chapter permits to use these ideas for optimization on any Lie group, not just on matrix Lie groups. For an arbitrary Lie group, the linear structure of its associated Lie algebra induces a convenient structure on the Lie group itself via the exponential map. In particular, the concept of straight lines in linear spaces is generalized to 1-parameter subgroups of non-linear groups. This provides an elegant way to perform line search-like algorithms on arbitrary Lie groups (see Section 1.2.2).

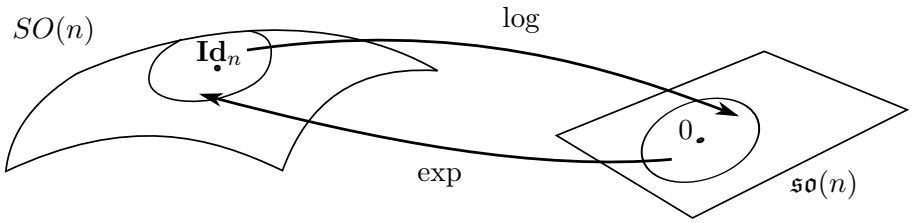


Figure 1.1: Lie Group $SO(n)$ and associated Lie algebra $\mathfrak{so}(n)$. Both sets are linked by the diffeomorphism exp.

1.2 Optimization on Lie groups - steepest descent

This section is concerned with finding a (local) *minimizer* x_* of a smooth function $f\colon \mathcal{M} \to \mathbb{R}$, i.e., a point $x_* \in \mathcal{M}$ such that $f(x_*) \leq f(x)$ for all x in a neighborhood of x_*. We can further restrict the problem by only considering points $x \in U \subset \mathcal{M}$ and thus minimizing f on a subset U. The problem then becomes a *constrained* optimization problem which we write in the standard way

$$\min_{x \in U} f(x).$$

In this setting, f is called *cost function* or *objective function* and $f(x_*)$ is the *minimum value* or just the *minimum*. The restriction $x \in U$ is referred to as the *constraint* and the set U as the *admissible set*. Note that U can be a submanifold and in particular, it might be a non-convex set.

If we study optimization problems with intricate constraints, it might be useful to exploit the structural properties on the set to make computations easier. The idea is to take advantage of special properties of the problem as e.g. symmetry or invariance in order to develop efficient numerical methods. These properties may concern both, constraints and/or cost functional, and they occur for example in matrix optimization. Often, the admissible sets of these problems have the structure of a non-linear matrix manifold.
One of the main challenges in solving these problems lies in the word *non-linear* since non-linear sets do not possess vector space properties. Classical optimization however, is based on additive iterative algorithms. This class of algorithms relies strongly on the Euclidean structure of the search space (see [2]). As a consequence of the non-linearity, the embedding of these manifolds in the $\mathbb{R}^{n^2}$ might lead to a non-convex set.
Additive iterative algorithms approach an extremum of the cost functional by adding at each step of the algorithm an update quantity to the current iterate. At each step the update direction and step-length needs to be computed. Typically, this computation is based on first and second order derivatives of the cost functional or on an approximation of these. In the case where the optimization is carried out on a manifold, this procedure needs to be translated to the language of differential geometry. This will be the subject of this section. We will concentrate on gradient descent line search algorithms on Lie groups. We will use the exponential map and the thereby defined 1-parameter subgroups. These curves on the Lie group can be used for generalized line search algorithms which are simple but completely sufficient for our purposes. In particular, these algorithms permit solving non-convex optimization problems on Lie groups (as e.g. on $SO(n)$).
Our precise description of the topic can be used as a basis for the further development of sophisticated optimization methods on Lie groups. It provides a universal formulation of the necessary ingredients to transfer standard methods into a more general framework. In this sense, it can be seen as a theoretical contribution to the active research field of optimization on manifolds and in particular on Lie groups. In comparison to Newton's method on Lie groups in [81] and the conjugate gradient method on $U(n)$ for a certain class of cost functions in [1], our approach is simpler to implement, nevertheless it leads to sophisticated results at relatively low computational cost.
We will now briefly introduce the steepest descent method in $\mathbb{R}^n$ before we address the core part of this section where we actually 'translate' this method to the more general setting of Lie groups.

1.2.1 Steepest descent in $\mathbb{R}^n$

In this section, we will briefly recall the concept of a *steepest descent* or *gradient descent* method in $\mathbb{R}^n$ in order to generalize it to Lie groups in the next section. We will rely on the text books [65] by Kelley and [43] by Geiger and Kanzow.
Let us consider the unconstrained optimization problem $\min_x f(x)$, where $f\colon \mathbb{R}^n \to \mathbb{R}$. A descent method is an iterative method to solve such problems. Its central idea is the following: At the current iterate x^k a direction ξ^k is determined in which the values of the cost function f decrease. To determine the next iterate one follows this direction until the functional value is small enough. This procedure is repeated in order

to approach a minimum. Such a direction ξ^k is called a *descent direction of f at x^k*. In mathematical terms, the descent direction ξ^k is a vector in $T_{x^k}\mathbb{R}^n$. Since $T_{x^k}\mathbb{R}^n = \mathbb{R}^n$, we can characterize ξ^k by

$$f(x^k + t\xi^k) < f(x^k) \qquad \text{for all sufficiently small } t > 0. \tag{1.13}$$

The determination of such a descent direction (at best optimal) is a crucial point. However, there is a sufficient condition for ξ^k being a descent direction involving the gradient $\nabla f(x^k) \in T_{x^k}\mathbb{R}^n$.

Lemma 1.53. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $x \in \mathbb{R}^n$ and $\xi \in T_x\mathbb{R}^n = \mathbb{R}^n$ with $\nabla f(x)^T\xi < 0$. Then, ξ is a descent direction of f at x.*

Proof. See [43]. □

Intuitively, we would probably choose ξ to be the negative gradient of the function f at x. Indeed, for $\nabla f(x) \neq 0$ the negative gradient $-\nabla f(x)$ is the *steepest descent (direction)*, i.e., the direction with the smallest directional derivative. To see this minimize $\langle \nabla f(x), \xi\rangle = \cos(\measuredangle(\nabla f(x), \xi))\, \|\nabla f(x)\|$ for $\|\xi\| = 1$. The minimum is attained for $\xi = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$ since $-1 = \cos(\pi) = \cos\left(\measuredangle(\nabla f(x), -\nabla f(x))\right)$. However, the restriction in Lemma 1.53 admits all directions which have an angle to the negative gradient that is smaller than $\frac{\pi}{2}$. For $\nabla f(x) = 0$ the point x is called *stationary*, i.e., a maximum, a minimum or a saddle point.
For applications often the normalized gradient $\xi = -\frac{\nabla f}{\|\nabla f\|}$ is used even though other choices may lead to better convergence properties (compare [43] Chapter 8).

Remark 1.54. Though the condition in Lemma 1.53 is sufficient, it is not necessary. If x is a strict local maximum of f, all directions $\xi \in \mathbb{R}^n \setminus \{0\}$ are descent directions, but none of them fulfills the condition $\nabla f(x)^T\xi < 0$.

Algorithm 1.55. A general descent method has the following structure:

1: **choose** $x^0 \in \mathbb{R}^n$ and **set** $k = 0$
2: **for** x^k does not satisfy a suitable stopping criteria **do**
3: **compute** descent direction ξ^k of f at x^k
4: **compute** step-length $t^k > 0$ with $f(x^k + t^k\xi^k) < f(x^k)$
5: **set** $x^{k+1} = x^k + t^k\xi^k$ and k to $k+1$
6: **end for**

As this general procedure shows, a descent method consists basically of two parts which are alternately repeated: The search of a descent direction and the choice of a step-length. So far we have discussed the choice of an appropriate update direction (even though the gradient might not be the optimal one), but what remains to be determined is the step-length t^k. An obvious approach is to follow the descent direction until the cost function is not descending any more, i.e., a line search along the descent direction. For the computation of the step-length several criteria are available. We refer to the Armijo rule (see [65]) which takes care of the chosen step-length to be not too small

and neither too large. Very small step-length may lead to an insufficient decrease of the function f and too large step-length may cause that the algorithm runs into another minimum.
Introducing the three parameters $\sigma \in]0,1[$, $\beta \in]0,1[$ and $\epsilon \geq 0$ and choosing the (normalized) gradient as descent direction and the Armijo rule for the step-length computation, Algorithm 1.55 reads as follows:

Algorithm 1.56. Gradient descent method with Armijo rule:
1: **choose** $x^0 \in \mathbb{R}^n$, σ, $\beta \in]0,1[$, $\epsilon \geq 0$ and **set** $k = 0$
2: **for** $\|\nabla f(x^k)\|_2 > \epsilon$ **do**
3: **set** $\xi^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|_2}$
4: **compute** $t^k = \max\{\beta^l : l \in \mathbb{N} \text{ and } f(x^k + \beta^l \xi^k) \leq f(x^k) - \sigma\beta^l \|\nabla f(x^k)\|_2\}$
5: **set** $x^{k+1} = x^k + t^k \xi^k$ and k to $k+1$
6: **end for**

It can be shown that for $\epsilon = 0$ each accumulation point of a sequence generated by Algorithm 1.56 is a stationary point of f (see [43]). However, the convergence is rather slow compared to other methods. But the steepest descent method provides the possibility to be generalized to Lie groups which is convenient for our applications.

1.2.2 Steepest descent on Lie groups

In the first part of this chapter (Section 1.1), we have introduced all necessary tools for generalizing the method of steepest descent to Lie groups. This is motivated by the fact that sometimes the set of constraints of an optimization problem is restricting the admissible set to a Lie group G. In order to solve such a problem of the form

$$\min_{x \in G} f(x), \tag{1.14}$$

where $f : G \to \mathbb{R}$, we cannot directly apply the steepest descent method since it involves an additive update step. With this step, most likely we move away from the manifold G and the new iterate is not admissible any more. To see this, consider the example of the Lie group $SO(n)$. Nevertheless, the theory is applicable for arbitrary Lie groups.

Example 1.57. Let f be a function from the set of real $n \times n$ matrices $\mathbb{R}^{n\times n}$ to the real line $\mathbb{R}$. Consider the optimization problem

$$\min_{X \in SO(n)} f(X).$$

As we know from Example 1.30, the set of special orthonormal matrices $SO(n)$ has a Lie group structure, but it is clearly not a vector space since in general the sum of two orthonormal matrices is not orthonormal. Even though for any update direction ξ and any $X \in SO(n)$ the matrix $X + t\xi$, $t \in \mathbb{R}$ is an element of $\mathbb{R}^{n\times n}$, it might not be one of $SO(n)$. Moreover, the expression $X + t\xi$ is critical because a priori the operation $+$ is not defined on a general manifold. This is why the classical steepest descent algorithm in $\mathbb{R}^{n\times n}$ will fail.

This example already points out two possible problems occurring when transferring the steepest descent method from $\mathbb{R}^n$ to a Lie group. In $\mathbb{R}^n$ a current iterate x is updated in the direction where the directional derivative of the cost functional f is most negative. This direction is computed by minimizing the directional derivative

$$Df(x)(\xi) = \lim_{t \to 0} \frac{f(x + t\xi) - f(x)}{t}$$

under all $\xi \in T_x\mathbb{R}^n = \mathbb{R}^n$ with $\|\xi\| = 1$. If now f operates on a general manifold G instead of $\mathbb{R}^n$, the argument $x + t\xi$ does not make sense due to the lack of the vector space structure. And as motivated above, even if we consider $G \subset \mathbb{R}^\nu$ for a suitable ν, we might leave G or even the domain of f. To circumvent this problem, we have introduced the tangential vector ξ_x at $x \in G$ as action on smooth functions $f \in \mathcal{F}_x(G)$ (compare Definition 1.13). This is a generalization of the directional derivative as in $\mathbb{R}^n$ we have

$$\xi_x f = Df(x)(\xi_x). \tag{1.15}$$

To make matters worse, there is a third critical aspect one has to face: How is the gradient defined and what is its length? This problem occurs when computing the descent direction itself. Therefore, we recall the formal definition of the gradient. To define the gradient on manifolds we need an inner product. This justifies the introduction of Riemannian manifolds (compare page 9) which allows also a rigorous definition of the length of a tangential vector.

Definition 1.58. Let $(\mathcal{M}, (\langle \cdot\,, \cdot \rangle_{T_x\mathcal{M}})_{x \in \mathcal{M}})$ be a Riemannian manifold and $f \colon \mathcal{M} \to \mathbb{R}$ a continuously differentiable function. The gradient ∇f of f is the unique vector field whose inner product with any tangential vector $\xi_x \in T_x\mathcal{M}$ is equal to the directional derivative of f at x for all $x \in \mathcal{M}$:

$$\langle \nabla f(x), \xi_x \rangle_{T_x\mathcal{M}} = \xi_x f, \qquad \text{for all } \xi \in T_x\mathcal{M} \text{ and all } x \in \mathcal{M}.$$

Remark 1.59. The definition of the gradient depends strongly on the choice of the inner product. Since both, G and $\mathfrak{g}$, are Riemannian manifolds we have to be careful to distinguish the different inner products and gradients. We might refer to them as $\nabla_G f$ and $\nabla_{\mathfrak{g}} f$, respectively.

To overcome the problem concerning the additive update, we define a generalized descent direction in the tangent space at x of a Lie group G.

Definition 1.60. We call a tangential vector $\xi_x \in T_xG$ a *descent direction of f at x* if

$$f(\gamma(t)) < f(x),$$

for $\gamma \colon \mathbb{R} \to G$ with $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi_x$ and sufficiently small $t > 0$. Such curves γ are called *descent curves of f at x*.

Remark 1.61. Notice that this definition is compatible with the previous definition in (1.13) for $\mathbb{R}^n$.

Now, we want to characterize a certain class of descent curves involving the exponential map. Therefore, we will consider the function $f \circ \exp\colon \mathfrak{g} \to \mathbb{R}$ and its gradient on $\mathfrak{g}$.

Theorem 1.62. *Let h, $q \in \mathfrak{g}$ be left-invariant vector fields with $\langle \nabla_{\mathfrak{g}}(f \circ \exp)(q), h\rangle_{\mathfrak{g}} < 0$. Then,*

$$\gamma(t) = \exp(q + th)$$

is a descent curve of f at $x = \exp(q)$.

Proof. It is sufficient to prove that $\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))\Big|_{t=0} < 0$ as $f \circ \gamma\colon \mathbb{R} \to \mathbb{R}$ and $f \circ \gamma(0) = f(\exp(q))$. Hence, we compute for $\beta\colon \mathbb{R} \to \mathfrak{g}$, with $\beta(t) = q + th$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t)) &= \frac{\mathrm{d}}{\mathrm{d}t} (f \circ \exp \circ \beta(t)) \\ &= \dot{\beta}(t)(f \circ \exp) \\ &= \langle \nabla_{\mathfrak{g}}(f \circ \exp)(\beta(t)), \dot{\beta}(t)\rangle_{\mathfrak{g}}. \end{aligned} \tag{1.16}$$

Furthermore, as $\mathfrak{g}$ is a vector space, by Remark 1.14 we have $\dot{\beta}(t) = \beta'(t) = h$ and thus, for $t = 0$ equation (1.16) reads

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))\Big|_{t=0} = \langle \nabla_{\mathfrak{g}}(f \circ \exp)(q), h\rangle_{\mathfrak{g}} < 0 \tag{1.17}$$

which completes the proof. □

Theorem 1.62 provides the basis for a steepest descent algorithm on Lie groups. It allows that the additive update step $q + th$ is not performed in the Lie group G but in the Lie algebra $\mathfrak{g}$. This translation to the Lie algebra is called *Lie group method* and provides an elegant generalization of the steepest descent algorithm to manifolds. This is illustrated in Figure 1.2. Recall that the logarithm is only defined on a neighborhood of $e \in G$. Remark once more that we do not need to compute a projection onto the manifold G in the iteration step.

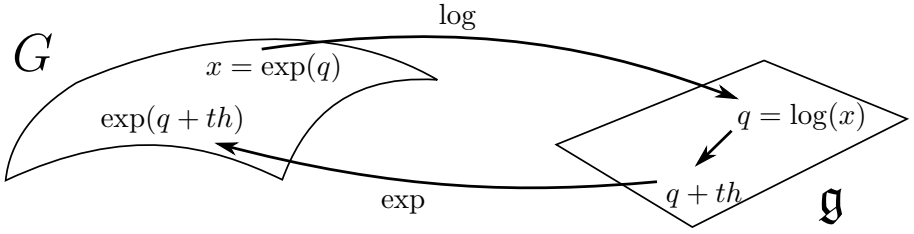


Figure 1.2: Lie group method. The additive update is performed in the Lie algebra $\mathfrak{g}$.

Corollary 1.63. *Among all descent curves γ of the form $\gamma(t) = \exp(q + th)$ the one with $h = -\frac{\nabla_{\mathfrak{g}}(f \circ \exp)(q)}{\|\nabla_{\mathfrak{g}}(f \circ \exp)(q)\|}$ has the steepest descent in $x = \exp(q)$.*

Proof. Due to equation (1.17) this can be proven analogously to the statement in $\mathbb{R}^n$. □

Now, the steepest descent on a Lie group can be thought of as following a geodesic curve in descent direction. Here, the geodesic is described as in Theorem 1.62 (for more details see [42]). This procedure is also called geometric flow method.
As in the vector space setting, the step size along a descent curve can be chosen by the Armijo rule. This is due to the fact that the steepest descent is the combination of several 'line' searches along curves on G in different directions. So to say, the optimization is performed on functions $f \circ \gamma \colon \mathbb{R} \to \mathbb{R}$, where γ is the descent curve of Theorem 1.62. Following the steps of Algorithm 1.56 we are now able to formulate a steepest descent algorithm on Lie groups.

Algorithm 1.64. Gradient descent method with Armijo rule: Let G be a Lie group and $\mathfrak{g}$ its associated Lie algebra.

1: **choose** $x^0 \in G$, σ, $\beta \in]0,1[$, $\epsilon \geq 0$ and **set** $k = 0$
2: **compute** $q^k = \log(x^k)$
3: **for** $\|\nabla_{\mathfrak{g}}(f \circ \exp)(q^k)\|_{\mathfrak{g}} > \epsilon$ **do**
4: **set** descent direction $h^k = -\frac{\nabla_{\mathfrak{g}}(f\circ\exp)(q^k)}{\|\nabla_{\mathfrak{g}}(f\circ\exp)(q^k)\|_{\mathfrak{g}}} \in \mathfrak{g}$
5: and descent curve $\gamma^k(t) = \exp(q^k + th^k)$
6: **compute** $t^k = \max\{\beta^l : l \in \mathbb{N} \text{ and } f(\gamma^k(\beta^l)) \leq f(x^k) - \sigma\beta^l\|\nabla_{\mathfrak{g}}(f \circ \exp)(q^k)\|_{\mathfrak{g}}\}$
7: **set** $x^{k+1} = \gamma^k(t^k)$ and k to $k+1$
8: **end for**

Remark 1.65. In line 3 of Algorithm 1.64, the logarithm of x^k needs to be computed since γ is a descent curve at $x^k = \exp(q^k)$ (compare Figure 1.2). This might be problematic as the logarithm is only defined on a neighborhood of $e \in G$. Furthermore, the computation of the logarithm is usually quite expensive. In the next section, we will see how to overcome this problem.

Analogously to the gradient descent method in $\mathbb{R}^n$ (Algorithm 1.56) the Algorithm 1.64 generates for $\epsilon = 0$ a sequence whose accumulation points are stationary points of f (compare [2] for a general theoretical result concerning Riemannian manifolds). This guarantees the convergence of a subsequence.

1.3 Implementation-friendly optimization on Lie groups

In the last section, we have introduced an update algorithm with the update step $x^{k+1} = \exp(q^k + t^k h^k)$, where $q^k = \log(x^k)$. Since we operate in a group, there exists a $g^k \in G$ with $x^{k+1} = x^k g^k$. We would prefer to use g^k in order to perform the update, as the computation of the logarithm is a bottleneck of the Algorithm 1.64. Therefore, the computation of g^k will be one objective of this section. Heuristically, the update

$$x^k \rightsquigarrow q^k = \log(x^k) \rightsquigarrow q^k + t^k h_1^k \rightsquigarrow x^{k+1} = \exp(q^k + t^k h_1^k) = x^k g^k$$

is equivalent to the update

$$e \curvearrowright \log(e) = 0 \curvearrowright t^k h_2^k \curvearrowright g^k = \exp(t^k h_2^k) \quad \text{and setting} \quad x^{k+1} = x^k g^k,$$

which shifts the problem from x^k in the origin and thus, omits the computation of the logarithm.
Note that in general we cannot conclude

$$\exp(q^k + t^k h_1^k) = \exp(q^k)\exp(t^k h_1^k)$$

to determine g^k. To see this, consider the matrix case and recall that $e^{q^k + t^k h_1^k} = e^{q^k} e^{t^k h_1^k}$ if $h_1^k q^k = q^k h_1^k$ (compare (1.12)). In particular, this implies in general $h_1^k \neq h_2^k$. The remaining task is to determine the direction $h_2^k = h^k$.
In the following, we will formalize this heuristic rigorously. Therefore, we further explore the left-invariance of the vector fields in $\mathfrak{g}$. Furthermore, we will discuss an example, where this algorithm is used to find a certain rotation for a data set in $\mathbb{R}^d$.

1.3.1 Multiplicative update

As already mentioned, we now aim to circumvent the computation of the logarithm in Algorithm 1.64 by shifting the problem from x^k to $e \in G$. This shift will be carried out by the left-translation $\ell_{(x^k)^{-1}}$.
To this end, let us now reconsider Theorem 1.62 for $\gamma(t) = \ell_x \circ \exp(th)$.

Theorem 1.66. *Let $h \in \mathfrak{g}$ and $x \in G$ with $\langle \nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0), h\rangle_{\mathfrak{g}} < 0$ then*

$$\gamma(t) = \ell_x \circ \exp(th)$$

is a descent curve of f at x.

Proof. For the proof we apply Theorem 1.62 to $\tilde{f} = f \circ \ell_x$ and $q = 0$. First, we note that the condition $\langle \nabla_{\mathfrak{g}}(\tilde{f} \circ \exp)(0), h\rangle_{\mathfrak{g}} < 0$ is fulfilled by the assumption on f and h. Thus, Theorem 1.62 is applicable and

$$\tilde{\gamma}(t) = \exp(th)$$

is a descent curve of $\tilde{f}$ at $\tilde{x} = e$. From this it follows directly that

$$\gamma(t) = \ell_x \circ \exp(th)$$

is a descent curve of f at x. □

Remark 1.67. We observe that equation (1.17) for $\tilde{f}$ and $q = 0$ reads

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} f\left(\gamma(t)\right)\Big|_{t=0} &= \frac{\mathrm{d}}{\mathrm{d}t} \tilde{f}\left(\exp(th)\right)\Big|_{t=0} \\ &\overset{(1.17)}{=} \langle \nabla_{\mathfrak{g}}(\tilde{f} \circ \exp)(0), h\rangle_{\mathfrak{g}} \\ &= \langle \nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0), h\rangle_{\mathfrak{g}}. \end{aligned} \tag{1.18}$$

Theorem 1.66 provides an alternative description of descent curves of f at x, where the logarithm is not involved. Analogously to Corollary 1.63, the steepest descent direction is $h = -\frac{\nabla_{\mathfrak{g}}(f\circ\ell_x\circ\exp)(0)}{\|\nabla_{\mathfrak{g}}(f\circ\ell_x\circ\exp)(0)\|_{\mathfrak{g}}}$. But the theorem gives no hint how to compute this descent direction in practice.
In the following, we want to derive an expression for this gradient in dependence on $\nabla_G f(x)$, since this might facilitate computations. Therefore, we define the left-invariance of a Riemannian metric on G.

Definition 1.68. A Riemannian metric is called *left-invariant* if

$$\langle \xi_x^1, \xi_x^2 \rangle_{T_xG} = \left\langle (\mathrm{d}\ell_g)_x \left[\xi_x^1\right], (\mathrm{d}\ell_g)_x \left[\xi_x^2\right] \right\rangle_{T_{gx}G} \qquad \text{for all } x, g \in G \text{ and } \xi_x^1, \xi_x^2 \in T_xG. \tag{1.19}$$

From the definition it is clear that a left-invariant Riemannian metric is uniquely determined by $\langle \cdot, \cdot \rangle_{T_eG}$ for the identity $e \in G$. Using the identification of the associated Lie algebra $\mathfrak{g}$ with the tangent space T_eG at the identity (see 1.39(ii)), it can be shown that there is a bijective correspondence between the inner products on the Lie algebra $\mathfrak{g}$ and the left-invariant Riemannian metrics on G (see [42] Chapter 17). This correspondence is explicitly given as

$$\langle h^1, h^2 \rangle_{\mathfrak{g}} = \langle h^1(e), h^2(e) \rangle_{T_eG} \qquad \text{for all } h^1, h^2 \in \mathfrak{g} \text{ and } e \in G. \tag{1.20}$$

In the following, we will always assume without mentioning that the Riemannian metric on G and the inner product of $\mathfrak{g}$ are related by equation (1.20). This will play an important role in the proof of the next lemma.

Lemma 1.69. *The gradient introduced in Theorem 1.66 is given by*

$$\nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0) = X^{\nabla_G f(x)}, \tag{1.21}$$

where $X^{\nabla_G f(x)}$ is the left-invariant vector field with $X^{\nabla_G f(x)}(e) = (\mathrm{d}\ell_{x^{-1}})_x [\nabla_G f(x)]$.

Proof. Let $h \in \mathfrak{g}$ be a left-invariant vector field. To prove the statement we compute $\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))\Big|_{t=0}$ with $\gamma(t) = \ell_x \circ \exp(th)$ and $\gamma(0) = x$ in a different way. A comparison with the previous computation in equation (1.18) will yield the result.
First observe for arbitrary $\tilde{f} \in C^\infty(G)$

$$\begin{aligned}
\dot{\gamma}(0)\tilde{f} &= \mathrm{d}\gamma \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_{r=0} \right] \tilde{f} \\
&= \mathrm{d}\ell_x \circ \mathrm{d}\exp_h \left[\frac{\mathrm{d}}{\mathrm{d}r}\Big|_{r=0} \right] \tilde{f} \\
&\overset{(1.8)}{=} \mathrm{d}\ell_x \circ h(\exp_h(0))\tilde{f} \\
&\overset{(1.4)}{=} h \circ \ell_x(\exp_h(0))\tilde{f} \\
&= h(x)\tilde{f}
\end{aligned} \tag{1.22}$$

and

$$(\mathrm{d}\ell_{x^{-1}})_x\,[h(x)] = h \circ \ell_{x^{-1}}(x) = h(e). \tag{1.23}$$

For both equations it is essential that h is left-invariant. In equation (1.18), we can consider $\gamma\colon \mathbb{R} \to G$ as curve and compute for the cost functional f

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} f\left(\gamma(t)\right)\Big|_{t=0} &= \dot{\gamma}(0) f \\
&\overset{\text{Def. 1.58}}{=} \left\langle \nabla_G f(x), \dot{\gamma}(0) \right\rangle_{T_x G} \\
&\overset{(1.22)}{=} \left\langle \nabla_G f(x), h(x) \right\rangle_{T_x G} \\
&\overset{(1.19)}{=} \left\langle (\mathrm{d}\ell_{x^{-1}})_x\,[\nabla_G f(x)]\,, (\mathrm{d}\ell_{x^{-1}})_x\,[h(x)] \right\rangle_{T_e G} \\
&\overset{(1.23)}{=} \left\langle (\mathrm{d}\ell_{x^{-1}})_x\,[\nabla_G f(x)]\,, h(e) \right\rangle_{T_e G} \\
&\overset{(1.20)}{=} \left\langle X^{\nabla_G f(x)}, h \right\rangle_{\mathfrak{g}},
\end{aligned} \tag{1.24}$$

where $X^{\nabla_G f(x)}$ is the left-invariant vector field with $X^{\nabla_G f(x)}(e) = (\mathrm{d}\ell_{x^{-1}})_x\,[\nabla_G f(x)]$. From equations (1.18) and (1.24) it follows

$$\langle \nabla_{\mathfrak{g}} (f \circ \ell_x \circ \exp)(0), h \rangle_{\mathfrak{g}} = \langle X^{\nabla_G f(x)}, h \rangle_{\mathfrak{g}}$$

and thus, since h was arbitrary

$$\nabla_{\mathfrak{g}} (f \circ \ell_x \circ \exp)(0) = X^{\nabla_G f(x)}.$$

□

This characterization of $\nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0)$ may seem quite technical, but one has to realize that the gradient $\nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0)$ is an element of $T_0\mathfrak{g}$ which is isomorphic to the Lie algebra $\mathfrak{g}$ itself and thus, both sides of equation (1.21) are left-invariant vector fields on G.
Lemma 1.69 provides a way to describe $\nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0)$ in dependence on $\nabla_G f(x)$. But still we would have to determine a left-invariant vector field in order to give an explicit expression for $\nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0)$. To avoid this, we can use the isomorphism $F\colon \mathfrak{g} \to T_e G$ from Theorem 1.39(ii) to identify the gradient with a tangent vector at $e \in G$.

Lemma 1.70. *For the isomorphism $F\colon \mathfrak{g} \to T_e G$, with $F(h) = h(e)$ from Theorem 1.39(ii), we have*

$$\nabla_{\mathfrak{g}} (f \circ \ell_x \circ \exp)(0_{\mathfrak{g}})(e) = \nabla_{T_e G} (f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_e G}).$$

Proof. Let $h \in \mathfrak{g}$ and set $\beta(t) = th$. Then, we can compute

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} f \circ \ell_x \circ \exp(\beta(t))\Big|_{t=0} &= \frac{\mathrm{d}}{\mathrm{d}t} f \circ \ell_x \circ \exp \circ F^{-1}(\beta(t)(e))\Big|_{t=0} \\
&= \left\langle \nabla_{T_e G} (f \circ \ell_x \circ \exp \circ F^{-1})(\beta(0)(e)), \dot{\beta}(0)(e) \right\rangle_{T_e G} \\
&= \left\langle \nabla_{T_e G} (f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_e G}), h(e) \right\rangle_{T_e G}.
\end{aligned}$$

Comparison with equation (1.18) yields

$$\begin{aligned}\langle \nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(\beta(0))(e), h(e)\rangle_{T_eG} &\overset{(1.20)}{=} \langle \nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(\beta(0)), h\rangle_{\mathfrak{g}} \\ &\overset{(1.18)}{=} \left\langle \nabla_{T_eG}(f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_eG}), h(e)\right\rangle_{T_eG}\end{aligned}$$

which completes the proof. □

Now, we combine Lemma 1.69 and Lemma 1.70 in order to characterize the gradient in T_eG.

Theorem 1.71. *It holds*

$$\nabla_{T_eG}(f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_eG}) = (\mathrm{d}\ell_{x^{-1}})_x \left[\nabla_G f(x)\right].$$

Proof. With the lemmas we can compute

$$\begin{aligned}\nabla_{T_eG}(f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_eG}) &\overset{\text{Lem. } 1.70}{=} \nabla_{\mathfrak{g}}(f \circ \ell_x \circ \exp)(0_{\mathfrak{g}})(e) \\ &\overset{\text{Lem. } 1.69}{=} X^{\nabla_G f(x)}(e) \\ &= (\mathrm{d}\ell_{x^{-1}})_x \left[\nabla_G f(x)\right].\end{aligned}$$

□

With Theorem 1.71 we have provided a way to describe ∇_{T_eG} in terms of ∇_G which is often simpler to compute as we will see in Section 1.3.2.

Remark 1.72. With the aid of ∇_{T_eG}, we can reformulate Theorem 1.66. For $h(e) \in T_eG$ and $x \in G$ with $\langle \nabla_{T_eG}(f \circ \ell_x \circ \exp \circ F^{-1})(0_{T_eG}), h(e)\rangle = \langle (\mathrm{d}\ell_{x^{-1}})_x \left[\nabla_G f(x)\right], h(e)\rangle < 0$ the curve $\gamma(t) = \ell_x \circ \exp \circ F^{-1}(th(e))$ is a descent curve of f at x. This can be seen using Lemma 1.70 and equation (1.20).

Together with Theorem 1.71, we are now able to compute the updates starting in the origin. This can be thought of as a shift to the origin as motivated in the beginning of this section (compare [97]). Let us rewrite Algorithm 1.64 to summarize these achievements.

Algorithm 1.73. Gradient descent method with Armijo rule and shifting: Let G be a Lie group with left-invariant Riemannian metric induced by the inner product of its associated Lie algebra $\mathfrak{g}$.

1: **choose** $x^0 \in G$, σ, $\beta \in]0,1[$, $\epsilon \geq 0$ and **set** $k = 0$
2: **for** $\theta^k = \left\| (\mathrm{d}\ell_{(x^k)^{-1}})_{x^k} \left[\nabla_G f(x^k)\right] \right\|_{T_eG} > \epsilon$ **do**
3: **set** descent direction $h^k = -\frac{(\mathrm{d}\ell_{(x^k)^{-1}})_{x^k}\left[\nabla_G f(x^k)\right]}{\theta^k} \in T_eG$
4: and descent curve $\gamma^k(t) = \ell_{x^k} \circ \exp \circ F^{-1}(th^k)$
5: **compute** $t^k = \max\{\beta^l : l \in \mathbb{N} \text{ and } f(\gamma^k(\beta^l)) \leq f(x^k) - \sigma\beta^l\theta^k\}$
6: **set** $x^{k+1} = \gamma^k(t^k)$ and k to $k+1$
7: **end for**

Once more observe that we do not need to compute the logarithm in Algorithm 1.73. In particular, we do not have to worry about staying in a neighborhood of $e \in G$, where the logarithm is defined. Furthermore, the multiplicative update g^k aimed for in the beginning of this section is explicitly given as

$$g^k = \exp(F^{-1}(t^k h^k)).$$

It is an element of the image of the 1-parameter subgroup $g(t) = \exp(F^{-1}(th))$. This 1-parameter subgroup describes the 'line' in direction h and generalizes the concept of line search. Due to the commutativity of the image of a 1-parameter subgroup (see Definition 1.45 et seq.) a line search along the descent curves $\ell_x \circ g(t)$ does not depend on the step size nor on the order of the steps.

1.3.2 Rotation of data clouds in $\mathbb{R}^d$

In this section we will discuss a concrete optimization problem in order to illustrate the theoretical explanations of this chapter. Let us consider a point cloud $X = \{x_1, \dots, x_n\}$ in $\mathbb{R}^d$. By X we might also refer to the matrix $(x_1, \dots, x_n) \in \mathbb{R}^{d \times n}$.
In applications, it is often desirable to compute a rotated configuration of the data with certain properties. In non-negative dimensionality reduction, for instance, the low-dimensional data is required to have non-negative coordinates, i.e., $X_{ij} \geq 0$ for $i = 1, \dots, d$ and $j = 1, \dots, n$ (see Chapter 3). If this requirement is not fulfilled, a rotation R can be applied to the data with the objective that $(RX)_{ij} \geq 0$ (see Figure 1.3).

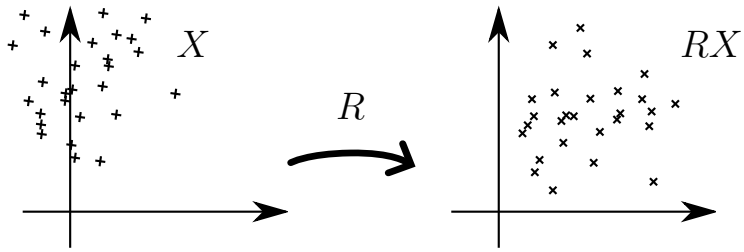


Figure 1.3: Rotation of a data set $X \in \mathbb{R}^d$. After the rotation the data is entry-wise non-negative.

Obviously, such a rotation only exists if the data is lying inside a cone with opening angle $\frac{\pi}{2}$, i.e., if for all pairs of data points x_i, $x_j \in X$ we have

$$\frac{\langle x_i, x_j \rangle}{\|x_i\| \, \|x_j\|} \geq \cos\left(\frac{\pi}{2}\right) = 0.$$

Otherwise, if the data points are spreading too much they cannot be rotated to the positive orthant. Even though for $d = 2$ the task of finding a rotation R to the positive

orthant is simple, it can be quite challenging for higher dimensions. Furthermore, note that such a rotation is in general not unique.
The above motivated task can be formulated as an optimization problem on the Lie group of special orthogonal matrices of dimension d

$$\min_{R\in SO(d)} f(R), \tag{1.25}$$

where the cost functional $f\colon SO(d) \to \mathbb{R}$ should penalize the negative entries of RX. Therefore, we choose

$$f(R) = \|(RX)_-\|_F^2$$

where $(RX)_-$ denotes the matrix with the negative entries of RX

$$((RX)_-)_{ij} = ((RX)_{ij})_- = \min\{(RX)_{ij}, 0\} = \begin{cases} 0, & \text{if } (RX)_{ij} \geq 0, \\ (RX)_{ij} & \text{if } (RX)_{ij} < 0. \end{cases}$$

This cost function reaches its minimum 0 if all coordinates of the rotated data are non-negative, i.e., if the data cloud is in the positive orthant. Of course, other choices of f are possible, but we stick to this one proposed in a similar context in [97]. The optimization problem (1.25) has the form of (1.14) discussed in Section 1.2.2 and thus, we can apply the theory of that section and in particular Algorithm 1.73 to find an optimal rotation R_*.
To implement the algorithm, we need to determine the involved objects. From Example 1.42 we know that the associated Lie algebra $\mathfrak{so}(d)$ of $SO(d)$ is isomorphic to the set of skew-symmetric matrices $\mathrm{Skew}(d) = T_{\mathbf{Id}_d}SO(d)$ and in Example 1.52 we have shown that the exponential map on $\mathfrak{so}(d)$ is the matrix exponential up to the isomorphism F with

$$\exp \circ F^{-1}(A) = e^A \qquad \text{for } A \in T_{\mathbf{Id}_d}SO(d).$$

It remains to define an inner product on $\mathfrak{so}(d)$ with its induced left-invariant Riemannian metric on $SO(d)$ and to compute the gradient

$$\begin{aligned} \nabla_{T_{\mathbf{Id}_d}SO(d)}(f \circ \ell_R \circ \exp \circ F^{-1})(0_{\mathfrak{so}(d)}) &= \nabla_{T_{\mathbf{Id}_d}SO(d)}(f \circ \ell_R \circ e)(0_{T_{\mathbf{Id}_d}SO(d)}) \\ &= (\mathrm{d}\ell_{R^{-1}})_R \left[\nabla_{SO(d)} f(R)\right]. \end{aligned}$$

To this end, we consider the inner product on $\mathfrak{so}(d)$

$$\begin{aligned} \langle h^1, h^2\rangle_{\mathfrak{so}(d)} &\overset{(1.20)}{=} \langle h^1(\mathbf{Id}_d), h^2(\mathbf{Id}_d)\rangle_{T_{\mathbf{Id}_d}SO(d)} \\ &= \langle B^1, B^2\rangle_{\mathrm{Skew}(d)} \\ &\overset{(1.42)}{=} \mathrm{tr}\left((B^1)^T B^2\right), \end{aligned}$$

for $h^i \in \mathfrak{so}(d)$ with $h^i(\mathbf{Id}_d) = B^i$. This defines an inner product $\langle\cdot,\cdot\rangle_{T_{\mathbf{Id}_d}SO(d)}$ on the tangent space $T_{\mathbf{Id}_d}SO(d)$ of $SO(d)$ at the identity. In order to compute $\nabla_{SO(d)}f(R)$ we need to extend $\langle\cdot,\cdot\rangle_{T_{\mathbf{Id}_d}SO(d)}$ to a left-invariant Riemann metric on $SO(d)$. Recall that

the tangent space of $SO(d)$ at R is given by $T_R SO(d) = \{RB \colon B \in \mathrm{Skew}(d)\}$. Furthermore, the differential of the left-translation $(\mathrm{d}\ell_{R^{-1}})_R$ is in this setting the multiplication by R^{-1} (compare Example 1.34). Then, equation (1.19) yields for $\xi_R^1, \xi_R^2 \in T_R SO(d)$

$$\begin{aligned}
\langle \xi_R^1, \xi_R^2 \rangle_{T_R SO(d)} &= \left\langle (\mathrm{d}\ell_{R^{-1}})_R \left[\xi_R^1\right], (\mathrm{d}\ell_{R^{-1}})_R \left[\xi_R^2\right] \right\rangle_{T_{R^{-1}R} SO(d)} \\
&= \langle R^{-1}\xi_R^1, R^{-1}\xi_R^2 \rangle_{T_{\mathbf{Id}_d} SO(d)} \\
&= \mathrm{tr}\left((\xi_R^1)^T R R^{-1} \xi_R^2 \right) \\
&= \mathrm{tr}\left((\xi_R^1)^T \xi_R^2 \right) \\
&= \langle \xi_R^1, \xi_R^2 \rangle_F .
\end{aligned}$$

This shows that the inner product on $T_R SO(d)$ is the Frobenius inner product. Using the definition of the gradient, this implies

$$\langle \nabla_{SO(d)} f(R), Y \rangle_F = \langle \nabla_{SO(d)} f(R), Y \rangle_{T_R SO(d)} \overset{(1.15)}{=} Df(R)(Y) = \langle \nabla_{M(n,\mathbb{R})} f(R), Y \rangle_F$$

for all $Y \in T_R SO(d) = \{RB \colon B \in \mathrm{Skew}(d)\}$. Here, $\nabla_{M(n,\mathbb{R})} f(R)$ denotes the gradient in the Euclidean space $M(n, \mathbb{R})$. This equation is equivalent to

$$\langle R^T \nabla_{SO(d)} f(R), B \rangle_F = \langle R^T \nabla_{M(n,\mathbb{R})} f(R), B \rangle_F \qquad \text{for all } B \in \mathrm{Skew}(d).$$

From this we conclude that the skew-symmetric part of the matrices $R^T \nabla_{SO(d)} f(R)$ and $R^T \nabla_{M(n,\mathbb{R})} f(R)$ has to be the same. The matrix $R^T \nabla_{SO(d)} f(R)$ is skew-symmetric since $\nabla_{SO(d)} f(R) \in T_R SO(d) = \{RB \colon B \in \mathrm{Skew}(d)\}$ and $R^T R = \mathbf{Id}_d$. And thus, we get

$$R^T \nabla_{SO(d)} f(R) = \mathrm{skew}(R^T \nabla_{M(n,\mathbb{R})} f(R)) = \frac{1}{2}\left(\nabla_{M(n,\mathbb{R})} f(R) - \left(\nabla_{M(n,\mathbb{R})} f(R) \right)^T \right).$$

Now, the gradient $\nabla_{SO(d)} f(R)$ can be computed by partial differentiation as in real analysis which yields

$$\begin{aligned}
(\mathrm{d}\ell_{R^{-1}})_R \nabla_{SO(d)} f(R) = R^{-1} \nabla_{SO(d)} f(R) &= \mathrm{skew}\left(R^T \nabla_{M(n,\mathbb{R})} f(R) \right) \\
&= \mathrm{skew}\left(R^T \begin{pmatrix} \frac{\partial}{\partial R_{11}} f & \cdots & \frac{\partial}{\partial R_{1d}} f \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial R_{d1}} f & \cdots & \frac{\partial}{\partial R_{dd}} f \end{pmatrix} \right) \\
&= \mathrm{skew}(-2R^T (RX)_- X^T) \\
&= X(RX)_-^T R - R^T (RX)_- X^T .
\end{aligned}$$

Now, that we have all ingredients to determine an optimal rotation R_*, we rewrite Algorithm 1.73.

Algorithm 1.74. Gradient descent method with Armijo rule and shifting in $SO(d)$:

1: **choose** $R^0 \in G$, σ, $\beta \in]0, 1[$, $\epsilon \geq 0$ and **set** $k = 0$

2: **for** $\theta^k = \left\| X(R^kX)_-^T R^k - (R^k)^T (R^kX)_- X^T \right\|_F > \epsilon$ **do**
3: **set** descent direction $h^k = -\frac{X(R^kX)_-^T R^k - (R^k)^T (R^kX)_- X^T}{\theta^k} \in \mathrm{Skew}(d)$
4: and descent curve $\gamma^k(t) = R^k e^{th^k}$
5: **compute** $t^k = \max \left\{ \beta^l : l \in \mathbb{N} \text{ and } \left\| \left(\gamma^k(\beta^l)X\right)_- \right\|_F^2 \leq \left\| \left(R^kX\right)_- \right\|_F^2 - \sigma\beta^l\theta^k \right\}$
6: **set** $R^{k+1} = \gamma^k(t^k)$ and k to $k+1$
7: **end for**

1.3.3 Summary

In this section, we have experienced the benefit of the precise theoretical considerations in the previous parts of this chapter. Here, we have worked at the interface of theory and praxis. In this field, we have attained a great achievement by formulating certain optimization problems on Lie groups in a rigorous way. We have shown how a multiplicative update algorithm on Lie groups can be formulated, where we used the special structure of Lie groups and their associated Lie algebras. A main advantage of this procedure is that at each iteration step the updated value is still a group element. This prevents us from imposing additional constraints making the optimization even more complex.
The core step towards the definition of a multiplicative update was the shifting of the current iterate to the origin (see Figure 1.4). This has the further advantage that the

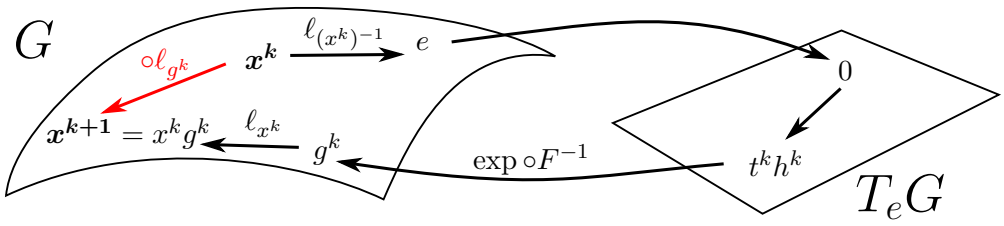


Figure 1.4: Update algorithm on the Lie group G. A multiplicative update is obtained by shifting the problem to the origin $e \in G$ and calculating from there an update step via transferring the problem to the tangent space at the identity and back. Note that the computation of the logarithm is not needed as $\log(e) = 0$.

computation of the logarithm is not needed any more and thus, we are not restricted to its domain of definition as before (compare Figure 1.2).
Furthermore, we have discussed an example which plays a key role in our non-negative dimensionality reduction approach (see Section 2.3). The optimization was performed on the matrix manifold $SO(d)$, but the theory we have provided is not restricted to this Lie group.

2 Non-negative dimensionality reduction

High-dimensional data sets occur frequently in many scientific disciplines such as physics, medicine and musicology, to mention just a few. Due to the extremely fast growing data storage capacity, analysis and interpretation of this kind of data sets pose new mathematical and computational challenges and there has been an increasing demand for effective methods to process high-dimensional data in the last decades. Especially in the fields of data analysis and machine learning new methods known as dimensionality reduction methods have been developed in order to understand, visualize and process the structure of such data sets.

Many high-dimensional data sets from applications possess the inherent property that the data's intrinsic dimensionality is actually low. The core idea of all dimensionality reduction methods is to make use of this property by embedding the data into a significant manifold of lower dimension within the high-dimensional space in order to encode important information of the data set. This lower dimension should ideally correspond to the intrinsic dimensionality of the data. Different strategies are available for estimating this dimensionality (see [75] or [119]).

This approach is motivated by the observation that in many cases less than all information contained in the data points is sufficient for understanding the underlying characteristics or properties of the data. A logical consequence is that for many applications a reduction of the data's dimensionality might improve the quality and speed up the computation of the data analysis. Also, low-dimensional data sets are much easier to operate with in case of classification, visualization or compression.

Within the recent years, researcher became aware that not only the data's intrinsic dimensionality is worth to be preserved beyond the reduction. In particular, several applications arose where the entry-wise non-negativity of the data is of major importance (see [3, 12, 51, 52, 89, 92]). Especially, if the dimensionality reduction is included in an established procedure for computational reasons, the non-negativity of the low-dimensional data might be necessary to proceed with the next step. This is for example the case for the application of dimensionality reduction in signal separation, where the classical decomposition tools require a non-negative input. The high-dimensional data set is typically obtained from a discrete signal transform and it is non-negative by construction. Therefore, this particular example requires non-negativity preserving dimensionality reduction methods.

Since there are plenty of efficient dimensionality reduction methods without the additional non-negativity constraint (see e.g. [73, 125]), the generic procedure is to use the knowledge about these methods for constructing non-negative dimensionality reduction methods. Especially, if we formulate the problem of finding a low-dimensional representation of the data as an optimization problem as in [18], we can benefit from both the

formulation of the problem and the known reduction map. In fact, only an additional constraint forcing the low-dimensional data to be non-negative needs to be included. This describes and motivates our approach to non-negative dimensionality reduction.
There have been several other attempts to develop methods of these kind. Basically, it can be differentiated between two main approaches. The first one applies only to linear dimensionality reduction methods, i.e., the reduction map is given by a matrix. In addition to the constraints of the dimensionality reduction problem this matrix is required to be non-negative. This strategy was studied for sparse principal component analysis (PCA) by Zass and Shashua in 2007 [133] and picked up in different contexts in [31], [53] and others. A similar approach to multidimensional PCA has been developed by Panagakis et al. in [92] for music genre classification, where the data is given as a tensor of higher order.
From our point of view, requiring the reduction map to be non-negative is not the best way to ensure that the reduction preserves the non-negativity of the data. First of all, this restricts the class of methods to choose from to linear reductions and secondly, the assumption itself is much stronger than only requiring the low-dimensional data to be non-negative.
This motivates the other of the above mentioned approaches to non-negative dimensionality reduction. Here, the additional constraint concerns directly the low-dimensional data set by forcing it to be entry-wise non-negative (as it is the high-dimensional data set). This approach was taken by [3, 4, 107] for developing non-negative PCA methods for sparse data. Since data from signal processing is not necessarily sparse, one of the main objectives of this work was to develop a better suited non-negativity preserving PCA method. Indeed, we have been able to formulate and solve a non-negative PCA problem and on top we achieve the same approximation errors as the classical PCA method. This is surprising as it means that the additional constraint does not affect the quality of the approximation. Our approach provides an efficient computation of the reduction map. The key idea is to divide the optimization problem into two steps. In the first step of this splitting approach a classical dimensionality reduction method (e.g. PCA) is performed and in the second step the non-negativity of the data is achieved through a rotation of the data set.
Different approaches to non-negative PCA can be found in [87] from 2014 based on theoretical statistical considerations and in [95] from 2004 based on a non-linear PCA due to [88].
The procedure we propose for the non-negative PCA can also be used for other dimensionality reduction techniques, no matter if linear or not, but the corresponding reduction map has to satisfy certain properties. In particular, we need a good approximative inverse of this map for both theoretical issues and the application of the developed method to signal separation.
Non-linear non-negative dimensionality reduction techniques have also been studied by other authors. Several methods are briefly introduced by Zafeiriou and Laskaris in [132] from 2010, a non-negative locally linear embedding (LLE) approach was developed in 2013 in [127] and a non-negative Laplacian eigenmaps (LE) can be found in [78]. These approaches are based on an optimization problem subject to constraints on the

low-dimensional data, where the conditions are posed in a very restrictive form. More precisely, the low-dimensional data is not only required to be non-negative but also to have orthogonal rows. The combination of both constraints seems rather limiting in signal separation.
Completely different approaches to non-linear non-negative dimensionality reduction are due to [23] where a non-negative LLE is constructed using a special neighborhood structure of the data and to a very recent paper [12] from 2015 where a method called prototype vector projection is proposed. The latter method has non-negative output data by construction. Furthermore, there are many non-negative dimensionality reduction methods based on non-negative matrix factorization which have not been studied for this work (e.g. [90, 116]).

In this chapter we will first briefly introduce the concept of dimensionality reduction and basic notations in Section 2.1. In Section 2.2 the general formulation of dimensionality reduction as an optimization problem is studied. We will discuss PCA in Section 2.2.1, multidimensional scaling (MDS) in Section 2.2.2, Isomap in 2.2.3 and other non-linear methods in 2.2.4. In Section 2.3 we will introduce the non-negative dimensionality reduction problem in form of an optimization problem. Here, we will propose a new approach to non-negative dimensionality reduction. This splitting approach is discussed in Section 2.3.2. We investigate the applicability of our approach to different dimensionality reduction methods in Section 2.4. In particular, in Section 2.4.1 our non-negative PCA algorithm is formulated and analyzed.

2.1 Basic notations

Mathematically, the problem of dimensionality reduction can be formulated as in [48]: Let $X = \{x_k\}_{k=1}^n \subset \mathbb{R}^D$ be a data set of dimensionality $D \in \mathbb{N}$. In abuse of notation we will denote by X also the matrix $X \in \mathbb{R}^{D\times n}$ with columns x_k. If much of the information described by X is redundant and can be neglected, we try to find a low-dimensional data set $Y \subset \mathbb{R}^d$ which best represents X while conserving the characteristics of the data such as distances, geometry or other features. The dimensionality d of Y is called *intrinsic dimensionality* of the data X and we assume that it satisfies $d \ll D$. This process is called *dimensionality reduction*.
In this context, the data is assumed to lie on (or nearby) a (smooth) manifold $\mathcal{M}$ embedded in a D-dimensional space. More precisely, we assume X to be sampled from $\mathcal{M}$, a ν-dimensional smooth compact manifold of $\mathbb{R}^D$. In mathematical terms, we search for a homeomorphism $\mathcal{B} : \mathbb{R}^D \supset \mathcal{M} \to \Omega \subset \mathbb{R}^d$, where Ω is a ν-dimensional submanifold of $\mathbb{R}^d$ (see Figure 2.1 for an illustration).
Recall that due to the Whitney Embedding Theorem every ν-dimensional smooth connected manifold can be embedded in $\mathbb{R}^d$, for all d with $d \geq 2\nu$ (compare Theorem 1.11). Now, the objective is to construct a low-dimensional data set Y representing X and its structure using the geometrical informations given by $\mathcal{M}$. The homeomorphism $\mathcal{B}$ maps the data set X with dimensionality D onto a new data set Y with dimensionality d preserving the main structure of the data.

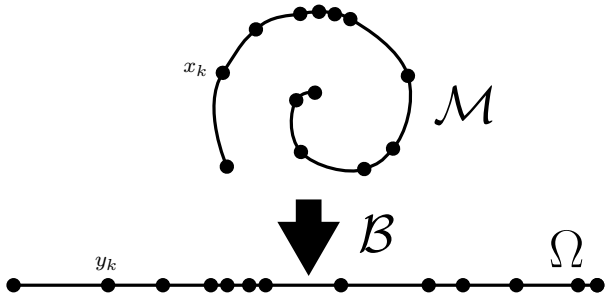


Figure 2.1: A manifold $\mathcal{M} \subset \mathbb{R}^D$ is embedded in a low-dimensional space $\mathbb{R}^d$ (here $D = 2, d = 1$). The pair $(\Omega, \mathcal{B})$ is a submanifold and the low-dimensional representation Ω of $\mathcal{M}$ has basically the same geometrical structure. The main structure of the data set X (here the geodesic distances between the data points) is preserved.

In practice, neither the manifold $\mathcal{M}$ nor its low-dimensional representation Ω is known. Therefore, we can only approximate the homeomorphism $\mathcal{B}$ by a dimensionality reduction mapping P as shown in the diagram in Figure 2.2.

$$\begin{array}{ccccc} X & \subset & \mathcal{M} & \subset & \mathbb{R}^D \\ \downarrow{\scriptstyle P} & & \downarrow{\scriptstyle \mathcal{B}} & & \\ Y & \subset & \Omega & \subset & \mathbb{R}^d \end{array}$$

Figure 2.2: The dimensionality reduction map P approximates the unknown homeomorphism $\mathcal{B}$.

These concepts are the basis of different dimensionality reduction methods developed in the last decades. They can be classified in linear and non-linear techniques and among them are PCA [93], MDS [115], Isomap [114], Laplacian eigenmaps [7] and locally linear embedding [104], (local tangent space alignment [136]) to mention just a few. In this context, linearity refers to the idea that each data point on the manifold is a linear combination of the original data points, i.e., we assume the manifold $\mathcal{M}$ to be a linear subspace (see [38]). For more information about dimensionality reduction we refer to [73].

2.2 Dimensionality reduction as an optimization problem

In this work we will concentrate on dimensionality reduction methods which can be computed as the solution of an optimization problem of the form

$$\min_{P \in \mathcal{U}} g(P), \tag{2.1}$$

where $\mathcal{U} \subset \{f\colon X \to \mathbb{R}^d\}$ and $g : \mathcal{U} \to \mathbb{R}^d$. The cost functional g can be interpreted as the measure of the distance of P to the homeomorphism $\mathcal{B}$. The choice of the pair $(\mathcal{U}, g)$ determines the dimensionality reduction method P and the minimization problem (2.1) is (in general) a non-convex problem. Let us summarize this in the following definition.

Definition 2.1. We call a problem of the form (2.1) (i.e., of computing $P\colon X \to \mathbb{R}^d$) a *dimensionality reduction problem*. A solution

$$P \in \operatorname*{arg\,min}_{\tilde{P}\in\mathcal{U}} g(\tilde{P})$$

of a dimensionality reduction problem is called a *dimensionality reduction method*. The solution is in general not unique, but it can be made unique by restricting $\mathcal{U}$ sufficiently.

Remark 2.2. Note, that we consider optimization problems with admissible sets consisting of reduction maps and not of low-dimensional data sets. A brief summary of dimensionality reduction as an optimization problem can be found in [18].

It may seem a bit cumbersome to formulate the dimensionality reduction problem in this way, but it will turn out that this form is convenient for further computations. Later in this work, we are interested in dimensionality reduction methods with a rotationally invariant cost functional g and an angle-preserving reduction map P. This can be verified easily if the dimensionality reduction problem has the form (2.1). Moreover, we will see that it is beneficial if $\mathcal{U}$ has some structure (e.g. a Lie group structure).
In the following, we will briefly discuss some dimensionality reduction methods, where we focus on writing the corresponding dimensionality reduction problems in the form of (2.1).

2.2.1 Principal Component Analysis - PCA

Principal component analysis (PCA) is probably one of the most frequently used techniques in multivariate data analysis. Due to its structure based on singular value decomposition it permits an efficient implementation. As PCA has many applications, it was discovered independently in different scientific fields and improved by many scientists. It was first introduced by Pearson [93] in 1901 in a biological framework. In the field of stochastic processes PCA is also known as the Karhunen-Loève transform.
As stated before, we consider a data matrix $X = (x_1, \dots, x_n) \subset \mathbb{R}^{D\times n}$. In the case of PCA the data points are assumed to lie on or nearby a d-dimensional linear subspace $\mathcal{M}$ of $\mathbb{R}^D$ and thus, we can choose $\Omega = \mathbb{R}^d$ as its low-dimensional representation. The aim is to find the subspace $\mathcal{M}$ such that the sum of the Euclidean distances from the points x_k to $\mathcal{M}$ is minimized. It is well-known that the nearest point $\hat{x}_k$ to x_k in $\mathcal{M}$ is given by the orthogonal projection of x_k to $\mathcal{M}$, i.e., $x_k - \hat{x}_k \perp \mathcal{M}$ or equivalently $\langle x_k - \hat{x}_k, p\rangle = 0$ for all $p \in \mathcal{M}$.
Since $\mathcal{M}$ is a linear subspace, we can choose an orthonormal basis $U \in \mathbb{R}^{D\times d}$ with $\mathcal{M} = U\mathbb{R}^d$. Note that we call a matrix *orthonormal* even though only its columns *or* rows are orthonormal, i.e., the matrix U is not necessarily quadratic. As a consequence, for $U \in \mathbb{R}^{D\times d}$ we have $U^TU = \mathbf{Id}_d$, but in general $UU^T \neq \mathbf{Id}_D$.

For $\hat{x}_k \in U\mathbb{R}^d$ we fix $y_k \in \mathbb{R}^d$ with $\hat{x}_k = Uy_k$ and obtain

$$\begin{aligned} & 0 = \langle x_k - Uy_k, Uz\rangle = (x_k^T U - y_k^T U^T U)z \quad \text{for all } z \in \mathbb{R}^d \\ \Leftrightarrow \quad & 0 = x_k^T U - y_k^T \\ \Leftrightarrow \quad & y_k = U^T x_k. \end{aligned}$$

Thus, $\hat{x}_k = Uy_k = UU^T x_k$ and the low-dimensional representation of x_k is given by $y_k = U^T x_k$. Altogether, the set $Y \in \mathbb{R}^{d\times n}$ is obtained by projecting the set X onto $\mathbb{R}^d$ by the orthonormal matrix $U^T \in \mathbb{R}^{d\times D}$ such that

$$Y = U^T X.$$

The projection U^T is then a minimizer of the squared Euclidean distance of the original data to the data points on the subspace, i.e., of the PCA problem

$$\min_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} \sum_{k=1}^{n} \left\| x_k - \tilde{U}\tilde{U}^T x_k \right\|_2^2 = \min_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} \left\| X - \tilde{U}\tilde{U}^T X \right\|_F^2. \tag{2.2}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm. To minimize this functional, we have to assure that the low-dimensional representation captures as much of the spreading (or scattering) of the data as possible and discards the redundancy in terms of correlation. This is the case if the first spanning vector of the subspace is pointing in the direction in which the data is scattering the most and each of the other $d-1$ spanning vectors is pointing in the direction of the widest spread under the constraint of being orthogonal to the previous ones. Figure 2.3 serves to illustrate the general idea of PCA. It depicts the distances $\|x_k - \hat{x}_k\|_2$, whose squared sum has to be minimized, and the directions v_i in which the data is distributed with maximum variance. To reduce the dimension, the data is projected on the first direction. Thus, PCA can be interpreted as a truncated principal axis transformation.
Since the variance of a data set is a measure for its range of spread, the minimization procedure (2.2) is equivalent to maximizing the variance of the low-dimensional data, which is the trace of the data's covariance matrix

$$\max_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} \operatorname{tr}(YY^T) = \max_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} \operatorname{tr}(\tilde{U}^T XX^T\tilde{U}) = -\min_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} -\operatorname{tr}(\tilde{U}^T XX^T\tilde{U}). \tag{2.3}$$

Theorem 2.3 (Principal component analysis). *The above introduced optimization problems* (2.2) *and* (2.3) *have the same minimizer. It is given by* $U^T = \mathbf{Id}_{d\times D} V^T = {V_d}^T$, *where* V *contains the eigenvectors of* XX^T *sorted in decreasing order by the size of the corresponding eigenvalues.*

To prove this theorem let us recall the Eckart-Young-Mirsky Theorem on the best d-rank approximation of a matrix relying on a special form of the singular value decomposition.

Theorem 2.4 (Eckart and Young(1936), Mirsky (1960)). *Let the singular value decomposition of matrix* $X \in \mathbb{R}^{D\times n}$ *with* $\operatorname{rk}(X) = r$

$$X = V\Sigma W^T = \sum_{i=1}^{r} \sigma_i v_i {w_i}^T,$$

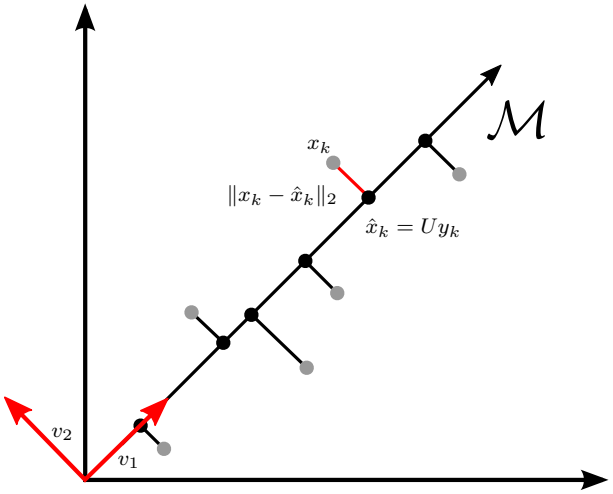


Figure 2.3: A data set $X \subset \mathbb{R}^D$ is embedded in a d-dimensional subspace (here $D = 2, d = 1$). The subspace $\mathcal{M}$ is spanned by v_1, the first principal component (first eigenvector) of XX^T. The approximation error is the sum over k of $\|x_k - \hat{x}_k\|_2^2$ and it is minimized by taking the orthogonal projection on the subspace spanned by the first principal components.

where $V = (v_1, \ldots, v_r) \in \mathbb{R}^{D\times r}$ *with* $V^TV = \mathbf{Id}_r$, $W = (w_1, \ldots, w_r) \in \mathbb{R}^{n\times r}$ *with* $W^TW = \mathbf{Id}_r$ *and* $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$ *is a diagonal matrix with* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. *Let*

$$X_* = \sum_{i=1}^{d} \sigma_i v_i {w_i}^T, \qquad d \leq r.$$

Then, $\operatorname{rk}(X_*) = d$ *and* X_* *is the best* d*-rank approximation under the Frobenius norm, namely,*

$$\|X - X_*\|_F^2 = \min_{\substack{B\in\mathbb{R}^{D\times n}\\ \operatorname{rk}(B)=d}} \|X - B\|_F^2,$$

with approximation error

$$\|X - X_*\|_F^2 = \sum_{l=d+1}^{r} \sigma_l^2.$$

Proof. For a proof see [125]. □

Proof of Theorem 2.3. First we show that the minimization problems

$$\min_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} \left\|X - \tilde{U}\tilde{U}^TX\right\|_F^2 \text{ and } \min_{\tilde{U}^T\tilde{U}=\mathbf{Id}_d} -\operatorname{tr}(\tilde{U}^TXX^T\tilde{U})$$

are equivalent. Bearing in mind the orthonormality property of $\tilde{U}$, by the well-known

Pythagoras' theorem follows that

$$\begin{aligned}\operatorname{tr}(\tilde{U}^T X X^T \tilde{U}) &= \sum_{k=1}^{n} \|\tilde{U}^T x_k\|_2^2 \\ &= \sum_{k=1}^{n} \|x_k\|_2^2 - \|x_k - \tilde{U}\tilde{U}^T x_k\|_2^2 \\ &= \|X\|_F^2 - \|X - \tilde{U}\tilde{U}^T X\|_F^2.\end{aligned}$$

This shows that

$$\underset{\tilde{U}^T\tilde{U}=\mathbf{Id}_d}{\arg\min} -\operatorname{tr}(\tilde{U}^T X X^T \tilde{U}) = \underset{\tilde{U}^T\tilde{U}=\mathbf{Id}_d}{\arg\min} \left\| X - \tilde{U}\tilde{U}^T X \right\|_F^2.$$

Now, it remains to prove that $U = V\mathbf{Id}_{D\times d}$ is indeed a minimizer of this problem. This is clear, since $\mathbf{Id}_{d\times D} V^T X$ is the best d-rank approximation of X with singular value decomposition $X = V\Sigma W^T$ (see Theorem 2.4). □

Remark 2.5. As a consequence of Theorem 2.3 the maximal variance of a d-dimensional representation is

$$\operatorname{tr}({V_d}^T X X^T V_d) = \sum_{i=1}^{d} \sigma_i^2,$$

where σ_i are the singular values of X. The matrix $V_d = V\mathbf{Id}_{D\times d}$ is given by the singular value decomposition of $X = V\Sigma W^T$.

Remark 2.6. If the PCA model is fully respected, i.e., the data is exactly lying on $\mathcal{M}$, the smallest d is given by $D - \dim(\ker(XX^T))$. The trace $\operatorname{tr}({V_d}^T X X^T V_d) = \operatorname{tr}(YY^T)$ is maximal if we keep all eigenvalues of XX^T apart from the zero eigenvalues. The number of zero eigenvalues of XX^T is given by $\dim(\ker(XX^T))$.
In contrast, in real situations we often observe some noise and thus, the PCA model might be not fully respected. This can result in a situation where all eigenvalues of XX^T are larger than zero. In this case, d cannot be estimated without loss of information. But assuming that the spreading of the data points within the manifold is much larger than the noise, it is a natural procedure to choose for V_d only the eigenvectors associated to the largest eigenvalues. Then, we have almost the same situation as before and the approximation error depending on the dimension d is given by

$$\operatorname{err}_{PCA}(d, X) = \|X - V_d {V_d}^T X\|_F^2 = \sum_{i=d+1}^{D} \sigma_i^2. \tag{2.4}$$

Remark 2.7 (Inverse projection). In the case where the PCA model is fully respected, it is obvious that there exists a back-lifting, projecting the data from the subspace back to the original high dimensional space using U: Since PCA is based on an orthonormal projection it is invertible on the low-dimensional subspace, i.e., $X = UY$, if the data exactly lies on the subspace. But if the model is not fully respected, i.e., the data is not

lying in the subspace (just near by), it might be difficult to find an exact back projection and sometimes it does not even exist. To deal with this problem, we assume the data to lie in the subspace, and if it does not, we neglect the error.

Summing up, we can specify the cost functional introduced in (2.1)

$$g_{PCA}(\tilde{U}^T) = -\operatorname{tr}(\tilde{U}^T X X^T \tilde{U}),$$

which describes the scattering of the data and the admissible set

$$\mathcal{U}_{PCA} = \{\tilde{U}^T \in \mathbb{R}^{d\times D} \colon \tilde{U}^T\tilde{U} = \mathbf{Id}_d\}$$

as the set of linear functions from X to $\mathbb{R}^d$ which have orthogonal rows. The minimizer U^T of $\min_{\tilde{U}^T\in\mathcal{U}_{PCA}} g_{PCA}(\tilde{U}^T)$ is given by the singular value decomposition of X.

Remark 2.8 (Uniqueness). The solution of the PCA problem is not unique since the columns of U can be permuted and multiplied by -1.

Remark 2.9. Usually, PCA is applied to *centered* data sets, i.e., $\sum_i x_i = 0$. However, our approach does not need this restriction. This is especially useful for the construction of non-negative PCA (see Section 2.3).

Remark 2.10 (Computation). Numerically, the problem can be solved by computing the singular value decomposition of X or the eigenvalue decomposition of the data's covariance matrix XX^T, respectively. Theorem 2.3 states explicitly how U has to be chosen.

2.2.2 Multidimensional Scaling - MDS

Another classical approach for dimensionality reduction is metric *multidimensional scaling (MDS)*. In this context, scaling refers to the attempt to determine a configuration of points in a certain (metric) space from information about pairwise dissimilarities of objects. So to say, we interpret the dissimilarities of objects as distances (not necessarily Euclidean) between pairs of points and we aim to compute a configuration of points in a Euclidean space with the same distance properties. MDS goes back to [131] by Young and Householder in 1938 and to [115] by Torgerson in 1952. For dimensionality reduction, MDS can be used in the sense that the low-dimensional representation of the high-dimensional data is computed trying to preserve the pairwise distances of the data points. Even though the approach of MDS differs quite a lot from the one of PCA, we will see that both methods are closely related and yield the same reduction map. In the following we will sketch a MDS approach to dimensionality reduction based on minimization. Additional information on MDS can be found in [125] and [29].
Let d_{ij} for $i, j \in \{1, \dots, n\}$ with $d_{ij} \geq 0$, $d_{ij} = d_{ji}$ and $d_{ii} = 0$ be the pairwise dissimilarities and let $\mathcal{D} = (d_{ij}^2)_{i,j=1,\dots,n} \in \mathbb{R}^{n\times n}$ be the *dissimilarity* (or squared distance) *matrix*. Similarly, for n data points $y_1, \dots, y_n \in \mathbb{R}^d$ we define their squared distance matrix as

$$\mathcal{D}^Y = \left((d_{ij}^Y)^2\right)_{i,j=1,\dots,n} = (\|y_i - y_j\|_2^2)_{i,j=1,\dots,n}. \tag{2.5}$$

In this definition, we used the Euclidean metric, but in general any metric can be used for MDS. A rather complete list of different MDS techniques can be found in Cox and Cox [29].
Clearly, for a given dissimilarity matrix $\mathcal{D}$ we can find n points with $\|y_i - y_j\|_2^2 \approx d_{ij}^2$ by minimizing

$$\sum_{i,j=1}^{n} \left| d_{ij}^2 - \|y_i - y_j\|_2^2 \right| \tag{2.6}$$

with respect to Y. The points Y obtained by this minimization are a configuration representing the dissimilarities given by $\mathcal{D}$. Note that here the dimension d of the Euclidean space is not yet chosen. In particular, if d is chosen large enough and if the dissimilarities d_{ij} are obtained from a Euclidean distance measure, the minimal value of (2.6) can be reduced to zero. In this form, MDS can be seen as a dimensionality reduction method.
To use this technique for dimensionality reduction, we compute the squared distance matrix $\mathcal{D}^X$ of the original high-dimensional data set X and use it as input for MDS. The low-dimensional data set Y is then computed as the minimizer of

$$\min_{Y} \sum_{i,j=1}^{n} \left| (d_{ij}^X)^2 - \|y_i - y_j\|_2^2 \right|. \tag{2.7}$$

This is a special application of metric MDS, where the dissimilarity matrix is given by a configuration in the high-dimensional space.
In the following we will characterize a minimizer of (2.7) under the constraint that $Y = PX$ for an orthogonal matrix $P \in \mathbb{R}^{d \times D}$. Here, the orthogonality of P is a modeling assumption. Furthermore, we will rewrite the cost functional as the trace of a matrix-valued function.
But before doing so, let us discuss the relation between $\mathcal{D}^X$ and the Gramian matrix $X^T X$. We observe that $(d_{ij}^X)^2 = \langle x_i, x_i \rangle - 2\langle x_i, x_j \rangle + \langle x_j, x_j \rangle$ and thus,

$$\mathcal{D}^X = A^X \mathbf{1}_{1\times n} - 2X^T X + \mathbf{1}_{n\times 1}(A^X)^T, \tag{2.8}$$

with $A^X = (\langle x_1, x_1 \rangle, \ldots, \langle x_n, x_n \rangle)^T$. Here, $\mathbf{1}_{m\times n}$ denotes the $m \times n$ matrix whose entries are 1. Moreover, let us introduce the *centering matrix* $H = \mathbf{Id}_n - \frac{1}{n}\mathbf{1}_n$ before we actually compute the minimizer of (2.7). Note that we can center a data set X by right multiplication with H since $XH = X - \frac{1}{n}\left(\sum_i x_i, \ldots, \sum_i x_i\right)$ and thus, $\sum_i (XH)_i = \sum_i x_i - \frac{1}{n} n \sum_i x_i = 0$. Additionally, we have $H^2 = H$ and thus, $XH = X$ implies that X is centered.
Since MDS is defined to be a linear projection method, we can restrict Y to be the image of an orthogonal projection and thus, we will minimize (2.7) subject to orthogonal $P \in \mathbb{R}^{d\times D}$, with $Y = PX$ instead of Y. Now we can prove the following theorem that shows the close relation between MDS and PCA.

Theorem 2.11 (Multidimensional scaling). *Let $X \in \mathbb{R}^{D\times n}$ be a high-dimensional data set and let $X^c = V\Sigma W^T$ be the singular value decomposition of the centered data*

set $X^c = XH$. Then, $P = {V_d}^T = \mathbf{Id}_{d\times D}V^T$ is a minimizer of

$$\min_{\substack{P\in\mathbb{R}^{d\times D},\\ PP^T=\mathbf{Id}_d}} \sum_{i,j=1}^{n} \left|(d_{ij}^X)^2 - (d_{ij}^{PX})^2\right|. \tag{2.9}$$

Furthermore, the approximation error is

$$\mathrm{err}_{MDS}(d, X) = \sum_{i,j=1}^{n} (d_{ij}^X)^2 - (d_{ij}^{{V_d}^T X})^2 = \sum_{i=d+1}^{D} \sigma_i^2,$$

where σ_i are the singular values of X in descending order.

Proof. We will rewrite the minimization problem in several steps so that the solution can be computed as the best d-rank approximation of XH, similarly to PCA.
For an arbitrary orthogonal matrix $P \in \mathbb{R}^{d\times D}$, i.e., $PP^T = \mathbf{Id}_d$, we compute

$$\begin{aligned}
\left|(d_{ij}^X)^2 - (d_{ij}^{PX})^2\right| &= \left|\|x_i - x_j\|_2^2 - \|P(x_i - x_j)\|_2^2\right| \\
&= \left|(x_i - x_j)^T(x_i - x_j) - (x_i - x_j)^T P^T P(x_i - x_j)\right| \\
&= \left|(x_i - x_j)^T(\mathbf{Id}_D - P^T P)(x_i - x_j)\right| \\
&= \left|(x_i - x_j)^T(\mathbf{Id}_D - P^T P)^T(\mathbf{Id}_D - P^T P)(x_i - x_j)\right| \\
&= \|(\mathbf{Id}_D - P^T P)(x_i - x_j)\|_2^2,
\end{aligned} \tag{2.10}$$

where we have used that $\mathbf{Id}_D - P^T P$ is an orthogonal projector.
Next, we observe that for $H = \mathbf{Id}_n - \frac{1}{n}\mathbf{1}_n$ and $Z \in \mathbb{R}^{D\times n}$ it holds

$$\begin{aligned}
-n\,\mathrm{tr}(H\mathcal{D}^Z H) &= -n\,\mathrm{tr}\left(\mathcal{D}^Z - \frac{1}{n}(\mathcal{D}^Z\mathbf{1}_n + \mathbf{1}_n\mathcal{D}^Z) + \frac{1}{n^2}\mathbf{1}_n\mathcal{D}^Z\mathbf{1}_n\right) \\
&= -n\sum_{i=1}^{n}\left(d_{ii}^Z - \frac{2}{n}\sum_{k=1}^{n} d_{ki}^Z + \frac{1}{n^2}\sum_{k,l=1}^{n} d_{kl}^Z\right) \\
&= \sum_{i,k=1}^{n} d_{ki}^Z = \|\mathcal{D}^Z\|_F^2
\end{aligned} \tag{2.11}$$

and with $H\mathbf{1}_{n\times 1} = \mathbf{1}_{1\times n}H = 0$ we have

$$\begin{aligned}
H\mathcal{D}^Z H &\overset{(2.8)}{=} H\left(A^Z\mathbf{1}_{1\times n} - 2Z^T Z + \mathbf{1}_{n\times 1}(A^Z)^T\right)H \\
&= -2HZ^T ZH.
\end{aligned} \tag{2.12}$$

Now we rewrite the cost functional in (2.9) as the Frobenius norm of the squared distance

matrix of the matrix $(\mathbf{Id}_D - P^T P)X$ and compute

$$\begin{aligned}\sum_{i,j=1}^{n} \left|(d_{ij}^X)^2 - (d_{ij}^{PX})^2\right| &\overset{(2.10)}{=} \sum_{i,j=1}^{n} \|(\mathbf{Id}_D - P^T P)(x_i - x_j)\|_2^2 \\ &\overset{(2.5)}{=} \|\mathcal{D}^{(\mathbf{Id}_D - P^T P)X}\|_F^2 \\ &\overset{(2.11)}{=} -n \operatorname{tr}(H\mathcal{D}^{(\mathbf{Id}_D - P^T P)X} H) \\ &\overset{(2.12)}{=} 2n \operatorname{tr}\left(\left((\mathbf{Id}_D - P^T P)XH\right)^T (\mathbf{Id}_D - P^T P)XH\right) \\ &= 2n\|(\mathbf{Id}_D - P^T P)XH\|_F^2 \\ &= 2n\|XH - P^T PXH\|_F^2. \end{aligned} \tag{2.13}$$

It is well-known, that the best d-rank approximation X_* of XH under the Frobenius norm is given by $X_* = \sum_{i=1}^{d} \sigma_i v_i u_i^T = V_d \Sigma_d {W_d}^T$ (see e.g. Theorem 2.4) and thus, for the minimizer P_* of (2.13) it holds $X_* = P_*^T P_* XH = V_d \Sigma_d {W_d}^T$. An easy calculation shows, that for $P_* = V_d^T$ this equality indeed holds. Thus, the minimizer of (2.9) is given by $P = {V_d}^T$. Furthermore, the error can be computed as in PCA case. □

Remark 2.12. The matrix $P^T P$ in the cost functional

$$\sum_{i,j=1}^{n} \|(\mathbf{Id}_D - P^T P)(x_i - x_j)\|_2^2 \tag{2.14}$$

is the orthogonal projection on the d-dimensional subspace $P^T P\mathbb{R}^D$ of $\mathbb{R}^D$. This subspace is spanned by the rows of P. In this sense we have computed an optimal d-dimensional subspace of $\mathbb{R}^D$ which is the same as for PCA. The value (2.14) is the sum of the pairwise squared distances of the orthogonal projection of X onto the orthogonal complement of $P^T P\mathbb{R}^D$.

Corollary 2.13. *The minimization problem in* (2.9) *can be equivalently formulated as*

$$\min_{\substack{P \in \mathbb{R}^{d \times D} \\ PP^T = \mathbf{Id}_d}} \operatorname{tr}(HX^T XH - HX^T P^T PXH).$$

Proof. This is a direct consequence of Theorem 2.11, since

$$\begin{aligned} 2n\|XH - P^T PXH\|_F^2 &= 2n \operatorname{tr}\left((XH - P^T PXH)^T (XH - P^T PXH)\right) \\ &= 2n \operatorname{tr}(HX^T XH - HX^T P^T PXH). \end{aligned}$$

□

Remark 2.14. In particular, Theorem 2.11 shows that MDS and PCA lead to the same reduction method for centered data, even though the heuristics are quite different. Comparing the cost functionals in trace-form, this is not surprising since for centered data

we have $XH = X$ and thus, due to the cyclic invariance of the trace it holds

$$\begin{aligned}\operatorname*{arg\,min}_{\substack{P\in\mathbb{R}^{d\times D}\\ PP^T=\mathbf{Id}_d}} \operatorname{tr}(HX^TXH - HX^TP^TPXH) &= \operatorname*{arg\,min}_{\substack{P\in\mathbb{R}^{d\times D}\\ PP^T=\mathbf{Id}_d}} \operatorname{tr}(X^TX - X^TP^TPX)\\ &= \operatorname*{arg\,min}_{\substack{P\in\mathbb{R}^{d\times D}\\ PP^T=\mathbf{Id}_d}} -\operatorname{tr}(X^TP^TPX)\\ &= \operatorname*{arg\,min}_{\substack{P\in\mathbb{R}^{d\times D}\\ PP^T=\mathbf{Id}_d}} -\operatorname{tr}(PXX^TP^T).\end{aligned}$$

In summary, the linear dimensionality reduction method MDS is given as

$$\operatorname*{arg\,min}_{P\in\mathcal{U}_{MDS}} g_{MDS}(P),$$

with cost functional $g_{MDS}(P) = \operatorname{tr}(HX^TXH - (PXH)^TPXH)$ and admissible set $\mathcal{U}_{MDS} = \{P \in \mathbb{R}^{d\times D} \colon PP^T = \mathbf{Id}_d\}$. By Theorem 2.11, the solution is given by the matrix containing row-wise the eigenvectors to the d largest eigenvalues of XHX^T.

Remark 2.15 (Uniqueness). In analogy to PCA, the solution of the MDS problem is not unique. Moreover, all other properties of PCA are also true for MDS.

2.2.3 Isomap

So far we have introduced dimensionality reduction methods, which are based on the assumption that the data set is concentrated around a linear subspace of the high-dimensional space $\mathbb{R}^D$. Such methods are called *linear* and they will most likely fail, if the data is concentrated around a non-linear (Riemannian) manifold $\mathcal{M}$. One reason for the failure is that the Euclidean metric is not suitable for measuring the distances of two points on a manifold (see Figure 2.4). To overcome this problem, we can use the

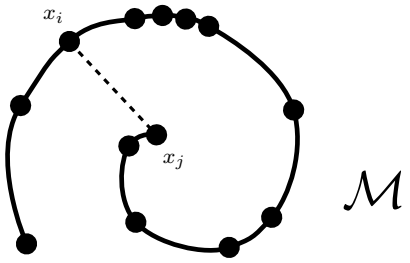


Figure 2.4: A configuration of points on a non-linear manifold. The Euclidean distance of x_i and x_j is much smaller than their distance on the manifold. If MDS is applied, the points x_i and x_j will be mapped closer together as they should and the geometrical structure of $\mathcal{M}$ is lost.

metric induced by the Riemannian metric (called the *geodesic metric*) on the underlying manifold instead of the Euclidean distance. As the manifold $\mathcal{M}$ is unknown, the computation of the geodesic distance is not possible but we can use the neighborhood structure of the data in order to approximate it. This neighborhood structure (see Step 1 on page 56) of the data induces a graph $\Gamma = (X, E)$ on the data set, whose vertices $x_i \in X$ are the data points and whose edges $e_{ij} = (x_i, x_j) \in E$ connect points which are in the same neighborhood, i.e., $e_{ij} \in E$ if x_j is a neighbor of x_i or x_i of x_j.
Now, we define the *graph distance* d_Γ of two points $x_i, x_j \in X$ as follows: For a path $\gamma = (x_0, x_1, \dots, x_m)$ connecting $x_i = x_0$ and $x_j = x_m$ we define its length as $d_\gamma(x_i, x_j) = \sum_{i=0}^{m-1} \|x_i - x_{i+1}\|_2$. Then, the graph distance of two points is defined as $d_\Gamma(x_i, x_j) = \min_{\gamma \in \Phi} d_\gamma(x_i, x_j)$, where Φ is the set of all paths connecting x_i and x_j.

Remark 2.16. If the data points are dense enough on $\mathcal{M}$, the graph distance approximates the geodesic distance well (for a proof see Section 8.5 in [125]).

In analogy to the previous section we can define a dissimilarity matrix from the graph distance

$$\mathcal{D}_\Gamma^X = \left((d_\Gamma^X)_{ij}^2\right)_{i,j=1,\dots,n} = \left(d_\Gamma(x_i, x_j)^2\right)_{i,j=1,\dots,n}.$$

The idea is now to use this dissimilarity matrix as an input for MDS. This proceeding is called Isomap and was introduced by Tenenbaum and co-workers in 2000 (see [114]). The name Isomap refers to *isometric mapping* since the dimensionality reduction is realized by an isometric mapping $P\colon \mathcal{M} \to \mathbb{R}^d$. Here, a mapping is called isometric if it preserves the pairwise distances of the data set.
Due to its construction Isomap strongly relies on MDS and is sometimes called the non-linear version of MDS.
The Isomap problem is given by

$$\min_Y \sum_{i,j=1}^{n} \left| (d_\Gamma^X)_{ij}^2 - (d_{ij}^Y)^2 \right|, \tag{2.15}$$

compare (2.7). Remark that we do not assume that $Y = PX$ for a linear P, so that Isomap is a non-linear dimensionality reduction method.
In the following we will specify the constraint set of the minimization problem (2.15). Note that $\mathcal{D}_\Gamma^X$ is not the squared Euclidean distance matrix and thus, Theorem 2.11 is not directly applicable and we need some further considerations.
First, we observe that for N sufficiently large (at most n) there exists a configuration of points $Z \in \mathbb{R}^{N \times n}$ with $\mathcal{D}^Z = \mathcal{D}_\Gamma^X$ as long as the so called *Isomap kernel* $-\frac{1}{2} H \mathcal{D}_\Gamma^X H$ is positive-semidefinite. This can be seen as follows. A necessary condition for $\mathcal{D}_\Gamma^X$ to be a squared Euclidean distance matrix of a configuration in $\mathbb{R}^N$ is by equation (2.12)

$$-\frac{1}{2} H \mathcal{D}_\Gamma^X H = (ZH)^T ZH.$$

From this it can be directly seen that the left-hand side needs to be positive-semidefinite. On the other hand, if it is positive-semidefinite, an eigendecomposition yields a desired

but centered configuration (compare [125]), i.e., $ZH = Z$. Moreover, we observe that the dimension N can be chosen as the rank of the Isomap kernel $-\frac{1}{2}H\mathcal{D}_\Gamma^X H$.
In the following we assume that $-\frac{1}{2}H\mathcal{D}_\Gamma^X H$ is indeed positive-semidefinite (if it is not compare Remark 2.18) and that $Z \in \mathbb{R}^{N\times n}$ is a corresponding configuration. Now, we apply the metric MDS from Section 2.2.2 to this data set Z in order to compute a low-dimensional representation $Y = PZ$ of X. The Isomap problem then reads

$$\min_{\substack{P\in\mathbb{R}^{d\times N}\\ PP^T=\mathbf{Id}_d}} \sum_{i,j=1}^{n} \left|(d_\Gamma^X)_{ij}^2 - (d_{ij}^{PZ})^2\right| \tag{2.16}$$

and with Theorem 2.11 a solution is given by the singular value decomposition of $ZH = V\Sigma W^T$ as $P = {V_d}^T$. This yields the low-dimensional representation $Y = {V_d}^T Z$.
From the practical point of view it is not necessary to compute Z because Y can be directly computed from the Isomap kernel. Consider the singular value decomposition of $ZH = V\Sigma W^T$ which yields the eigendecomposition $-\frac{1}{2}H\mathcal{D}_\Gamma^X H = W\Sigma^T\Sigma W^T = W\Lambda^2 W^T$. Then, it follows

$$Y = {V_d}^T Z = {V_d}^T ZH = {V_d}^T V\Sigma W^T = \mathbf{Id}_{d\times N}\Sigma W^T = \mathbf{Id}_{d\times n}\Lambda W^T.$$

Thus, Y can be directly obtained from the eigendecomposition of $-\frac{1}{2}H\mathcal{D}_\Gamma^X H$.

Remark 2.17. The low-dimensional representation of X is a centered data set even though X was not centered.

Remark 2.18. If the points are not distributed densely enough on $\mathcal{M}$, the Isomap kernel might not be positive-semidefinite and thus, Y cannot be computed as described above. To overcome this problem we can use the *constant shift technique* as in [125] which consists in adding a positive integer $\delta > 0$ to the graph distance $d_\Gamma(x_i, x_j)$ for $i \neq j$. It is possible to choose δ in a way that the resulting Isomap kernel is positive-semidefinite (see Chapter 8 in [125] for details).

Remark 2.19. Of course, other metrics beside the graph distance can be used to construct the dissimilarity matrix of X approximating the geodesic metric. This will lead to other methods.

In summary, Isomap is based on the same cost functional as MDS. Only the starting distances are computed differently. Thus, we have $g_{Isomap}(P) = \sum_{i,j=1}^n \left|(d_\Gamma^X)_{ij}^2 - (d_{ij}^{PZ})^2\right|$ and the admissible set $\mathcal{U}_{Isomap} = \{P \in \mathbb{R}^{d\times N} \colon PP^T = \mathbf{Id}_d\}$. In contrast to MDS and PCA, when formulating Isomap as an optimization problem, we need to make an intermediate step where we compute the configuration Z.
To end this section we describe the Isomap algorithm.

Algorithm

As mentioned above, to solve the Isomap problem the graph metric d_Γ on the data set X is computed to approximate the geodesic metric of the underlying manifold before a linear dimensionality reduction method (here MDS or PCA) is applied in order to find

a low-dimensional representation preserving the graph metric. Since this computation involves more steps as solving the PCA or MDS problem, where a simple singular value decomposition is sufficient to compute the minimizer, we will outline the Isomap algorithm in the following steps (compare [125]).

Step 1. Definition of neighboring points. To fix the neighborhood structure of the data set X, we need to determine the neighboring points of a data point x_i. To do so we can use either its k-nearest neighbors or all points in an ϵ-neighborhood. Let us denote the set of neighboring points of x_i by $N(i)$. Note that for k-nearest neighbors in general $x_i \in N(j)$ does not imply $x_j \in N(i)$.

Step 2. Computation of graph distance. As already explained, the neighborhood structure of X induces a graph $\Gamma = (X, E)$ on X called the *adjacency graph* of X. The graph metric on X is computed as the pairwise graph distance $d_\Gamma(x_i, x_j)$ for each pair of points (x_i, x_j). If there are unconnected points, we set their distance to infinity. Define the dissimilarity matrix $\mathcal{D}_\Gamma^X$ by the graph distances.

Step 3. Construction of the Isomap kernel. Compute the Isomap kernel $G^c = -\frac{1}{2} H \mathcal{D}_\Gamma^X H$. If it is not positive-semidefinite, change it according to Remark 2.18.

Step 4. Eigen decomposition of the kernel. Since G^c is positive-semidefinite it has an eigendecomposition $W \Lambda^2 W^T$, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d, \ldots, \lambda_n)$ with $\lambda_i \geq 0$ ordered by size. The low-dimensional data set Y is then given by $Y = \mathbf{Id}_{d \times n} \Lambda W^T$.

2.2.4 Other non-linear methods

Of course Isomap is just one of many non-linear dimensionality reduction methods that have been developed in the last decades. Due to the high demand for sophisticated reduction methods the creation and combination of techniques is a very active field of research. Many of these methods follow the same pattern as Isomap: the neighborhood structure of X is defined and used to construct a certain graph. Then, a kernel is defined and decomposed.

Surely not all dimensionality reduction methods can be formulated as an optimization problem of the above form in (2.1). But there are quite a few. To discuss all of them would go beyond the scope of this work, but we would like to mention briefly two other very popular methods.

Locally Linear Embedding - LLE

The first one is *locally linear embedding (LLE)*. LLE was introduced in 2000 by Roweis and Saul [104]. The basic idea in LLE is to compute the embedding of the data in the low-dimensional space by preserving the locally Euclidean structure of the neighborhood of the data points. The intuition behind this is as follows. If the data points are distributed densely enough on $\mathcal{M}$, each data point x_i and its neighbors are in (or close to) a locally linear patch of the manifold. Thus, we can compute weights w_{ij} to reconstruct x_i from its neighbors by minimizing $\|x_i - \sum_j w_{ij} x_j\|_2^2$. If x_j is not a neighbor of x_i, the weight w_{ij} is set to zero. Furthermore, we require $\sum_j w_{ij} = 1$. For the complete data set, the weights can be computed by summing up the reconstruction error of all data points

which leads to

$$\min_{\substack{W=(w_{ij})_{i,j=1,\dots,n} \\ W\mathbf{1}_{n\times 1}=\mathbf{1}_{n\times 1} \\ w_{ij}=0 \text{ for } x_j\notin N(i)}} \sum_{i=1}^{n} \Big\| x_i - \sum_{j=1}^{n} w_{ij} x_j \Big\|_2^2. \tag{2.17}$$

This is a simple least square problem which is solved in closed form by

$$w_{ij} = \frac{\sum_{x_k\in N(i)} \langle x_i - x_j, x_i - x_k\rangle^{-1}}{\sum_{x_l,x_m\in N(i)} \langle x_i - x_l, x_i - x_m\rangle^{-1}}, \qquad \text{for } x_j \in N(i)$$

(compare [106]). The constraint $W\mathbf{1}_{n\times 1} = \mathbf{1}_{n\times 1}$ leads to a translation invariant cost functional in the sense that a translation of X leads to the same weights W. Due to the structure of the cost functional it is furthermore invariant under rotations and scalings of X.
A low-dimensional representation is now computed by mapping each patch (point x_i and its neighbors) linearly to the low-dimensional space. On each patch this map consists of rotation, translation and scaling and thus, the weights W are kept fixed.
Due to this construction, we expect that the low-dimensional representation y_i of x_i can be reconstructed from its neighbors y_j by the same weights as in the original space. Thus, we can obtain Y by minimizing the same cost functional as in (2.17) but subject to Y

$$\min_{\substack{Y\in\mathbb{R}^{d\times n} \\ YY^T=\mathbf{Id}_d}} \sum_{i=1}^{n} \Big\| y_i - \sum_{j=1}^{n} w_{ij} y_j \Big\|_2^2.$$

Constraining Y to have unit covariance, i.e., $YY^T = \mathbf{Id}_d$, forces the rank of Y to be not smaller than d and especially excludes the trivial solution $Y = 0$. Moreover, this optimization problem has a unique solution if Y is assumed to be centered, compare [106].
Let $M = (\mathbf{Id}_n - W)^T(\mathbf{Id}_n - W)$, then the cost functional can be reformulated as a trace

$$\sum_{i=1}^{n} \Big\| y_i - \sum_{j=1}^{n} w_{ij} y_j \Big\|_2^2 = \operatorname{tr}(YMY^T) \tag{2.18}$$

and thus, it can be solved by solving an eigenvalue problem. The centered solution Y is given by the eigenvectors to the d smallest non-vanishing eigenvalues (see [106]).
All in all, we do not write the LLE problem directly in the form of (2.1) but in a similar one

$$\min_{Y\in\mathcal{U}_{LLE}} g_{LLE}(Y), \tag{2.19}$$

where $\mathcal{U}_{LLE} = \{Y \in \mathbb{R}^{d\times n} \colon YY^T = \mathbf{Id}_d\}$ is not a set of reduction maps and where $g_{LLE}(Y) = \operatorname{tr}(YMY^T)$ is the corresponding cost functional.

Laplacian Eigenmaps - LE

The second method we would like to mention here is *Laplacian eigenmaps (LE)* developed by Belkin and Niyogi in 2001 in [7]. LE shows some parallels to LLE but the involved

kernel W is defined differently. LE is derived from the Laplace-Beltrami operator on the manifold, which is approximated by the Laplace operator on the neighborhood graph of the high-dimensional data set. This operator is represented by the graph's weighted Laplacian matrix.
The motivation behind LE is to preserve the neighborhood structure of the data set. First, a neighborhood graph is constructed as in LLE and then a weight matrix W is chosen in a way that nearby points x_i and x_j have a larger weight w_{ij} than faraway points. Next, from this weight matrix we define a cost functional which takes a larger value if nearby points x_i and x_j are mapped on faraway points y_i and y_j. A low-dimensional representation can be obtained by minimizing

$$\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\|y_i - y_j\|_2^2. \tag{2.20}$$

The cost function in (2.20) can be reformulated in terms of a trace and the graph Laplacian which is defined as

$$L = D - W = \operatorname{diag}(\sum_i w_{1i}, \dots, \sum_i w_{ni}) - W.$$

A short calculation leads to

$$\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\|y_i - y_j\|_2^2 = \operatorname{tr}(YLY^T).$$

For details on this we refer to [8].
Thus, the LE problem is given by

$$\min_{\substack{Y\in\mathbb{R}^{d\times n}\\ YDY^T=\mathbf{Id}_d}} \operatorname{tr}(YLY^T), \tag{2.21}$$

where the constraint ensures that $\operatorname{rk}(Y) = d$, in particular $Y = 0$ is not admissible. Uniqueness can be obtained by requiring the solution to be centered since it is already normalized by the condition $YDY^T = \mathbf{Id}_d$. The values $d_{ii} = \sum_j w_{ij}$ corresponding to the ith vertex of the neighborhood graph can be interpreted ass a measure for the importance of this vertex. The larger the value the more important the vertex since the adjacent edges have larger weights and contribute more to the cost functional's value.
A minimizer of (2.21) is provided by the solution of the generalized eigenvalue problem

$$Lf = \lambda Df.$$

The minimizer is explicitly given by $Y = (f_1, \dots, f_d)^T$, where f_i are the generalized eigenvectors to the d smallest non-zero generalized eigenvalues. For a rigorous derivation see e.g. [68].
As for LLE, we do not write the LE problem in the form of (2.1) but in the form (2.19)

$$\min_{Y\in\mathcal{U}_{LE}} g_{LE}(Y).$$

Here, the admissible set is given by $\mathcal{U}_{LE} = \{Y \in \mathbb{R}^{d\times n} \colon YDY^T = \mathbf{Id}_d\}$ and the cost functional by $g_{LE}(Y) = \mathrm{tr}(YLY^T)$.
Last but not least, let us briefly mention the choice of the weights w_{ij}. Due to the analogy to the Laplace-Beltrami operator and its relation to the heat equation they are usually chosen to be a modified heat kernel with $t \in \mathbb{R}$ and $\epsilon > 0$

$$w_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|_2^2}{4t}} & \text{if } \|x_i - x_j\|_2 < \epsilon \\ 0 & \text{otherwise,} \end{cases}$$

as proposed in [7]. Here, other choices for the weight matrix W are possible and the similarity to LLE becomes apparent if we choose the weights accordingly.
The weights of LLE have the constraint $\sum_j w_{ij} = 1$ and thus, the corresponding matrix D_{LLE} is the identity. This yields $M = L^2$, where M is the matrix from equation (2.18). Since the eigenvectors of L and L^2 are the same, for this special choice of weights LLE and LE lead to the same reduction map.

2.3 Non-negative dimensionality reduction as an optimization problem

In the first part of this chapter we have introduced the general concept of dimensionality reduction, which is based on the crucial assumption that the intrinsic dimension of the data is much lower than the dimension of the space wherein the data is embedded. We have discussed different dimensionality reduction methods and we have seen that all of them are constructed in a way which seeks to preserve some geometrical properties of the data (as e.g. distances, neighborhoods, scattering in terms of variance).
However, there are other properties of the data which are worth to be preserved beyond the dimensionality reduction step. One of these properties is the *non-negativity* of the data. In this context, non-negativity refers to the data matrix being entry-wise non-negative, i.e., $X = (x_{ij})_{i=1,\dots,d,j=1,\dots,n} \geq 0$ if $x_{ij} \geq 0$ for all i, j. For example, when dimensionality reduction is used as a preprocessing step, further computations might require non-negative input data. This is for instance the case in signal separation (compare Chapter 3). Here, time-frequency data is obtained by a signal transform and stored in a data matrix, the so called spectrogram of the source signal. The spectrogram is non-negative by construction. To decompose the low-dimensional data set with methods like independent component analysis or non-negative matrix factorization we want the reduced data to be non-negative as well. Other applications for non-negative dimensionality reduction can be found in [52, 90, 92].

2.3.1 Motivating example

The following illustrative example shows that even for an elementary linear dimensionality reduction method like PCA we cannot expect the low-dimensional data to be non-negative although the high-dimensional data was so.

Example 2.20. Consider the non-negative data set

$$X = \begin{pmatrix} \frac{1}{2} & \frac{3}{2} & 1 & 2 \\ \frac{3}{2} & \frac{1}{2} & 1 & 2 \\ 1 & 1 & 1 & 2 \end{pmatrix} \in \mathbb{R}^{3\times 4}$$

and compute a 2-dimensional representation Y using PCA. The covariance matrix XX^T has the eigendecomposition

$$XX^T = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix} \begin{pmatrix} 21 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix}$$

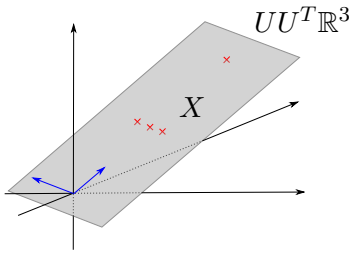
and thus, by Theorem 2.3 a minimizer of the PCA problem is given as

$$U^T = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}.$$

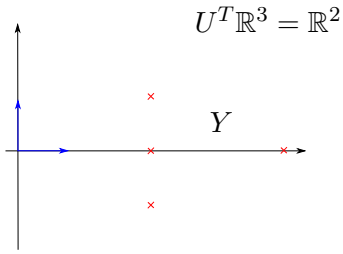
This yields

$$Y = U^T X = \begin{pmatrix} \sqrt{3} & \sqrt{3} & \sqrt{3} & 2\sqrt{3} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \end{pmatrix},$$

which clearly has a negative entry.
The reason for this phenomenon is the principal axes transformation that is performed when PCA is applied to the data set (compare Figure 2.5).



(a) The data set X (red crosses) lies in the plane $UU^T\mathbb{R}^3$ (gray). The first two principal axes (main scattering directions) are depicted in blue.



(b) 2-dimensional representation Y (red crosses) and principal axes (blue).

Figure 2.5: Principal component analysis of non-negative data set $X \in \mathbb{R}^{3\times 4}$ which yields the low-dimensional representation Y with a negative entry.

This example shows that in general the application of dimensionality reduction methods might cause negative entries in the low-dimensional representation. Therefore, non-negative dimensionality reduction methods are essentially required to guarantee non-negative output data from non-negative input data. Accordingly, the construction of

non-negative dimensionality reduction methods is of particular interest. One possible approach to develop such methods is to modify already well-understood dimensionality reduction techniques in a way that they preserve non-negativity. Seeking a universal procedure to extend existent dimensionality reduction methods to non-negativity preserving ones is one of the main objectives of this work.
The main advantage of this approach is that we can benefit from the knowledge of dimensionality reduction without non-negativity constraint. For many methods of this type not only the formulation as an optimization problem is known but also a solution. Nevertheless, we have to be aware that this ansatz entails some restrictions concerning the dimensionality reduction methods that can be forced to preserve the non-negativity. In the following, we develop and discuss this framework.

2.3.2 Splitting approach

As motivated above, for the further processing of the reduced data set we are interested in preserving the non-negativity so that the low-dimensional representation of the data is likewise non-negative. From Example 2.20 we have learned, that there is absolutely no reason why the reduced data set Y should be non-negative if we apply any dimensionality reduction method. Thus, in order to preserve this property, we need to enforce the dimensionality reduction method to do so. This is where our formulation of the dimensionality reduction problem as an optimization of the form (2.1) pays off because we simply need to include an additional constraint.

Definition 2.21. The problem

$$\min_{\substack{P\in\mathcal{U}\\ P(X)\geq 0}} g(P) \tag{2.22}$$

is called *non-negative dimensionality reduction problem*. A solution of this problem is called *non-negative dimensionality reduction method*.

By requiring $P(X) \geq 0$ we guarantee that the low-dimensional representation is indeed non-negative. This is a completely different ansatz as introduced in [133] and as used by many others [31, 51, 52, 92] where a non-negative PCA is developed under the assumption that $U \geq 0$ instead of $U^T X \geq 0$. Compared to that one, our approach is less restrictive. In particular, we will see that we obtain the same error as for the usual PCA (see Theorem 2.41) which is not true in general for the methods using $U \geq 0$.
A local solution of (2.22) can be found using standard optimization methods for constraint optimization. But searching for a global minimizer makes this typically non-convex problem much more complex and difficult to solve since, in general, descent methods do not result in a global minimum. We should keep in mind that this complexity basically results from the additional constraint $P(X) \geq 0$ as without this constraint the problem reduces to the usual dimensionality reduction problem, for which we assume that a minimizer is known. This assumption is reasonable since we have seen in Section 2.2 that for many dimensionality reduction methods a minimizer can be computed analytically. Precisely this observation motivates our approach to non-negative dimensionality reduction. The idea is to treat both constraints separately in two steps

by splitting the problem into an ordinary dimensionality reduction problem and a second step where we take care of the non-negativity of the data.
In the following, we will refer to this ansatz by calling it *splitting approach*. More precisely, this approach can be summarized like this:

Approach 2.22 (Splitting approach)**.** For a dimensionality reduction method determined by $(\mathcal{U}, g)$ we

(i) solve the dimensionality reduction problem $\min_{P \in \mathcal{U}} g(P)$ and

(ii) force the low-dimensional representation to be non-negative by applying a post-processing without changing the value of the cost functional.

Remark 2.23. The success of this approach strongly depends on the dimensionality reduction problem itself, i.e., on the pair $(\mathcal{U}, g)$. The possibility to uncouple both constraints is a very powerful tool and it allows for reducing the computational costs drastically. However, it is not applicable to all problems of the form (2.22).

We want to dedicate the remaining sections of this chapter to the classification of (non-negative) dimensionality reduction methods that can be treated by this approach.

Splitting approach: translation

For the second part of Approach 2.22 several approaches are conceivable. The most simple one would be a translation of the data since by adding a constant c to all entries of the data matrix we would achieve that $Y + c\mathbf{1}_{d\times n} \geq 0$ for $c > 0$ large enough.
This idea is motivated by the observation that some cost functionals from Section 2.2 are indeed translationally invariant.

Definition 2.24. We say that a dimensionality reduction problem has a *translationally invariant* cost functional $g\colon \mathcal{U} \to \mathbb{R}$ if for all $P \in \mathcal{U}$ and all constant vectors $c \in \mathbb{R}^d$

$$P + c \in \mathcal{U} \text{ and } g(P + c) = g(P).$$

In this case, we call the reduction method *translationally invariant*.

All cost functionals of the dimensionality reduction methods from Section 2.2, except for PCA, are based on the pair-wise distances $\|y_i - y_j\|_2$ of the low-dimensional data points. Due to the translational invariance of this distance measure these cost functions are translationally invariant by construction.
Unfortunately, this ansatz is not suitable for the application we have in mind even though it might be useful in other situations. Our approach to signal separation requires the separation of the low-dimensional data set Y. More precisely, we want to efficiently decompose a signal (represented by a high-dimensional data set X) by reducing its dimension, decomposing it with standard methods and mapping the thereby obtained components Y_1 and Y_2 back to the high-dimensional data space. By translating the low-dimensional data Y by $c\mathbf{1}_{d\times n}$, we need to be aware of the fact that we only get a decomposition of $Z = Y + c\mathbf{1}_{d\times n}$ into $Z_1 + Z_2$, which does not yield a decomposition of Y into Y_1 and Y_2. This proceeding is shown in Figure 2.6.
These considerations require the search for another more sophisticated approach.

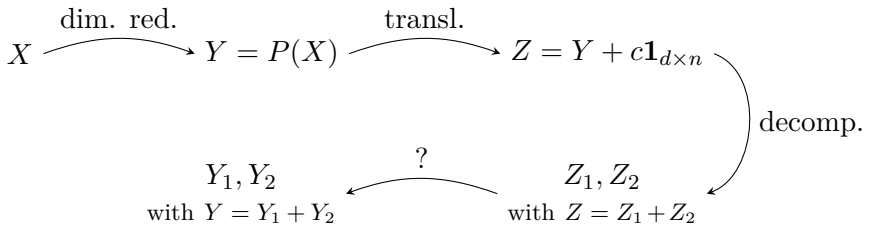


Figure 2.6: The general proceeding in signal separation with translation in order to obtain a non-negative low-dimensional data set. Here, it is not clear how to obtain Y_1 and Y_2 from Z_1 and Z_2 since Z_1 and Z_2 would need to be back translated somehow.

Splitting approach: rotation

To overcome the above discussed obstacle we reconsider Example 2.20, which motivates another approach. We observe that the points of the low-dimensional data set are scattered in a way that the angle between each pair of vectors is not larger than $\frac{\pi}{2}$. This observation makes us think of rotating the data to the positive quadrant of the coordinate system and motivates the following variant of the *splitting approach* 2.22 to solve the non-negative dimensionality reduction problem (2.22). Recall that any rotation of a data set in $\mathbb{R}^d$ can be described by a matrix $R \in SO(d)$ (compare Example 1.30).

Approach 2.25 (Splitting approach with rotation)**.** For a dimensionality reduction method determined by $(\mathcal{U}, g)$ we

(i) solve the dimensionality reduction problem $\min_{P\in\mathcal{U}} g(P)$ and

(ii) find a rotation matrix $R \in SO(d)$, i.e., $R^T R = \mathbf{Id}_d$, such that $RP(X) \geq 0$ without changing the value of the cost functional.

The drawback described in Figure 2.6 does not occur here since if $Z = RY$ is decomposed in Z_1 and Z_2 with $RY = Z_1 + Z_2$, we obtain the decomposition

$$Y = R^{-1}RY = R^{-1}(Z_1 + Z_2) = R^{-1}Z_1 + R^{-1}Z_2 = Y_1 + Y_2.$$

Of course this approach is not suitable for all data sets X and all dimensionality reduction methods P. It will only lead to a solution of the minimization problem (2.22) if we can guarantee the existence of such a rotation and that its application to the low-dimensional data set does not affect the value of the cost functional. Thus, we now want to characterize dimensionality reduction methods that allow for this ansatz.
In the following, we will formulate a sufficient condition for obtaining a global minimizer of (2.22) by using the splitting approach 2.25. This condition will include two aspects: first, the cost functional's independence of the application of a rotation and second, the existence of a suitable rotation.

Definition 2.26. We say that a dimensionality reduction problem has a *rotationally invariant* cost functional $g\colon \mathcal{U} \to \mathbb{R}$ if for all $P \in \mathcal{U}$ and all $R \in SO(d)$ it holds

$$RP \in \mathcal{U} \text{ and } g(RP) = g(P).$$

Then, we call the dimensionality reduction method *rotationally invariant*.

The dimensionality reduction methods presented in the previous section are all rotationally invariant as we will see in Section 2.4. Nevertheless, the rotation invariance of the cost functional is not sufficient to justify the usage of the splitting approach 2.25 since the question of the existence of a rotation is still not answered. To do so, we introduce the notion of a cone and its opening angle.

Definition 2.27. A set $K \subset \mathbb{R}^d$ is called a *cone with apex at 0* if for all $x \in K$ we have $\lambda x \in K$ for all $\lambda \geq 0$. Furthermore, we define the *opening angle* $\theta \in [0, \pi]$ of a cone with apex at 0 as

$$\theta = \sup \left\{ \arccos\left(\frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} \right) : x, y \in K \setminus \{0\} \right\}.$$

Note that we also refer to such a cone by calling it a *cone of angle* θ.

Remark 2.28. The opening angle of a cone only coincides with the geometrical picture (see Figure 2.7) for $\theta < \pi$ since all other cones have opening angle $\theta = \pi$ according to our definition.

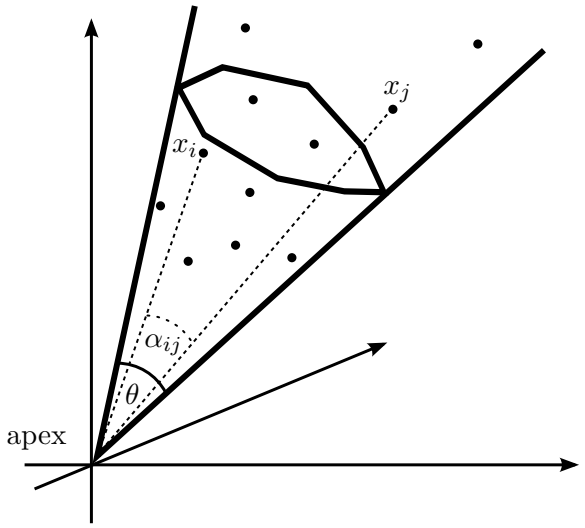


Figure 2.7: Data set lying inside a cone of angle θ. For the angle α_{ij} between x_i and x_j holds that $\alpha_{ij} \leq \theta$.

Let us now use the above definition to characterize the geometry of the discrete point set X.

Lemma 2.29. *A data set $X \neq \{0\}$ is lying inside a cone K of angle θ, i.e., $X \subset K$, if and only if*

$$\frac{\langle x_i, x_j\rangle}{\|x_i\|_2\|x_j\|_2} \geq \cos(\theta) \qquad \text{for all } x_i, x_j \in X \setminus \{0\}.$$

Proof. For $X \subset K$ it follows immediately from Definition 2.27 that $\frac{\langle x_i, x_j\rangle}{\|x_i\|_2\|x_j\|_2} \geq \cos(\theta)$ for $x_i, x_j \in X \setminus \{0\}$.
Conversely, consider $K = \operatorname{conv}\{\lambda x_i \colon \lambda \geq 0, x_i \in X\} = \{\sum_i \alpha_i x_i \colon \alpha_i \geq 0, x_i \in X\}$, the convex hull of all half-lines $\{\lambda x_i \colon \lambda \geq 0\}$. Then, clearly $x_i \in K$ and K is a cone with opening angle θ satisfying

$$\theta \geq \rho = \max\left\{\arccos\left(\frac{\langle x_i, x_j\rangle}{\|x_i\|_2\|x_j\|_2}\right) : x_i, x_j \in X \setminus \{0\}\right\}. \tag{2.23}$$

Actually, we even have $\theta = \rho$. To see this, consider two points $v_1, v_2 \in K$ with $\|v_1\|_2 = \|v_2\|_2 = 1$. The normalization effects no loss of generality since it does not affect the angle between v_1 and v_2. Then, we have $v_1 = \sum_i \alpha_i \frac{x_i}{\|x_i\|_2}$ and $v_2 = \sum_j \beta_j \frac{x_j}{\|x_j\|_2}$ with $\alpha_i, \beta_j \geq 0$ and $x_i, x_j \in X \setminus \{0\}$ and thus, $1 = \|v_1\|_2 \leq \sum_i \alpha_i$ and analogously $1 \leq \sum_j \beta_j$. This yields

$$\langle v_1, v_2\rangle = \sum_{i,j} \alpha_i\beta_j \frac{\langle x_i, x_j\rangle}{\|x_i\|_2\|x_j\|_2} \geq \sum_{i,j} \alpha_i\beta_j \cos(\rho) \geq \cos(\rho),$$

where we used the monotonicity of the cosine on $[0, \pi]$.
This shows that $\rho \geq \arccos(\langle v_1, v_2\rangle)$ and Definition 2.27 yields

$$\theta = \sup\left\{\arccos\left(\frac{\langle v_1, v_2\rangle}{\|v_1\|_2\|v_2\|_2}\right) : v_1, v_2 \in K \setminus \{0\}\right\} \leq \rho.$$

Together with (2.23) we get $\theta = \rho$. □

Remark 2.30. A non-negative data set lies inside a cone of angle $\theta = \frac{\pi}{2}$.
From the motivation it is already clear that a rotation of the data set to the positive orthant only exists if the low-dimensional data is lying inside a cone with apex at 0 and opening angle of at most $\theta = \frac{\pi}{2}$. If the opening angle of the cone would be larger, the scattering of the data would contradict the existence of a suitable rotation.
Accordingly, in order to solve a non-negative dimensionality reduction problem of the form (2.22) with the splitting approach 2.25 we need to ensure that the low-dimensional data set is also lying inside such a cone. Thus, we need to characterize dimensionality reduction methods that preserve the property of the data set to lie inside a cone of a certain angle.

Definition 2.31 (Cone condition)**.** Let the data set X lie inside a cone of angle θ. We say that a dimensionality reduction method P fulfills the *cone condition for* θ if the low-dimensional data points $y_i = P(x_i)$, $i = 1, \ldots, n$ are lying inside a cone of the same angle, i.e.,

$$\frac{\langle x_i, x_j\rangle}{\|x_i\|_2\,\|x_j\|_2} \geq \cos(\theta) \quad \Rightarrow \quad \frac{\langle P(x_i), P(x_j)\rangle}{\|P(x_i)\|_2\,\|P(x_j)\|_2} \geq \cos(\theta). \tag{2.24}$$

Remark 2.32. This condition is weaker than requiring P to be angle-preserving since only the opening angle of the cone containing the data is required to not increase. In particular, every angle-preserving map P fulfills the cone condition.

Now that we have introduced the appropriate concepts, let us formulate a sufficient condition for solving problem (2.22) with Approach 2.25.

Theorem 2.33 (Sufficient condition). *Let the pair $(\mathcal{U}, g)$ define a dimensionality reduction method P_*. If*

(i) *g is rotationally invariant and*
(ii) *P_* fulfills the cone condition for $\theta = \frac{\pi}{2}$,*

a solution of (2.22) *can be computed with the splitting approach 2.25. Moreover, it holds*

$$\min_{\substack{P\in\mathcal{U}\\ P(X)\geq 0}} g(P) = \min_{P\in\mathcal{U}} g(P).$$

Proof. We will show that indeed a solution of (2.22) can be constructed via 2.25. Let $P_* \in \arg\min_{P\in\mathcal{U}} g(P)$ be the dimensionality reduction method. Since X lies inside a cone with opening angle $\frac{\pi}{2}$, the low-dimensional representation $P_*(X)$ also lies inside a cone of the same angle due to the cone condition (condition (ii)). Hence, there exists a rotation $R \in SO(d)$ with $RP_*(X) \geq 0$. The rotational invariance of g (condition (i)) implies

$$RP_* \in \mathcal{U} \qquad \text{and} \qquad g(RP_*) = g(P_*).$$

Now, it follows from $\min_{P\in\mathcal{U}} g(P) \leq \min_{\substack{P\in\mathcal{U}\\ P(X)\geq 0}} g(P)$ that

$$\min_{\substack{P\in\mathcal{U}\\ P(X)\geq 0}} g(P) \leq g(RP_*) = g(P_*) = \min_{P\in\mathcal{U}} g(P) \leq \min_{\substack{P\in\mathcal{U}\\ P(X)\geq 0}} g(P),$$

which shows that RP_* is a minimizer of (2.22). □

Remark 2.34. The sufficient condition in Theorem 2.33 consists of two conditions which are of different nature. Condition (i) is a constraint concerning the cost functional of the optimization problem, whereas condition (ii) is a constraint on the solution of the optimization problem. Thus, the first one is much easier to check since for the second one a minimizer needs to be known explicitly.

Theorem 2.33 paves the way for using the splitting approach 2.25 in non-negative dimensionality reduction problems. This elegant approach provides the possibility of extending classical dimensionality reduction methods to non-negativity preserving ones. In contrast to [132], we can use the theory and algorithms developed for these classical methods.
Even though the rotation invariance is naturally fulfilled by many cost functionals due to their construction based on the preservation of the geometrical structure of the data set, the sufficient condition 2.33 is quite restrictive. The condition that the data is contained in a certain cone does not apply to that many methods.
In particular, from non-linear methods this cannot be expected without further requirements on the manifold as P is an approximation of B (compare the diagram in Figure

2.2). Linear methods, however, are more likely to satisfy the cone condition. Despite these limitations, our approach is a step ahead and can be used in many applications (e.g. compare Chapter 3).
Another crucial aspect is the *non-centering* of the low-dimensional data within the dimensionality reduction since otherwise the data is centered around zero and not contained in a cone with apex at 0 of angle smaller than $\frac{\pi}{2}$. Unfortunately, many dimensionality reduction methods include a centering of the data in order to uniquely identify a minimizer. The centering constrained $Y\mathbf{1}_{n\times 1} = 0$ can be dropped (as we did in Section 2.2) but the thereby obtained methods are not unique. However, the uniqueness is important for constructing an inverse reduction map, if this is possible at all.
We will now formulate a further condition that guarantees the validity of the cone condition with $\theta = \frac{\pi}{2}$ for P. This condition is motivated by the fact that in applications the high-dimensional points are often not exactly lying on the manifold $\mathcal{M}$ but only nearby. The following definition characterizes this deviation.

Definition 2.35. A map $Q\colon \mathbb{R}^d \to \mathbb{R}^D$ with

$$Q \circ P(x_i) = x_i + \epsilon_{x_i} \qquad \text{for all } x_i \in X \tag{2.25}$$

is called an *approximative left-inverse of* P with *perturbation vectors* ϵ_{x_i}.

The setting of this definition is depicted in Figure 2.8 in order to illustrate the perturbation ϵ_{x_i}.

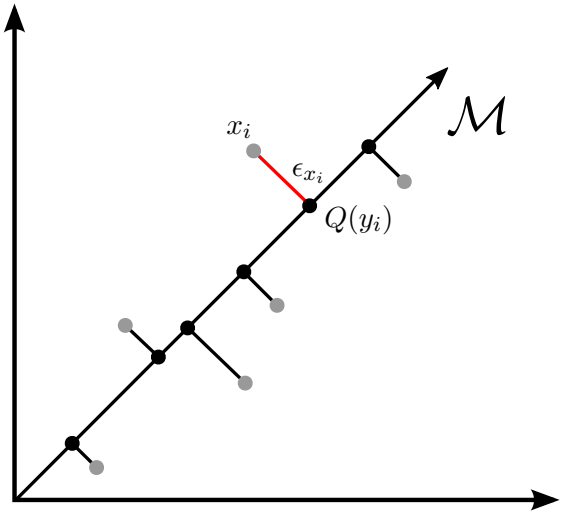


Figure 2.8: The high-dimensional data set X is not exactly lying on the manifold $\mathcal{M}$. The perturbation ϵ_{x_i} is the vector between point x_i and the point $Q(y_i) = Q(P(x_i))$.

Remark 2.36. Provided that P well approximates $\mathcal{B}$ from the diagram in Figure 2.2, the approximative left-inverse Q of P can be interpreted as an approximation of $\mathcal{B}^{-1}$.

Remark 2.37. We will see in Section 2.4.1 that for PCA the map Q with $Q(Y) = UY$ is an approximative left-inverse of P.

With this definition we can state an alternative condition to 2.33 (ii).

Theorem 2.38. *Let Q be an approximative left-inverse of P with perturbation vector ϵ_{x_i} bounded by*

$$\|\epsilon_{x_i}\|_2 \leq \min_{j=1,\dots,n} \left\{ \frac{1}{3} \frac{\langle x_i, x_j \rangle}{\|x_j\|}, \sqrt{\frac{1}{3}\langle x_i, x_j \rangle} \right\} \tag{2.26}$$

and

$$\frac{\langle Q(y_i), Q(y_j) \rangle}{\|Q(y_i)\|_2 \|Q(y_j)\|_2} \leq \frac{\langle y_i, y_j \rangle}{\|y_i\|_2 \|y_j\|_2} \qquad \text{for all } y_i = P(x_i),\ i, j = 1, \dots, n. \tag{2.27}$$

Then, P fulfills the cone condition for $\theta = \frac{\pi}{2}$.

Proof. Let X be a data set inside a cone of angle $\frac{\pi}{2}$. From the properties of the absolute value and the Cauchy-Schwartz inequality it follows $-x_i^T \epsilon_{x_j} \leq |x_i^T \epsilon_{x_j}| \leq \|x_i\|_2 \|\epsilon_{x_j}\|_2$ and thus,

$$x_i^T \epsilon_{x_j} \geq -\|x_i\|_2 \|\epsilon_{x_j}\|_2. \tag{2.28}$$

Hence, we get

$$\begin{aligned}
\frac{\langle P(x_i), P(x_j) \rangle}{\|P(x_i)\|_2 \|P(x_j)\|_2} &\overset{(2.27)}{\geq} \frac{\langle Q \circ P(x_i), Q \circ P(x_j) \rangle}{\|Q \circ P(x_i)\|_2 \|Q \circ P(x_j)\|_2} \\
&\overset{(2.25)}{=} \frac{\langle x_i + \epsilon_{x_i}, x_j + \epsilon_{x_j} \rangle}{\|x_i + \epsilon_{x_i}\|_2 \|x_j + \epsilon_{x_j}\|_2} \\
&= \frac{\langle x_i, x_j \rangle + \langle x_i, \epsilon_{x_j} \rangle + \langle x_j, \epsilon_{x_i} \rangle + \langle \epsilon_{x_i}, \epsilon_{x_j} \rangle}{\|x_i + \epsilon_{x_i}\|_2 \|x_j + \epsilon_{x_j}\|_2} \\
&\overset{(2.28)}{\geq} \frac{\langle x_i, x_j \rangle - \|x_i\|_2 \|\epsilon_{x_j}\|_2 - \|x_j\|_2 \|\epsilon_{x_i}\|_2 - \|\epsilon_{x_i}\|_2 \|\epsilon_{x_j}\|_2}{\|x_i + \epsilon_{x_i}\|_2 \|x_j + \epsilon_{x_j}\|_2} \\
&\geq \frac{\langle x_i, x_j \rangle - \frac{1}{3}\langle x_i, x_j \rangle - \frac{1}{3}\langle x_i, x_j \rangle - \frac{1}{3}\langle x_i, x_j \rangle}{\|x_i + \epsilon_{x_i}\|_2 \|x_j + \epsilon_{x_j}\|_2} \\
&= 0.
\end{aligned}$$

For $\theta = \frac{\pi}{2}$ this leads to

$$\cos(\theta) = 0 \leq \frac{\langle P(x_i), P(x_j) \rangle}{\|P(x_i)\| \|P(x_j)\|},$$

which completes the proof. □

Remark 2.39. Condition (2.27) can be thought of as an inverse version of the cone condition (2.24). It requires that the angle between two vectors y_i, y_j is not decreasing when Q is applied.

Remark 2.40. The bound on $\|\epsilon_{x_i}\|_2$ basically requires the data set to lie in a cone with a slightly smaller angle $\theta < \frac{\pi}{2}$. Careful rearranging of (2.26) leads to

$$\theta = \max_{i,j=1,\dots,n} \measuredangle(x_i, x_j) \leq \max_{i,j=1,\dots,n} \left\{ \arccos\left(3\frac{\|\epsilon_{x_i}\|_2}{\|x_i\|_2}\right), \arccos\left(3\frac{\|\epsilon_{x_i}\|_2^2}{\|x_i\|_2\|x_j\|_2}\right) \right\}.$$

This shows that even for a small perturbation, the maximal angle $\measuredangle(x_i, x_j)$ between x_i and x_j cannot be equal to $\frac{\pi}{2}$. This matches with our intuition.

Numerical considerations and summary

So far we have discussed the feasibility of the splitting approach analytically. Since we have a concrete application in mind, also the question of the numerical realization is of particular interest. The main issue is the computation of a suitable rotation R. As discussed in detail in Section 1.3.2, this can be done by solving an optimization problem on the Lie group $SO(d)$ of special orthogonal matrices. In the Sections 1.2 and 1.3 we have developed an efficient algorithm to compute R which is in general for $d > 2$ a non-trivial task. This substantiates the necessity of the theoretical considerations in Chapter 1.
Solving minimization problem (2.22) is a crucial task since we aim for a global minimizer. This is where many classical numerical methods fail. We have proposed an approach which avoids this problem, as it resorts to the known global solution of a related problem and uses this as a basis to compute a solution of the problem itself. By this, a global solution of the minimization problem (2.22) is obtained.

2.4 Methods for non-negative dimensionality reduction

In the following we reconsider the dimensionality reduction methods introduced in the first part of this chapter in order to figure out whether they fit into the above discussed framework (see splitting approach 2.25). This means that for the different methods we will verify the requirements of Theorem 2.33 and Theorem 2.38, respectively. It will turn out that all methods presented in Section 2.2 possess a rotationally invariant cost functional whereas the cone condition for $\theta = \frac{\pi}{2}$ is rarely fulfilled. This is due to the fact that non-linear methods are in general not angle-preserving. Non-linear methods try to unfold manifolds like the Swiss role (compare [73]) and thus, they can only try to preserve the angle between data points locally (as e.g. LLE). Furthermore, the centering of the data contradicts the cone condition as mentioned above.

2.4.1 Non-Negative Principal Component Analysis - NNPCA

Recall from Section 2.2.1 the cost functional g_{PCA} and the admissible set $\mathcal{U}_{PCA}$. With this, the *non-negative principal component analysis (NNPCA) problem* according to (2.22) reads

$$\min_{\substack{U^T \in \mathbb{R}^{d\times D}, U^T U = \mathbf{Id}_d \\ U^T X \geq 0}} -\operatorname{tr}(U^T X X^T U). \tag{2.29}$$

NNPCA problems have been studied in different forms by several authors. Some of them require the matrix U to be non-negative (e.g. [31, 52, 133]). This constraint is stronger than ours and thus, the minimal value of the cost functional is in general expected to be larger. The optimization problem itself is then solved by a relaxation method. A similar formulation is used in [92] for a multi-linear NNPCA which is solved by optimization on the Grassmannian manifold. Some of the earliest references on NNPCA are due to Plumbley and Oja [89, 95], where NNPCA is considered as a special case of a non-linear PCA [88], which leads to a different cost functional.
In [3, 4, 107] algorithms for a *sparse* NNPCA are developed. The NNPCA problem formulation therein is similar to ours, but the approaches for finding minimizers and thus, the algorithms are quite different. In particular, they require the minimizer to be a sparse minimizer.
All in all, none of the above mentioned references studies the NNPCA problem as stated in (2.29). Let us now analyze our splitting approach for this NNPCA problem. Due to the preliminary considerations we have done, verifying the sufficient condition 2.33 is straightforward.

Theorem 2.41 (Non-negative principal component analysis). *Let $X \in \mathbb{R}^{D\times n}$ be a non-negative data set and choose d such that*

$$\operatorname{err}_{PCA}(d,X) \overset{(2.4)}{=} \sum_{k=d+1}^{D} \sigma_k^2 \leq \min_{i,j=1,\dots,n} \left\{ \frac{1}{9}\frac{\langle x_i, x_j\rangle^2}{\|x_j\|_2^2}, \frac{1}{3}\langle x_i, x_j\rangle \right\}. \tag{2.30}$$

Then, a solution of the NNPCA problem (2.29) *can be computed by the splitting approach 2.25. It is given by $P = RV_d{}^T$ for a suitable $R \in SO(d)$, where $V_d{}^T$ is the solution of the PCA problem for X. Moreover, we have*

$$\operatorname{err}_{NNPCA}(d,X) = \operatorname{err}_{PCA}(d,X).$$

Proof. We will use Theorems 2.33 and 2.38 in order to show that the splitting approach yields indeed a solution of the NNPCA problem (2.29). We start with verifying 2.33 (i). Let $P = U^T \in \mathcal{U}_{PCA}$ and $R \in SO(d)$ be two matrices and compute for $RU^T \in \mathbb{R}^{d\times D}$

$$RU^T(RU^T)^T = RR^T = \mathbf{Id}_d \qquad \Rightarrow \qquad RU^T \in \mathcal{U}_{PCA}$$

and

$$\begin{aligned} g_{PCA}(RU^T) &= -\operatorname{tr}\left(RU^T XX^T (RU^T)^T\right) \\ &= -\operatorname{tr}\left(R^T RU^T XX^T U\right) \\ &= g_{PCA}(U^T). \end{aligned}$$

For the last equation we used the cyclic invariance of the trace. This shows that g_{PCA} is indeed rotationally invariant. Next, to prove 2.33 (ii), i.e., that the minimizer $P_* = V_d{}^T$ of (2.29) fulfills the cone condition for $\theta = \frac{\pi}{2}$, we exploit Theorem 2.38. First,

we observe that V_d is an approximative left-inverse of ${V_d}^T$ with perturbation vectors $\epsilon_{x_i} = (V_d {V_d}^T - \mathbf{Id}_d)x_i$. Then, it follows from (2.4) that

$$\|\epsilon_{x_i}\|_2^2 \leq \sum_{i=1}^{n} \|\epsilon_{x_i}\|_2^2 = \|(V_d {V_d}^T - \mathbf{Id}_d)X\|_F^2 = \sum_{i=d+1}^{D} \sigma_i^2,$$

where σ_i are the smallest $D-d$ singular values of X. Now, (2.30) yields

$$\|\epsilon_{x_i}\|_2 \leq \min_{i,j=1,\dots,n} \left\{ \frac{1}{3}\frac{\langle x_i, x_j\rangle}{\|x_j\|_2}, \sqrt{\frac{1}{3}\langle x_i, x_j\rangle} \right\}.$$

Furthermore, we have to check inequality (2.27). Therefore, we compute

$$\langle V_d y_i, V_d y_j\rangle = y_i^T {V_d}^T V_d y_j = \langle y_i, y_j\rangle,$$

which yields $\|V_d y_i\|_2 = \|y_i\|_2$ and thus,

$$\frac{\langle V_d y_i, V_d y_j\rangle}{\|V_d y_i\|_2 \|V_d y_j\|_2} = \frac{\langle y_i, y_j\rangle}{\|y_i\|_2 \|y_j\|_2}.$$

This shows that $P_* = {V_d}^T$ fulfills 2.33 (ii).
All in all, Theorem 2.33 is applicable and the splitting approach leads to a minimizer of the NNPCA problem. The equality of the approximation errors follows directly from the same theorem. □

If the high-dimensional data is exactly lying on a d-dimensional subspace of $\mathbb{R}^D$, i.e., $\mathrm{rk}(X) = d$, it is sufficient that the data set is lying in the positive orthant to apply the previous theorem. Because the reduction map ${V_d}^T$ preserves angles between elements of the subspace $V_d {V_d}^T \mathbb{R}^D$, the cone condition for $\theta = \frac{\pi}{2}$ is true for X. Let us formulate this as a corollary.

Corollary 2.42. *Let $X \in \mathbb{R}^{D\times n}$ be a non-negative data set with* $\mathrm{rk}(X) = d$. *Then, the assumptions of Theorem 2.41 are fulfilled. Moreover, the approximation error is*

$$\mathrm{err}_{NNPCA}(d, X) = 0.$$

Proof. This is a direct consequence of Theorem 2.3, Remark 2.6 and Theorem 2.41. Since X has rank d, it has exactly d non-vanishing singular values. Thus, the approximation error satisfies $\mathrm{err}_{PCA}(d, X) = 0$. From Theorem 2.41 now follows that $\mathrm{err}_{NNPCA}(d, X) = \mathrm{err}_{PCA}(d, X) = 0$. □

Remark 2.43. If the data set is not exactly lying on a d-dimensional subspace, it has to be contained in a cone with opening angle $\theta < \frac{\pi}{2}$ in order to make the cone condition still accomplishable. The value of θ depends on the size of the perturbation.

The statement of Theorem 2.41 is much stronger than the one of Corollary 2.42 since it allows for some perturbation of the data. This is very useful for practical applications, where usually some noise is involved.

2.4.2 Non-Negative Multidimensional Scaling - NNMDS

One could think that the theoretical results for *non-negative multidimensional scaling (NNMDS)* are the same as for NNPCA since their solutions for the classical dimensionality reduction problem are closely related. But the centering of X, from which the reduction map P_{MDS} is computed (compare Theorem 2.11), prevents proving a result similar to Theorem 2.41.
However, if the non-negative data set X is lying inside a d-dimensional subspace (and not just nearby), we will show that the splitting approach 2.25 yields a minimizer of the NNMDS problem

$$\min_{\substack{P\in\mathbb{R}^{d\times D}, PP^T=\mathbf{Id}_d \\ PX\geq 0}} \operatorname{tr}(HX^TXH - HX^TP^TPXH). \tag{2.31}$$

When applying the splitting approach to NNMDS, we have to bear in mind that the minimizer of the classical MDS problem needs to fulfill the cone condition for $\theta = \frac{\pi}{2}$. Since the minimizer of the MDS problem is not unique, we may choose the right one according to the condition.
To this end, note that the centered data set X^c has not necessarily rank d but $\operatorname{rk}(X^c) = \delta \leq d$. Furthermore, $X^c \in \operatorname{span}(X)$ since $X^c = XH$ and thus, the columns of X^c are linear combinations of the columns of X. Recall, that the solution ${V_\delta}^T$ of the classical δ-dimensional MDS problem (compare Theorem 2.11) is given by the singular vectors to the δ non-zero singular values of X^c. It can be extended to an orthonormal basis V_d of $\operatorname{span}(X)$ by adding $d-\delta$ singular vectors of X^c corresponding to the singular value 0. In particular, ${V_{d-\delta}}^T X^c = 0$ holds and $V_\delta {V_\delta}^T X^c = X^c$.
Let us show that

$${V_d}^T = (V_\delta, V_{d-\delta})^T \tag{2.32}$$

is indeed a minimizer of the MDS cost functional in Corollary 2.13 by proving that $g_{MDS}({V_d}^T) = 0$:

$$\begin{aligned} g_{MDS}({V_d}^T) &= \operatorname{tr}\left((X^c)^TX^c - (X^c)^TV_d{V_d}^TX^c\right) \\ &= \operatorname{tr}\left((X^c)^TX^c - (X^c)^T(V_\delta, V_{d-\delta})(V_\delta, V_{d-\delta})^TX^c\right) \\ &= \operatorname{tr}\left((X^c)^TX^c - (X^c)^TV_\delta{V_\delta}^TX^c - (X^c)^TV_{d-\delta}{V_{d-\delta}}^TX^c\right) \\ &= 0. \end{aligned}$$

In summary, we observe that

$$\operatorname{err}_{MDS}(\delta, X) = \sum_{i=\delta+1}^{D} \sigma_i^2 = 0 = \sum_{i=d+1}^{D} \sigma_i^2 = \operatorname{err}_{MDS}(d, X)$$

and, hence, the choice of the $(\delta+1)$th to dth basis vector does not affect the cost functionals value.
Using this particular minimizer ${V_d}^T$ of the classical MDS problem, we can apply the splitting approach to the NNMDS problem as stated in the following theorem.

Theorem 2.44 (Non-negative multidimensional scaling). *Let the data set $X \in \mathbb{R}^{D\times n}$ be non-negative with $\mathrm{rk}(X) = d$. Then, a solution of the NNMDS problem (2.31) can be computed by the splitting approach and it is explicitly given by $P = RV_d{}^T$ for a suitable rotation $R \in SO(d)$ and the solution $V_d{}^T$ of the corresponding MDS problem as defined in (2.32).*

Proof. We will apply Theorem 2.33. Therefore, we first observe that g_{MDS} is rotationally invariant: Let $P \in \mathbb{R}^{d\times D}$ with $PP^T = \mathbf{Id}_d$ and $R \in SO(d)$ then, for $RP \in \mathbb{R}^{d\times D}$

$$RP(RP)^T = RPP^TR^T = \mathbf{Id}_d \qquad \Rightarrow \qquad RP \in \mathcal{U}_{MDS}$$

and

$$\begin{aligned} g_{MDS}(RP) &= \mathrm{tr}(HX^TXH - HX^T(RP)^TRPXH) \\ &= \mathrm{tr}(HX^TXH - HX^TP^TR^TRPXH) \\ &= g_{MDS}(P). \end{aligned}$$

This proves the rotational invariance 2.33 (i). For condition 2.33 (ii) it is sufficient to show $\langle V_d{}^T x_i, V_d{}^T x_j\rangle \geq 0$. In the preliminary considerations we have seen that due to the assumption $\mathrm{rk}(X) = d$ and the construction of $V_d{}^T$ the data X is lying in the subspace $V_dV_d{}^T\mathbb{R}^D$ and thus, $V_dV_d{}^TX = X$. Hence,

$$\langle V_d{}^T x_i, V_d{}^T x_j\rangle = \langle x_i, V_dV_d{}^T x_j\rangle = \langle x_i, x_j\rangle \geq 0.$$

This completes the proof. □

A consequence of this theorem is that for X with $\mathrm{rk}(X) = d$ the approximation error is again the same for both, the non-negative and the classical method.

2.4.3 Splitting approach and non-linear methods

As already indicated, the splitting approach is not suitable for non-linear methods. This has several reasons. Even though the cost functionals for Isomap (2.16), LLE (2.18) and LE (2.21) are rotationally invariant (mostly due to the cyclic invariance of the trace), the sufficient condition from Theorem 2.33 is not applicable since these methods are in general not angle-preserving. The alternative condition from Theorem 2.38 is not helpful since we are neither aware of appropriate approximative left-inverses for these methods, nor are these given explicitly in the literature.
Another aspect concerns the formulation of the non-linear dimensionality reduction problem of LE and LLE. In Section 2.2.4 we have seen that these problems could not be formulated as an optimization problem on the map P but only on the representation Y. In this setting, imposing the additional constraint $Y \geq 0$ is very restrictive since the combination of $YDY^T = \mathbf{Id}_d$ for a diagonal matrix D and $Y \geq 0$ forces Y to have at most one non-vanishing entry per column. Thus, the solution is sparse which is not suitable for our applications.
Nevertheless, several authors have addressed problems with these constraints. In [78, 127, 132] they have been discussed for NNLE and NNLLE and the corresponding optimization problems were solved by update algorithms.

3 Applications to signal separation

Many audio-related applications take advantage of the ability to separate sources from a mixture without a prior knowledge about the mixing process. Thus, the analysis and separation of audio signals into their source components is an important tool for the extraction of meta data from audio data as for example separating musical instruments from a polyphonic ensemble, music restoration or extracting speech from a noisy background. In all these situations, an efficient method to analyze the auditory scene in order to extract essential information is needed. This concept is known as blind signal separation (BSS) and was the topic of many recent research projects as already discussed in the introduction of this thesis.
In the case of detection or separation of certain sources from a mixture of signals, time-frequency information about the data is collected and used to decompose the signal into different components corresponding each to one of the source signals. This decomposition is based on the assumption that the different source signals can be characterized by their frequency distribution. There are different methods for the decomposition of time-frequency data available (e.g. independent component analysis (ICA) or non-negative matrix factorization (NNMF)).
Time-frequency data is typically given by a spectrogram obtained from a signal transform, such as short-time Fourier transform (STFT). Of course, other transforms can be used for computing a time-frequency representation, but we will stick to the classical STFT. For high-energy signals, the time-frequency data is characterized by the high dimensionality of the Euclidean space in which the data is embedded. More precisely, the dimension of this space is defined by the frequency range of the original signal and the size of the signal transform. A standard value for the frequency range would be 256. Therefore, it suggests itself that a reduction of the data's dimensionality might improve the method and speed up the computation of the data analysis. We observe that in many cases not all information contained in the data points is relevant for understanding the underlying characteristics or properties of the data. Many signals can be sufficiently described by a few dominant frequencies. Also, low-dimensional data sets are much easier to operate with in view of classification, visualization or decomposition. As a consequence, we would like to reduce the dimensionality of the given data in a preprocessing step before we apply a decomposition method. Thus, we focus on the interaction of dimensionality reduction and decomposition methods such as ICA or NNMF. The idea of combining dimensionality reduction and ICA is not a new concept (see [36, 48, 69, 117]). But to improve these strategies, a better mathematical understanding of these procedures is needed. Also, the substitution of ICA by a non-statistic based method such as NNMF could improve the results.
An important aspect we have to take into account is non-negativity. The amplitude

spectrogram, output from the STFT, is non-negative. In this context, non-negativity refers to the fact that the data matrix is entry-wise non-negative. This fits very well with the decomposition by non-negative matrix factorization which requires, as the name suggests, non-negative input data. But the application of an intermediate dimensionality reduction step might cause negative entries in the low-dimensional representation. Thus, there is a need of reduction methods which are able to preserve non-negativity. To this end, we have developed an approach for non-negative dimensionality reduction (compare Chapter 2).
In the present section, we will introduce a signal separation procedure which includes this dimensionality reduction step. We will combine different techniques and discuss several numerical examples to illustrate the algorithm's applications. We will focus on the separation of single channel drum and percussion tracks which are typically high-energy signals. There has been done some research in this direction (see [36, 117]) but so far the combination of non-negative dimensionality reduction and NNMF has not been considered by other authors.
It will turn out that the combination of PCA or our NNPCA with ICA yields the best separation. However, our new NNPCA combined with NNMF does perform almost as good as PCA and ICA, whereas Isomap followed by any of the two decomposition methods shows very poor separation qualities. In the latter, we used a naive kernel approach (compare [86]) for approximating the inverse reduction map.
In Section 3.1, we introduce the concept of signal detection and separation and review the involved methods in several subsections. Section 3.1.1 is dedicated to the generation of time-frequency data, where we introduce the short-time Fourier transform. In Section 3.1.2, we discuss some difficulties which arise when it comes to dimensionality reduction in signal separation. Thereafter, we study decomposition techniques focusing on independent component analysis and non-negative matrix factorization (Section 3.1.3).
In Section 3.2, we will discuss three examples. We will use the before-explained algorithm in order to separate different mixtures of single-channel audio recordings. The examples are introduced in Section 3.2.1 and the results are discussed in 3.2.2.

3.1 Signal separation procedure

Given a mixture $f = \sum_i f_i$ of band-limited source signals $f_i \in L^2(\mathbb{R})$, signal separation aims to estimate the different components f_i by using specific assumptions on the time-frequency or statistical characteristics.
For the matter of signal separation, also the identification of the time intervals during which a certain source signal is active is a crucial aspect. This procedure is called signal detection and can be used for further analysis. In fact, provided the time locations where a certain source is active are known, separation algorithms could concentrate on these regions and perform the source extraction with higher resolution, but this is not the objective of this work.
For the numerical examples discussed in Section 3.2, we used the signal detection and separation procedure which is illustrated in Figure 3.1: First, the spectrogram X of the

signal to be separated is computed by a short-time Fourier transform. The spectrogram, a data matrix whose columns are representing the time steps of the signal, contains information about the frequencies which are present in the signal at each time step. A spectrogram can be obtained by any time-frequency transform but we stick to the classical discrete STFT. Usually, the data matrix X is very high-dimensional. In order to make computation and interpretation easier, it is therefore convenient to apply a (non-negative) dimensionality reduction method (the map P) in a second step which reduces the dimension of the data drastically. Then, the reduced data Y is decomposed by assuming it to be a linear mixture $Y = AS$ of the unknown source components S. From this decomposition we obtain data matrices which need to be back-lifted to the high-dimensional space in order to get the spectrograms of the sources and to apply the inverse time-frequency transform for a complete separation.
In the following, we will give further information on the involved methods. We will focus on our algorithm where we used STFT, PCA, NNPCA, Isomap, ICA and NNMF.

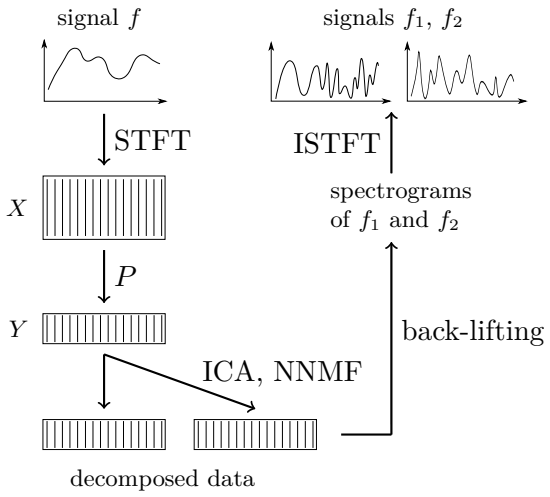


Figure 3.1: Signal separation with dimensionality reduction. The map P is used to first reduce the dimension of the data X obtained from a STFT, before the reduced data Y is decomposed into different components, each assigned to one of the source signals. The decomposed data is then back-lifted to the initial space before an ISTFT leads to the output signals.

3.1.1 Generation of time-frequency data

We are interested in implementing a separation algorithm for music recordings, especially for percussion tracks. Digital signals are time-limited and discrete (usually sampled from a continuous signal). For the implementation of an algorithm to separate this particular class of signals we will recall some basic definitions and tools from discrete Fourier

analysis. In the following, let $\mathbb{Z}_N$ denote the set $\{0, \dots, N-1\} \subset \mathbb{N}$ and thus, such a signal can be thought of as an element $\ell^\infty(\mathbb{Z}_N) \simeq \mathbb{R}^N$.
An approach for the extraction of meta data from an audio mixture is to use local information about the signal. Provided a data set at each point in time, the idea is to assign this information to the different source signals which yield a separation of the signal. One possible type of information we can use here is frequency information. We assume that each source signal has a characteristic time-frequency distribution which can be used to distinguish different source signals. This is based on the time-frequency representation of the signal which is the evolution in time of a signal's spectral content. From Fourier analysis we know that the frequency spectrum of an $L^1(\mathbb{R})$ or $\ell^\infty(\mathbb{Z}_N)$ signal is given by the Fourier transformation. However, since the Fourier transform is time-independent this it not precisely what we need. But we will see that we can use the Fourier transform to generate a time-frequency representation anyway.
In this section, we rely on the textbooks [85, 130]. For more information, especially on the continuous time-frequency analysis we refer to [46, 135]. Let us briefly recall the discrete Fourier transform and its inverse. These can be defined analogously to the *continuous Fourier transform* $\mathcal{F}F(\omega) = \int_{\mathbb{R}} F(t) e^{-2\pi \mathbf{i} \omega t} \mathrm{d}t$ and its inverse.

Definition 3.1. For a discrete function $f \in \ell^\infty(\mathbb{Z}_N)$, its *discrete Fourier transform* $\mathfrak{F}f \in \ell^\infty(\mathbb{Z}_N)$ is defined as

$$(\mathfrak{F}f)_j = \sum_{k=0}^{N-1} f_k e^{-\frac{2\pi \mathbf{i} j k}{N}}, \qquad \text{for } j \in \mathbb{Z}_N,$$

with $\mathbf{i}$ denoting the imaginary unit.

The values f_k can be obtained as samples $f_k = F(t_k)$ from a continuous function F. The sampling points in time $t_k = \frac{kT}{N}$ with sampling rate $\frac{T}{N}$ have to be chosen equispaced and according to the length T of the signal. Recall that a continuous function F can be exactly reconstructed if sampled at *Nyquist rate*. From the famous *Nyquist-Shannon Sampling Theorem* (see [84]), we know that this optimal sampling rate is closely related to the bandwidth of the signal F.

Definition 3.2. Let $F \in L^1(\mathbb{R})$ be a function. The length of the support of the continuous Fourier transform $\mathcal{F}F$ of F is called *total bandwidth*. If $\mathcal{F}F(\omega) = 0$ for $\omega \notin]-\pi\delta, \pi\delta[$, the function F has total bandwidth $2\pi\delta$ and is called *band-limited to* $[-\pi\delta, \pi\delta]$.

The Sampling Theorem now states, that a continuous function $F \in L^1(\mathbb{R})$ which is band-limited to $[-\delta\pi, \delta\pi]$ can be completely reconstructed from its samples at $t_k = \frac{k}{\delta}$ for $k \in \mathbb{N}$ with the formula

$$F(t) = \sum_{k=-\infty}^{\infty} F(t_k) \frac{\sin(\pi\delta(t - t_k))}{\pi\delta(t - t_k)} = \sum_{k=-\infty}^{\infty} f_k \operatorname{sinc}(\delta t - k).$$

The sampling frequency δ is known as the *Nyquist rate*.

A signal cannot be both band-limited and time-limited. As in practice all signals are time-limited, band-limited signals are only a theoretical concept which is used for analytical purposes. A common technique in application is the truncation of the signal's Fourier transform if it decreases fast enough. Moreover, this truncation can be justified by recalling that the human hearing range is roughly given as 20Hz to 20000Hz.
The continuous Fourier transform is not convenient for implementation and thus, the truncation and the Sampling Theorem are fundamental for digital signal processing since they enable us to use the discrete Fourier transform.
In Definition 3.1, we have not only sampled in time but also in the frequency domain. The frequency samples are $\omega_j = \frac{2\pi j\delta}{N}$. The value $(\mathfrak{F}f)_j$ is a complex number which has in polar coordinates the form $|(\mathfrak{F}f)_j|e^{\mathbf{i}\arg((\mathfrak{F}f)_j)}$. The value $|(\mathfrak{F}f)_j|$ is called the *amplitude* and $\arg((\mathfrak{F}f)_j)$ the *phase* of $(\mathfrak{F}f)_j$.
As for the continuous Fourier transform, there is an inverse discrete Fourier transform.

Definition 3.3. For a discrete function $g \in \ell^\infty(\mathbb{Z}_N)$ its *discrete inverse Fourier transform* $\mathfrak{F}^{-1}g$ is defined by

$$(\mathfrak{F}^{-1}g)_k = \frac{1}{N}\sum_{j=0}^{N-1} g_j e^{\frac{2\pi\mathbf{i}jk}{N}}, \qquad \text{for } k \in \mathbb{Z}_N.$$

Indeed, the inverse discrete Fourier transform is the inverse of the Fourier transform as the following theorem states.

Theorem 3.4. *For a discrete function $f \in \ell^\infty(\mathbb{Z}_N)$ the discrete Fourier inversion formula holds:*

$$f_k = (\mathfrak{F}^{-1}\mathfrak{F}f)_k = (\mathfrak{F}\mathfrak{F}^{-1}f)_k \qquad \textit{for all } k \in \mathbb{Z}_N.$$

Proof. See [130]. □

Remark 3.5. It is easy to see that a straightforward computation of the discrete Fourier transform is of complexity $\mathcal{O}(N^2)$ as the computation for each of the N components is of complexity $\mathcal{O}(N)$. In order to compute the discrete Fourier transform efficiently, the so called *fast Fourier transform (FFT)* can be used. There are different algorithms to perform the FFT, among them the Cooley-Tukey algorithm proposed in 1965 [28].

As already motivated, in signal detection we would like to have some local properties of f on which we can base our separation algorithm. In particular, we are interested in a 'local frequency spectrum'. Since for a continuous F the frequency spectrum computed by a Fourier transform is only given for a time interval and not for a single point in time, the idea is to choose the length of the interval to be short in order to approximate the frequency spectrum at a point. To this end, we restrict F to an interval by multiplication with a so called *window function* φ and compute the Fourier transform of this restriction. We choose the window function to be smooth as this avoids problems at the ends of the interval. As shown in Figure 3.2, we consider a segmentation of the signal into small patches of length L at distance h. For the discrete setting, this segmentation is obtained by multiplication of the signal by a discrete, compactly supported window of length L

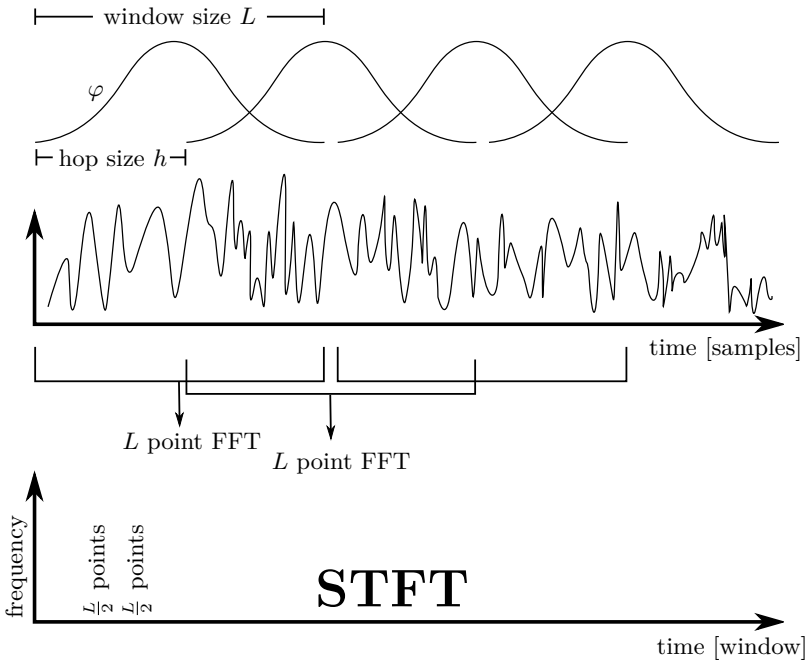


Figure 3.2: Short-time Fourier transform and construction of spectrogram.

with center $\frac{L}{2} + lh$. Subsequently, the FFT algorithm is applied to the segments in order to compute a discrete time-frequency representation.
This motivation leads to the definition of the discrete short-time Fourier transform.

Definition 3.6. Assume that $\varphi \in \ell^\infty(\mathbb{Z}_L)$ is a discrete window with $\varphi_k \neq 0$ and $f \in \ell^\infty(\mathbb{Z}_N)$. For n and $h \in \mathbb{N}$ with $(n-1)h = N - 1 - L$, we define the *discrete short-time Fourier transform (STFT)* $\mathfrak{F}_\varphi f$ of f by

$$(\mathfrak{F}_\varphi f)_{j,l} = \sum_{k=0}^{L-1} f_{k+lh}\varphi_k e^{-\frac{2\pi \mathbf{i} jk}{L}} = \left(\mathfrak{F}\left(f_{k+lh}\varphi_k\right)_{k=0}^{L-1}\right)_j, \qquad \text{for } j \in \mathbb{Z}_L, l \in \mathbb{Z}_n.$$

The parameter h is called *hop size* and L is the *window length*.

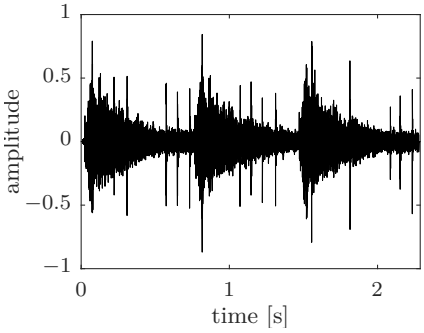
Remark 3.7. There is also a continuous version of the short-time Fourier transform (see e.g. [46]). Therefore, the discrete STFT is also called DSTFT in the literature.

The localization in Definition 3.6 gives us the frequency content of the signal in a concrete window φ with center $\frac{L}{2} + lh$ so that the discrete short-time Fourier transform depends on two indices, j for the frequency and l for the position of the window. Obviously, for a fixed l, we have $(f_{k+lh}\varphi_k)_k \in \ell^\infty(\mathbb{Z}_L)$ and thus, the STFT has properties analogue to the properties of the discrete Fourier transform.
By means of the discrete short-time Fourier transform we compute the frequency range of a signal f as a discrete function of time: the *(amplitude) spectrogram* of f. The spectrogram displays the values $|(\mathfrak{F}_\varphi f)_{j,l}|$ in a time-frequency diagram. Since we are

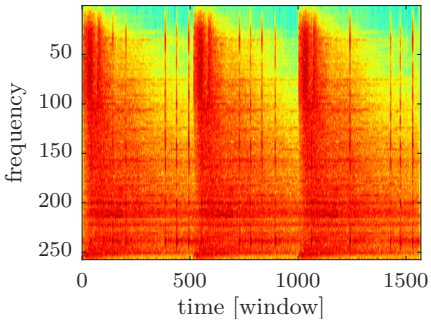
considering real-valued signals, the absolute value is symmetric in ω_j and thus, we only use the positive part of the spectrum and not the total bandwidth. For a fixed l, the values $|(\mathfrak{F}_\varphi f)_{j,l}|$ can be interpreted as the frequency range of f at time $\frac{L}{2} + lh$. Compared to the frequency spectrum obtained by a classical discrete Fourier transform, the spectrogram makes a lot more information contained in f accessible. In order to completely describe the STFT, the *phase spectrogram* $\arg((\mathfrak{F}_\varphi f)_{j,l})$ is needed as well.
In Figure 3.3, an example for the spectrogram of a signal is shown. The data matrix in Figure 3.3b contains column-wise the approximate frequency information for a point in time and row-wise the behavior in time of a certain frequency. The signal has been sampled with 44100Hz. According to that, the distance in time between two sampling points is therefore given as $2.27 \cdot 10^{-5}$s. When using $j = 1, \ldots, 256$ equispaced frequency samples, their distance is 86.13Hz. In the time-frequency plot we refer to a frequency sample by its number j. In a slight abuse of the notation we call this number nonetheless 'frequency'.
We say that a frequency is active at a certain time, if it contributes to the Fourier transform of the signal, i.e., the coefficient corresponding to this frequency does not vanish. In Figure 3.3b red colored entries correspond to a high value whereas blue correspond to a low value. The idea is to assign the active frequencies at each time step to one of the source signals. From the figure it can be seen, that for this particular example a lot of frequencies are active when a peak (high amplitude) is recorded. This is what we mean by *high-energy* or *transient signal*. The precise definition for a transient signal differs from this heuristic as a transient has a continuous and unbounded spectrum. Thus, the discrete Fourier transform seems not to be the optimal choice but this is no problem in practice due to the above-mentioned truncation.
An extreme example is a δ-distribution which is not even an $L^2(\mathbb{R})$ function. Therefore, the term high-energy signal is a better choice of denomination since this implies that the signal is at least in $L^2(\mathbb{R})$. The discrete Fourier transform leads to a discrete, finite spectrum which circumvents the above explained problem.
Previous to Definition 3.6, we mentioned the window function φ. Usually, a window function is a continuous, compactly supported, non-negative and symmetric function. In fact, this definition can be generalized claiming that the function decreases sufficiently fast to zero away from the origin. In the discrete setting, we sample the window function with the same sampling rate as the signal. The STFT was first used by Gábor in 1946. In [41], Gábor considers a truncation of the Gaussian window. Due to the importance of the STFT in many applications, the STFT using this special window is called *Gábor transform*.
From the huge class of window functions we like to introduce the *Hann window*

$$h(t) = \frac{1}{2}\left(1 + \cos\left(\frac{2\pi t}{L}\right)\right)\chi_{[-\frac{L}{2},\frac{L}{2}]}(t)$$

where L is the window size, i.e., $\operatorname{supp} h \subset [-\frac{L}{2}, \frac{L}{2}]$ (see [11]). This window is often chosen in signal processing as it has very low aliasing effects. We will use this window for our applications in Section 3.2. Of course, there are many other possible window functions (see e.g. [84]) but the comparison of those is not the objective of this work.



(a) A monophonic recording of 2.27 seconds length sampled at 44100Hz. The peaks in the signal correspond to one of the source signals and its extraction seems rather difficult.



(b) The spectrogram is a data matrix of size 256×1569, the redder the color the higher the Fourier coefficient, whereas blue corresponds to no contribution. The amplitude peaks of the signal are clearly recognizable as a lot of frequencies contribute to the signal at these particular time steps.

Figure 3.3: A signal f and its corresponding spectrogram. For the computation a 512-point FFT was used and a discrete Hann window with hop size 64.

Let us now introduce the inverse discrete short-time Fourier transform.

Definition 3.8. For φ, h and n as in Definition 3.6 with $h \leq L$ and $g \in \ell^\infty(\mathbb{Z}_L \times \mathbb{Z}_n)$ the *discrete inverse short-time Fourier transform (ISTFT)* is defined by

$$\left(\mathfrak{F}_\varphi{}^{-1} g\right)_k = \frac{1}{c_k} \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\colon\, j+lh=k} \left(\mathfrak{F}^{-1}\left(g_{i,l}\right)_{i=0}^{L-1}\right)_j, \qquad \text{for } k \in \mathbb{Z}_N,$$

where

$$c_k = \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\colon\, j+lh=k} \varphi_j.$$

Remark 3.9. The sum in Definition 3.8 is not empty if $h \leq L$. This follows from the decomposition of k by Euclidean division by h. This restriction is reasonable since otherwise the hop size would be larger than the window size and application of the discrete STFT would cause the loss of parts of the function f.

Theorem 3.10. *For a function $f \in \ell^\infty(\mathbb{Z}_N)$ and φ, h and n as in Definition 3.6 with $h \leq L$ the inversion formula holds:*

$$f_k = \left(\mathfrak{F}_\varphi{}^{-1}\mathfrak{F}_\varphi f\right)_k \qquad \textit{for all } k \in \mathbb{Z}_N.$$

Furthermore, for $g \in \ell^\infty(\mathbb{Z}_L \times \mathbb{Z}_n)$ it holds

$$g_{j,l} = \left(\mathfrak{F}_\varphi \mathfrak{F}_\varphi{}^{-1} g\right)_{j,l} \qquad \textit{for all } j \in \mathbb{Z}_L, l \in \mathbb{Z}_n.$$

Proof. Computation leads to

$$\begin{aligned}
\left(\mathfrak{F}_\varphi{}^{-1}\mathfrak{F}_\varphi f\right)_k &= \frac{1}{c_k} \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\,:\,j+lh=k} \left(\mathfrak{F}^{-1}\left(\left(\mathfrak{F}_\varphi f\right)_{i,l}\right)_{i=0}^{L-1}\right)_j \\
&= \frac{1}{c_k} \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\,:\,j+lh=k} \left(\mathfrak{F}^{-1}\left(\left(\mathfrak{F}\left(f_{m+lh}\varphi_m\right)_{m=0}^{L-1}\right)\right)\right)_j \\
&= \frac{1}{c_k} \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\,:\,j+lh=k} f_{j+lh}\varphi_j \\
&= \frac{1}{c_k} \sum_{(j,l)\in\mathbb{Z}_L\times\mathbb{Z}_n\,:\,j+lh=k} f_k\varphi_j \\
&= f_k.
\end{aligned}$$

The other equality can be proved analogously. □

3.1.2 Dimensionality reduction in signal separation

The spectrogram of a signal introduced in the previous section can be interpreted as a data set in a high-dimensional space $\mathbb{R}^{\frac{L}{2}}$. Each column of the spectrogram is one data point in that space. On this data set we wish to apply dimensionality reduction tools as motivated before.

This section is concerned with dimensionality reduction in signal separation. The core assumption for the application of dimensionality reduction in signal separation is that only some frequencies are necessary to represent the main content of a signal. By reducing the dimension of the frequency range, we only keep some frequency information while the rest is discarded.

One of the objectives of this work is to combine and compare the interaction of dimensionality reduction with different decomposition techniques (compare Section 3.1.3). Recall that a core requirement for the combination of dimensionality reduction and NNMF is the non-negativity preservation beyond the reduction.

To carry out the dimensionality reduction step we use the *Matlab Toolbox for Dimensionality Reduction* [118] from van der Maaten. This toolbox includes 34 different dimensionality reduction techniques but no specific non-negativity preserving ones, for more information see [119]. Using our approach from Chapter 2, we transfer methods of the toolbox into non-negativity preserving ones.

As we have already introduced and discussed different dimensionality reduction techniques in Chapter 2, we will now concentrate on the 'back-lifting' of the data to the high-dimensional space. This can be interpreted as something like an 'inverse dimensionality reduction'. Since we have thrown away a lot of information when reducing the data's dimension, we will not be able to really *invert* the reduction. But as we have already discussed in Remark 2.7, we have some results in this sense for PCA. If the high-dimensional data is indeed lying in a d-dimensional subspace of $\mathbb{R}^D$, we have shown that the inverse PCA mapping is given by $P^{-1} = V_d$ since this is an exact left-inverse for $P = V_d{}^T$ on $V_dV_d{}^T\mathbb{R}^D$, i.e., $V_dV_d{}^Tx = x$ for all $x \in V_dV_d{}^T\mathbb{R}^D$. Furthermore, we

have shown (compare Section 2.4.1) that for small perturbations this particular choice of an approximate left-inverse still ensures the applicability of Theorem 2.38 and thus, P still fulfills the cone condition. However, the perturbations are smoothed out as the back-lifted data is always lying inside the subspace $V_d V_d{}^T \mathbb{R}^D$.
These considerations justify the use of PCA for dimensionality reduction in signal separation. In contrast, we will not use MDS since the centering of the data makes an application of this method in the context of NNMF impossible, except for the special situation discussed in Section 2.4.2 which is rarely happening in practice.
The inversion of non-linear dimensionality reduction methods is far more difficult than for liner techniques. However, in both cases the inversion of the rotation of the low-dimensional data set (which guarantees the non-negativity of the data) is not problematic. Since a rotation matrix $R \in SO(d)$ is orthogonal, it has a natural inverse, namely R^T.

Inverting non-linear dimensionality reduction

When it comes to the application of non-linear non-negative dimensionality reduction in signal separation, the computation of an approximative left-inverse is a serious issue since in general this is not at hand. To overcome this shortcoming, the original data and its low-dimensional representation can be used to obtain a high-dimensional reconstruction of the separated low-dimensional data sets. This is the typical problem one is confronted with in scattered data approximation [62].
One possible approach is to use interpolation by radial functions [63]. In the following let us briefly sketch this approach using [86] as a reference. So far, we have that the data points $x_i \in \mathbb{R}^D$ are embedded in $\mathbb{R}^d$ via the non-linear discrete mapping $P\colon X \to \mathbb{R}^d$ with $P(x_i) = y_i$. As the non-linear mapping P is usually only defined on the discrete data points x_i for $i = 1, \ldots, n$, an inverse mapping P^{-1} is for now - if at all - given only on the discrete data. For the 'back-lifting' of the separated data sets we seek an extension of the inverse mapping to $\mathbb{R}^D$. If we assume that there is an underlying homeomorphism (or more general a continuous operator) $\mathcal{B}\colon \mathcal{M} \to \mathbb{R}^d$ with $\mathcal{B}|_X = P$, we aim to find an approximative left-inverse $\mathcal{B}^{-1}$ of this homeomorphism with $\mathcal{B}^{-1}\colon \mathcal{B}(\mathcal{M}) \to \mathbb{R}^D$ and $\mathcal{B}^{-1}(y_i) = x_i$.
In signal separation, this can be applied if we assume the spectrograms of the source signals to lie on the same manifold as the spectrogram of the mixture. In this situation the separated low-dimensional representation of the source signals can be lifted back to the high-dimensional space via $\mathcal{B}^{-1}$.
This approximation of the inverse mapping can be done in several ways (e.g. [33, 71]). However, we rely on the above mentioned paper which uses a *radial interpolation function*.
Let us introduce radial and positive definite functions in order to define the interpolant of a function $f\colon \mathbb{R}^d \to \mathbb{R}$. Each component of the function $\mathcal{B}^{-1}\colon \mathbb{R}^d \to \mathbb{R}^D$ will then be approximated by such an interpolant. A function $\phi\colon \mathbb{R}^d \to \mathbb{R}$ is called *radial* if there exists a function $\varphi\colon [\,0, \infty\,) \to \mathbb{R}$ such that $\phi(y) = \varphi(\|y\|_2)$ for all $y \in \mathbb{R}^d$. A radial function ϕ defines a bivariate function $\Phi\colon \Omega \times \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^d$ with $\Phi(y, c) = \phi(y - c)$.

Obviously, for a fixed $c \in \Omega$ the function Φ is a radial function centered in c.

Definition 3.11. A continuous function $\Phi\colon \Omega \times \Omega \to \mathbb{R}$ is called *positive definite on* Ω if for all $n \in \mathbb{N}$, all pairwise distinct points in $Y = \{y_1, \dots, y_n\} \subset \Omega$, and all $\alpha \in \mathbb{R}^n \setminus \{0\}$ we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \Phi(y_i, y_j) > 0.$$

Then, the matrix $K = (\Phi(y_i, y_j))_{i,j=1,\dots,n}$ of function values is positive definite.

Definition 3.12. For a positive definite radial function $\Phi\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and centers $Y = (y_1, \dots, y_n)$ with corresponding evaluations $f_Y = (f(y_1), \dots, f(y_n))$ we call

$$s_{f,Y}(y) = \sum_{i=1}^{n} \alpha_i \Phi(y, y_i)$$

the *radial interpolation function*, where the weights $\alpha = (\alpha_1, \dots, \alpha_n)$ are defined by solving

$$\alpha K = f_Y. \tag{3.1}$$

Remark 3.13. Due to the construction of the weights we have $s_{f,Y}(y_i) = f(y_i)$. Note, that the linear system in equation (3.1) is uniquely solvable since K is positive definite. Now, $\mathcal{B}^{-1}$ can be approximated by interpolating each coordinate in $\mathbb{R}^D$ with such a function. This can be done simultaneously by solving first the system

$$AK = X$$

for $A = (\alpha_{ij})_{\substack{i=1,\dots,D \\ j=1,\dots,n}}$ which are the weights for the different coordinates. Then, for $y \in \mathcal{B}(\mathcal{M})$ the corresponding $x \in \mathbb{R}^D$ is computed as

$$\mathcal{B}^{-1}(y) = A\left(\Phi(y, y_1), \dots, \Phi(y, y_n)\right)^T = XK^{-1}\left(\Phi(y, y_1), \dots, \Phi(y, y_n)\right)^T.$$

Note that using A for the computation of $\mathcal{B}^{-1}(y)$ is more efficient than using K^{-1} as it does not involve a direct inversion of any matrix.
There are several results on the error estimation for radial interpolation functions but this was not the objective of this work. Thus, we refer to [129] for further details on this topic. For the application of Isomap in our separation procedure we will use this technique for interpolating in the high-dimensional frequency space. A commonly used positive definite radial function is the Gaussian

$$\Phi(y_i, y_j) = e^{-\epsilon^2 \|y_i - y_j\|_2^2}, \qquad \text{with } \epsilon > 0$$

but we use a cubic function proposed in [86] for inverting Laplacian Eigenmaps:

$$\Phi(y_i, y_j) = \|y_i - y_j\|_2^3.$$

Other positive definite radial functions can be found in [17].
This is a very rough approach to the search of an approximative left-inverse of $\mathcal{B}$ which is still an open topic. Recall that we assumed the separated low-dimensional data to be lying in $\mathcal{B}(\mathcal{M})$ which is kind of audacious. Nevertheless, we will try this approach but we will see that the results are not satisfactory.

3.1.3 Decomposition techniques

In Figure 3.4, three frequency components of the spectrograms of a cymbal (red) and a castanet (blue) recording are depicted. It can be seen that the two data sets clearly separate into different clusters. This justifies the assumption that different signals can be distinguished by their frequency representation. From this heuristic the idea arises that a mixture of ρ signals can be separated by decomposing its frequency representation.

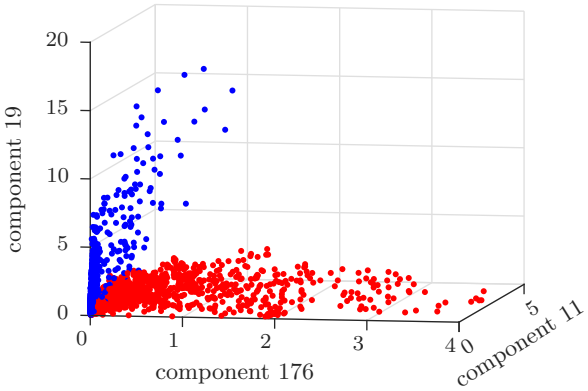


Figure 3.4: Three arbitrary components of the high-dimensional spectrograms of a cymbal (red) and a castanet (blue) recording.

To decompose a spectrogram $Y \in \mathbb{R}^{d \times n}$ we assume a linear mixing of the sources, i.e.,

$$Y = AS$$

for a *mixing matrix* $A \in \mathbb{R}^{d \times \rho}$ and a *source matrix* $S \in \mathbb{R}^{\rho \times n}$. This problem is highly under-determined since both, A and S are unknown. There are several possibilities to further restrict the problem in order overcome this limitation.
In the following, let us discuss two approaches, namely independent component analysis and non-negative matrix factorization.

Independent Component Analysis - ICA

The problem of *independent component analysis (ICA)* was first proposed and so named by Herault and Jutten in [55] around 1986 because of its similarities to PCA.
ICA is a stochastical method for decomposing a given data set into a set of statistically (i.e., mutually) independent components. This statistical independence can be achieved by maximization of the non-Gaussianity or equivalently by minimization of the mutual information. This ansatz is based on the Central Limit Theorem [101] which states that the distribution of a sum of independent random variables tends to a Gaussian distribution.
The core idea of ICA is to make some statistical assumptions on the source signals in order to balance the disproportion of equations and unknowns in the BSS problem. In

concrete terms we assume the signals to be statistically independent. This does not need to be completely true in practice [60].
To give a mathematical formulation of the just explained situation, we follow [27] and consider d weighted sums $y_1, \ldots, y_d$ of ρ source signals $s_1, \ldots, s_\rho$ called independent components. The discrete functions in time s_j and y_i can be interpreted as the realization of random variables which leads to the following linear statistical model:

$$\mathcal{Y} = A\mathcal{S}, \tag{3.2}$$

where $\mathcal{Y}$ and $\mathcal{S}$ are random vectors with values in $\mathbb{R}^d$ and $\mathbb{R}^\rho$ respectively and $A \in \mathbb{R}^{d\times\rho}$. The components of the vector $\mathcal{S}$ are maximizing a 'contrast' function. The contrast of a random vector is maximal if its components are statistically independent. Thus, the ICA of a random vector consists of searching a linear transformation such that the statistical dependence between its components is minimized.
Given n realizations of the random vector $\mathcal{Y}$, equation (3.2) becomes $Y = AS$ where we aim to estimate both, the mixing matrix A and the corresponding realizations S of $\mathcal{S}$. This can be done by minimizing the mutual information.
The *mutual information* is a distance measure between the density function $p_\mathcal{S}$ of a random vector $\mathcal{S}$ and the product of its marginal densities $\prod_{i=1}^d p_{\mathcal{S}_i}$. The random variables $\mathcal{S}_i$ are stochastically independent if and only if this distance vanishes. Thus, minimizing the mutual information will lead to a maximally independent set of random variables $\mathcal{S}$ and to the mixing matrix A. From A and Y then S can be computed.
Consequently, the remaining task is to specify an appropriate distance measure. Usually, ICA relies on the *Kullback-Leibler divergence*

$$\delta(p_\mathcal{X}, p_\mathcal{Z}) = \int p_\mathcal{X}(x) \ln\left(\frac{p_\mathcal{X}(x)}{p_\mathcal{Z}(x)}\right) \mathrm{d}x \tag{3.3}$$

introduced in 1951 by Kullback and Leibler [70]. The Kullback-Leibler divergence is not a true metric. However, it is $\delta(p_\mathcal{X}, p_\mathcal{Z}) = 0$ if and only if $p_\mathcal{X} = p_\mathcal{Z}$.
As in application $p_\mathcal{S}$ and $p_{\mathcal{S}_i}$ are typically unknown, we need to estimate these quantities from the realizations Y of the random vector $\mathcal{Y}$. One possible estimation can be computed via the *Edgeworth expansion* for $p_\mathcal{S} = p_{A^{-1}\mathcal{Y}}$, where A^{-1} is a pseudo inverse of A (for a derivation of this expansion see [66]). For more information on ICA compare [27, 68].
For the numerical implementation we rely on the *Joint Approximate Diagonalization for Eigenmatrices (JADE)* algorithm developed by Cardoso (see [19] for the code). This algorithm is based on Givens rotations which are subsequently applied until a solution is reached. For details we refer to [21].

Non-Negative-Matrix Factorization

The ICA approach uses stochastical assumptions about the source signals. In order to decompose the reduced spectrogram Y the data is modeled as a random process, i.e., the data matrix Y is interpreted as n realizations of a random vector. However, signals

(especially high-energy signals) are very often deterministic and therefore, using a non-statistic based separation method could improve the results. Moreover, ICA decomposes into components that are not necessarily non-negative. Thus, in practice, the point-wise absolute value is often taken. In [54] Helén and Virtanen stated that an NNMF approach to signal separation outperforms ICA, at least for drum tracks. However, dimensionality reduction in combination with NNMF was not studied therein.
Starting again from the problem of decomposing a given data set Y into a mixing matrix $A \in \mathbb{R}^{d\times\rho}$ and the source signals or source components $S \in \mathbb{R}^{\rho\times n}$, i.e., $Y = AS$, the task is to find a matrix factorization of Y. The time-frequency data we use for signal separation is non-negative, so that we have the additional information that Y is non-negative. Moreover, we would like the extracted source components S to be spectrograms and therefore, we want S to be likewise non-negative. Due to these facts *non-negative matrix factorization (NNMF)* seems to be a promising ansatz.
NNMF has its origin in the 1970s but was back then used in a completely different context. In 1999 NNMF has been reconsidered by Lee and Seung in [72] who developed efficient algorithms and established the name. The first to apply NNMF to audio signals have been Smaragdis and Brown in [108]. In recent years, this ansatz has been used successfully in the context of signal separation by many working groups (see e.g. [34, 37, 54, 91, 122]).
NNMF computes a factorization of Y

$$Y \approx AS$$

by minimizing an error function depending on Y and AS under the constraint that A and S are non-negative. As error function, a norm (e.g. $\|Y - AS\|_F$) or another measure (e.g. a divergence) can be chosen. We use the normalized discrete Kullback-Leibler divergence

$$\hat{\delta}(Y, AS) = \sum_{i=1}^{d}\sum_{j=1}^{n}\left(Y_{ij}\ln\left(\frac{Y_{ij}}{(AS)_{ij}}\right) - Y_{ij} + (AS)_{ij}\right).$$

Note that for $\sum_{i,j} Y_{ij} = \sum_{ij}(AS)_{ij} = 1$ this is the discrete Kullback-Leibler divergence (compare equation (3.3)).
In order to minimize this measure, we perform the multiplicative update algorithm proposed in [72] which is based on a gradient descent method. For the implementation we rely on *NMFlib* [45], a Matlab toolbox provided by Grindlay.

Remark 3.14. NNMF can also be used for dimensionality reduction if the dimensions of the factor matrices are chosen accordingly (compare e.g. [90, 116]).

Independent Subspace Analysis - ISA

A commonly used enhancement in the decomposition step is *independent subspace analysis (ISA)*. This concept was introduced in [58] by Hyvärinen and Hoyer as an extension for ICA but recently, similar clustering techniques have also been combined with other decomposition tools such as NNMF.

Classical decomposition methods are based on the assumption that the number of sources is known. In practice, this is not always the case and therefore the extraction of sources of a data set Y might be inaccurate. As a consequence it could happen that we detect more components as the true number of sources. In this case, two or more of the separated components belong to the same source. In other words, the source is contained in the subspace spanned by these components.
The general proceeding in ISA is to first extract some source components of a given data set Y using ICA or NNMF. In a second step these components are grouped (or partitioned) into subspaces, each one corresponding to a source. Finally, the sources are reconstructed from these multi-component subspaces (see [22, 54, 58, 64]).
Thus, ISA can be seen as an upgrade of ICA or NNMF which partitions the different components into groups, each of which is spanning a subspace. This procedure avoids the above-explained problem of extracting more sources as really underlying. The main difficulty in the concept of ISA - beside the decomposition itself - is to identify the components that belong to the same multi-component subspace. This can be done by some type of grouping and is not an objective of this thesis. The 'independent' in the denomination of ISA comes from its first appearance when it was designed for ICA.
Let us formalize these considerations. For a given data set $Y = ({\eta_1}^T, \dots, {\eta_d}^T)^T \in \mathbb{R}^{d\times n}$ we suppose as before each row $\eta_i \in \mathbb{R}^{1\times n}$ to be the weighted sum of ρ independent components $\sigma_j \in \mathbb{R}^{1\times n}$:

$$\eta_i = \sum_{j=1}^{\rho} a_{ij}\sigma_j = a_i S$$

where $S = ({\sigma_1}^T, \dots, {\sigma_\rho}^T)^T \in \mathbb{R}^{\rho\times n}$ and $A = ({a_1}^T, \dots, {a_d}^T)^T \in \mathbb{R}^{d\times\rho}$. Remark that now we consider rows η_i of Y, while before we denoted the columns by y_l. The unknown matrices S and A can be estimated with a decomposition method. Note that ρ can be chosen such that $\mathrm{rk}(S) = \rho$. Different as before, at this point we do not assume the σ_j to be the sources of the mixed signal. More precisely, we have c source signals, where $c \leq \rho$.
The core idea of ISA is that each source is a linear combination of σ_j, $i = 1, \dots, \rho$. Assume that each σ_j corresponds to only one of the c different unknown sources. Then, the ρ-dimensional subspace U spanned by the σ_j is the internal direct sum of subspaces U_k, each associated to one of the sources. Hence, we get a partition of $Z = \{\sigma_1, \dots, \sigma_\rho\}$

$$Z = \bigcup_{k=1}^{c} Z_k, \quad Z_k \cap Z_j = \emptyset \ \text{ for all } k \neq j.$$

This leads simultaneously to a partition of the index set $I = \{1, \dots, \rho\}$ into sets I_k. Each of the collections of components Z_k defines a matrix $S_k \in \mathbb{R}^{\rho_k\times n}$ whose rows are the ρ_k components belonging to the kth source $S_k = ((\sigma^k{}_1)^T, \dots, (\sigma^k{}_{\rho_k})^T) = ({\sigma_i}^T)_{i\in I_k}$.
This partition of Z yields a decomposition $Y = \sum_{k=1}^{c} Y_k$ with

$$Y_k = A_k S_k,$$

where $A_k = (a_{ij})_{i=1,\dots d,\ j\in I_k} \in \mathbb{R}^{d\times \rho_k}$ is a submatrix of A obtained by deleting some columns and only keeping the columns corresponding to the index set I_k.
So far, we reviewed how to reconstruct the data sets Y_k from Y, where Y_k represents the kth source, provided S and an adequate partition has been found. However, the main difficulty in the concept of ISA is to identify this partition. This can be done by some type of grouping. For our numerical results we used the grouping method introduced by Casey and Westner in [22] combined with a manual refinement. This method is based on calculating the similarities of the components σ_j and sorting them by using their pairwise dissimilarities.
The dissimilarity measure used in [22] is again based on the *Kullback-Leibler divergence* (see equation (3.3)) which is a dissimilarity measure for density functions. To apply this, we assume again that the extracted components are realizations of a random variable. The idea is to combine those components whose generating random variables are most similar. Since the probability densities of the underlying variables $\mathcal{S}_j$ are unknown, we need to estimate those from the realizations σ_j. As for ICA, this can be done by the Edgeworth expansion involving the central moments of the random variable. For a rigorous derivation compare [68].
Due to this particular choice of the dissimilarity measure, this grouping is especially suitable for ICA but it can be likewise applied to components extracted by NNMF. The obtained dissimilarity values are the basis for the clustering algorithm relying on a cost functional proposed by Hofmann and Buhmann in [56].

Remark 3.15. Our algorithm only separates the amplitude spectrogram and not the phase spectrogram. As both are needed for the ISTFT, we naively use for all components the phase of the recorded mixture in the reconstruction step.

3.2 Numerical examples

In this section, we will consider three different examples of mixed high-energy audio tracks in order to study the performance of the proposed separation procedure. Before introducing the examples, let us discuss the quality measures we will use. Unfortunately, it will turn out that we are not aware of any adequate error indicator to quantify the deviation of the separated sources s_i to the original sources f_i.

L^∞-error

The L^∞*-error* in the time-amplitude space is the maximum in time of the difference of the amplitudes of f_i and s_i:

$$\mathrm{err}_\infty(f_i, s_i) = \max_k |f_i(t_k) - s_i(t_k)|.$$

For audio signals, the maximal value of the L^∞-error is usually 2 as these signals are bounded between -1 and 1.
For the reconstruction we only used the phase of the original mixture instead of separating also the phase spectrogram. Due to this fact, the extracted sources can have

a different phase than the original sources which typically causes an increase of the L^∞-error.
Nevertheless, this error can be used to quantify the effect of the dimensionality reduction. If we just reduce the dimension and lift the data back to the high-dimensional space, i.e., we simply sum up the separated sources, we get a reconstruction of the mixture. This reconstruction does not suffer from a possible phase shift as the phase should still be the same. Thus, with the L^∞-error the reconstructed mixture can be compared to the original signal.

Signal to Noise Ratio - SNR

In signal processing the *signal to noise ratio (SNR)* is another frequently used quality measure. It estimates the portion of noise in a signal measured in decibel. Here, the error $f_i(t_k) - s_i(t_k)$ can be considered as noise and thus, the SNR error [44] is given by

$$\mathrm{err}_{SNR}(f_i, s_i) = 10 \log_{10} \left(\frac{\sum_k |f_i(t_k) - s_i(t_k)|^2}{\sum_k |f_i(t_k)|^2} \right).$$

For the SNR-error it holds: the smaller the error, the less the noise and the better the reconstruction. Note that the SNR is sometimes also defined using the reciprocal of the logarithm's argument. This error measure can also suffer from a possible phase shift. Nevertheless, we will use these measures to compare the results due to the lack of more sophisticated ones.

3.2.1 Examples

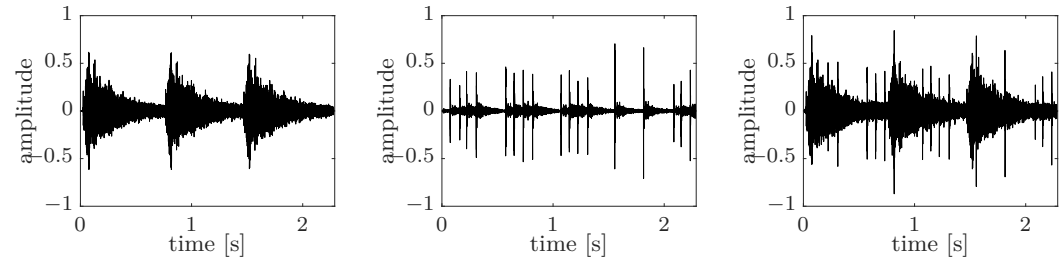
We will now briefly present the examples we will use for showing the separation qualities of the algorithm in Section 3.2.2.

Example 1

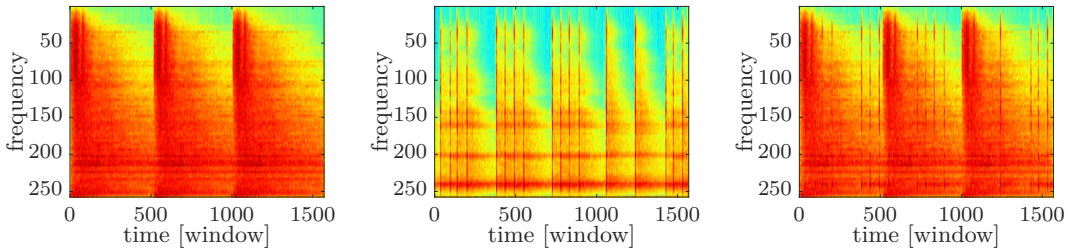
The first example is a short sequence of a mixture of a cymbal's and a castanet's recording (see Figure 3.5). The first signal consists of three strokes of the cymbal, where each stroke has a relatively slow decay behavior in time. These rapid oscillations superpose the short clicks of the castanets quite considerably. In particular, the clicks coinciding with the stroke of the cymbal are not distinguishable from the rest of the mixture (see Figure 3.5a).
From the separation algorithm we expect that the reconstruction of the cymbal will not be too noisy whereas we consider the extraction of the castanets as a challenge.

Example 2

As a second example we consider again a mixture of two sources, a base drum and a finger flipping (see Figure 3.6). As before, one of the sources is active during longer periods (drum) than the other (finger flips). In the frequency representation we observe

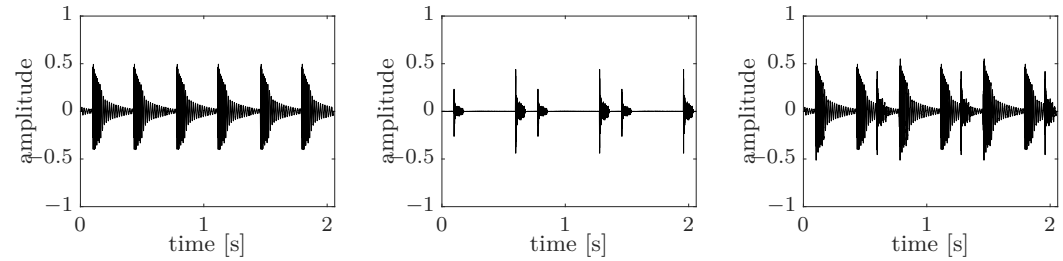


(a) Time-amplitude plots of cymbal, castanets and the mixture (from left to right).

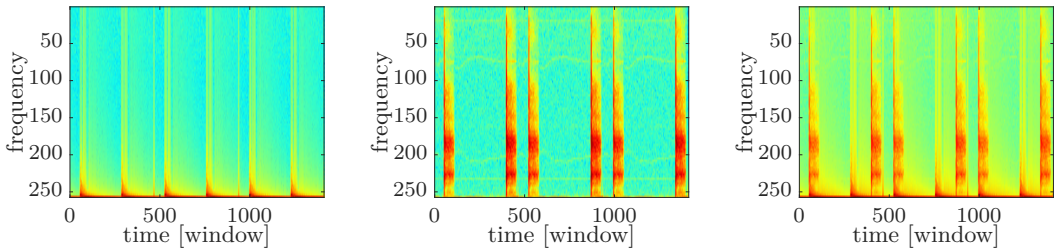


(b) Time-frequency plots of cymbal, castanets and the mixture (from left to right).

Figure 3.5: Signals of Example 1.

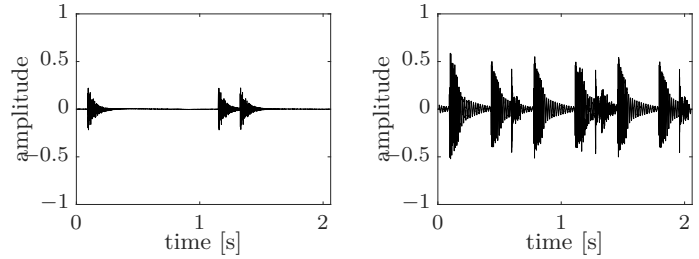


(a) Time-amplitude plots of base drum, finger flips and the mixture (from left to right).

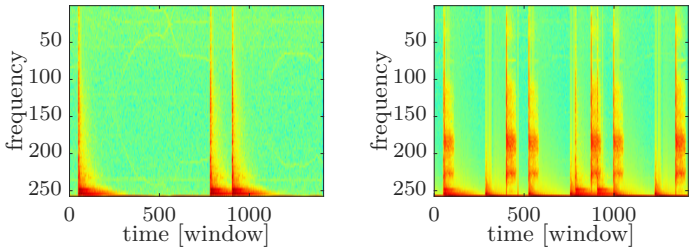


(b) Time-frequency plots of base drum, finger flips and the mixture (from left to right).

Figure 3.6: Signals of Example 2.



(a) Time-amplitude plots of bongo and the mixture of base drum, bongo and finger flips (from left to right).



(b) Time-frequency plots of bongo and the mixture of base drum, bongo and finger flips (from left to right).

Figure 3.7: Signals of Example 3.

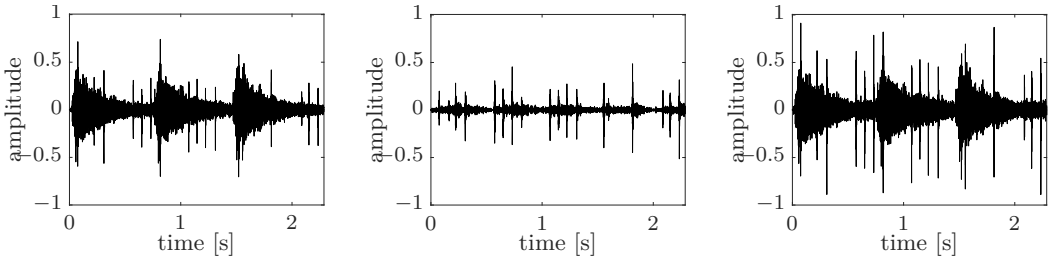
likewise the superposition of some base drum beats by the broader finger flip. However, the frequency behavior differs from the previous example as the frequency activity is sparser and also the occurrence in time is shorter.
This example is meant to analyze the performance of the algorithm in the case that the characteristics of the sources are not that well distinct and that their spectrograms are more similar.

Example 3

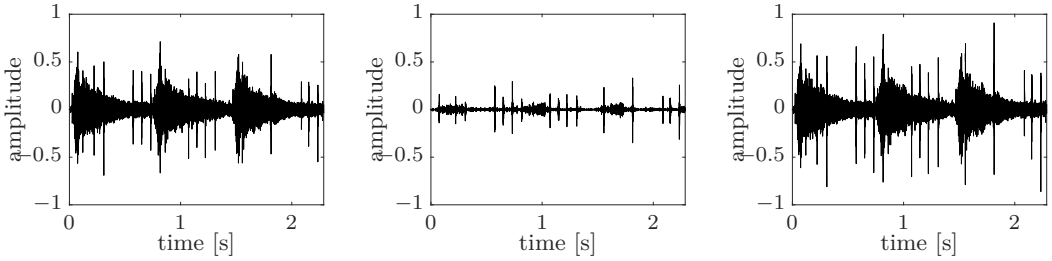
In the third example, we consider the mixture of three sources. We use again the base drum and the flip of Example 2 (see Figure 3.6) and combine these with a bongo sound (see Figure 3.7). The spectrogram of the bongo is quantitatively the same as the spectrogram of the base drum. Comparing this mixture with the one from Example 2 the bongo sound is only visible if we are looking for this difference. Thus, this example is another level of difficulty not only due to the higher number of sources but also due to the severe overlay.

3.2.2 Results

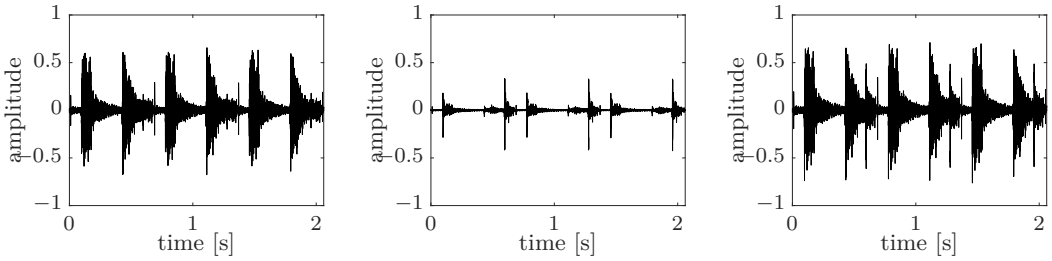
Let us first compare the performance of the separation algorithm when using NNPCA & NNMF and PCA, entry-wise absolute value & NNMF (referred to as PCA & NNMF in the following). The latter is a naive alternative to NNPCA & NNMF in order to guarantee non-negative entries in the low-dimensional data. For the comparison, we



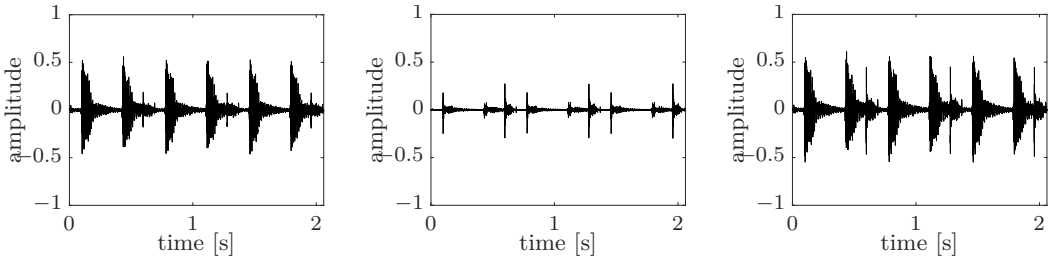
(a) By NNPCA & NNMF extracted sources and reconstructed mixture from Example 1 (from left to right).



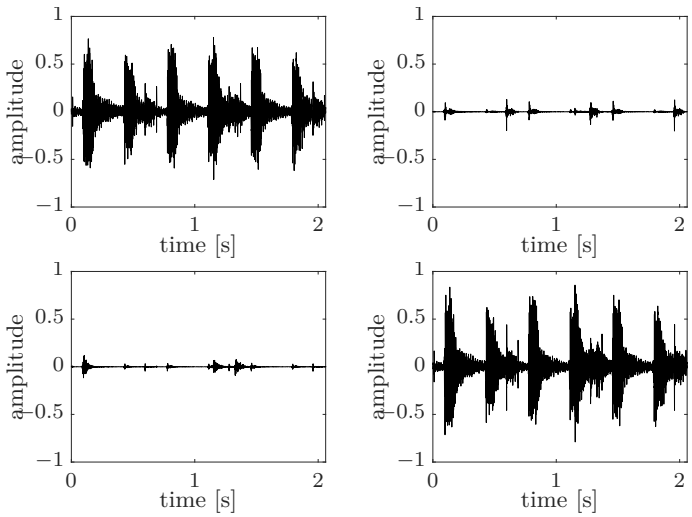
(b) By PCA with entry-wise absolute value and NNMF extracted sources and reconstructed mixture from Example 1 (from left to right).



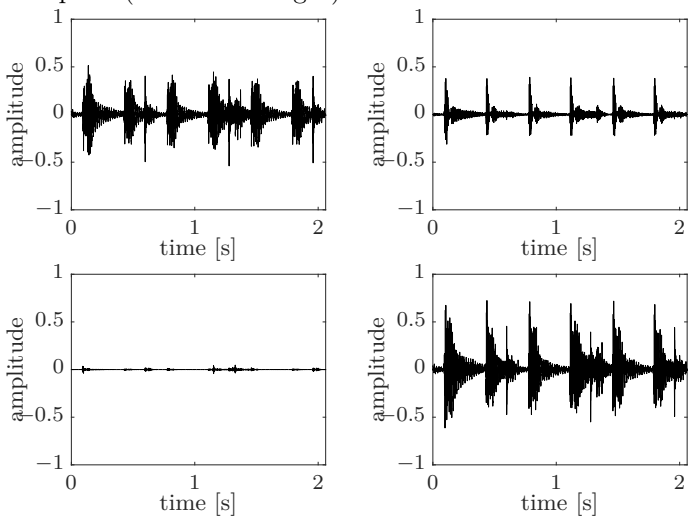
(c) By NNPCA & NNMF extracted sources and reconstructed mixture from Example 2 (from left to right).



(d) By PCA with entry-wise absolute value and NNMF extracted sources and reconstructed mixture from Example 2 (from left to right).



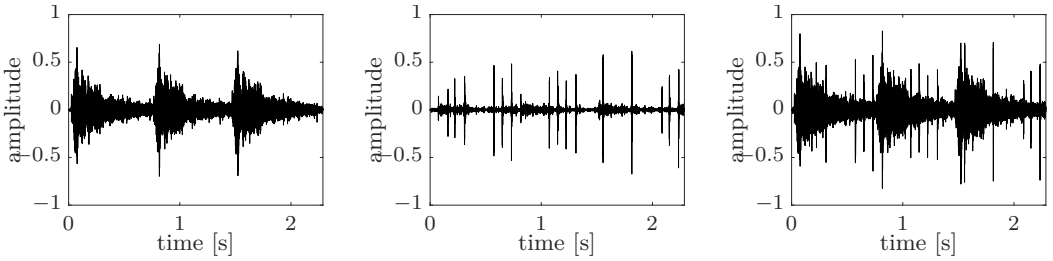
(e) By NNPCA & NNMF extracted sources and reconstructed mixture from Example 3 (from left to right).



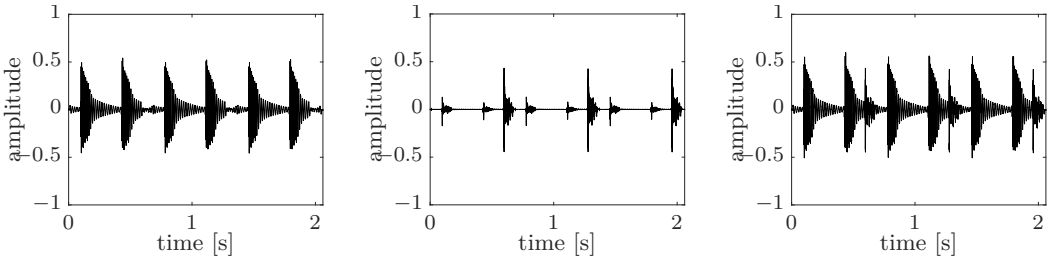
(f) By PCA with entry-wise absolute value and NNMF extracted sources and reconstructed mixture from Example 3 (from left to right).

Figure 3.9: Results of all three examples for the reduction to 10 dimensions by NNPCA or PCA and decomposition by NNMF.

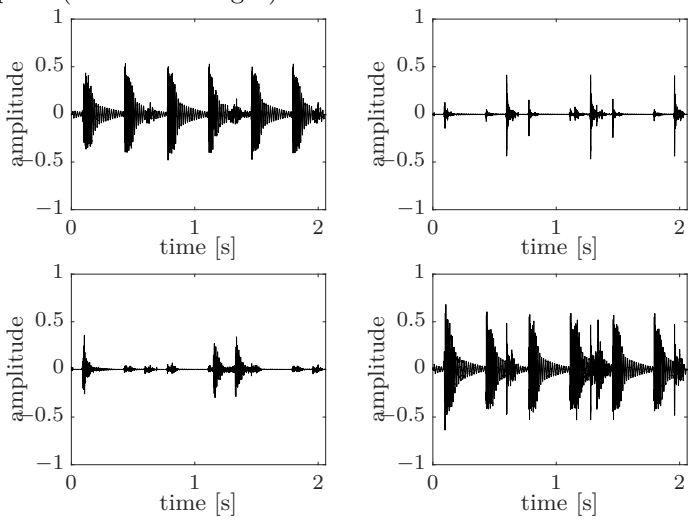
reduce the spectrograms of the mixture of sources from 256 frequency samples to only 10 and apply the separation algorithm. The results are depicted in Figure 3.9.
From the figure we can see that the algorithm with NNPCA & NNMF is able to detect the source signals for all three examples. The algorithm with PCA & NNMF does not perform that well. Surprisingly, also the well-hidden peaks of the castanets (Figures 3.8a and 3.8b) and of the finger flips (Figures 3.8c and 3.8d) can be located. With NNPCA & NNMF the bongo and the finger flips (Figure 3.9e) are detected, whereas with PCA & NNMF (Figure 3.9f) this is not the case. Especially for the finger flips not even the rhythm is recognized and thus, the detection of the sources can be said to have failed. This shows that our approach outperforms the naive approach.
Furthermore, we compare the combination of NNPCA & NNMF with PCA & ICA. The results are shown in Figure 3.10. We reduce again the dimension of the spectrograms from 256 to 10.
It can be seen that PCA & ICA also detects the source signals for all three examples. As before, also the well-hidden peaks of the castanets (Figure 3.10a), of the finger flips (Figure 3.10b) and of the bongo (Figure 3.10c) can be distinguished.
If we compare the reconstructed mixture (i.e., the sum of the extracted sources, Figures 3.9 and 3.10, last column) with the original input signal, we can hardly see any difference, especially for ICA. This justifies the application of dimensionality reduction in signal separation, as not too much information seems to be lost when reducing and back-lifting the spectrogram. In other words, the high-dimensional data is not too far away from a 10-dimensional subspace of $\mathbb{R}^{256}$.
We observe that some peaks have a different amplitude as in the original signal. This is due to the fact that frequencies are assigned to only one of the source signals. Consequently, if they occur in several signals at the same time instant, this is not captured. This also causes artifacts of one source signal in the other which can be observed for example in the cymbal's signal when separated by NNMF.
Moreover, PCA & ICA seems to produce better results than NNPCA & NNMF. This could be due to the fact that we consider high-energy signals whose frequencies follow a certain distribution. Nevertheless, NNMF can still be used for signal detection. Noteworthy, in Example 3 the bongo sound could be detected by PCA & ICA even though it was hardly visible in the mixed signal or in its spectrogram. This shows impressively the superior performance of ICA.
To further study these observations, let us reconsider the first example and perform all variants of the algorithm with a reduction to different dimensions. In Table 3.1 we compare the L^∞-error and the SNR for different techniques and dimensions.
The table is organized as follows. In the first block the results for NNPCA & NNMF with a reduction to 3, 10, 20 dimensions and without the reduction are listed and in the second block the same for PCA & ICA can be found. Moreover, we tested the combination of PCA with NNMF and NNPCA with ICA, the results are shown in the last blocks of Table 3.1. Note that the L^∞-error is displayed only for the reconstructed mixture, as this comparison does not suffer from the phase-shift.
From Table 3.1 it can be seen that reducing less (i.e., staying in a higher-dimensional space) and keeping more information does not necessarily lead to better results. Es-



(a) By PCA & ICA extracted sources and reconstructed mixture from Example 1 (from left to right).



(b) By PCA & ICA extracted sources and reconstructed mixture from Example 2 (from left to right).



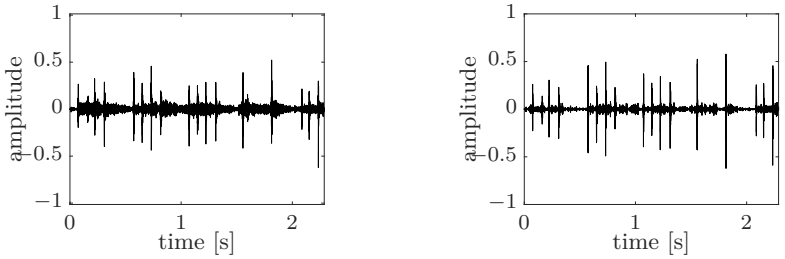
(c) By PCA & ICA extracted sources and reconstructed mixture from Example 3 (from left to right).

Figure 3.10: Results of all three examples for the reduction to 10 dimensions by PCA and decomposition ICA.

Table 3.1: Comparison of the results for Example 1. The length of the signal is 2.27s and the size of the spectrogram is 256×1569.

no. of comp.	comput. time	err_∞	SNR of recon. signal	SNR of cymbal	SNR of castanets
NNPCA & NNMF					
3	14.8s	0.57	-2.899	-2.339	-0.978
10	35.6s	0.55	-2.239	-1.262	-0.553
20	248s	0.68	-2.170	-1.942	-0.237
all	115min	0.014	-36.576	-3.948	3.024
PCA & ICA					
3	2.3s	0.23	-8.661	-7.424	-5.016
10	8s	0.23	-10.524	-8.660	-4.574
20	10.7s	0.21	-12.517	-9.765	-4.647
all	∞				
PCA & absolute value & NNMF					
3	3.6s	0.44	-4.949	-4.066	-2.992
10	13.1s	0.48	-4.708	-3.430	-2.149
20	23.2s	0.47	-4.367	-3.184	-2.313
NNPCA & ICA					
3	8s	0.70	-3.212	-2.421	-1.114
10	20s	0.58	-2.227	-1.368	-1.888
20	201s	0.64	-2.207	-2.082	-0.109

pecially for the castanet reconstruction, we see that the SNR value is actually getting better for fewer components.
To substantiate this observation we depicted the extracted castanet signals in Figure 3.11 where the dimension of the spectrogram was reduced to 3 dimensions. Remarkably, also for this drastic reduction, the time locations of the castanets are well-detected and for the PCA & ICA result, there is hardly any difference to the 10-dimensional case.
From Table 3.1 it also becomes apparent that both quality measures (L^∞-error and SNR) are not appropriate for these examples. The L^∞-error is very high if the dimension is reduced to 10, even if the resulting source signals seem not to be that different from the originals (compare Figures 3.8a and 3.10a). In the case of NNMF, this can be partly explained by the not exact approximation of $Y \approx AS$. Not surprisingly, the NNMF separation without reduction yields a very small L^∞-error.
Qualitatively, the same is true for the SNR values. As expected, the SNR value for the reconstructed signal is in all cases better than the one of the extracted sources. The best results in the SNR sense are clearly obtained for the extraction of the cymbal with reduction to 20 dimensions for PCA & ICA.
The errors for NNPCA & NNMF (third block of Table 3.1) do not mirror the observations in Figure 3.9. More precisely, the SNR values suggest a better performance of the naive approach while the pictures clearly show the contrary.



(a) NNPCA & NNMF. **(b)** PCA & ICA.

Figure 3.11: Extracted castanets signals of Example 1 from a reduction to 3 dimensions.

Concerning the computational time, we have to admit that the NNPCA & NNMF version of the algorithm is more expensive. This is due to the computation of the rotation matrix when performing NNPCA. We also observe that ICA cannot be used without dimensionality reduction due to high storage requirements, at least if carried out by the JADE algorithm. NNMF itself however, shows quite good behavior even if it is also expensive.

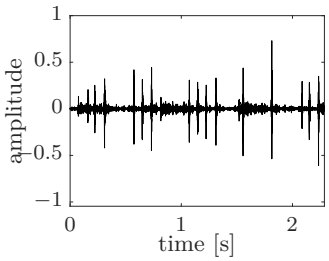
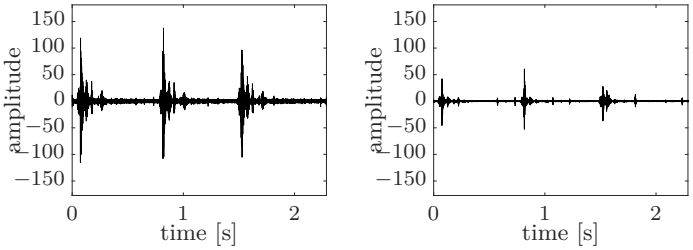


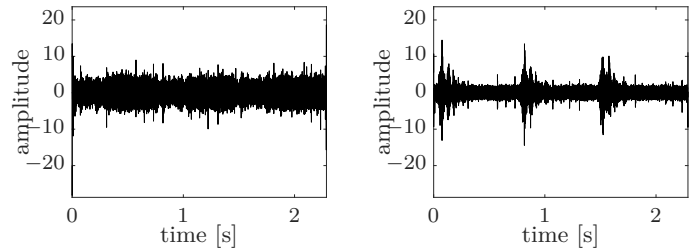
Figure 3.12: By NNPCA & ICA extracted castanets signals of Example 1 from a reduction to 10 dimensions.

Furthermore, we have tested the combination of NNPCA & ICA and the results (compare Table 3.1) seem competitive with NNPCA & NNMF. In Figure 3.12, the extracted castanet sound is depicted and it can be seen that the extraction of the castanets is almost as good as for PCA & ICA. This is another hint telling us that the SNR might not be an appropriate measure.
When applying Isomap and using the naive kernel approach described in Section 3.1.2 the results are useless (see Figure 3.13). No matter for which, NNMF or ICA, we are not even able to decide which of the extracted sources corresponds to the castanets and which to the cymbal. Moreover, the amplitude scaling of the sources is completely nonsense due to a bad spectrogram reconstruction. Here, a more sophisticated inverse reduction map is needed.

In summary, we can say that our approach to signal separation is able to detect peaks of the unknown source signals, even if they are well hidden. Furthermore, different sources



(a) Isomap with entry-wise absolute value and NNMF.



(b) Isomap & ICA.

Figure 3.13: Extracted source signals of Example 1 from a reduction by Isomap to 10 dimensions.

can be detected also if the original signals were quite similar. When NNMF is used for decomposition we improved the results by applying our NNPCA instead of PCA and entry-wise absolute value. Nevertheless, the combination of PCA or NNPCA with ICA outperforms NNMF.

4 Conclusion

This work was concerned with the improvement of algorithms in signal processing. More precisely, the involvement of dimensionality reduction in signal separation has been studied. The starting point for these studies was the observation that classical dimensionality reduction methods (in particular principal component analysis) cannot be used as a preprocessing step if combined with non-negative matrix factorization. Therefore, we provided a general framework for enhancing classical reduction techniques to non-negativity preserving ones. This framework consists of splitting the non-negative dimensionality reduction problem in two subproblems which are solved successively.
We formulated a condition under which a non-negative dimensionality reduction problem can be solved by this splitting approach. This condition restricts the class of possible dimensionality reduction tools in two ways. On the one hand the involved cost functional needs to be rotationally invariant, a requirement that is automatically fulfilled by many methods. On the other hand, the reduction map may not increase the angles between the data points, a restriction that is rarely satisfied. Nonetheless, this theory is provably applicable to PCA and MDS.
Furthermore, we discussed the numerical implementation of our signal separation procedure and in particular of the splitting approach which can be done in a smart way using the Lie group structure of the admissible set. Here, the exponential map plays an important role since it permits the generalization of a steepest descent method to Lie groups.
With this approach to signal separation we were able to improve our results from [48, 68]. Moreover, we compared the results of signal separation with NNPCA & NNMF with the results for PCA & ICA. Although NNMF without dimensionality reduction seems to be a promising alternative to ICA, it does not show this potential when combined with NNPCA. Nevertheless, we improved the performance of the combination of NNMF with dimensionality reduction. Possibly, the use of non-linear dimensionality reduction tools might lead to a better separation but the lack of sophisticated non-linear non-negative methods and a suitable approximative left-inverse made a comparison impossible.

Of course, there are several possibilities to further improve signal separation with dimensionality reduction concerning the implementation, the theory or the procedure itself.
Theory: In order to better understand the mechanism of the algorithm, one of the first steps is to find an appropriate measure for the separation error. The analysis of the interaction of the involved techniques could lead to a better understanding which can be used for improving the algorithm through a combination of more sophisticated tools. Furthermore, we have seen that there is a strong demand for non-linear non-negative dimensionality reduction methods and approximative inverses of the reduction maps for not being restricted to linear tools.

Procedure: The signal separation procedure includes a grouping when reconstructing the source signals from the extracted components. This is a fundamental issue which we have not considered so far. Additionally, the influence of other signal transforms (e.g. wavelet transform) to the quality of the separation could be studied.
Implementation: Last but not least, the implementation could be done in a more efficient way. We have seen, that the ICA algorithm we used is quite costly. Thus, one could think of including a FastICA algorithm instead. It could be also considered to use a Newton-like or conjugate gradient method when computing the rotation.

Bibliography

[1] T. Abrudan, J. Eriksson, and V. Koivunen. Conjugate gradient algorithm for optimization under unitary matrix constraint. *Signal Processing*, 89(9):1704–1714, 2009.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[3] G. I Allen and M. Maletić-Savatić. Sparse non-negative generalized PCA with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035, 2011.

[4] M. Asteris, D. S. Papailiopoulos, and A. G. Dimakis. Nonnegative Sparse PCA with Provable Guarantees. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1728–1736, 2014.

[5] D. Barry, E. Coyle, D. FitzGerald, and R. Lawlor. Drum source separation using percussive feature detection and spectral modulation. In *Proceedings of Irish Signals and Systems Conference*, pages 13–17, Dublin, Ireland, 2005.

[6] D. Barry, E. Coyle, and B. Lawlor. Sound Source Separation: Azimuth Discrimination and Resynthesis. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX-04)*, Naples, Italy, 2004.

[7] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591. MIT Press, 2001.

[8] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.

[9] R. Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

[10] E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *International journal of neural systems*, 10:1–8, 2000.

[11] R. B. Blackman and J. W. Tukey. *The measurement of power spectra: from the point of view of communications engineering*. Dover books on engineering and engineering physics. Dover Publications, Dover, 1959.

[12] D. Bollegala, G. Kontonatsios, and S. Ananiadou. A Cross-Lingual Similarity Measure for Detecting Biomedical Term Translations. *PLOS ONE*, 10(6), 2015.

[13] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab Toolbox for Optimization on Manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[14] G. E. Bredon. *Topology and Geometry*, volume 139 of *Graduate Texts in Mathematics*. Springer, New York, 1993.

[15] R. W. Brockett. Lie Algebras and Lie Groups in Control Theory. In D. Q. Mayne and R. W. Brockett, editors, *Geometric Methods in System Theory*, volume 3 of *NATO Advanced Study Institutes Series*, pages 43–82. Springer, Netherlands, 1973.

[16] G. J. Brown and M. Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994.

[17] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Number 12 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.

[18] K. Bunte, M. Biehl, and B. Hammer. A General Framework for Dimensionality-reducing Data Visualization Mapping. *Neural Compututation*, 24(3):771–804, 2012.

[19] J.-F. Cardoso. JADE. `perso.telecom-paristech.fr/~cardoso/guidesepsou.html`.

[20] J.-F. Cardoso. Blind signal separation: statistical principles. In *Proceedings of the IEEE*, volume 86, pages 2009–2025, 1998.

[21] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11:157–192, 1999.

[22] M. A. Casey and A. Westner. Separation of Mixed Audio Sources by Independent Subspace Analysis. In *Proceedings of the International Computer Music Conference*, pages 154–161, Berlin, 2000.

[23] D. Chen, J. C. Lv, and Z. Yi. A Local Non-Negative Pursuit Method for Intrinsic Manifold Structure Preservation. In *AAAI Conference on Artificial Intelligence*, 2014.

[24] J.-T. Chien and B.-C. Chen. A new independent component analysis for speech recognition and separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1245–1254, 2006.

[25] A. Cichocki, R. Zdunek, and S.-I. Amari. New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, 2006.

[26] C. E. Colin. Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, 25(5):975–979, 1953.

[27] P. Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314, 1994.

[28] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90):297–301, 1965.

[29] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*, volume 88 of *Monographs on Statistics & Applied Probability*. CRC Press, 2000.

[30] S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, and H. Yserentant. *Extraction of Quantifiable Information from Complex Systems*. Lecture Notes in Computational Science and Engineering. Springer International Publishing, 2014.

[31] L. Deng, K.-K. Cheng, J. Dong, J. L. Griffin, and Z. Chen. Non-negative principal component analysis for NMR-based metabolomic data analysis. *Chemometrics and Intelligent Laboratory Systems*, 118:51–61, 2012.

[32] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[33] A. Elgammal and C.-S. Lee. Nonlinear manifold learning for dynamic shape and dynamic appearance. *Computer Vision and Image Understanding*, 106(1):31–46, 2007.

[34] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis. *Neural Computation*, 21(3):793–830, 2009.

[35] D. FitzGerald, E. Coyle, and B. Lawlor. Sub-band Independent Subspace Analysis for Drum Transcription. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX-02)*, Hamburg, Germany, 2002.

[36] D. FitzGerald, E. Coyle, and B. Lawlor. Independent subspace analysis using locally linear embedding. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX-03)*, pages 13–17, London, UK, 2003.

[37] D. FitzGerald, M. Cranitch, and E. Coyle. Non-negative tensor factorisation for sound source separation. In *Proceedings of Irish Signals and Systems Conference*, pages 8–12, Dublin, Ireland, 2005.

[38] I. Fodor. A Survey of Dimension Reduction Techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.

[39] J. B. J. Fourier. *Théorie analytique de la chaleur*. Firmin Didot Père et Fils, Paris, 1822.

[40] D. Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[41] D. Gabor. Theory of communication. Part 3: Frequency compression and expansion. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):445–457, 1946.

[42] J. Gallier. Notes on Differential Geometry and Lie Groups. e-book, `http://www.seas.upenn.edu/~jean/diffgeom.pdf`, 9 May 2015.

[43] C. Geiger and C. Kanzow. *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer-Lehrbuch. Springer Berlin Heidelberg, 1999.

[44] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Education, 2011.

[45] G. Grindlay. NMFlib. `http://www.ee.columbia.edu/~grindlay/code.html`.

[46] K. Gröchenig. *Foundations of Time-Frequency Analysis*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2001.

[47] M. Guillemard, D. Heinen, A. Iske, S. Krause-Solberg, and G. Plonka. Adaptive Approximation Algorithms for Sparse Data Representation. In S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, and H. Yserentant, editors, *Extraction of Quantifiable Information from Complex Systems*, volume 102 of *Lecture Notes in Computational Science and Engineering*, pages 281–302. Springer International Publishing, 2014.

[48] M. Guillemard, A. Iske, and S. Krause-Solberg. Dimensionality Reduction Methods in Independent Subspace Analysis for Signal Detection. In *Proceedings of the 9th International Conference on Sampling Theory and Applications (SampTA)*, Singapore, 2011.

[49] M. Guillemard, A. Iske, and U. Zölzer. Geometric Data Manipulation with Clifford Algebras and Möbius Transforms. *Advances in Applied Clifford Algebras*, pages 1–12, 2015.

[50] B. C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222 of *Graduate Texts in Mathematics*. Springer, New York, 2003.

[51] H. Han. Nonnegative principal component analysis for mass spectral serum profiles and biomarker discovery. *BMC Bioinformatics*, 11(S-1), 2010.

[52] X. Han. Nonnegative Principal Component Analysis for Proteomic Tumor Profiles. In *Proceedings of the SIAM International Conference on Data Mining*, pages 269–280, USA, 2010.

[53] X. Han and J. Scazzero. Protein Expression Molecular Pattern Discovery by Nonnegative Principal Component Analysis. In *Pattern Recognition in Bioinformatics*, volume 5265 of *Lecture Notes in Computer Science*, pages 388–399. Springer, Berlin, 2008.

[54] M. Helén and T. Virtanen. Separation of drums from polyphonic music using nonnegative matrix factorization and support vector machine. In *Proceedings of 13th European Signal Processing Conference*, pages 1091–1094, Istanbul, Turkey, 2005.

[55] J. Herault and C. Jutten. Space or time adaptive signal processing by neural network models. In *AIP Conference Proceedings 151 on Neural Networks for Computing*, pages 206–211, New York, 1987.

[56] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[57] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[58] A. Hyvärinen and P. Hoyer. Independent subspace analysis shows emergence of phase and shift invariant features from natural images. In *Proceedings of the International Joint Conference on Neural Networks*, 1999.

[59] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9:1483–1492, 1997.

[60] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[61] A. Iserles, H. Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, pages 215–365, 2000.

[62] A. Iske. *Multiresolution Methods in Scattered Data Modelling*. Lecture Notes in Computational Science and Engineering. Springer, 2004.

[63] A. Iske. Scattered data approximation by positive definite kernel functions. *Rendiconti del Seminario Matematico*, 69(3):217–246, 2011.

[64] R. Jaiswal, D. FitzGerald, D. Barry, E. Coyle, and S. Rickard. Clustering NMF basis functions using Shifted NMF for monaural sound source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 245–248, 2011.

[65] C. T. Kelley. *Iterative Methods for Optimization*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1999.

[66] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*. Charles Griffin & Company Limited, 1977.

[67] P. Kisilev, M. Zibulevsky, and Y. Y. Zeevi. A Multiscale Framework for Blind Separation of Linearly Mixed Signals. *Journal of Machine Learning Research*, 4(7-8):1339–1364, 2004.

[68] S. Krause-Solberg. Dimensionality Reduction Methods in Independent Subspace Analysis for Signal Detection. `http://www.math.uni-hamburg.de/home/krause-solberg/Diplomarbeit_SaraKrause-Solberg.pdf` , diploma thesis, Universität Hamburg, August 2011.

[69] S. Krause-Solberg and A. Iske. Non-negative dimensionality reduction for audio signal separation by NNMF and ICA. In *International Conference on Sampling Theory and Applications (SampTA)*, pages 377–381, 2015.

[70] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[71] D. Kushnir, A. Haddad, and R. R. Coifman. Anisotropic diffusion on sub-manifolds with application to Earth structure classification. *Applied and Computational Harmonic Analysis*, 32(2):280–294, 2012.

[72] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562. MIT Press, 1999.

[73] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Information Science and Statistics Series. Springer, London, 2010.

[74] T.-W. Lee, B.-U. Koehler, and R. Orglmeister. Blind Source Separation of Nonlinear Mixing Models. In *Neural networks for Signal Processing VII*, pages 406–415, 1997.

[75] E. Levina and P. J. Bickel. Maximum Likelihood Estimation of Intrinsic Dimension. In *Advances in Neural Information Processing Systems*, volume 17, pages 777–784. MIT Press, 2005.

[76] M. G. López, H. M. Lozano, F. Sánchez, P. Luis, and L. Moreno. Blind Source Separation of audio signals using independent component analysis and wavelets. In *21st International Conference on Electrical Communications and Computers (CONIELECOMP)*, pages 152–157, 2011.

[77] D. G. Luenberger. The Gradient Projection Method Along Geodesics. *Management Science*, 18(11):620–631, 1972.

[78] D. Luo, C. Ding, H. Huang, and T. Li. Non-negative Laplacian Embedding. In *Ninth IEEE International Conference on Data Mining*, pages 337–346, 2009.

[79] R. C. Maher. Evaluation of a method for separating digitized duet signals. *Journal of the Audio Engineering Society*, 38:956–979, 1990.

[80] R. E. Mahony. The Constrained Newton Method on a Lie Group and the Symmetric Eigenvalue Problem. *Linear Algebra and its Applications*, 248:67–89, 1996.

[81] R. E. Mahony and J. H. Manton. The Geometry of the Newton Method on Non-Compact Lie Groups. *Journal of Global Optimization*, 23(3-4):309–327, 2002.

[82] J. H. Manton. Optimization Algorithms Exploiting Unitary Constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, 2002.

[83] J. H. Manton. On the various generalisations of optimisation algorithms. In *16th International Symposium of Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[84] R. J. Marks II. *Handbook of Fourier Analysis & Its Applications*. Oxford University Press, USA, 2009.

[85] A. Mertins. *Signaltheorie:*. Springer, 2010.

[86] D. N. Monnig, B. Fornberg, and F. G. Meyer. Inverting nonlinear dimensionality reduction with scale-free radial basis function interpolation. *Applied and Computational Harmonic Analysis*, 37(1):162–170, 2014.

[87] A. Montanari and E. Richard. Non-negative Principal Component Analysis: Message Passing Algorithms and Sharp Asymptotics. arXiv:1406.4775, 18 Jun 2014.

[88] E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–45, 1997.

[89] E. Oja and M. Plumbley. Blind Separation Of Positive Sources Using Non-Negative PCA. In *Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 11–16, 2003.

[90] O. Okun, H. Priisalu, and A. Alves. Fast Non-negative Dimensionality Reduction for Protein Fold Recognition. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Machine Learning: ECML 2005*, volume 3720 of *Lecture Notes in Computer Science*, pages 665–672. Springer, Berlin, Heidelberg, 2005.

[91] A. Ozerov and C. Févotte. Multichannel Nonnegative Matrix Factorization in Convolutive Mixtures for Audio Source Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2010.

[92] Y. Panagakis, C. Kotropoulos, and G. R. Arce. Non-Negative Multilinear Principal Component Analysis of Auditory Temporal Modulations for Music Genre Classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):576–588, 2010.

[93] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[94] M. Pflaum. *Analytic and Geometric Study of Stratified Spaces: Contributions to Analytic and Geometric Aspects*. Lecture Notes in Mathematics. Springer, 2001.

[95] M. Plumbley and E. Oja. A 'nonnegative PCA' algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 15(1):66–76, 2004.

[96] M. D. Plumbley. Lie Group Methods for Optimization with Orthogonality Constraints. In C. G. Puntonet and A. Prieto, editors, *Independent Component Analysis and Blind Signal Separation*, volume 3195 of *Lecture Notes in Computer Science*, pages 1245–1252. Springer, Berlin, Heidelberg, 2004.

[97] M. D. Plumbley. Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, 67:161–197, 2005.

[98] M.D. Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543, 2003.

[99] I. Potamitis and A. Ozerov. Single channel source separation using static and dynamic features in the power domain. In *16th European Signal Processing Conference*, pages 1–5, 2008.

[100] L. R. Rabiner and R. W. Schafer. *Introduction to digital speech processing*. Foundations and Trends in Signal Processing. Now Publishers Inc., Hanover, MA, USA, 2007.

[101] J. Rice. *Mathematical Statistics and Data Analysis*. Thomson Brooks/Cole, 2007.

[102] W. Ring and B. Wirth. Optimization Methods on Riemannian Manifolds and Their Application to Shape Space. *SIAM Journal on Optimization*, 22(2):596–627, 2012.

[103] J. B. Rosen. The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

[104] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[105] A. A. Sagle and R. E. Walde. *Introduction to Lie Groups and Lie Algebras*. Pure and applied mathematics. Academic Press, 1973.

[106] L. K. Saul and S. T. Roweis. An Introduction to Locally Linear Embedding. Technical report, 2000.

[107] C. D. Sigg and J. M. Buhmann. Expectation-maximization for Sparse and Non-negative PCA. In *Proceedings of the 25th International Conference on Machine Learning*, pages 960–967, 2008.

[108] P. Smaragdis and J. C. Brown. Non-Negative Matrix Factorization for Polyphonic Music Transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[109] S. T. Smith. *Geometric Optimization Methods for Adaptive Filtering*. PhD thesis, Harvard University, 1993.

[110] M. Solazzi, F. Piazza, and A. Uncini. Nonlinear blind source separation by spline neural networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2781–2784, 2001.

[111] M. Spiertz and V. Gnann. Iterative monaural audio source separation for subspace grouping. In *International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, pages 1–4, 2009.

[112] F. Taringoo, P. M. Dower, D. Nesic, and Y. Tan. Optimization Methods on Riemannian Manifolds via Extremum Seeking Algorithms. arXiv:1412.2841, 9 Dec 2014.

[113] C. J. Taylor and D. J. Kriegman. Minimization on the Lie Group SO(3) and Related Manifolds. Technical report, Yale University, 1994.

[114] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290:2319–2323, 2000.

[115] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[116] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita. Dimensionality reduction using non-negative matrix factorization for information retrieval. In *2001 IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 960–965, 2001.

[117] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using Independent Subspace Analysis. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 843–848, Nara, Japan, 2003.

[118] L. van der Maaten. Matlab Toolbox for Dimensionality Reduction. `https://lvdmaaten.github.io/drtoolbox/`.

[119] L. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality Reduction: A Comparative Review. Technical report, Tilburg University, 2009.

[120] V. S. Varadarajan. *Lie Groups, Lie Algebras, and their Representations*. Graduate Texts in Mathematics 102. Springer, Berlin, 1974.

[121] T. Virtanen. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007.

[122] T. Virtanen. Monaural sound source separation by perceptually weighted non-negative matrix factorization. Technical report, 2007.

[123] T. Virtanen and A. Klapuri. Separation of harmonic sound sources using sinusoidal modeling. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II765–II768. IEEE, 2000.

[124] B. von Querenburg. *Mengentheoretische Topologie*. Springer-Lehrbuch. Springer, Berlin, Heidelberg, 2001.

[125] J. Wang. *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Springer, Berlin, Heidelberg, 2012.

[126] F. W. Warner. *Foundations of Differentiable Manifolds and Lie Groups*. Graduate Texts in Mathematics. Springer, 1983.

[127] L. Wei, N. Guan, X. Zhang, Z. Luo, and D. Tao. Orthogonal Nonnegative Locally Linear Embedding. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2134–2139, 2013.

[128] J. Wellhausen. Audio Signal Separation Using Independent Subspace Analysis and Improved Subspace Grouping. In *Proceedings of Nordic Signal Processing Symposium NORSIG*, pages 310–313, Rejkjavik, Iceland, 2006.

[129] H. Wendland. *Scattered Data Approximation*. Number 17 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.

[130] M. W. Wong. *Discrete Fourier Analysis*. Birkhäuser, Basel, 2011.

[131] G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.

[132] S. Zafeiriou and N. A. Laskaris. Nonnegative Embeddings and Projections for Dimensionality Reduction and Information Visualization. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 726–729, 2010.

[133] R. Zass and A. Shashua. Nonnegative Sparse PCA. In *Advances in Neural Information Processing Systems*, volume 19, pages 1561–1568. MIT Press, 2007.

[134] M. Zaunschirm, J. D. Reiss, and A. Klapuri. A sub-band approach to modification of musical transients. *Computer Music Journal*, 36(2):23–36, 2012.

[135] A. I. Zayed. *Advances in Shannon's Sampling Theory*. CRC Press, Boca Raton, 1993.

[136] Z. Zhang and H. Zha. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. *SIAM Journal of Scientific Computing*, 26:313–338, 2002.

Index

Summary

In this thesis, we studied the application of (non-negative) dimensionality reduction methods in signal separation. In single-channel separation, the decomposition techniques as e.g. non-negative matrix factorization (NNMF) or independent component analysis (ICA) are typically applied to time-frequency data of the mixed signal obtained by a signal transform.
Starting from this classical separation procedure in the time-frequency domain, we considered an additional preprocessing step, in which the dimension of the data is reduced in order to facilitate the computation. Depending on the separation methods, different properties of the dimensionality reduction technique are required. We focused on the non-negativity of the low-dimensional data or - since the time-frequency data is non-negative - rather on the non-negativity preservation beyond the reduction step, which is mandatory for the application of NNMF.
We proposed an approach to non-negative dimensionality reduction that modifies classical dimensionality reduction techniques, which can be written as an optimization problem with rotationally invariant cost functional. By adding a non-negativity constraint to the optimization problem, we enforce the low-dimensional data to be non-negative. If furthermore the reduction map does not increase the angles between data points, these conditions enable us to first solve the classical dimensionality reduction problem before applying a rotation in order to obtain non-negativity of the low-dimensional data set. We discuss the applicability of this *splitting approach* to different dimensionality reduction techniques, especially to principal component analysis (PCA).
For the second step of the splitting approach, a suitable rotation map is needed, which we compute by solving an auxiliary optimization problem on the set of special orthogonal matrices $SO(d)$. This set is not a vector space and thus, standard optimization methods such as steepest descent or Newton's method are not directly applicable. To overcome the lack of additive update algorithms, we used the Lie group properties of $SO(d)$ in order to construct a multiplicative update algorithm. This construction strongly relies on the exponential map which links $SO(d)$ with its associated Lie algebra. We rigorously derive a steepest descent method on Lie groups, which iterates along curves on the group starting in the direction of a tangent vector. Usually, it is quite difficult to determine such curves explicitly but the structure of a Lie group and the exponential map offer a simple and efficient way to do so.
Finally, we discuss the application of the developed non-negative dimensionality reduction techniques to signal separation. We present some numerical results when using our non-negative PCA (NNPCA) and compare its performance with other versions of PCA and different separation techniques, namely NNMF and ICA. From the results, it can be seen that our NNPCA performs better than the rather naive alternative of taking the absolute value of the low-dimensional data set before applying NNMF. Furthermore, the separation with NNPCA in combination with NNMF is almost as good as the one with PCA and ICA.

Some results of this thesis are published in [47, 69].

Kurzfassung

In der vorliegende Arbeit untersuchten wir die Anwendung von Methoden zur (nichtnegativen) Dimensionsreduktion (NNDR) im Gebiet der Signaltrennung. Typischerweise werden für die Trennung von Monosignalen Zeit-Frequenz-Daten benutzt, die durch eine Signaltransformation aus dem gemischten Signal berechnet werden. Die Trennung selber kann mit Hilfe von verschiedene Methoden wie z. B. nichtnegativer Matrix Faktorisierung (NNMF) oder Independent Component Analysis (ICA) durchgeführt werden.
Ausgehend hiervon betrachteten wir einen zusätzlichen Schritt, in welchem die Dimension der Daten im Zeit-Frequenz-Bereich reduziert wird, um Berechnungen zu vereinfachen. In Abhängigkeit von der Trennmethode können unterschiedliche Eigenschaften der Reduktionsmethode im Vordergrund stehen. Weil schon die Zeit-Frequenz-Daten nichtnegativ sind, konzentrierten wir uns auf die Erhaltung der Nichtnegativität der Daten über die Reduktion hinaus, da dies für die Anwendung von NNMF notwendig ist.
Wir entwickelten eine Methode zur NNDR, die darauf beruht klassische Reduktionstechniken, welche als Optimierungsproblem (OP) mit rotationsinvariantem Kostenfunktional formuliert werden können, abzuwandeln. Durch Hinzufügen einer Nichtnegativitätsbedingung zu dem OP können wir garantieren, dass die niedrigdimensionalen Daten nichtnegativ sind. Wenn außerdem durch die Reduktion die Winkel zwischen Datenpunkten nicht vergrößert werden, können wir das OP lösen, indem wir erst die klassische Reduktion durchführen und dann die Daten ins Positive rotieren. Überdies diskutierten wir die Anwendbarkeit dieses *Splitting Ansatzes* auf verschiedene Dimensionsreduktionstechniken, insbesondere auf Principal Component Analysis (PCA).
Dieser Ansatz basiert auf einer Rotationsabbildung, die wir durch Lösen eines weiteren OPs auf der speziellen orthogonalen Gruppe $SO(d)$ berechneten. Durch die fehlende Vektorraumstruktur können auf dieser Menge Standardmethoden wie z. B. das Verfahren des steilsten Abstiegs oder das Newtonverfahren nicht ohne Weiteres angewendet werden. Wir können jedoch die Eigenschaften von $SO(d)$ als Lie Gruppe verwenden, um einen multiplikativen Update-Algorithmus zu konstruieren. Diese Konstruktion basiert maßgeblich auf der Exponentialabbildung, die $SO(d)$ mit ihrer assoziierten Lie Algebra verknüpft. Auf Grund dieser Verknüpfung konnten wir ein Verfahren des steilsten Abstiegs auf Lie Gruppen von Grund auf herleiten, bei dem wir entlang von Kurven, die in Richtung eines Tangentialvektors verlaufen, iterieren. Im Allgemeinen ist es nicht leicht solche Kurven explizit zu bestimmen, jedoch bietet die Exponentialabbildung eine einfache und effiziente Möglichkeit hierfür.
Schlussendlich diskutierten wir die Anwendung der entwickelten Methoden im Bereich der Signaltrennung. Wir stellten einige numerische Ergebnisse vor und verglichen unsere nichtnegative PCA (NNPCA) mit anderen PCA-Versionen sowie unterschiedlichen Trenntechniken (NNMF und ICA). Die Ergebnisse zeigen, dass unsere NNPCA geeigneter ist als die naive Alternative, bei welcher der Absolutbetrag auf PCA-reduzierte Daten angewendet wird. Des Weiteren zeigte sich, dass die Trennung mit NNPCA und NNMF fast ebenso gute Ergebnisse liefert wie die Trennung mit PCA und ICA.

Einige Resultate dieser Arbeit sind in [47, 69] veröffentlicht.