Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Datenverarbeitung

Structured Co-sparse Analysis Operator Learning for Inverse Problems in Imaging

Julian Wörmann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende(r): Prof. Dr. Gerhard Kramer

Prüfer der Dissertation:

1. Priv.-Doz. Dr. rer. nat. Martin Kleinsteuber
2. apl. Prof. Dr.-Ing. Walter Stechele

Die Dissertation wurde am 10.04.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 25.10.2019 angenommen.

Julian Wörmann. *Structured Co-sparse Analysis Operator Learning for Inverse Problems in Imaging*. Dissertation, Technische Universität München, Munich, Germany, 2020.

Acknowledgements

This work would not have been possible without continuous encouragement and support I received from many people within the past years. I would like to take this opportunity to express my sincere gratitude to all the people who supported me in writing this thesis.

First of all, I would like to thank my supervisor PD Dr. rer. nat. Martin Kleinsteuber, for giving me the opportunity to pursue my doctoral study in such an interesting research field. His ongoing support and advice motivated me a lot and eventually allowed me to successfully complete my thesis.

Special thanks also go to Dr. Hao Shen who gave me the chance to continue my work in his research group and who always encouraged me to keep on track. I also would like to thank Prof. Dr.-Ing. Klaus Diepold for all his advice during my time at the Chair for Data Processing. My thanks also go to Prof. Dr.-Ing. Walter Stechele for reviewing my thesis.

Needless to say, many thanks to my colleagues at GOL and LDV, in particular Martin and Peter, as well as Clemens, Dominik, Rori, Simon, Uli, Alex, Julian, Tim, Marko and Martin for all their support and for making the past couple of years such a great experience. Furthermore, thanks to my colleagues at fortiss for all the fruitful discussions and helpful comments in the final stage of this thesis.

Last but not least, I would like to thank my family and friends for their support, encouragement and motivation during all the stages of my doctoral study. Finally, my heartfelt thanks go to Birte who enriches my life every day and without whom accomplishing this dissertation would not have been possible. Thank you so much for supporting me at all times.

Abstract

Sparse data models have been proven very valuable for various tasks in signal processing. Basically, these models aim to represent the signal with only a few nonzero coefficients. Especially the field of image processing has been influenced in a large extent. This is mainly due to the reason that the sparse image code provides a more efficient representation with regard to frequent tasks, e.g. storage and transmission, structural analysis or inverse problem regularization. Commonly, analytically given transformations are used because they are efficient to apply and they can be deployed in a distributed manner. However, this comes at the cost of not being well adapted to the particular image class of interest. This issue motivates the research branch of learning sparse data models to better capture the structural information of the image data. In this thesis, I attempt to combine the best from both worlds, i.e., efficient computations and adaptability to the underlying structure. To that end, a separability constraint is imposed on the model in order to address both requirements. Moreover, the structural constraint allows to efficiently handle multidimensional data which is often intractable with classical learning approaches. Instead of focusing on the well established sparse synthesis model, I concentrate on the co-sparse analysis model, where it is assumed that the multiplication of the analysis operator with the present signal leads to a sparse representation.

First, an algorithm that is able to learn a separable co-sparse analysis operator from clean training signals is provided. The proposed Stochastic Gradient Descent on manifolds approach efficiently accounts for the geometrical structure of the operators. Furthermore, the sequential processing of the samples avoids the problem of determining a training set of suitable size in advance. Robustness of the algorithm with respect to parameter changes makes the presented learning scheme easily applicable. Finally, the competitive performance compared to state-of-the-art algorithms further motivates the presented approach.

In the second part, the sample complexity of the algorithm along with the generalization behavior of the learned model is explored empirically. An operator recovery experiment based on synthetic data is used to verify that separable operators can be reliably learned from less training samples compared to non-structured ones. In order to assess the generalization, the estimated divergence between the distribution of the training signals and the distribution of samples where the learned operator serves as a generative model is an-

alyzed. This strategy renders the process of evaluating the suitability of the learned model entirely task independent.

An extension of the learning algorithm such that it can cope with corrupted or undersampled measurements is addressed in the third part of this thesis. The formulation as a blind learning problem where the operator and the image are recovered simultaneously allows the model to adapt to the image content. Especially the extension to multidimensional data reveals the strength of the separable model to easily take into account the available information from all dimensions.

Finally, the last contribution in this thesis concerns Sparse Auto-Encoders. I explore how to integrate the co-sparse analysis model assumption into the encoder function, such that the Sparse Auto-Encoder framework can be leveraged to learn meaningful analysis operators. It is shown that simple weight normalization constraints are sufficient to permit the algorithm to learn useful encoder matrices, which is verified by means of the same experimental setup that is used to assess the performance of the conventional learning scheme.

Contents

List of Publications

M. Seibert*, J. Wörmann*, R. Gribonval, and M. Kleinsteuber
Learning Co-Sparse Analysis Operators with Separable Structures
IEEE Transactions on Signal Processing 64(1), pp. 120-130, January 2016.

M. Seibert*, J. Wörmann*, R. Gribonval, and M. Kleinsteuber
Separable Cosparse Analysis Operator Learning
European Signal Processing Conference (EUSIPCO), September 2014.

J. Wörmann, S. Hawe, and M. Kleinsteuber
Analysis Based Blind Compressive Sensing
IEEE Signal Processing Letters 20(5), pp. 491-494, May 2013.

* These authors contributed equally to this work.

List of Figures

List of Tables

List of Symbols

$\mathcal{U}, \mathcal{V}, \mathcal{Q}$	Tensors (multiway or multimode signals), written as capital calligraphic letters.
$\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Xi}$	Matrices, written as capital boldface letters.
$\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\beta}$	Column vectors, written as lowercase boldface letters.
a, γ, P	Scalars, written as lowercase or capital letters.
t	Iteration.
$\mathcal{U}_j$	j-th multimode signal from set of tensors $\{\mathcal{U}_i\}_{i=1}^{V}$.
$\boldsymbol{u}_j$	j-th column of matrix $\boldsymbol{U}$.
u_i	i-th entry of vector $\boldsymbol{u}$.
$u_{i,j}$	j-th entry of vector $\boldsymbol{u}_i$.
U_{ij}	i-th element in the j-th column or (i, j)-th entry of matrix $\boldsymbol{U}$.
$\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$	First, Second, Third matrix out of the set of matrices $\{\boldsymbol{U}_i\}_{i=1}^{V}$.
$\boldsymbol{u}_{\mathbf{d}}, \boldsymbol{U}_{\mathbf{e}}, \boldsymbol{\Xi}_{\mathbf{z}}$	Special vectors/matrices identified through indices $\mathbf{d}, \mathbf{e}, \mathbf{z}$.
$\boldsymbol{\Xi}^{(t)}, \boldsymbol{u}^{(t)}, v^{(t)}$	Elements at the t-th iteration.
$\boldsymbol{U}^\top, \boldsymbol{v}^\top$	Matrix/Vector transposed.
$\boldsymbol{U}^\dagger$	Pseudo-Inverse of $\boldsymbol{U}$.
N, n	Signal dimension.
I_1, I_2, J_3	Signal dimension along the first, second, third mode of a tensor.
K, k	Number of atoms, filters.
M, m	Number of measurements.
T	Number of samples.
$\mathcal{S}, \boldsymbol{S}, \boldsymbol{s}, s$	Signal in tensor, matrix, vector and scalar form.
$\boldsymbol{Y}, \boldsymbol{y}, y$	Measurements in matrix, vector and scalar form.
$\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$	System matrix.
$\boldsymbol{\phi} \in \mathbb{R}^{N}$	Single row of system matrix (as column vector).
$\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$	Sparsifying basis, e.g. DCT basis, Fourier basis, Wavelet basis.
$\boldsymbol{\psi}_j \in \mathbb{R}^{N}$	j-th basis vector from $\boldsymbol{\Psi}$.
$\boldsymbol{D} \in \mathbb{R}^{N \times K}$	Synthesis Dictionary.
$\boldsymbol{d}_j \in \mathbb{R}^{N}$	j-th atom from Dictionary $\boldsymbol{D}$.
$\boldsymbol{d}_{v,j} \in \mathbb{R}^{N}$	j-th atom from Dictionary $\boldsymbol{D}_v$.

$\mathcal{X}, \boldsymbol{X}, \boldsymbol{x}$	Synthesis sparse code in tensor, matrix and vector form.
$\boldsymbol{\Omega} \in \mathbb{R}^{K \times N}$	Analysis Operator.
$\boldsymbol{\omega}_j \in \mathbb{R}^N$	j-th filter/row from Analysis Operator $\boldsymbol{\Omega}$ (as column vector).
$\boldsymbol{\omega}_{v,j} \in \mathbb{R}^N$	j-th filter/row from Analysis Operator $\boldsymbol{\Omega}_v$ (as column vector).
$\mathcal{A}, \boldsymbol{A}, \boldsymbol{a}$	Analysis co-sparse code in tensor, matrix and vector form.
$\boldsymbol{I}_k$	Identity matrix of dimension $(k \times k)$.
$\mathbf{0}_m, \mathbf{0}_{m \times n}$	All zero vector and matrix of dimension m and $(m \times n)$, respectively.
$\boldsymbol{e}_i$	Vector with the i-th element equal to 1 and 0 elsewhere.
$\mathbb{S}^{N-1}$	Unit Euclidean sphere in $\mathbb{R}^N$.
$\mathsf{S}(N, K)$	Product of spheres manifold, K spheres residing in $\mathbb{S}^{N-1}$.
$\mathsf{OB}(N, K)$	Oblique manifold, equivalent to $\mathsf{S}(N, K)$.
$\nabla_{\boldsymbol{\Xi}} f(\boldsymbol{\Xi}, \cdots)$	Euclidean Gradient with respect to the variable $\boldsymbol{\Xi}$ (multivariate function).
$\nabla f(\boldsymbol{\Xi})$	Euclidean Gradient with respect to the variable $\boldsymbol{\Xi}$.
$\mathtt{G}(\boldsymbol{\Xi})$	(Riemannian) Gradient with respect to the variable $\boldsymbol{\Xi}$.
$\mathtt{T}_{\boldsymbol{\xi}} \mathbb{S}^{N-1}$	Tangent space of the manifold $\mathbb{S}^{N-1}$ at point $\boldsymbol{\xi}$.
$\mathtt{P}(\boldsymbol{\Upsilon}, \boldsymbol{\Xi}, \boldsymbol{H}, l)$	Parallel transport of $\boldsymbol{\Upsilon}$ along a parametrized geodesic, emanating from $\boldsymbol{\Xi}$ in the direction $\boldsymbol{H}$.
$\mathrm{vec}(\mathcal{U})$	The vec-operator rearranges the entries of a tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_V}$ into the column vector $\mathrm{vec}(\mathcal{U}) = \boldsymbol{u} \in \mathbb{R}^{I_1 I_2 \cdots I_V}$.
$\mathrm{vec}^{-1}(\boldsymbol{u})$	Inverse vectorization operator.
$\mathrm{unf}(\mathcal{U}, v)$	The unf-operator unfolds the entries of a tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_V}$ along the v^{th} dimension into the matrix $\mathrm{unf}(\mathcal{U}, v) = \boldsymbol{U} \in \mathbb{R}^{I_v \times (\prod_{j \neq v} I_j)}$.
$\mathrm{unf}^{-1}(\boldsymbol{U})$	Inverse unfolding operator.
$\mathrm{cond}(\cdot)$	Matrix condition.
$\mathrm{rk}(\cdot)$	Matrix rank.
$\mathrm{tr}(\cdot)$	Trace of a matrix.
$\det(\cdot)$	Matrix determinant.
$\mathrm{diag}(\cdot)$	Vector containing the diagonal entries of a square matrix.
$\log(\cdot)$	Natural logarithm (elementwise).
$\exp(\cdot)$	Exponential (elementwise).
$\lVert \cdot \rVert_p$	ℓ_p-norm.
$\lVert \cdot \rVert_F$	Frobenius norm (elementwise).

List of Abbreviations

AE	Auto-Encoder.
AWGN	Additive White Gaussian Noise.
BCS	Blind Compressed Sensing.
CG	Conjugate Gradient.
CS	Compressed Sensing.
CT	Computed Tomography.
DCT	Discrete Cosine Transform.
DFT	Discrete Fourier Transform.
DWT	Discrete Wavelet Transform.
EKL	Estimated Kullback-Leibler Divergence.
GD	Gradient Descent.
MAP	Maximum A Posteriori.
MLE	Maximum Likelihood Estimation.
MRI	Magnetic Resonance Imaging.
MSSIM	Mean Structural Similarity.
OB	Oblique Manifold.
OMP	Orthogonal Matching Pursuit.
PCA	Principal Component Analysis.
PSNR	Peak Signal to Noise Ratio.
SAE	Sparse Auto-Encoder.
SGD	Stochastic Gradient Descent.
SVD	Singular Value Decomposition.
TV	Total Variation.
UNTF	Uniform Normalized Tight Frame.
UoS	Union-of-Subspace.

Chapter 1.

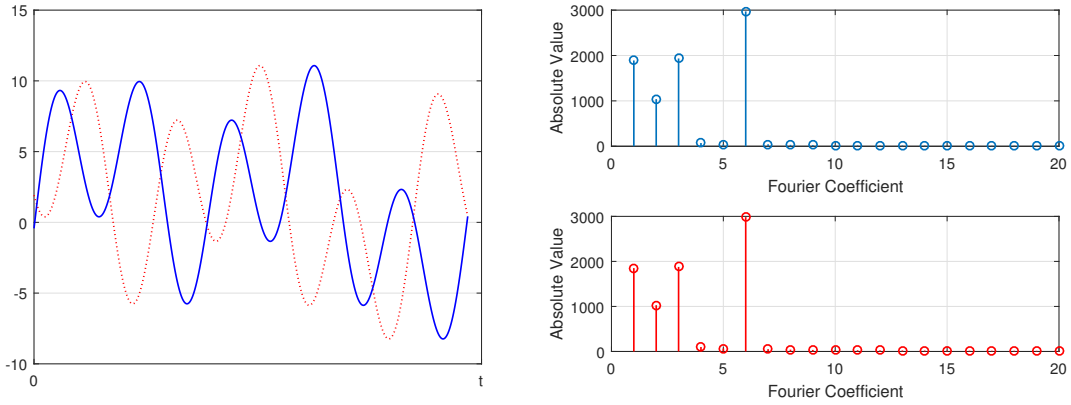
Introduction

The most natural representation of a digitally sampled signal is the sum of Dirac delta functions in time or space domain. While very convenient for purposes like visual inspection of the waveform, this representation might be useless for automatic processing or analysis tasks, e.g. classification or recognition. Even storage or transmission might be impractical due to the high redundancy present in the sampled signal. Consequently, transforming the signal content in a more useful representation that captures all the necessary information is highly desired. The core assumption for tackling the aforementioned problems is that structured signals can be represented concisely with respect to a convenient basis [20].

The Discrete Fourier Transformation (DFT) for example can be considered as one of the most important transformations in signal processing. Discrete and periodic signals are represented by a series of complex exponentials, or equivalently, as a weighted sum of sines and cosines. Given a signal described in the canonical basis, expanding the signal in the Fourier basis reveals its frequency information. Mathematically speaking, a signal $\boldsymbol{s} \in \mathbb{R}^N$ is a linear combination of the basis elements of the orthonormal Fourier basis $\{\boldsymbol{\psi}_i\}_{i=1}^N$, i.e.

$$\boldsymbol{s} = \sum_{i=1}^{N} c_i \boldsymbol{\psi}_i, \tag{1.1}$$

where $\boldsymbol{c}$ is the Fourier coefficient sequence of $\boldsymbol{s}$ obtained via $c_i = \langle \boldsymbol{s}, \boldsymbol{\psi}_i \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Let $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N] \in \mathbb{C}^{N \times N}$ be the matrix composed of the Fourier basis elements as its columns, we have $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{c}$. Figure 1.1 exemplary illustrates the applicability of the Fourier representation for the aforementioned tasks. The left figure shows a smooth target signal $\boldsymbol{s} \in \mathbb{R}^{1000}$ in the time domain that is composed of four sinusoids with varying frequencies. In the same plot, a shifted version of the same signal is given. On the right side, the corresponding absolute values of the Fourier coefficient sequences are plotted for each input signal. First, consider the problem of assessing the similarity of both signal curves. Naively comparing the samples in the time domain seems



(a) Original (blue) and shifted (red) input signal

(b) Absolute values of Fourier coefficients of the original signal (top) and shifted signal (bottom)

Figure 1.1.: The representation of a smooth signal that is composed of sinusoids.

very cumbersome to tackle the problem. However, a comparison of the significant entries in the Fourier coefficient sequence immediately reveals that the waveforms of both signals are essentially equal. Regarding the compression problem, Figure 1.1 also depicts a solution. We can directly deduce that only the position and the values of the prominent Fourier coefficients have to be stored or transmitted instead of the 1000 samples that constitute the signal in the time domain. The only assumption is that the decoder knows in which basis we have represented our signal. In [138] Rubinstein et al. describe this behavior of the linear transformation as *compaction*, i.e., the ability to express the signal with only a few coefficients. To conclude, the given example has shown that for smooth signals, the Fourier basis is well suited to represent this class of signals. On the other hand, if we know that our signal at hand is a member of this class, i.e., signals that reside in a low dimensional subspace spanned by only a few Fourier basis elements, then this representation constitutes a *signal or data model* - a way to mathematically characterize the signal [46].

The right choice of the data model is crucial for the success of the intended application. In [46] Elad describes an evolution of models over the last decades from simple ℓ_2 based methods like Thikonov Regularization [158], over Anisotropic Diffusion [166] and Total Variation [144] to sparsity promoting models [138].

Data models are also indispensable to stabilize the solution of inverse problems. In fact, many signal reconstruction tasks can be formulated as an inverse problem where the original signal $\boldsymbol{s} \in \mathbb{R}^N$ has to be inferred from corrupted and possibly incomplete observed measurements $\boldsymbol{y} \in \mathbb{R}^M$. Assuming linear measurements, the process of observing (measuring) a signal can be modeled via

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s} + \boldsymbol{e}, \tag{1.2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ describes a linear measuring process. Additive measuring noise is reflected by the vector $\boldsymbol{e} \in \mathbb{R}^M$.

Intuitively, one seeks for a solution $\boldsymbol{s}^\star$ that best explains the observations $\boldsymbol{y}$, which from a Bayesian perspective corresponds to the Maximum Likelihood Estimation (MLE) approach. In the case of Additive Gaussian White Noise (AWGN), i.e., the entries of the vector $\boldsymbol{e}$ follow an *i.i.d.* standard normal Gaussian distribution $\mathcal{N}(0, \boldsymbol{I}_M)$, it can be shown easily that maximizing the likelihood is equal to solving the problem

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s} \in \mathbb{R}^N} \tfrac{1}{2} \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{s}\|_2^2. \tag{1.3}$$

Now, if for example the number of measurements equals the dimension of the signal, i.e., we have $M = N$, and if we set the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ to be the identity $\boldsymbol{I}_N$, Eq. (1.3) reduces to a classical Denoising problem. However, following the MLE approach, without further knowledge about the structure of the signal $\boldsymbol{s}$ there is no hope to disentangle the true signal from the noise since the optimal solution to problem (1.3) is the noisy signal $\boldsymbol{y}$ itself. Another problem arises if we have more unknowns that observations, i.e., if $M < N$. In this scenario, infinitely many solutions exist that explain the measurements.

As a consequence, what we need is an adequate regularizer that models the *a priori* distribution of the signal and thus stabilizes the solution. With prior knowledge about the signal structure at hand, we can obtain the Maximum A Posteriori (MAP) estimate of the signal. To illustrate this approach, recall the example given in Figure 1.1. If $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{c}$ is a signal that can be compactly represented in the Fourier basis $\boldsymbol{\Psi}$, only a small amount of the Fourier coefficients are non-zero. Indeed, the compaction described in this example can be considered as a *sparse* representation model. According to [160], the sparse data model enjoys great popularity since many fundamental questions in electrical engineering, statistics and applied mathematics can be posed as sparse approximation problems.

A signal is said to be L_s-sparse if at most L_s samples are non-zero, i.e., $\|\boldsymbol{c}\|_0 \leq L_s$, where the function $\|\cdot\|_0 : \mathbb{R}^N \to \mathbb{R}^+$ counts the number of non-zero components. Thus, a natural approach to regularize the solution of problem (1.3) is to penalize the number of non-zero

entries in the signal's Fourier coefficient sequence, which ends up in the problem

$$\boldsymbol{c}^\star \in \arg\min_{\boldsymbol{c}\in\mathbb{C}^N} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{c}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{c}\|_0 \leq L_s. \tag{1.4}$$

After solving (1.4), the desired signal can be easily obtained via $\boldsymbol{s}^\star = \boldsymbol{\Psi}\boldsymbol{c}^\star$. A survey on practical algorithms to solve the sparse approximation problem can be found in [160].

In practice most of the signals tend to be only approximately sparse or compressible. Compressible signals exhibit the property that the magnitude of the sorted entries decays rapidly such that only a few samples are significant while the rest is close to zero. From a probability theory perspective, the coefficient sequence $\boldsymbol{c}$ follows a Laplace distribution, i.e., $\boldsymbol{c} \sim \text{Laplace}(0, b)$ and small or zero values appear with high probability. The probability density function reads $L(c_i|0,b) = \frac{1}{2b}\exp\Big(-\frac{|c_i|}{b}\Big)$. This prior knowledge can be also exploited in the MAP recovery approach. After applying Bayes theorem, and taking the logarithm of the probability distributions, fidelity to the Gaussian distributed measurements and adherence to the assumption of Laplacian distributed coefficients can be achieved via minimizing the problem

$$\boldsymbol{c}^\star \in \arg\min_{\boldsymbol{c}\in\mathbb{C}^N} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{c}\|_2^2 + \lambda\|\boldsymbol{c}\|_1, \tag{1.5}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm. This problem is also known as LASSO [157, 63] or Basis Pursuit [27]. The convexity of (1.5) has led to many theoretical investigations of the problem in the noiseless as well as noisy setting. The interested reader is referred to e.g. [54], [18] and references in [160]. Eventually, the theory of Compressed Sensing (CS) [42, 19, 121] which covers sparse approximation results for the case where the number of measurements is significantly smaller than the signal dimension, i.e., $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ with $M \ll N$, further popularized the ℓ_1 regularized problem (1.5) [22].

Usually, the sparsifying basis $\boldsymbol{\Psi}$ is an orthogonal matrix that allows for a fast computation of the coefficient sequence via $\boldsymbol{c} = \boldsymbol{\Psi}^\top\boldsymbol{s}$. With this property at hand, the recovery process can be equivalently stated in its analysis form that reads

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}\in\mathbb{R}^N} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{s}\|_2^2 + \lambda\|\boldsymbol{\Psi}^\top\boldsymbol{s}\|_1. \tag{1.6}$$

Analogous to the synthesis sparse approximation problem, there exist various algorithms that tackle the signal recovery problem following the analysis model framework. For an in-depth discussion and investigation on the uniqueness properties of these recovery algorithms in the context of image reconstruction, the reader is referred to e.g. [55, 58, 57, 101, 108].

Both (1.5) and (1.6) encourage solutions that on the one hand have a sparse representation with only a small amount of significant non-zero coefficients, and on the other hand fit to the measurements. Yet, we have assumed that the measurements are corrupted by AWGN. While this assumption is widely used in practice and various algorithms have been developed to tackle this problem, there are applications where the noise follows a different distribution. For instance Poisson noise appears in low-light photography, medical imaging and microscopy [56]. Rayleigh or Gamma distributed noise that is multiplied rather than added to the signal can be observed in synthetic aperture radar, sonar, ultrasound and laser imaging [8]. The presence of impulsive or salt-and-pepper noise is considered in [23]. To account for these different noise distributions, let $d(\boldsymbol{s}) : \mathbb{R}^N \to \mathbb{R}$ denote a general fidelity measure that, in contrast to the ℓ_2 data fitting term introduced in (1.3), does not necessarily need to be a function of the difference $\boldsymbol{\Phi s} - \boldsymbol{y}$ but only of the signal $\boldsymbol{s}$. Besides a general formulation of the data fidelity, in [48] the general prior distribution of the signal representation is modeled as a Gibbs-like distribution

$$\boldsymbol{s} \sim \text{const.} \cdot e^{[-\alpha \cdot g(\boldsymbol{s})]}, \tag{1.7}$$

with $g(\boldsymbol{s}) : \mathbb{R}^N \to \mathbb{R}^+$ being a function that takes low values if the signal representation of $\boldsymbol{s}$ corresponds to the distribution and high values if not. Besides the convex ℓ_1-norm, typical choices for $g(\boldsymbol{s})$ include the ℓ_p-norm with $0 < p \leq 1$, or the Huber loss [72]. With these two functions at hand, the general perhaps non-convex sparsity based reconstruction problem can be formulated as

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s} \in \mathbb{R}^N} d(\boldsymbol{s}) + \lambda g(\boldsymbol{s}), \tag{1.8}$$

where the parameter λ weights between the noise dependent data fidelity term $d(\boldsymbol{s})$ and the sparsity inducing function $g(\boldsymbol{s})$.

In the recent years, sparse data models have attracted high attention. On the one hand, sparsity allows for a simple interpretation of the model. In an era where data collection and storage is widespread and can be carried out easily and cheap, a signal representation that directly discovers the underlying features/predictors that explain the data is of highest interest. On the other hand, sparsity as a versatile tool to regularize the solution of many signal processing applications has been proven very useful. Even before the theory of Compressed Sensing has evolved, Donoho has shown in [43] that reconstruction from sub-Nyquist [1] sampled data is feasible with a sparsity constraint. A simple approach for

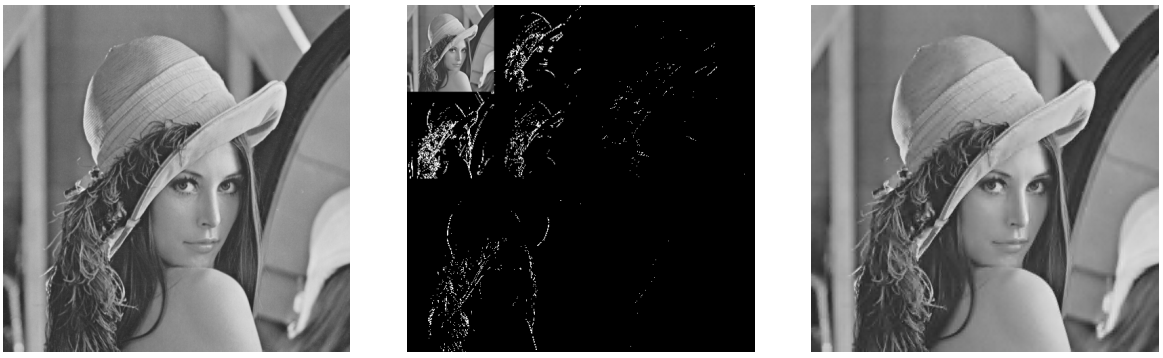
[1] The Nyquist Theorem states that for exact reconstruction, the sampling rate has to be more than twice the maximum frequency present in the signal.

denoising a signal has been encouraged by Donoho & Johnstone in [44], who proposed to simply threshold the wavelet representation of a noisy signal such that its expansion becomes sparse. Wavelet thresholding is also applied in a deconvolution setting in [153]. The combination of Principal Component Analysis (PCA) and sparsity is introduced in [182]. Again, the additional sparsity constraint on the loadings is used to make the model more interpretive. Revealing the structure of multivariate data collected by multi-channel sensors is another example where sparsity can improve the accessibility. This problem is also referred to as Blind Source Separation (BSS) where the data is separated into sources and their corresponding mixing factors. Sparsity in the context of BSS was first introduced by [181], while more recent results can be found in [9, 62].

The success of sparse representation modeling can be also attributed to its impact in the field of image processing. The observations that the receptive fields of simple cells in the mammalian primary visual cortex are localized, oriented and bandpass have strongly influenced the evolution of sparse image representations. According to Daugman [35] and Marcelja [96], these properties of the cells are well described by the basis functions of the Gabor-Wavelet transform. Indeed, filtering an image with these functions results in a coefficient sequence whose entries are sparsely distributed which also gives evidence that natural images may be described as a collection of localized and oriented structural primitives like edges, lines or other elementary features [51, 52, 84]. Eventually, the seminal work of Olshausen & Field [104, 105] shows that a sparse linear coding of images containing natural scenes results in learned basis functions that resemble the structure of Gabor-Wavelet like functions which further supports this particular image model assumption.

The sparse or compressible representation of images has also paved the way for efficient coding strategies. In 1992, the Joint Photographic Experts Group presented a lossy image compression algorithm called JPEG [163]. The core of this scheme is a decomposition of small non-overlapping 8×8 image patches into the basis signals of the two dimensional Discrete Cosine Transform (DCT) basis. Thus, the resulting coefficients encode the 2D spatial frequencies present in the patch. Since most of the energy is concentrated in the low frequency parts, the near-zero coefficients that represent the high frequency information can be neglected without any substantial loss in image quality. The compression framework has been developed further resulting in the JPEG2000 standard [156] that is based on the Discrete Wavelet Transformation (DWT). Figure 1.2 depicts the two-level DWT representation of an image along with a plot of the magnitude of the coefficients in descending order. The compressibility of the image in the DWT domain can be observed by the fact that the reconstruction of the image from only $\sim 11\%$ of the most significant coefficients, as shown in Figure 1.2c, is close to the input image and artifacts are hardly noticeable.

Sparse image representations have also been extensively used to regularize the solution



(a) Input Image

(b) Two-level Wavelet Decomposition

(c) Reconstructed Image

Figure 1.2.: Discrete Wavelet Transform (DWT) of an image. **(a)** Input image. **(b)** The multi resolution decomposition where the image is decomposed into two different scales. Only significant coefficients are shown. **(c)** Reconstruction of the image with only $\sim 11\%$ of the most significant coefficients as indicated in (b).

of many common image processing tasks that can be formulated as linear inverse problems. Depending on the specific application, the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ given in equation (1.2) represents the underlying measurement operation. Classical image applications include Denoising [41, 98, 53], Deblurring [16], Super-Resolution [176], Inpainting [50] or Image Separation [151].

As already mentioned above, the analysis of the general and very interesting recovery task where the dimension of the measurements is significantly smaller than the signal dimension has been investigated under the term Compressed Sensing. Especially medical imaging has profited from the theory which led Compressed Sensing become a highly active research area in recent years. In Magnetic Resonance Imaging (MRI) the formulation of the image recovery problem as a sparse approximation problem can lead to a significant reduction in scanning time [87, 86]. In Computed Tomogrpahy (CT) imaging the same principle can be used to lower the radiation dose the patient is exposed to cf. [26, 29, 177]. Eventually, much research has been conducted to further introduce CS into different medical imaging modalities. In [112], the reconstruction of Photo-Acoustic Tomography (PAT) from a reduced set of measurements is investigated. Ultrasound (US) images are considered in [118].

While sparse representations have been primarily used for image reconstruction and recovery, applications of these techniques can be also found in the field of computer vision. In this context, the main focus lies on the extraction of semantic information. The work reported in [171] addresses the problem of automatic face recognition. For this purpose,

vectorized face images from different subjects under varying illumination conditions constitute the columns of the dictionary $\boldsymbol{\Psi}$. It is assumed that images of the same face lie on a low-dimensional so-called face subspace. Thus the significant coefficients of the sparse representation of some query image are likely to be present at indices that belong to prototype faces from the same subject. As a result, classification or validation can be simply done via thresholding. Further references can be found in [170].

All of the aforementioned problems indicate the versatility of sparse representation modeling, especially for image processing tasks. However, while the concepts of sparsity are easily applicable to a vast amount of different problems, the critical role of choosing a suitable basis or dictionary has been left aside so far. The next sections address this issue with the focus on learning sparse data models.

1.1. Background on Learning Sparse Data Models

In the simplest case, the signal of interest is a linear combination of basis vectors from an orthogonal basis. The example given in Eq. (1.1) shows the expansion of the signal in the orthogonal Fourier basis. Figure 1.2 illustrates another famous transform, namely the wavelet decomposition [94]. The wavelet basis is constructed based on the repeated translation and scaling of a pair of localized functions. This pair is referred to as the low frequency scaling and the high frequency wavelet function. Both approaches share the important advantage that the desired representation can be computed efficiently even without forming and applying the matrix $\boldsymbol{\Psi}$. However, efficiency is achieved at the cost of loosing adaptivity. While the Fourier basis is suitable to sparsely represent smooth signals without sharp discontinuities, the wavelet basis is efficient in describing piecewise smooth signals. To account for these drawbacks, the literature offers a wide variety of other signal transforms. Candès and Donoho [21] proposed the curvelet transform which is efficient in representing smooth curves. The original description as a continuous transform was further extended to the discrete case in [152]. To directly account for the discrete two-dimensional nature of images, the contourlet transform has been introduced by Do and Vetterli [37]. Compared to the curvelet approach, the discrete formulation of contourlets reduces the complexity and redundancy. The non-adaptivity of the aforementioned transforms was addressed by Le Pennec and Mallat, who introduced the bandelet transform [109]. This signal adaptive transform is able to directly exploit the existing geometric structures, e.g. edges in an image. Further references on analytic transforms can be found in [138].

1.1.1. Sparse Synthesis Model

The quest for better adapted transforms has led to the emergence of so-called *dictionaries*. A dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K] \in \mathbb{R}^{N \times K}$ is a matrix whose columns $\{\boldsymbol{d}_i\}_{i=1}^K$ represent the basic elements also known as prototypes or atoms. Analogous to the expansion in a basis, a signal can be represented as a linear combination of the columns of $\boldsymbol{D}$, i.e., we have

$$\boldsymbol{s} = \boldsymbol{D}\boldsymbol{x}, \quad \boldsymbol{x} \text{ is sparse.} \tag{1.9}$$

In other words, the signal is synthesized from the basic elements provided by the dictionary. That is why in the literature, this model is known as the *sparse synthesis model*. The geometric interpretation of this model is straightforward. Notice that the coefficient vector $\boldsymbol{x}$ is assumed to be sparse, thus only a subset of the possible L_s columns out of $\boldsymbol{D}$ is selected to describe the signal and consequently, the subspace the signal lies in is determined by the non-zero entries in $\boldsymbol{x}$.

The index set that contains the locations of the significant coefficients is denoted as the support Y, i.e., we have $\mathrm{Y}(\boldsymbol{x}) = \{i | x_i \neq 0\}$ with cardinality $|\mathrm{Y}(\boldsymbol{x})| = L_s$. Depending on the sparsity level L_s, the number of possible subspaces also varies. If we denote the subspace spanned by the columns of $\boldsymbol{D}_\mathrm{Y}$ by $\mathfrak{A}_\mathrm{Y} := \mathrm{span}(\boldsymbol{d}_i, i \in \mathrm{Y})$, the sparse synthesis model comprises the union of all possible $\binom{K}{L_s}$ subspaces (Union-of-Subspaces (UoS) model [85]):

$$\boldsymbol{s} \in \bigcup_{\mathrm{Y}:|\mathrm{Y}|=L_s} \mathfrak{A}_\mathrm{Y}. \tag{1.10}$$

Beginning with the work of Mallat and Zhang [95], especially overcomplete dictionaries or frames, where $K > N$, have gained a lot of attention in the research community. The terminology of dictionaries provides a significant difference to bases in the sense that the signal representation is no longer unique meaning that dictionaries exhibit some sort of redundancy. The easiest way to realize an overcomplete dictionary is a concatenation of two different bases, i.e., $\boldsymbol{D} = [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2] \in \mathbb{R}^{N \times 2N}$, which is closely related to the signal separation approach where different representations are used to encode different content types. Evidently, with two bases at hand the expressiveness of the dictionary is significantly increased.

So far, the design of a suitable dictionary solely relies on a mathematical model of the data. This strategy implies that the characteristics of the signals are known a priori and that the sought signals indeed have a sparse representation with respect to the chosen dictionary. An even better approach is to learn a representation that directly adapts to the particular signal class. The resulting dictionary should provide more accurate signal approximations

with higher sparsity rates (less significant coefficients) and thus a better performance in signal processing applications. However, the advantage of trained dictionaries comes at the cost of two serious drawbacks. First, the decomposition requires additional algorithms that determine the sparse code. Second, to tackle the dictionary learning problem only small sized signals can be considered. As a consequence, in the context of image processing most of the classical approaches follow a patch-based approach where the image is decomposed in (non-)overlapping patches of small size.

The classical dictionary learning problem amounts to minimizing a cost function that is composed of two global objectives, namely the error between the input signal and its approximation with respect to the learned dictionary and second, a function that measures the activity of the coefficient sequence. If we let $\boldsymbol{S} \in \mathbb{R}^{n \times T}$ denote the matrix which holds T vectorized image patches of size $\sqrt{n} \times \sqrt{n}$ as its columns, the classical learning problem can be stated as

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg \min_{\substack{\boldsymbol{D} \in \mathbb{R}^{n \times k} \\ \boldsymbol{X} \in \mathbb{R}^{k \times T}}} \|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \lambda g(\boldsymbol{X}). \tag{1.11}$$

Section 1.2.1 briefly summarizes the major dictionary learning methods presented in the literature.

1.1.2. Co-Sparse Analysis Model

The sparse synthesis model, while well established in the research community and often used in practice, has a closely related counterpart called the *co-sparse analysis model*, whose impact in sparse data modeling has been left unconsidered a very long time. As the name already suggests, instead of synthesizing the signal content the analysis model draws its representational power from an operator that analyzes the signal. To put things formally, the analysis operator $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K]^\top \in \mathbb{R}^{K \times N}$ is applied to the signal $\boldsymbol{s} \in \mathbb{R}^N$ which results in the co-sparse outcome $\boldsymbol{a} \in \mathbb{R}^K$, i.e.,

$$\boldsymbol{\Omega}\boldsymbol{s} = \boldsymbol{a}, \quad \boldsymbol{a} \text{ is co-sparse.} \tag{1.12}$$

In contrary to the synthesis dictionary, the rows $\{\boldsymbol{\omega}_i\}_{i=1}^K$ of the analysis operator constitute the basic elements that describe the structure of the data. The rows of $\boldsymbol{\Omega}$ are also often referred to as *filters*, where the filter response indicates if the signal belongs to the model. The term co-sparsity originates from the fact that in the analysis model the essential information of the subspace the signal lies in is encoded in the zero elements, which is in sharp contrast to the synthesis model. Formally, the co-sparsity is defined as the number of zeros

in $\boldsymbol{\Omega s}$, i.e.,

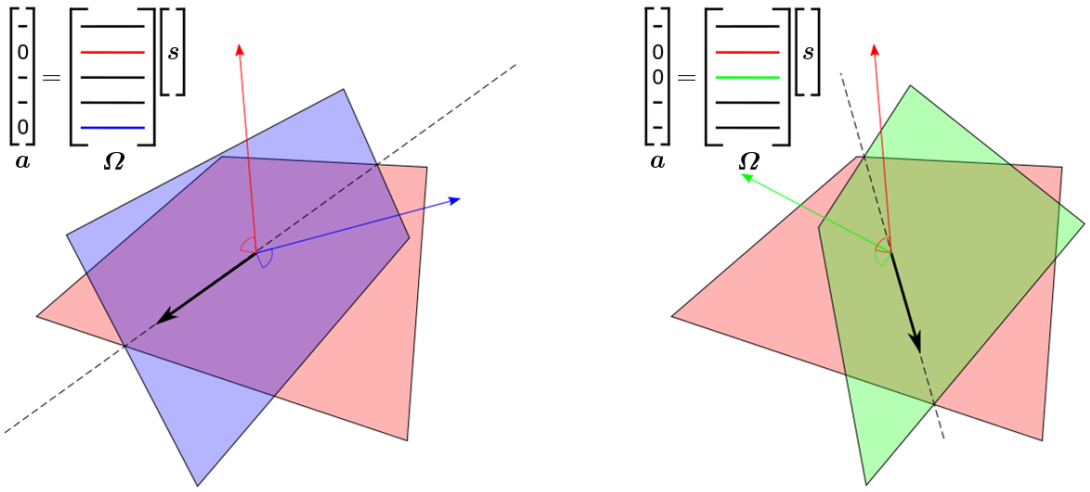
$$\text{co-sparsity}: \quad L_a := K - \|\boldsymbol{\Omega s}\|_0. \tag{1.13}$$

Hence, a signal is considered co-sparse when L_a is large meaning that many coefficients are close to zero. Analogously to the synthesis model, the index set of the zero elements in $\boldsymbol{\Omega s}$ is denoted as the co-support $\Lambda(\boldsymbol{s}) = \{i | \langle \boldsymbol{\omega}_i, \boldsymbol{s} \rangle = 0\}$ with $|\Lambda(\boldsymbol{s})| = L_a$. From a geometrical perspective, a signal that is exactly L_a-co-sparse lies in the orthogonal complement of the rows of $\boldsymbol{\Omega}$ indexed by Λ, i.e., $\boldsymbol{\Omega}_\Lambda \boldsymbol{s} = \boldsymbol{0}$. Consequently, if we denote $\mathfrak{B}_\Lambda := \text{span}(\boldsymbol{\omega}_i, i \in \Lambda)^\perp = \text{Null}(\boldsymbol{\Omega}_\Lambda)$, we have that for the analysis model the signals reside in the union of all possible $\binom{K}{L_a}$ subspaces

$$\boldsymbol{s} \in \bigcup_{\Lambda : |\Lambda| = L_a} \mathfrak{B}_\Lambda, \tag{1.14}$$

each with dimension $N - L_a$. Figure 1.3 schematically illustrates the subspace identification in the analysis model based on signals that reside in $\mathbb{R}^3$. As pointed out above, the signal is characterized by the zero entries in L_a, which in the given example are associated to the colored filters in $\boldsymbol{\Omega}$. Assuming that $\boldsymbol{s}, \boldsymbol{\omega}_i \neq \boldsymbol{0}_N$, a zero filter response is achieved if $\boldsymbol{\omega}_i$ is orthogonal to the signal. Thus, each of the rows can be considered as normal vectors, describing hyperplanes the signal resides in (illustrated by the colored planes in Figure 1.3). Eventually, the signal lies in the hyperplane identified by the collection of normal vectors that lead to zero analysis coefficients. Regarding the $\mathbb{R}^3$ example in Figure 1.3, the signals originate from a one-dimensional subspace, where the different subspaces are clearly determined through the colored filters in $\boldsymbol{\Omega}$.

Although the analysis and synthesis approach look similar, the seminal work of Elad et al. [48] has shed light on the properties of both models, revealing the strong difference between them. The authors first show that in the square non-singular case, i.e., the operator $\boldsymbol{\Omega} \in \mathbb{R}^{N \times N}$ has full rank, both models are equivalent in the sense that $\boldsymbol{\Omega} = \boldsymbol{D}^{-1}$. Even for the undercomplete non-singular case where $K < N$, when $\boldsymbol{s}$ is in the column span of $\boldsymbol{D}$ equivalence is achieved when $\boldsymbol{\Omega} = \boldsymbol{D}^\dagger$, where $\boldsymbol{D}^\dagger$ denotes the Pseudo-Inverse of $\boldsymbol{D}$. The most interesting case however, are overcomplete representations where $K \geq N$. Here, the increased expressiveness achieved through the redundancy of the atoms/filters allows to better reflect the true underlying structure of the signals. Assuming a fixed number of atoms/filters the authors in [101] vividly describe the strong difference between both models in the overcomplete scenario. While the synthesis model includes few low dimensional subspaces and an increasing number of high dimensional subspaces, the analysis model behaves contrary, i.e., the number of low dimensional subspaces representable by



(a) Subspace #1 **(b)** Subspace #2

Figure 1.3.: Analysis Union-of-Subspace (UoS) Model. **(a)** Subspace #1. **(b)** Subspace #2.

the operator $\boldsymbol{\Omega}$ clearly exceeds the number of high dimensional ones. Finally, simulations showing the potential superiority of the analysis model compared to its synthesis counterpart can be found in [48, 101], where numerical results on various image processing applications are given in e.g. [64, 122, 168].

Typical choices for analytically given operators are on the one hand the finite difference operator, that calculates the difference between neighboring pixels. This approach is closely related to the Total Variation (TV) norm minimization [144] which enjoys great popularity in image processing applications. The shift invariant wavelet transform on the other hand is another representative of this group [101]. However, just like for the synthesis model, learning the co-sparse analysis model is a field of active research. Again, typically a patch-based approach is the method of choice to keep the number of free parameters in an acceptable range. With $\boldsymbol{S} \in \mathbb{R}^{n \times T}$ being the same matrix as defined in (1.11) the general analysis operator learning problem reads

$$\boldsymbol{\Omega}^{\star} \in \arg \min_{\boldsymbol{\Omega} \in \mathbb{R}^{k \times n}} g(\boldsymbol{\Omega S}), \tag{1.15}$$

with $g(\cdot)$ denoting again the sparsity inducing function. Various approaches to tackle problem (1.15) are described in the following section.

1.2. State-of-the-Art Learning Algorithms

1.2.1. Dictionary Learning

One of the first dictionary learning approaches is the work presented by Olshausen and Field [104], who tackle the problem in a two stage fashion. First, the coefficients $\boldsymbol{X}$ are determined by minimizing (1.11) with fixed atoms $\boldsymbol{D}$ which can be done for each vectorized patch separately. In the second stage, the current residuals between the input and the approximation are used to globally update the columns in $\boldsymbol{D}$ via iterative gradient descent. The scale ambiguity between the product $\boldsymbol{DX}$, however, causes the norm of $\boldsymbol{d}_i$ to grow to infinity while the sparse coefficients in $\boldsymbol{X}$ approach zero. That is why an additional constraint has to be imposed on the norm of $\boldsymbol{d}_i$ which is usually set to $\|\boldsymbol{d}_i\|_2 = 1$. Although being a relatively simple algorithm, the authors could impressively show that it is possible to learn localized, oriented and bandpass prototype signals from examples that resemble the behavior of cell receptive fields in the mammalian primary visual cortex [104, 105].

The Method of Optimal Directions (MOD) is an algorithm proposed by Engan et al. [49]. The MOD differs from [104] in terms of a more efficient update of the sparse codes and a modified update of the dictionary elements. To be precise, assuming the coefficients $\boldsymbol{X}^{(i)}$ to be known, the next iteration of the dictionary is obtained via

$$\boldsymbol{D}^{(i+1)} = \boldsymbol{S}\boldsymbol{X}^{(i)}(\boldsymbol{X}^{(i)}\boldsymbol{X}^{(i)^\top})^{-1}. \tag{1.16}$$

This update rule can be easily derived by considering the derivative of the residual error $\|\boldsymbol{S} - \boldsymbol{DX}\|_F^2$ with respect to $\boldsymbol{D}$.

A dictionary learning algorithm named *K-SVD* was developed by Aharon et al. [2]. The proposed method is inspired by the vector quantization framework which can be considered as the extreme case where the columns of $\boldsymbol{X}$ are taken from the canonical basis, i.e., all $\boldsymbol{x}_i$ are one-sparse with $\boldsymbol{x}_i = \boldsymbol{e}_j$. If also the codebook $\boldsymbol{D}$ is updated additionally, the whole procedure is known as K-means which gave rise to the name K-SVD. The second part of the name originates from the update scheme. To be precise, the objective of the method reads

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg\min_{\substack{\boldsymbol{D}\in\mathbb{R}^{n\times k}\\ \boldsymbol{X}\in\mathbb{R}^{k\times T}}} \|\boldsymbol{S} - \boldsymbol{DX}\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{x}_i\|_0 \leq L_s \,\forall i, \|\boldsymbol{d}_j\|_2 = 1 \,\forall j, \tag{1.17}$$

where L_s denotes a fixed upper limit on the number of active coefficients in each sparse code representation. Analogous to the aforementioned methods, the algorithm alternates between two steps. In the sparse coding stage, the coefficients are usually determined by applying suitable pursuit algorithms, e.g. Orthogonal Matching Pursuit (OMP). The

second stage comprises a simultaneous dictionary and active coefficients update. For this purpose, the error induced by the k-th atom is considered, which can be calculated via

$$\|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{X}\|_F^2 = \left\| \left(\boldsymbol{S} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{X}_j^{\text{row}} \right) - \boldsymbol{d}_k \boldsymbol{X}_k^{\text{row}} \right\|_F^2 = \|\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{X}_k^{\text{row}}\|_F^2. \tag{1.18}$$

Here, $\boldsymbol{X}_k^{\text{row}}$ denotes the k-th row of the matrix $\boldsymbol{X}$. The aim of the second stage is to find an update of $\boldsymbol{d}_k$ and $\boldsymbol{X}_k^{\text{row}}$ such that the error $\|\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{X}_k^{\text{row}}\|_F^2$ is minimized. This amounts to finding the closest rank-1 approximation of $\boldsymbol{E}_k$ which can be derived easily by means of the Singular Value Decomposition (SVD). However, naively updating both the atom and the selected sparse coefficients, will result in a new row vector $\boldsymbol{X}_k^{\text{row}}$ that is non-sparse. That is why the SVD of the reduced matrix $\widetilde{\boldsymbol{E}}_k$, that only contains the columns indexed by the active non-zero entries in $\boldsymbol{X}_k^{\text{row}}$, is calculated. Eventually, the decomposition $\widetilde{\boldsymbol{E}}_k = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is utilized to determine the new iterate of the atom $\boldsymbol{d}_k = \boldsymbol{u}_1$ and the simultaneous update of the sparse coefficients $\widetilde{\boldsymbol{X}}_k^{\text{row}} = \sigma_{(1,1)} \boldsymbol{v}_1^\top$.

Further references on the topic include the works of Mairal et al. [91, 90, 89] as well as the survey papers [15] and [159].

1.2.2. Analysis Operator Learning

The work of Roth & Black [137] can be considered one of the first attempts that address the problem of learning analysis priors that capture the statistics of natural images. Their approach, named Fields-of-Experts, aims at modeling the probability density distribution of the whole image content as the product of several experts distributions, which are much easier to retrieve. For this purpose, a set of filters is learned whose response to a neighborhood of image patches follows a Student-t probability distribution.

In [106], Ophir et al. propose a simple analysis operator learning scheme that sequentially updates the filters by identifying directions that are orthogonal to a subset of the training data. Denoting $\boldsymbol{S}_t$ as a randomly chosen subset of the training samples the core step of the presented algorithm consists in finding a vector $\boldsymbol{\omega}_i$ that is nearly orthogonal to the set $\boldsymbol{S}_t$. This vector can be easily determined in closed from by taking the eigenvector associated with the smallest eigenvalue of the matrix $\boldsymbol{S}_t \boldsymbol{S}_t^\top$. While being optimal with regard to the orthogonality property, sequentially finding the rows of $\boldsymbol{\Omega}$ in this manner is prone to produce repetitions and linear combinations. That is why the authors propose to prune atoms that do not differ in a certain extent from previously found ones. These heuristics, however, may result in tedious pruning steps which in the worst case end up in deadlock situations.

A successor of the K-SVD algorithm transferred to the analysis model is presented by Ru-

binstein et al. [140, 141]. Comparable to its synthesis counterpart, the proposed approach alternates between updating the columns of the signal matrix $\boldsymbol{S}$ - which emphasizes the algorithm's property to handle noise contaminated measurements - and a sequential update of the filters in $\boldsymbol{\Omega}$. After projecting the signals $\boldsymbol{S}$ onto the orthogonal complement of a subset of rows from $\boldsymbol{\Omega}$ in the first step, a filter update rule similar to [106] is pursued:

$$\boldsymbol{\omega}_j^\star \in \arg\min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}_j^\top \boldsymbol{Y}_J\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\omega}_j\|_2 = 1. \tag{1.19}$$

While $\boldsymbol{Y}$ denotes the matrix of noisy observations as its columns, the index set J identifies the subset of the columns of $\boldsymbol{S}$ that are orthogonal to $\boldsymbol{\omega}_j$. The singular vector corresponding to the smallest singular value of $\boldsymbol{Y}_J$ constitutes the solution. In this way, only those samples $\boldsymbol{Y}_J$ contribute to the update $\boldsymbol{\omega}_j$ that are most orthogonal to it (deviations are only due to noise). As a drawback, to resolve deadlock situations, after each update step the rows of $\boldsymbol{\Omega}$ which are too close to each other or that have too few associated samples are replaced by random vectors. Furthermore, although being able to simultaneously denoise the input while learning the analysis operator, the presented approach remains completely patch-based without any global data fidelity term. Hence, the resulting denoised image must be obtained via simple patch averaging.

Analogous to the task-driven synthesis based approach of Mairal et al. [89], Peyré & Fadili [110] propose a learning algorithm that aims at finding a suitable analysis operator for a given task, e.g. Denoising. To achieve this goal, pairs of clean and noisy signal samples $(\boldsymbol{s}_k, \boldsymbol{y}_k)$ serve as an input for a bi-level programming optimization problem. This framework optimizes the lower level operator learning problem, whose solution simultaneously minimizes the higher level Denoising problem, i.e.,

$$\underset{\boldsymbol{\Omega}}{\text{minimize}} \ \sum_k \tfrac{1}{2}\|\boldsymbol{s}(\boldsymbol{\Omega}, \boldsymbol{y}_k) - \boldsymbol{s}_k\|_2^2, \tag{1.20}$$

$$\text{where} \quad \boldsymbol{s}(\boldsymbol{\Omega}, \boldsymbol{y}_k) = \arg\min_{\boldsymbol{s}} \tfrac{1}{2}\|\boldsymbol{s} - \boldsymbol{y}_k\|_2^2 + g(\boldsymbol{\Omega}\boldsymbol{s}), \tag{1.21}$$

with $g(\cdot)$ denoting the sparsity measure. However, since the desired operator takes the form of a circular convolution, the learning part consists in determining only one filter γ whose shifted versions constitute the rows of $\boldsymbol{\Omega}$.

A similar approach is followed by Chen et al. [28]. Instead of reconstructing individual image patches that are combined via averaging to form the final image, the authors extend the bi-level programming framework such that it provides global image support. One of the main advantages of the bi-level learning problem is the avoidance of any further constraints on $\boldsymbol{\Omega}$. This is because although the trivial solution $\boldsymbol{\Omega} = \boldsymbol{0}_{k \times n}$ minimizes the

lower level problem (1.21), the resulting solution $\boldsymbol{s}(\boldsymbol{\Omega}, \boldsymbol{y}_k) = \boldsymbol{y}_k$ is not a minimizer of the higher level Denoising problem, which forces the algorithm to learn meaningful filters. However, the applicability of the algorithm is rather limited due to its extensive training time. The authors mention a training time of 20 days even for moderate filter sizes of 9×9.

While the aforementioned algorithms avoid trivial solutions either based on heuristics, e.g. pruning or replacement of certain rows in $\boldsymbol{\Omega}$, or by means of a bi-level programming approach, there is a third category that directly imposes constraints on the operator itself. In the work of Yaghoobi et al. [173, 174, 175] constrained analysis operator learning is expressed as an optimization problem that is able to cope with noisy observations $\boldsymbol{Y}$. Formally, the proposed patch based approach reads

$$\{\boldsymbol{\Omega}^\star, \boldsymbol{S}^\star\} \in \arg\min_{\substack{\boldsymbol{\Omega}\in\mathbb{R}^{k\times n}\\ \boldsymbol{S}\in\mathbb{R}^{n\times T}}} \|\boldsymbol{\Omega}\boldsymbol{S}\|_1 + \frac{\lambda}{2}\|\boldsymbol{Y} - \boldsymbol{S}\|_F^2, \qquad \text{s.t.} \quad \boldsymbol{\Omega} \in \mathfrak{C}, \tag{1.22}$$

with the Lagrangian multiplier λ and some constraint set $\mathfrak{C}$ to avoid trivial or meaningless solutions. The authors propose a Uniform Normalized Tight Frame (UNTF) constraint $\mathfrak{C} = \{\boldsymbol{\Omega} \in \mathbb{R}^{k\times n} : \boldsymbol{\Omega}^\top\boldsymbol{\Omega} = \boldsymbol{I}_{n\times n}, \|\boldsymbol{\omega}_i\|_2 = c\,\forall i\}$, which results in a full rank operator whose rows $\boldsymbol{\omega}_i$ exhibit norm c. Equation (1.22) is minimized in an alternating fashion that optimizes for one variable while keeping the other fixed. To fulfill the constraint, at each gradient descent step the operator is projected onto the intersection of the Tight Frames (TF) manifold and the manifold of frames that are Uniform Normalized (UN).

Hawe et al. [64] present an analysis operator learning scheme that utilizes weighted penalty functions to relax the strict UNTF constraint. The suggested constraint set includes full rank matrices with normalized rows, i.e., $\mathfrak{C} = \{\boldsymbol{\Omega} \in \mathbb{R}^{k\times n} : \mathrm{rk}(\boldsymbol{\Omega}) = n, (\boldsymbol{\Omega}\boldsymbol{\Omega}^\top)_{ii} = 1\}$. Since in the overcomplete case, the full rank constraint alone does not prevent the algorithm to learn identical filters, a log barrier function $r(\boldsymbol{\Omega})$ that penalizes the pairwise mutual coherence of the filters has been included, resulting in the optimization problem

$$\boldsymbol{\Omega}^\star \in \arg\min_{\boldsymbol{\Omega}\in\mathbb{R}^{k\times n}} g(\boldsymbol{\Omega}\boldsymbol{S}) + \kappa h(\boldsymbol{\Omega}) + \mu r(\boldsymbol{\Omega}) \qquad \text{s.t.} \quad \boldsymbol{\Omega} \in \mathfrak{C}, \tag{1.23}$$

where $g(\cdot)$ denotes the ℓ_p-norm sparsity inducing function, and $h(\cdot)$ serves as a penalty on the singular values of $\boldsymbol{\Omega}$ to enforce full rank matrices. The constraint set admits a manifold structure known as the oblique manifold. This property has been directly utilized in the optimization strategy, resulting in an efficient conjugate gradient on manifolds approach. Thus, at each iteration the algorithm updates the whole operator at once without any additional projection step as required in [175]. The authors present state-of-the-art results when

applying the learned analysis operator as a regularizer in various inverse problems. However, the presented learning framework only handles vectorized noise free image patches.

1.2.3. Transform Operator Learning

Recently, a closely related concept termed transform operator learning has evolved. In this model, a signal $\boldsymbol{s}$ is assumed to be approximately sparse in the transform domain, i.e., $\boldsymbol{W}\boldsymbol{s} = \boldsymbol{b} + \boldsymbol{e}$, with $\boldsymbol{W}$ denoting the transform operator, $\boldsymbol{b}$ is the sparse representation, and $\boldsymbol{e}$ denotes the representation error in the transform domain that is assumed to be small. In contrary to the classical analysis model, the co-sparse representation in the transform model is not restricted to lie in the range space of $\boldsymbol{W}$. The transform operator learning problem reads

$$\{\boldsymbol{W}^\star, \boldsymbol{B}^\star\} \in \arg\min_{\substack{\boldsymbol{W}\in\mathbb{R}^{k\times n}\\ \boldsymbol{B}\in\mathbb{R}^{k\times T}}} \|\boldsymbol{W}\boldsymbol{S} - \boldsymbol{B}\|_F^2 \qquad \text{s.t.} \quad \|\boldsymbol{b}_i\|_0 \leq L_a \,\forall i, \tag{1.24}$$

where again $\boldsymbol{S}$ and $\boldsymbol{B}$ hold the signals and the transform sparse representations as their columns, respectively. When optimizing for the sparse code $\boldsymbol{B}$ while keeping $\boldsymbol{W}$ fixed, a significant advantage of this formulation consists in the fact that a solution $\boldsymbol{B}^\star$ can be analytically obtained via hard thresholding the product $\boldsymbol{W}\boldsymbol{S}$ such that in each column only the L_a largest entries remain. This motivates an alternating optimization strategy to solve the problem as given in Eq. (1.24). In the following, various approaches that address the transform operator learning problem including the avoidance of trivial solutions are discussed.

Similar to the work of Hawe et al. [64], optimization on the unit sphere to fulfill the row-norm constraint $\|\boldsymbol{w}_i\|_2 = 1\,\forall i$ is also pursued in the work of Dong et al. [38, 39]. However, the authors neglect any possible rank deficiency of the learned operator.

To avoid rank deficient solutions, Ravishankar et al. [123, 126] propose to add a weighted negative log-determinant penalty $-\lambda \log(\det(\boldsymbol{W}))$ to the cost (1.24). To control the scale ambiguity in $\boldsymbol{W}$, an additional squared Frobenius norm penalty on $\boldsymbol{W}$ is inserted in the cost, resulting in the problem

$$\{\boldsymbol{W}^\star, \boldsymbol{B}^\star\} \in \arg\min_{\substack{\boldsymbol{W}\in\mathbb{R}^{k\times n}\\ \boldsymbol{B}\in\mathbb{R}^{k\times T}}} \|\boldsymbol{W}\boldsymbol{S} - \boldsymbol{B}\|_F^2 - \lambda \log(\det(\boldsymbol{W})) + \mu\|\boldsymbol{W}\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{b}_i\|_0 \leq L_a \,\forall i. \tag{1.25}$$

The authors observe that penalizing the Frobenius norm results in superior transforms compared to restricting each of the rows to unit norm. Additionally, the penalties allow to

control the condition number of $\boldsymbol{W}$. Denoting the largest and smallest singular value of $\boldsymbol{W}$ as $\sigma_{\max}$ and $\sigma_{\min}$, respectively, the condition number is defined as $\operatorname{cond}(\boldsymbol{W}) = \sigma_{\max}/\sigma_{\min}$. Especially for signal denoising, the authors argue that a low condition number of the operator is beneficial.

Obviously, the minimization problem (1.25) is only applicable to learn square sparsifying transforms of the form $\boldsymbol{W} \in \mathbb{R}^{n \times n}$. In [125] the learning problem is modified such that it can cope with overcomplete operators. For this purpose, the log-penalty is changed to $-\lambda \log(\det(\boldsymbol{W}^\top \boldsymbol{W}))$ to ensure full column rank of $\boldsymbol{W}$. Furthermore, an additional incoherence penalty $\sum_{j \neq k} |\langle \boldsymbol{w}_j, \boldsymbol{w}_k \rangle|^p$ promotes non-repeating rows. In the work of Wen et al. [168], overcompleteness is modeled as a union of square matrices, i.e., the algorithm learns a collection of K well-conditioned square transforms $\{\boldsymbol{W}_i\}_{i=1}^K$. This is due to the fact that the solution of the transform update sub problem can be derived in closed form if $\boldsymbol{W}$ is square. The union of transforms model requires a clustering step, that assigns each sample to the transform $\boldsymbol{W}_i$ that provides the smallest sparsification error compared to all existing transforms. Extensions of the sparsifying transform learning approach to mini-batch, or one-sample online learning are given in [130, 131, 128].

1.3. Formulation of the Research Problem

Nowadays, the collection and storage of large multidimensional data sets can be achieved at very low costs. On the other hand, processing the data and extracting useful information becomes more elaborate especially in terms of memory storage resources and computation load. Since the number of data entries grows exponentially with the number of dimensions, applying standard representation learning approaches that require vectorized samples are no longer suitable from a computational point of view. Not only learning the model, but also applying the model to the data at hand rapidly becomes prohibitive. Moreover, these vectorization approaches neglect the intrinsic structure of multidimensional patterns frequently present especially in image data.

Many analytically defined sparse transformations like DFT, DCT, and DWT enjoy great popularity in the image processing community because they offer fast implementations. This can be attributed to the fact that multidimensional transformations can be easily calculated by sequentially computing one-dimensional transformations along each dimension separately. This *separability* property significantly reduces the computational complexity. Consequently, imposing a separability constraint on the model is of highest interest to combine the best of both worlds, i.e., fast computations and accelerated learning schemes on the one hand, and a better adaption to the signal class on the other hand.

In this thesis, I will focus on the analysis model. The following challenges and problems regarding the task of learning a structured co-sparse data model for multidimensional signals are addressed in this work:

Computational Complexity. In order to reduce the computational complexity during learning and to handle multidimensional signals, the structural constraint has to be directly integrated into the learning process. The aim is to obtain a separable analysis operator that shares the same beneficial properties that have already be shown useful in the literature of non-structured learning approaches.

The computational effort for iterative gradient descent learning approaches is also strongly influenced by the size of the training set. Redundant samples will unnecessarily slow down the learning process due to expensive calculations of the objective function and the gradient. In addition, theoretical considerations show that restricting the set of feasible solutions will have a direct impact on the number of samples needed for a reliable estimate of the model. Hence, the learning algorithm has to be adjusted accordingly.

Although being efficiently learnable, it remains unclear if the separable analysis model provides compatible results when applied as a regularizer in common inverse problems in imaging. That is why the potential tradeoff between the reduced computational complexity on the one hand, and the achievable performance on the other hand has to be explored.

Applicability. One of the most important aspects concerning learning algorithms is the determination of suitable parameters. A universally applicable algorithm is characterized by the fact that the parameters do not have to be fine-tuned with respect to the task at hand. Moreover, the utilized optimization framework should easily adapt to varying settings without the need for a tedious parameter search.

Usually, the usefulness of a particular parameter set is assessed based on the performance of the learned model with respect to some particular problem, e.g. Denoising, which further decelerates the evaluation process. This motivates to examine the generalization behavior of the learned co-sparse analysis model from a task independent perspective. By implication, an operator that generalizes well should also provide good performance if it is used in a sparsity prior to regularize inverse problems.

Mapping the input signal to a sparse representation is also the key in the Sparse Auto-Encoder framework. That is why I want to explore the required ingredients to leverage this concept to regularize the learning process such that a meaningful Co-sparse Auto-Encoder whose encoder part follows the co-sparse analysis model is attained.

Blind Learning. In many scenarios, noise-free training signals are difficult to acquire or even simply not available. Besides, the underlying measurement process often demands global image support which prohibits to tackle the model learning problem in a purely patch-based manner. For that reason, the learning framework has to be extended in order to meet these requirements.

1.4. Contributions

In the first part of this thesis, I introduce a separable analysis operator learning algorithm that allows to efficiently learn filters for multidimensional training data. An example of a learned separable operator, that is applicable to 2D signals is shown in Figure 1.4. The applied cost function is motivated by the work of Hawe et al. [64] who use log-barrier penalty functions to regularize the solution. I show that the separability constraint can be straightforwardly integrated into the learning framework by restricting the single filter matrices, which are applied to each signal dimension separately, to the constraint set. To avoid the normalization of the filters after each update step, the optimization algorithm directly utilizes the product of spheres manifold structure. As opposed to [64], I propose a geometric Stochastic Gradient Descent (SGD) implementation that processes the training samples in a mini-batch fashion. This approach avoids the determination of a fixed training set with appropriate size beforehand which also renders the algorithm ready for online learning scenarios. Together with an Armijo based line search technique adapted to the SGD setting, the algorithm converges quickly without the need for manually adjusting the step size. Furthermore, numerical results show that the performance of the separable analysis model, when it is used as a prior in a Denoising problem, is on par compared to various non-separable approaches, which highlights the benefit of the proposed approach.

The sequential processing of the samples in the SGD implementation is also advantageous with regard to the sample complexity. In the second part, I show empirically that a separable ground truth operator can be retrieved from far fewer training samples if the structural constraint is directly exploited in the optimization. This result confirms the assumption that reducing the number of the model parameters goes hand in hand with a significant reduction in the amount of samples needed in the learning task.

In order to assess the generalization behavior of the learned model, I utilize the Estimated Kullback-Leibler divergence between the distribution of the original training data and the distribution of signals that are generated with respect to the learned operator. It can be observed that this measure correlates with the performance of the model in inverse problem regularization, which eventually enables a task independent evaluation of the applicability.

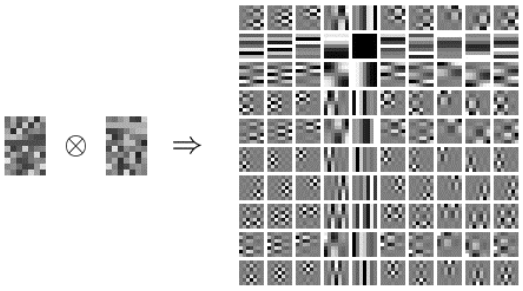


Figure 1.4.: Example of a Separable Analysis Operator. For visualization purposes, the obtained separable filters are as shown as 2D kernels. Gray pixel values correspond to zero filter entries.

Within this thesis, I also present a combined learning and reconstruction approach. Experiments on different image reconstruction tasks show that the framework is able to adaptively learn the analysis model even from undersampled and corrupted measurements. The applied Conjugate Gradient (CG) optimization scheme allows to flexibly exchange the data fidelity term such that it accounts for varying noise distributions, e.g. additive, impulsive or multiplicative noise. This feature renders the presented approach a powerful universal learning framework for various scenarios in imaging, which is not restricted to clean image data or specific measurement matrices.

The last part of the thesis considers learning a separable analysis operator from a Sparse Auto-Encoder (SAE) perspective. Interestingly, by restricting the norm of the weights in the decoder, the condition number of the encoder matrix can be controlled, which has been proven beneficial if the operator is utilized to regularize inverse problems in imaging. Analogously to the experiments conducted before, the learned encoder matrix is used in a co-sparsity prior to regularize the solution of a Compressed Sensing and Denoising problem. The numerical results show that the SAE learned analysis operator indeed provides stable reconstruction performance comparable to operators obtained with the conventional learning setting, however, without the need for additional penalty functions.

1.5. Thesis Outline

The thesis continues with an introduction to multidimensional signals, separability, and optimization on manifolds in Chapter 2. This chapter is intended to explain the basic concepts used throughout the thesis. Afterwards, Chapter 3 summarizes the related work on learning sparse data models with separable structures. Additionally, the prior art on adaptive learning strategies and approaches to combine artificial neural networks and sparse data models are considered. In Chapter 4, the proposed separable analysis operator learn-

ing algorithm is introduced and evaluated. Chapter 5 presents an empirical evaluation of the sample complexity in relation to the standard non-separable case. Furthermore, the model generalization is addressed. The simultaneous learning and reconstruction approach is outlined in Chapter 6. The chapter includes various experiments on solving inverse problems with varying noise distributions. In Chapter 7, the learning problem is examined from the Sparse Auto-Encoder perspective. The thesis concludes with a summary in Chapter 8.

Chapter 2.

Mathematical Preliminaries

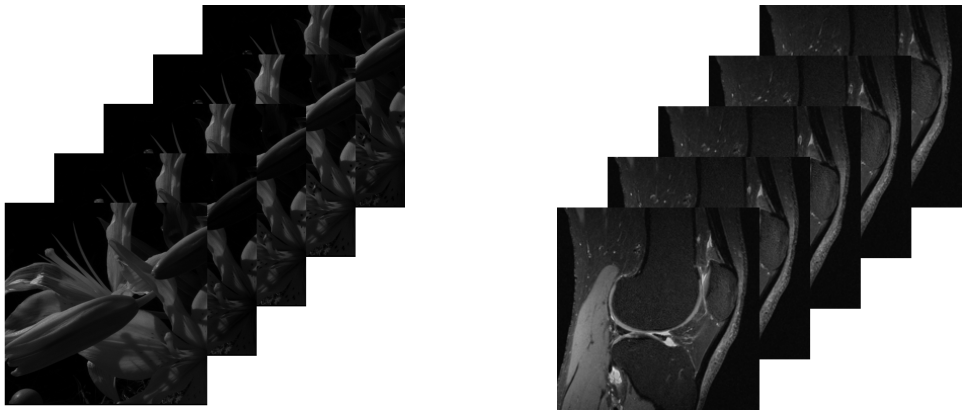
In order to provide a self-contained work, this chapter introduces some important concepts that will be used throughout the entire thesis. In the first part, the structure of multidimensional signals as well as the separability of matrices is explained. Afterwards, the key components of a geometric gradient descent on manifolds along with an extension to the conjugate gradient method are presented.

2.1. Multidimensional Signals and Separability

Multidimensional signals are referred to as tensors, where the order of a tensor is the number of dimensions, also known as ways or modes. Following this notation, vectors can be considered as tensors of order one, while matrices represent tensors of order two. Tensors, whose modes are all the same size are called *cubical*.

A typical representative of multidimensional signals in image processing could be for example a hyperspectral image, where I_1 and I_2 describe the spatial resolution of the image with I_3 indicating the spectral resolution, i.e., the number of acquired wavelengths as shown in Figure 2.1a. As another example, think of a 3D Magnetic Resonance Image of the knee as illustrated in Figure 2.1b. Here, all three dimensions I_1, I_2, and I_3 encode the spatial information with the same physical meaning in all dimensions.

Multiway component analysis addresses the demand of (i) retaining the multidimensional structure of the data and (ii) providing tools that scale well with the number of dimensions, i.e., being computationally tractable. As a result, various tensor decomposition techniques have been proposed in the literature, where the reader is referred to the work of Cichocki et al. [30] who provide a comprehensive introduction into this topic. A survey of general methods to decompose a tensor into low dimensional components can be found in the work of Kolda & Bader [76]. For ease of notation, in this work I will stick to the *n-mode (matrix) product* which denotes the multiplication of an arbitrary tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_V}$ with a matrix $\boldsymbol{Q} \in \mathbb{R}^{J \times I_n}$ along the n-th dimension by $\mathcal{U} \times_n \boldsymbol{Q}$. The result



(a) Hyperspectral image **(b)** Volumetric MRI scan

Figure 2.1.: Example of image data that has a tensor structure. **(a)** Hyperspectral image with several wavelength subbands. **(b)** Volumetric MRI scan of the human knee.

is of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_V$ and can be obtained elementwise via

$$[\mathcal{U} \times_n \boldsymbol{Q}]_{i_1 \cdots i_{n-1}\, j\, i_{n+1} \cdots i_V} = \sum_{i_n=1}^{I_n} u_{i_1 i_2 \cdots i_V}\, q_{j i_n}. \tag{2.1}$$

To offer a better understanding of this concept, the n-mode product for a three dimensional tensor is illustrated in Figure 2.2, where the tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is multiplied by the three matrices $\{\boldsymbol{Q}_i\}_{i=1}^{3}$ along each dimension which results in the output $\mathcal{W}$.

The n-mode product $\mathcal{W} = \mathcal{U} \times_1 \boldsymbol{Q}_1 \times_2 \boldsymbol{Q}_2 \ldots \times_V \boldsymbol{Q}_V$ can be rewritten as a matrix-vector product using the Kronecker product [1] $\otimes$ and the vec-operator[2] such that

$$\operatorname{vec}(\mathcal{W}) = (\boldsymbol{Q}_1 \otimes \boldsymbol{Q}_2 \otimes \cdots \otimes \boldsymbol{Q}_V) \cdot \operatorname{vec}(\mathcal{U}). \tag{2.2}$$

[1] The Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ of the matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times L}$ yields a matrix of size $(IK) \times (JL)$ defined by

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1J}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2J}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & a_{I2}\boldsymbol{B} & \cdots & a_{IJ}\boldsymbol{B} \end{bmatrix}$$

[2] The vec-operator rearranges the entries of a tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_V}$ into the column vector $\operatorname{vec}(\mathcal{U}) \in \mathbb{R}^{I_1 I_2 \cdots I_V}$

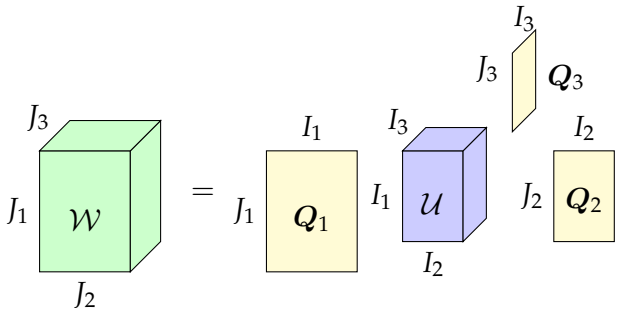


Figure 2.2.: The n-mode product $\mathcal{W} = \mathcal{U} \times_1 \boldsymbol{Q}_1 \times_2 \boldsymbol{Q}_2 \times_3 \boldsymbol{Q}_3$ with tensors $\mathcal{W}, \mathcal{U}$ and matrices $\boldsymbol{Q}_i$, $i = 1, 2, 3$.

In addition to the n-mode product, many authors make use of the so-called *n-mode matrix unfolding* that enables to map a given tensor to a matrix of appropriate size and to apply simple matrix-matrix operations. The matrix unfolding along the V^{th} dimension of the tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_V}$ is denoted by $\text{unf}(\mathcal{U}, V) \in \mathbb{R}^{I_V \times (\prod_{j \neq V} I_j)}$. With the help of this concept, equation (2.2) can be rewritten as

$$\text{unf}(\mathcal{W}, V) = \boldsymbol{Q}_V \cdot \text{unf}(\mathcal{U}, V) \cdot (\boldsymbol{Q}_1 \otimes \cdots \otimes \boldsymbol{Q}_{V-1})^\top . \tag{2.3}$$

In Figure 2.3, the unfolding of the third order tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is illustrated.

In this context, the matrix $\boldsymbol{Q} = (\boldsymbol{Q}_1 \otimes \boldsymbol{Q}_2 \otimes \cdots \otimes \boldsymbol{Q}_V) \in \mathbb{R}^{\prod_i J_i \times \prod_i I_i}$ is referred to as a *separable* matrix, that can be fully described by the small matrices $\{\boldsymbol{Q}_i\}_{i=1}^V$. To enhance the readability, in the subsequent chapters I will frequently make use of the mapping

$$\begin{aligned} \iota : \mathbb{R}^{J_1 \times I_1} \times \mathbb{R}^{J_2 \times I_2} \times \cdots \times \mathbb{R}^{J_V \times I_V} &\to \mathbb{R}^{\prod_i J_i \times \prod_i I_i} \\ (\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_V) &\mapsto \boldsymbol{Q}_1 \otimes \boldsymbol{Q}_2 \otimes \cdots \otimes \boldsymbol{Q}_V. \end{aligned} \tag{2.4}$$

In this manner, the construction of a separable matrix $\boldsymbol{Q} = \boldsymbol{Q}_1 \otimes \boldsymbol{Q}_2 \otimes \cdots \otimes \boldsymbol{Q}_V$ can be compactly written as $\boldsymbol{Q} = \iota(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_V)$.

Synthesis Model

In the case of a separable dictionary, equivalence to the model presented in (1.9) is achieved by means of the Kronecker product of the dictionaries $\{\boldsymbol{D}_i \in \mathbb{R}^{N_i \times K_i}\}_{i=1}^V$. Denoting $\text{vec}(\mathcal{S}) \in \mathbb{R}^{N_1 N_2 \cdots N_V}$ and $\text{vec}(\mathcal{X}) \in \mathbb{R}^{K_1 K_2 \cdots K_V}$ as the vectorized versions of the input data

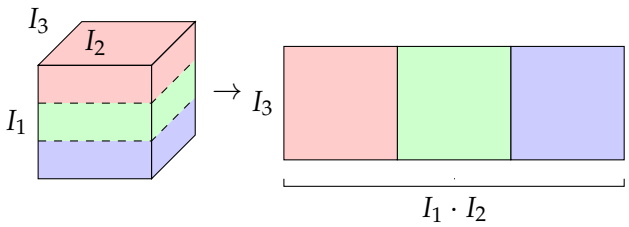


Figure 2.3.: The 3-mode matrix unfolding of a 3-tensor.

and the sparse code, respectively, (3.1) can be rewritten as

$$\operatorname{vec}(\mathcal{S}) = (\boldsymbol{D}_1 \otimes \boldsymbol{D}_2 \otimes \cdots \otimes \boldsymbol{D}_V) \operatorname{vec}(\mathcal{X}). \tag{2.5}$$

Analysis Model

Analogously to (2.5), the co-sparse analysis representation can be obtained in vectorized form via

$$\operatorname{vec}(\mathcal{A}) = (\boldsymbol{\Omega}_1 \otimes \boldsymbol{\Omega}_2 \otimes \cdots \otimes \boldsymbol{\Omega}_V) \operatorname{vec}(\mathcal{S}), \tag{2.6}$$

where $\{\boldsymbol{\Omega}_i \in \mathbb{R}^{K_i \times N_i}\}_{i=1}^{V}$ denote the analysis operators for each of the V dimensions.

2.2. Geometric Optimization

The ultimate goal of analysis operator learning is to find an operator $\boldsymbol{\Omega} \in \mathbb{R}^{k \times n}$ that provides co-sparse representations of the input signals. Referring to the learning problem (1.15), minimizing the sparsity measure $g(\boldsymbol{\Omega}\boldsymbol{S})$ will result in the zero matrix, i.e., $\boldsymbol{\Omega}^\star = \boldsymbol{0}_{k \times n}$. Obviously, this solution does not provide any useful information about the analyzed signals. Hence, to avoid this trivial solution, it is common practice to restrict the norm of the rows $\boldsymbol{\omega}_i \in \mathbb{R}^n$ of the operator to unit value, i.e., we have $\|\boldsymbol{\omega}_i\|_2 = 1 \, \forall i$.

Interestingly, with the unit norm constraint at hand, the transpose $\boldsymbol{\Omega}^\top$ admits a well defined manifold structure. The set of matrices $\boldsymbol{\Xi}$ whose columns have unit ℓ_2-norm is known as the Oblique Manifold (OB) denoted as $\mathsf{OB}(n,k) = \{\boldsymbol{\Xi} \in \mathbb{R}^{n \times k} : (\boldsymbol{\Xi}^\top \boldsymbol{\Xi})_{ii} = 1, \, i = 1, \ldots, k\}$. Since each column in $\boldsymbol{\Xi}$ resides in $\mathbb{S}^{n-1} = \{\boldsymbol{\xi} \in \mathbb{R}^n : \|\boldsymbol{\xi}\|_2 = 1\}$, geometrically, the same manifold can be described by the product of k spheres identified via $\mathsf{S}(n,k)$. Throughout this work, both definitions $\mathsf{OB}(n,k)$ and $\mathsf{S}(n,k)$ are used interchangeably.

In this work, I will focus on iterative methods to find the optimizer of the learning problem. In general, given the function $f(\boldsymbol{\Xi}) : \mathbb{R}^{n\times k} \to \mathbb{R}$, line search methods aim at iteratively finding an update of the matrix $\boldsymbol{\Xi}$ in the search direction $\boldsymbol{H} \in \mathbb{R}^{n\times k}$ such that the function value decreases, i.e., $f(\boldsymbol{\Xi}^{(t+1)}) < f(\boldsymbol{\Xi}^{(t)})$. The standard line search method applied in the ambient Euclidean space reads

$$\boldsymbol{\Xi}^{(t+1)} = \boldsymbol{\Xi}^{(t)} + \alpha^{(t)}\boldsymbol{H}^{(t)}, \tag{2.7}$$

where $\alpha^{(t)}$ denotes the step size that leads to a sufficient decrease of the cost function. Eventually, with the choice of $\boldsymbol{H}^{(t)} = -\nabla_{\boldsymbol{\Xi}} f(\boldsymbol{\Xi}^{(t)})$ equation (2.7) gives rise to the Gradient Descent (GD) or Steepest Descent method for optimizing the function $f(\boldsymbol{\Xi})$.

Following this optimization strategy, the probably easiest way to fulfill the requirement of normalized columns is to project the update $\boldsymbol{\Xi}^{(t+1)}$ back to the product of spheres manifold via normalizing the columns to unit norm after each update step. However, since in the considered setting where the geometry of the manifold is known, another promising strategy consists in directly optimizing on the unit sphere. This approach avoids the projection step introduced above, which potentially result in a reduced number of iterations required until convergence.

In the following, necessary ingredients for an optimization strategy that utilizes the geometric structure are outlined. Besides a simple gradient descent algorithm, I will also present the adaptation of the CG method to the manifold setting. For more general insights into optimization on manifolds, the reader is referred to [1] and [10].

2.2.1. Optimization on the Sphere

To derive the elements of the proposed geometric optimization framework, the concept of tangent spaces has to be introduced first. Formally, the set $\mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1} = \{\boldsymbol{v} \in \mathbb{R}^n : \boldsymbol{v}^\top\boldsymbol{\xi} = 0\}$ denotes the tangent space at the point $\boldsymbol{\xi}$, with $\boldsymbol{v}$ being a tangent vector. In order to calculate distances and lengths, the tangent space is endowed with an inner product which in the considered case is simply given by the Euclidean metric $\boldsymbol{v}^\top\boldsymbol{\zeta}$ for $\boldsymbol{v}, \boldsymbol{\zeta} \in \mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1}$. Equipped with this metric, the manifold is called *Riemannian* manifold.

The element in the tangent space $\mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1}$ that points in the direction of steepest ascent of some cost function f defined on a Riemannian manifold is denoted as the *Riemannian* gradient. Starting from the Euclidean gradient, the Riemannian gradient at $\boldsymbol{\xi}$ can be computed via the orthogonal projection of the Euclidean gradient onto the tangent space $\mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1}$. The

orthogonal projection of a vector $\boldsymbol{q} \in \mathbb{R}^n$ onto the tangent space $\mathtt{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1}$ is obtained via

$$\Pi_{\mathtt{T}_{\boldsymbol{\xi}}\mathbb{S}^{n-1}}(\boldsymbol{q}) = \left(\boldsymbol{I}_n - \boldsymbol{\xi}\boldsymbol{\xi}^\top\right)\boldsymbol{q}. \tag{2.8}$$

Eventually, the Riemannian gradient for the j-th column $\boldsymbol{\xi}_j$ of the matrix $\boldsymbol{\Xi}$ is denoted as

$$\mathtt{G}(\boldsymbol{\xi}_j) = \Pi_{\mathtt{T}_{\boldsymbol{\xi}_j}\mathbb{S}^{n-1}}\left(\nabla_{\boldsymbol{\xi}_j} f(\boldsymbol{\Xi}^{(t)})\right), \tag{2.9}$$

which will serve as a suitable choice for the search direction. However, searching along the line $-\mathtt{G}(\boldsymbol{\xi}_j)$ still neglects the spherical structure of the set of feasible solutions. That is why in the geometric optimization framework, the line search is performed along geodesics, which can be considered the generalization of a straight line on the manifold. Regarding the given product of spheres setting, the geodesics reduce to great circles on the sphere that allow for a computationally feasible parametrization. The geodesic from $\boldsymbol{\xi}_j$ along the direction $\boldsymbol{h}_j \in \mathtt{T}_{\boldsymbol{\xi}_j}\mathbb{S}^{n-1}$ is denoted as $\gamma(\boldsymbol{\xi}_j, \boldsymbol{h}_j, l)$, where l denotes the arc length. Thus, the parametrization of the great circle reads

$$\gamma(\boldsymbol{\xi}_j, \boldsymbol{h}_j, l) = \begin{cases} \boldsymbol{\xi}_j, & \text{if} \|\boldsymbol{h}_j\|_2 = 0 \\ \boldsymbol{\xi}_j \cos(l\|\boldsymbol{h}_j\|_2) + \boldsymbol{h}_j \frac{\sin(l\|\boldsymbol{h}_j\|_2)}{\|\boldsymbol{h}_j\|_2)}, & \text{otherwise.} \end{cases} \tag{2.10}$$

All the aforementioned concepts can be straightforwardly extended to the product of spheres manifold via applying (2.9) and (2.10) to all columns of $\boldsymbol{\Xi}$ consecutively.

To conclude, regarding the geodesics for all columns in $\boldsymbol{\Xi}$, the generalization of the update formula (2.7) for an update step on the product of spheres manifold can be compactly written as

$$\boldsymbol{\Xi}^{(t+1)} = \Gamma(\boldsymbol{\Xi}^{(t)}, \boldsymbol{H}^{(t)}, \alpha^{(t)}), \tag{2.11}$$

where $\Gamma(\boldsymbol{\Xi}, \boldsymbol{H}, \alpha)$ denotes the set of geodesics for each of the $j = 1, \ldots, k$ columns of $\boldsymbol{\Xi}$. Finally, setting $\boldsymbol{H}^{(t)} = -\mathtt{G}(\boldsymbol{\Xi}^{(t)})$ results in a Gradient Descent step on the manifold.

2.2.2. Conjugate Gradient on Manifold

In CG methods, the search direction $\boldsymbol{h}_j^{(t)} \in \mathtt{T}_{\boldsymbol{\xi}_j^{(t)}}\mathbb{S}^{n-1}$ is a linear combination of the current gradient at iteration (t) and the previous search direction $\boldsymbol{h}_j^{(t-1)} \in \mathtt{T}_{\boldsymbol{\xi}_j^{(t-1)}}\mathbb{S}^{n-1}$. Since the current gradient and the previous search direction do not lie in the same tangent

space, linearly combining them only makes sense after mapping $\boldsymbol{h}_j^{(t-1)}$ onto $\mathtt{T}_{\boldsymbol{\xi}_j^{(t)}}\mathbb{S}^{n-1}$. The identification of different tangent spaces is done by the so-called parallel transport which is denoted by $\mathtt{p}_{\boldsymbol{v}}^{(t)} := \mathtt{p}(\boldsymbol{v}, \boldsymbol{\xi}_j^{(t-1)}, \boldsymbol{h}_j^{(t-1)}, l)$, which transports a tangent vector $\boldsymbol{v}$ along a geodesic $\gamma(\boldsymbol{\xi}_j^{(t-1)}, \boldsymbol{h}_j^{(t-1)}, l)$ emanating from $\boldsymbol{\xi}_j^{(t-1)}$ in the direction $\boldsymbol{h}_j^{(t-1)}$ to the tangent space $\mathtt{T}_{\boldsymbol{\xi}_j^{(t)}}\mathbb{S}^{n-1}$. The transport along a great circle on the unit sphere reads

$$\mathtt{p}(\boldsymbol{v}, \boldsymbol{\xi}_j, \boldsymbol{h}_j, l) = \boldsymbol{v} - \frac{\boldsymbol{v}^\top \boldsymbol{h}_j}{\|\boldsymbol{h}_j\|_2} \left(\boldsymbol{\xi}_j \|\boldsymbol{h}_j\|_2 \sin(l\|\boldsymbol{h}_j\|_2) + \boldsymbol{h}_j (1 - \cos(l\|\boldsymbol{h}_j\|_2)) \right). \tag{2.12}$$

Again, the generalization to all the columns of $\boldsymbol{\Xi}^{(t-1)}$ is denoted by $\mathtt{P}_{\boldsymbol{\Upsilon}}^{(t)} := \mathtt{P}(\boldsymbol{\Upsilon}, \boldsymbol{\Xi}^{(t-1)}, \boldsymbol{H}^{(t-1)}, l)$. With the concept of parallel transport at hand, the new search direction $\boldsymbol{H}^{(t)}$ in the CG update can be obtained via

$$\boldsymbol{H}^{(t)} = -\mathtt{G}(\boldsymbol{\Xi}^{(t)}) + \beta^{(t)} \mathtt{P}_{\boldsymbol{H}^{(t-1)}}^{(t)}, \tag{2.13}$$

with the CG-update parameter $\beta^{(t)}$. Typical choices for $\beta^{(t)}$ that can be found in the literature are the Hestenes-Stiefel (HS), Polak-Ribière (PR), and Fletcher-Reeves (FR) update formulae [103]. Exemplarily, with the shorthand notations $\mathtt{G}^{(t)} := \mathtt{G}(\boldsymbol{\Xi}^{(t)})$, as well as $\mathtt{U}^{(t)} = \mathtt{G}^{(t)} - \mathtt{P}_{\mathtt{G}^{(t-1)}}^{(t)}$, the manifold adaption of the Hestenes-Stiefel formula reads

$$\beta_{\mathrm{HS}}^{(t)} = \frac{\langle \mathtt{G}^{(t)}, \mathtt{U}^{(t)} \rangle}{\langle \mathtt{P}_{\boldsymbol{H}^{(t-1)}}^{(t)}, \mathtt{U}^{(t)} \rangle}, \tag{2.14}$$

where $\langle \boldsymbol{U}, \boldsymbol{V} \rangle := \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{V})$ denotes the Riemannian metric on $\mathtt{OB}(n,k)$ induced by the Euclidean metric of the embedding space $\mathbb{R}^{n \times k}$.

Chapter 3.

Related Work

3.1. Separable Dictionary and Analysis Operator Learning

In the subsequent part, sparsity-inspired learning approaches that address the multiway structure of the data are considered. Compared to the vectorization approaches, these methods provide crucial advantages. First, compared to non-structured approaches, fewer parameters have to be estimated in the learning stage. Second, the separable structure allows for an efficient application of the model.

3.1.1. Synthesis Model

One straightforward application of tensor decompositions with regard to sparse multidimensional data representations is proposed by Caiafa & Cichocki [17]. If the input samples are given as V-dimensional tensors $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_V}$, the sparse data model reads

$$\mathcal{S} = \mathcal{X} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \ldots \times_V \boldsymbol{D}_V, \quad \mathcal{X} \text{ is sparse.} \tag{3.1}$$

Here, the dictionaries $\boldsymbol{D}_v \in \mathbb{R}^{n_v \times k_v}$ with $v = 1, \ldots, V$ are two dimensional matrices applied to the V distinct dimensions of the sparse multidimensional tensor $\mathcal{X} \in \mathbb{R}^{k_1 \times k_2 \times \ldots \times k_V}$. Since the authors in [17] use analytically given dictionaries, their main concern is about efficient sparse coding schemes to compute the representation $\mathcal{X}$ without the explicit storage of the dictionary $\boldsymbol{D} = \iota(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_V)$. This can be achieved by sequentially choosing the corresponding mode atoms that are most correlated with the residual. Solving for the sparse code can be subsequently done via solving a least squares problem that involves the Khatri-Rao product [76] of the determined atoms. The complexity is further reduced by imposing a block sparsity assumption on $\mathcal{X}$. This is motivated by the observation that non-zero entries are not evenly distributed but occur grouped into blocks [17]. Thus, only a subset of atoms is used in the decomposition, which can be quickly identified by their proposed V-way Block OMP algorithm.

Extending dictionary learning algorithms that are originally designed for the non-separable case, e.g. MOD [49] or K-SVD [2], to the multidimensional setting is proposed by Roemer et al. in [136]. The alternating sparse code and dictionary update fashion, however, is left untouched which renders the method rather limited to low-dimensional scenarios. This is mostly due to a standard OMP in the sparse recovery step, that relies on the Kronecker dictionary $\boldsymbol{D} = \iota(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_V)$. Additionally, the MOD dictionary update involves laborious tensor unfolding and Kronecker products, which renders this approach rather expensive. Moreover, the proposed Higher-Order SVD used to compute the rank-1 tensor approximation in the multidimensional K-SVD implementation is computationally more expensive than the standard two dimensional SVD update from [2].

In [116], Qi et al. also focus on the 2D separable dictionary learning problem, that is solved in the same alternating manner. Opposed to the aforementioned approach, the authors utilize a 2D-OMP which maintains the two dimensional structure of the data samples. While the outcome is equivalent to the ordinary OMP, the recovery complexity and memory usage is reduced. A generalization of the model to multidimensional data is proposed in the follow-up work [113]. However, the presented approach has a severe drawback as the sparse coding problem is converted into a standard 1D problem that involves vectorization of the multidimensional data along with building the Kronecker dictionary $\boldsymbol{D} = \iota(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_V)$. Depending on the sparsity penalty, the solution is found via classical OMP or BP. The subsequent dictionary update is performed in a K-SVD type manner where each row of $\boldsymbol{D}_v$, $v = 1, \ldots, V$ is updated sequentially. A modified approach that considers the inherent structure of the multidimensional data is presented by part of the same authors in [115]. Here, the sparse coding problem is tackled via an Tensor-based Iterative Shrinkage Thresholding Algorithm (TISTA) that avoids building the Kronecker product of all dictionaries prior to solving the sparse coding problem. Instead, computing the gradient with respect to the sparse code simply involves the n-mode product between the data and the dictionaries, which can be performed efficiently. Depending on the employed sparsity penalty, i.e., $\|\mathcal{X}\|_1$ or $\|\mathcal{X}\|_0$, the next iterate is easily determined by soft- or hard-thresholding, respectively. Comparable to [116] and [113], the dictionaries $\{\boldsymbol{D}_v\}_{v=1}^V$ are updated sequentially by solving a least squares problem for each mode via Newton's method or conjugate gradients. For each $\boldsymbol{D}_v$, this strategy implies calculating and unfolding the n-mode product $\widetilde{\mathcal{X}} = \mathcal{X} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \ldots \times_{v-1} \boldsymbol{D}_{v-1} \times_{v+1} \boldsymbol{D}_{v+1} \ldots \times_V \boldsymbol{D}_V$ for each mode separately, such that $\mathcal{S} \approx \widetilde{\mathcal{X}} \times_v \boldsymbol{D}_v$. To summarize, the aforementioned approaches still largely rely on standard methods during optimization without explicitly utilizing the multidimensional structure of the data.

In contrary to the alternating sparse coding and sequential dictionary update frameworks above, Hawe et al. [65] propose to learn a set of dictionaries $\{\boldsymbol{D}_v\}_{v=1}^V$ while jointly optimizing over the dictionaries and the sparse code $\mathcal{X}$. To be precise, since each of the

columns of $\boldsymbol{D}_v$, $v = 1, \ldots, V$ is restricted to unit norm to avoid the scale ambiguity, the resulting matrices admit a manifold structure, namely the product of spheres $\mathsf{S}(n_v, k_v)$. This property is directly exploited in the optimization procedure through updates along geodesics on the unit sphere. Thus, each $\boldsymbol{D}_v$ is updated as a whole without violating the unit norm constraint. The desired moderate coherence between the columns of $\boldsymbol{D}_v$ is enforced via differentiable log-barrier penalties $r(\boldsymbol{D}_v)$ that are incorporated in the cost function. Finally, the objective for the investigated problem of learning a sparse model for two dimensional signals reads

$$\begin{aligned} &\{\mathcal{X}^\star, \boldsymbol{D}_1^\star, \boldsymbol{D}_2^\star\} \in \\ &\arg\min_{\mathcal{X}, \boldsymbol{D}_1, \boldsymbol{D}_2} \; \tfrac{1}{2T} \sum_{j=1}^{T} \|\boldsymbol{D}_1 \boldsymbol{X}_j \boldsymbol{D}_2^\top - \boldsymbol{S}_j\|_F^2 + \tfrac{\lambda}{T} g(\mathcal{X}) + \kappa_1 r(\boldsymbol{D}_1) + \kappa_2 r(\boldsymbol{D}_2), \end{aligned} \tag{3.2}$$

where T denotes the number of two-dimensional training samples. Analogous to [64], the employed conjugate gradient approach permits the usage of any smooth sparsity inducing function, resulting in $g(\mathcal{X}) = \sum_i \log(1 + \nu x_i^2)$, with ν controlling the slope of the log-function. Eventually, the product manifold structure allows to jointly update $(\mathcal{X}, \boldsymbol{D}_1, \boldsymbol{D}_2)$ at each iteration step. The presented approach allows to learn a separable dictionary even from noisy samples, however it is restricted to the AWGN assumption and it does not provide global image support.

Zhang et al. [179] take the same line of penalizing the incoherence and exploiting the manifold structure to update the dictionaries as in [65]. They further include the full-rank log barrier function mentioned in [64], however without proper motivation. Indeed enforcing the full rank in the synthesis model seems counterintuitive and ineffective in the case when the signals reside in a low dimensional space. Like most of the other algorithms, the proposed optimization is based on an alternating approach: (1) Sparse coding via Separable Fast Iterative Shrinkage-Thresholding algorithm (SFISTA), (2) Conjugate Gradient update on the oblique manifold as suggested in [65]. An extension can be found in [180], however with 2D-OMP in the sparse coding stage.

Another interesting strategy to learn separable filter banks is proposed by Rigamonti et al. [134] and Sironi et al. [149] who adopt a convolutional approach that replaces the matrix-vector (tensor) product by a convolution. Sticking to the two dimensional case, if we let $\boldsymbol{S}_i \in \mathbb{R}^{N \times N}$ denote an entire 2D image, a set of K two dimensional kernels $\{\boldsymbol{D}_j\}_{j=1}^K$

along with their corresponding sparse feature *maps* $\{\boldsymbol{X}_{j,i}\}_{j=1}^{K}$ can be retrieved via solving

$$\{\{\boldsymbol{D}_j^{\star}\}_{j=1}^{K}, \{\boldsymbol{X}_{j,i}^{\star}\}_{j=1}^{K}\} \in \arg\min_{\{\{\boldsymbol{D}_j\}_{j=1}^{K}, \{\boldsymbol{X}_{j,i}\}_{j=1}^{K}\}} \sum_i \left(\|\boldsymbol{S}_i - \sum_{j=1}^{K} \boldsymbol{D}_j * \boldsymbol{X}_{j,i}\|_F^2 + \lambda \sum_{j=1}^{K} \|\boldsymbol{X}_{j,i}\|_1 \right), \tag{3.3}$$

with λ weighting the sparsity penalty against the data fidelity term. Stochastic Gradient Descent and soft-thresholding are used to alternatingly optimize over both, the kernels and the sparse feature maps, respectively. In the second step, the non-separable learned kernels $\{\boldsymbol{D}_j\}_{j=1}^{K}$ are approximated by means of a linear combination of a smaller set of separable ones. Despite their first approach in [134] that relies on minimizing the nuclear norm, in [149] a strategy based on tensor decompositions is utilized to decompose $\{\boldsymbol{D}_j\}_{j=1}^{K}$ as the weighted sum of separable kernels. For this purpose, the K matrices $\boldsymbol{D}_j \in \mathbb{R}^{N \times N}$ are arranged as a 3-dimensional tensor $\mathcal{D} \in \mathbb{R}^{N \times N \times K}$. Now, representing each of the $\boldsymbol{D}_j$ kernels as linear combinations of rank-1 matrices is equivalent to writing the tensor $\mathcal{D}$ as a linear combination of rank-1 tensors, also known as the Canonical Polyadic Decompostion (CPD) [76] that reads

$$\mathcal{D} \approx \sum_{m=1}^{M} \boldsymbol{d}_m^{\text{col}} \circ \boldsymbol{d}_m^{\text{row}} \circ \boldsymbol{w}_m. \tag{3.4}$$

The symbol $\circ$ denotes the tensor or outer product of the one dimensional vectors $\boldsymbol{d}_m^{\text{col}} \in \mathbb{R}^N$, $\boldsymbol{d}_m^{\text{row}} \in \mathbb{R}^N$, and $\boldsymbol{w}_m \in \mathbb{R}^K$. Finally, the approximation of the matrices $\{\boldsymbol{D}_j\}_{j=1}^{K}$ as a weighted sum of M separable tensors can be estimated via minimizing

$$\{\boldsymbol{d}_m^{\text{col}\,\star}, \boldsymbol{d}_m^{\text{row}\,\star}, \boldsymbol{w}_m^{\star}\}_{m=1}^{M} \in \arg\min_{\{\boldsymbol{d}_m^{\text{col}}, \boldsymbol{d}_m^{\text{row}}, \boldsymbol{w}_m\}} \left\| \mathcal{D} - \sum_{m=1}^{M} \boldsymbol{d}_m^{\text{col}} \circ \boldsymbol{d}_m^{\text{row}} \circ \boldsymbol{w}_m \right\|_F^2, \tag{3.5}$$

which is also easily extendable to higher dimensional data. Empirically, the authors have shown in [149] that small values of M suffice to well approximate the original non-separable kernels without significantly degrading the representation accuracy. For the purpose of pixel classification, like vessel detection in biomedical image data, an additional linear classifier is learned given the feature maps of training images. With separable kernels at hand, correlating the matrices with query images (volumes) to obtain the feature maps can be performed much faster than with conventional non-separable ones. However, while reducing the computational complexity when the kernels are applied to the

signal, the proposed learning scheme still remains extensive because the step involving (3.3) ignores the separable structure imposed later in (3.5).

In [34], Dantas et al. propose a scheme similar to [134]. The learned two dimensional dictionary is built as the sum taken over M separable matrices, where each term is composed of the Kronecker product of two sub-dictionaries $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, i.e.,

$$\boldsymbol{D} = \sum_{m=1}^{M} \boldsymbol{D}_{2,m} \otimes \boldsymbol{D}_{1,m}. \tag{3.6}$$

Consequently, the separable dictionary as proposed in [65] is a special case with $m = 1$. Instead of directly enforcing the structure given in (3.6), the objective function involves a rearrangement $R(\cdot)$ of the entries in $\boldsymbol{D}$ such that

$$R(\boldsymbol{D}) = \sum_{m=1}^{M} R(\boldsymbol{D}_m) = \sum_{m=1}^{M} \operatorname{vec}(\boldsymbol{D}_{2,m}) \operatorname{vec}(\boldsymbol{D}_{1,m})^\top. \tag{3.7}$$

An additional nuclear norm rank penalty $\|R(\boldsymbol{D})\|_*$ is added to the cost of the sparse coding problem to enforce the desired degree of separability via soft thresholding the singular values of $R(\boldsymbol{D})$.

3.1.2. Analysis Model

Recently, exploiting the inherent structure of multidimensional data has also been pursued regarding the analysis model. Again, given the V-dimensional input signals $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_V}$, the analysis co-sparse data model assumes

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{\Omega}_1 \times_2 \boldsymbol{\Omega}_2 \ldots \times_V \boldsymbol{\Omega}_V \quad \mathcal{A} \text{ is co-sparse}, \tag{3.8}$$

with the analysis operators $\boldsymbol{\Omega}_v \in \mathbb{R}^{k_v \times n_v}$ with $v = 1, \ldots, V$ and the multidimensional co-sparse representation $\mathcal{A} \in \mathbb{R}^{k_1 \times k_2 \times \ldots \times k_V}$.

Ongoing from their synthesis approach presented in [116], some of the authors propose a separable two dimensional analysis sparse model in [114] which is extended to the multidimensional case in [113]. Given T multidimensional noisy observations

$\mathcal{Y} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_V \times T}$, their learning objective reads

$$\begin{aligned} \{\{\boldsymbol{\Omega}_v^\star\}_{v=1}^V, \{\mathcal{S}_j^\star\}_{j=1}^T\} \in \arg \min_{\{\boldsymbol{\Omega}_v\}_{v=1}^V, \{\mathcal{S}_j\}_{j=1}^T} & \sum_{j=1}^T \|\mathcal{S}_j - \mathcal{Y}_j\|_2^2, \\ \text{s.t.} \quad & \|\mathcal{S}_j \times_1 \boldsymbol{\Omega}_1 \times_2 \boldsymbol{\Omega}_2 \ldots \times_V \boldsymbol{\Omega}_V\|_0 \leq L_a, \\ & \|\boldsymbol{\omega}_{v,i}\|_2 = 1 \ \forall v, i \end{aligned} \tag{3.9}$$

where L_a is some pre-defined co-sparsity and the vector $\boldsymbol{\omega}_{v,i}$ denotes the i-th row of $\boldsymbol{\Omega}_v$. As within the synthesis scenario, the optimization is performed in an alternating fashion. Retrieving an estimate of the multiway signal $\mathcal{S}_j$ is done via the Backward-Greedy algorithm as described in [141]. However, this method employs the vectorization approach (2.6) which neglects the separable structure of the operators. Clearly, the construction of the Kronecker operator $\boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)$ and the formulation as a 1D analysis sparse coding problem does not reduce the computational complexity at all which is a severe drawback of the proposed approach. After the sparse code estimation, the update of the analysis operators is executed sequentially with respect to both, the modes of the tensors and the rows of each $\boldsymbol{\Omega}_v$, respectively. For this purpose, the authors utilize a modification of the update formula already given in (1.19). First, define the auxiliary matrices $\boldsymbol{Q}_v = [\boldsymbol{Q}_{v,1}, \boldsymbol{Q}_{v,2}, \ldots, \boldsymbol{Q}_{v,T}]$ and $\boldsymbol{W}_v = [\boldsymbol{W}_{v,1}, \boldsymbol{W}_{v,2}, \ldots, \boldsymbol{W}_{v,T}]$, with $\boldsymbol{Q}_{v,j} = \text{unf}(\mathcal{S}_j, v)(\boldsymbol{\Omega}_V \otimes \cdots \otimes \boldsymbol{\Omega}_{v+1} \otimes \boldsymbol{\Omega}_{v-1} \otimes \cdots \boldsymbol{\Omega}_1)$, and $\boldsymbol{W}_{v,j} = \text{unf}(\mathcal{Y}_j, v)(\boldsymbol{\Omega}_V \otimes \cdots \otimes \boldsymbol{\Omega}_{v+1} \otimes \boldsymbol{\Omega}_{v-1} \otimes \cdots \boldsymbol{\Omega}_1)$ which involve unfolded versions of the signals along dimension v. Sequentially updating the rows of the n-mode operator $\boldsymbol{\Omega}_v$ amounts to solving the problem

$$\boldsymbol{\omega}_{v,i}^\star \in \arg\min_{\boldsymbol{\omega}_{v,i}} \|\boldsymbol{\omega}_{v,i}^\top \boldsymbol{W}_{v,J}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\omega}_{v,i}\|_2 = 1. \tag{3.10}$$

Analogous to (1.19), the index set J identifies the disjoint set of columns in $\boldsymbol{Q}_v$ that are orthogonal to $\boldsymbol{\omega}_{v,i}$. Although the number of parameters that have to be estimated is reduced due to the separable structure, the calculation of the auxiliary matrices and the SVD computation for each row of each mode remains a computationally demanding task.

3.1.3. Transform Model

The literature of sparse transform learning lacks any work that focuses on separability of the transform. In Wen et al. [167] the authors propose to learn sparsifying transforms for spatio-temporal video data. However, in their algorithm they maintain the strategy of vectorizing the 3D video cubes. This procedure allows to use standard algorithms that are designated to learn square transform matrices.

3.1.4. Further Approaches

The aforementioned related work on separable sparse data models indicates that imposing a structural constraint on the dictionary or the analysis operator has several advantages. Besides separability, another interesting approach called double sparsity has been proposed for the synthesis and transform sparse representation model to alleviate the limitations of unstructured models.

These methods usually utilize a fixed simple mathematical model of the data that allows to efficiently compute the representation. Since this comes at the cost of loosing adaptivity to the signal class of interest, a sparse mixing matrix $\boldsymbol{B}$ that linearly combines a few weighted atoms/filters of the analytically given base dictionary/transform is learned to alleviate the drawback of non-adaptivity. The main appeal of these methods is that the sparsity of $\boldsymbol{B}$ allows to efficiently compute the forward and adjoint operators. Simultaneously, this approach facilitates the use of larger dictionaries and thus to handle higher-dimensional data. Even more, a compact explicit matrix representation of the analytical base dictionary $\boldsymbol{\Psi}$ is ensured by choosing separable dictionaries like DCT, overcomplete DCT, or Wavelet dictionaries. Eventually, following [142], the double sparse data model for the T signals in $\boldsymbol{S} \in \mathbb{R}^{n \times T}$ can be found via solving the optimization problem

$$\begin{aligned} \{\boldsymbol{X}^\star, \boldsymbol{B}^\star\} \in \arg\min_{\boldsymbol{X},\boldsymbol{B}} \; & \|\boldsymbol{S} - \boldsymbol{\Psi}\boldsymbol{B}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{x}_i\|_0 \leq L_s \, \forall i \\ & \|\boldsymbol{b}_j\|_0 \leq p, \; \|\boldsymbol{\Psi}\boldsymbol{b}_j\|_2 = 1 \; \forall j. \end{aligned} \tag{3.11}$$

Yaghoobi et al. [172] follow a similar approach called Compressible Dictionary Learning (CDL) where the dictionary is decomposed into the base dictionary $\boldsymbol{\Psi}$ and a sparse (or compressible, hence CDL) matrix $\boldsymbol{B}$. Opposed to [142], who enforce fixed sparsity for each column $\boldsymbol{b}_i$ separately via ℓ_0 constraints, CDL utilizes an ℓ_1-sparsity measures for both $\boldsymbol{X}$ and $\boldsymbol{B}$ to induce sparsity over the whole matrices which is more flexible.

The work of Ophir et al. [107] is also closely related to [142] and [172], however they utilize the multi-scale structure of the Wavelet transform in their learning process. Furthermore, instead of expressing the data in the image domain with linear combinations from atoms of the Wavelet synthesis operator, the authors propose to model the sparsity in the Wavelet analysis domain.

Another algorithm that is capable of handling high dimensional signals is proposed by Sulam et al. called Trainlets [155]. Primarily, the work follows the idea of a double sparse dictionary as already proposed in [142]. The main difference consists in the choice of the analytically given base dictionary $\boldsymbol{\Psi}$, which is represented by a cropped separable Wavelet

transform that resolves the border issues which is a serious limitation of the traditional Wavelet transform.

Sparsely coding the image in the Wavelet analysis domain as in [107] naturally leads one to think of the sparse transform model. Indeed, double sparsity for the transform learning approach is proposed by Ravishankar & Bresler in [124]. Analogously to the synthesis model of [142] and [172], the authors impose a similar structural constraint on their square transform matrix $\boldsymbol{W}$ such that it is the product of a sparse matrix $\boldsymbol{C}$ and an analytical transform $\boldsymbol{\Psi}$, i.e., $\boldsymbol{W} = \boldsymbol{C}\boldsymbol{\Psi}$.

Instead of factorizing the dictionary $\boldsymbol{D} = \boldsymbol{\Psi}\boldsymbol{B}$ into a fixed analytically given transform $\boldsymbol{\Psi}$ and a sparse matrix $\boldsymbol{B}$, Magoarou & Gribonval [88] propose to directly factorize a given dense dictionary $\boldsymbol{D}$ in multiple sparse factors $\widetilde{\boldsymbol{D}}_j$, such that $\boldsymbol{D} = \prod_{j=1}^{J} \widetilde{\boldsymbol{D}}_j$. The approach allows to use any input $\boldsymbol{D}$ which might have been learned with some classical dictionary learning algorithm like K-SVD [2]. The corresponding optimization problem reads

$$\{\widetilde{\boldsymbol{D}}_j^{\star}\}_{j=1}^{J} = \arg\min_{\widetilde{\boldsymbol{D}}_j} \|\boldsymbol{D} - \prod_{j=1}^{J} \widetilde{\boldsymbol{D}}_j\|_F^2 + \sum_{j=1}^{J} g(\widetilde{\boldsymbol{D}}_j) \tag{3.12}$$

where $g(\cdot)$ denotes a sparsity inducing function. The presented optimization strategy borrows some ideas from the pre-learning of the layers of a deep neural network. Instead of finding all the sparse factors simultaneously, the proposed hierarchical algorithm iteratively factorizes the current input dictionary into two factors. To be precise, at the first iteration the decomposition reads $\boldsymbol{D} = \boldsymbol{T}_1\widetilde{\boldsymbol{D}}_1$ with $\boldsymbol{T}_1 = \prod_{j=2}^{J} \widetilde{\boldsymbol{D}}_j$ and the sparse factor $\widetilde{\boldsymbol{D}}_1$. At the subsequent $l = 2, \ldots, J$ iterations this process is repeated by means of factorizing the matrix $\boldsymbol{T}_{l-1}$ again into two factors. Thus, the number of non-zero coefficients in $\boldsymbol{T}$ is continuously decreased until the desired number J of sparse factors is attained. After solving (3.12), the reduced number of parameters beneficially influences the storage cost and the computational effort when applying the dictionary. However, although being flexible with regard to the input dictionary $\boldsymbol{D}$, the proposed structure is imposed after the learning stage. Hence, the computational complexity during learning remains unchanged.

Finally, an approach related to convolutional sparse coding is presented in [24]. To obtain an estimate of the signal $\boldsymbol{s}$, the sparse code $\boldsymbol{x}$ is sequentially convolved with $k = 1, \ldots, K$ sparse kernels $\widehat{\boldsymbol{d}}_k$ such that the error $\|\boldsymbol{s} - \boldsymbol{x} * \widehat{\boldsymbol{d}}_1 * \cdots * \widehat{\boldsymbol{d}}_K\|_2^2$ is minimized. The proposed convolution based strategy results in numerical efficient dictionaries - the matrix-vector multiplications with $\boldsymbol{D}$ and its adjoint are replaced with convolutions involving kernels with sparse support - that additionally permits to learn large atoms.

3.2. Adaptive or Blind Learning

From image denoising it is known that the reconstruction accuracy can be improved when the dictionary or operator is not only learned based on some general and representative clean training signals, but also directly on the specific signal $\boldsymbol{s} \in \mathbb{R}^n$ that has to be reconstructed [47, 141, 175, 90, 168, 126]. Typically, all these approaches assume measurements generated via $\boldsymbol{y} = \boldsymbol{\Phi s} + \boldsymbol{e} \in \mathbb{R}^m$, with $m = n$, $\boldsymbol{\Phi} = \boldsymbol{I}_n$ and $\boldsymbol{e} \sim \mathcal{N}(0, \boldsymbol{I}_n)$. In general, however, we often only have access to possibly undersampled measurements, generated from a non-diagonal, usually dense system matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ with $m < n$. The scenario of reconstructing $\boldsymbol{s}$ from undersampled measurements $\boldsymbol{y}$, while simultaneously learning a (co-)sparse data model can be attributed to the theory of Blind Compressed Sensing (BCS) [59, 148]. In contrast to the classical CS scenario, where the sparsity inducing matrix is known a priori, BCS is based on the assumption that the signal is sparse under some unknown representation that has to be determined during signal recovery. In the following, related approaches that follow the BCS framework are discussed.

The work of Gleichman & Eldar [59] investigates the blind recovery problem from a theoretical perspective with the aim to derive basic conditions that allow to reconstruct the signal from compressed measurements. Following the sparse synthesis approach, the observations can be modeled via $\boldsymbol{y} = \boldsymbol{\Phi D x} + \boldsymbol{e}$. Intuitively, any signal $\boldsymbol{s}$ is perfectly sparse with respect to some representation $\boldsymbol{D}$ that contains the signal itself. Thus, the theory either requires a set of signals, each being sparse under the same representation, or additional constraints on the sparsity inducing dictionary $\boldsymbol{D}$. For example, one of the analyzed constraints is directly related to the double sparsity approach discussed above, where the sought atoms are assumed to be a sparse weighted linear combination of some elements from the given dictionary $\boldsymbol{\Psi}$. Finally, for each of the applied constraints the authors provide conditions to guarantee the uniqueness of the solution.

While the authors in [59] are only interested in the product $\boldsymbol{DX}$ itself, the work presented in [148] directly tackles the problem of estimating both, the dictionary $\boldsymbol{D}$ along with the sparse code $\boldsymbol{x}$. In their work, the authors derive an algorithm that performs dictionary learning and signal recovery simultaneously. Besides, numerical results regarding an image inpainting problem are given. The same question how to learn the sparse synthesis model from compressed measurements is addressed in [154]. The presented approach utilizes the alternating optimization strategy as proposed in the K-SVD learning algorithm [2].

An application oriented view on the BCS framework is presented by Ravishankar & Bresler. In [122], they propose to adaptively learn a dictionary from undersampled k-space data, which describes the data space in Magnetic Resonance Image acquisition. Samples

in the k-space correspond to Fourier coefficients of the imaged object. Opposed to a purely patch-based adaptive learning scheme, like it is usually deployed in Denoising scenarios, the authors combine the patch-based sparse synthesis model with a global image fidelity term which leads to the following optimization problem:

$$\{\boldsymbol{s}^\star, \boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg\min_{\boldsymbol{s},\boldsymbol{D},\boldsymbol{X}} \sum_{ij} \|\boldsymbol{R}_{ij}\boldsymbol{s} - \boldsymbol{D}\boldsymbol{x}_{ij}\|_2^2 + \lambda\|\boldsymbol{F}_\mathbf{u}\boldsymbol{s} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{x}_{ij}\|_0 \leq L_s \,\forall i, j. \tag{3.13}$$

Here, the dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times k}$ is learned from patches that are extracted from the vectorized image $\boldsymbol{s} \in \mathbb{R}^N$ via the operator $\boldsymbol{R}_{ij} \in \mathbb{R}^{n\times N}$. Consequently, $\boldsymbol{x}_{ij}$ denotes the sparse code for the patch at position i, j stacked as column vectors into the matrix $\boldsymbol{X}$. The matrix $\boldsymbol{F}_\mathbf{u} \in \mathbb{C}^{M\times N}$ represents the undersampled Fourier encoding matrix, i.e., $M \ll N$, that models the MRI sampling process such that the Fourier representation of the reconstructed signal $\boldsymbol{s}$ can be compared to the acquired measurements $\boldsymbol{y} \in \mathbb{C}^M$. Finally, problem (3.13) is solved using an alternating minimization procedure.

In the follow-up work [132], the same idea of reconstructing MR images is tackled via a blind sparse transform learning approach. The extension of the model outlined in Section 1.2.3 to the blind setting reads

$$\begin{aligned}\{\boldsymbol{s}^\star, \boldsymbol{W}^\star, \boldsymbol{B}^\star\} \in \arg\min_{\boldsymbol{s},\boldsymbol{W},\boldsymbol{B}} \sum_{ij} \|\boldsymbol{W}\boldsymbol{R}_{ij}\boldsymbol{s} - \boldsymbol{b}_{ij}\|_2^2 + \lambda\|\boldsymbol{F}_\mathbf{u}\boldsymbol{s} - \boldsymbol{y}\|_2^2 - \gamma\left(\log(\det(\boldsymbol{W})) - \tfrac{1}{2}\|\boldsymbol{W}\|_F^2\right) \\ \text{s.t.} \quad \|\boldsymbol{b}_{ij}\|_0 \leq L_a \,\forall i, j, \;\; \|\boldsymbol{s}\|_2 \leq c,\end{aligned} \tag{3.14}$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ denotes a square sparsifying transform matrix. To better capture the diversity of MRI features, in [129] the same authors propose to learn a union of sparsifying transforms. The core idea behind this approach is to cluster image patches into distinct classes. Each image patch is assumed to correspond to only one of the square transforms that best sparsifies the particular patch. Consequently, the transform update step is carried out for each class separately involving only those patches assigned to the current cluster. While being very efficient in the update (closed form solution) this strategy does not prevent the algorithm from learning repetitive filters across the clusters.

An adaptive sparsifying transform regularizer to sparsely represent CT images is presented in [111].

3.3. Sparse Data Models and Neural Networks

Instead of solving the sparse approximation problem (1.4) via classical pursuit methods, there has been several attempts in the literature that try to approximately solve the sparse coding step by means of learning a direct mapping from the input to its sparse representation. Formally, this process can be described as an encoding $\boldsymbol{h} = f_e(\boldsymbol{s})$, where $\boldsymbol{h}$ is assumed to be sparse. If this representation is intended for signal reconstruction, the decoding amounts to finding the function $f_d(\boldsymbol{h}) \approx \boldsymbol{s}$. Combining both steps, we have $\boldsymbol{s} \approx f_d(f_e(\boldsymbol{s}))$, which in the machine learning community is well known as an Auto-Encoder or Auto-Associator framework [14, 67]. A detailed investigation of this approach concerning the co-sparse analysis perspective in shown in Chapter 7. In the remainder of this section, the related work following the introduced paradigm is summarized.

Ongoing from their previous work [120], Ranzato et al. propose an algorithm named Sparse Encoding Symmetric Machine (SESM) that aims at predicting the sparse code via a simple feed-forward propagation through the encoder. To this end, they define the sparse approximator as the mapping $f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}, \boldsymbol{b}_\mathbf{e}) = \boldsymbol{\Theta}^\top \boldsymbol{s}_i + \boldsymbol{b}_\mathbf{e} = \boldsymbol{h}_i$, with $\boldsymbol{\Theta} \in \mathbb{R}^{n \times k}$ denoting the encoder matrix that maps the input to its feature representation. With the decoder $f_d(\boldsymbol{h}_i, \boldsymbol{\Theta}, \boldsymbol{b}_\mathbf{d}) = \boldsymbol{\Theta}\sigma(\boldsymbol{h}_i) + \boldsymbol{b}_\mathbf{d}$ at hand, with $\sigma(\cdot)$ denoting a point-wise logistic non-linearity, the following loss including some additional weights $\alpha_{e,s,r}$ is minimized

$$\{\boldsymbol{\Theta}^\star, \boldsymbol{X}^\star, \boldsymbol{b}_\mathbf{e}^\star, \boldsymbol{b}_\mathbf{d}^\star\} \in \arg\min_{\{\boldsymbol{\Theta}, \boldsymbol{X}, \boldsymbol{b}_\mathbf{e}, \boldsymbol{b}_\mathbf{d}\}} \sum_{i=1}^{T} \Big[\alpha_e \|\boldsymbol{x}_i - f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}, \boldsymbol{b}_\mathbf{e})\|_2^2 + \|\boldsymbol{s}_i - f_d(\boldsymbol{h}_i, \boldsymbol{\Theta}, \boldsymbol{b}_\mathbf{d})\|_2^2 + \alpha_s g(\boldsymbol{x}_i) + \alpha_r \|\boldsymbol{\Theta}\|_1\Big], \tag{3.15}$$

where the function $g(\cdot)$ measures the sparsity of $\boldsymbol{x}_i$. To avoid the scaling ambiguity between the encoder and the decoder, the authors use weight sharing (or tied weights) to achieve automatic scaling of filters. The same idea is pursued in the work of Kavukcuoglu et al. [75]. Instead of a logistic non-linearity that provides sparse outcomes if most of the entries in the latent code exhibit negative entries, they utilize the Tanh activation function which requires the latent representation to be sparse as well. The encoder reads $f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}_\mathbf{e}, \boldsymbol{b}, \boldsymbol{G}) = \boldsymbol{G}\tanh(\boldsymbol{\Theta}_\mathbf{e}^\top \boldsymbol{s}_i + \boldsymbol{b})$, where the matrix $\boldsymbol{G} \in \mathbb{R}^{k \times k}$ denotes an additional diagonal matrix that compensates for the input scaling when dealing with normalized de-

coder weights. The joint optimization framework reads

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star, \boldsymbol{\Theta}_\mathbf{e}^\star, \boldsymbol{b}^\star, \boldsymbol{G}^\star\} \in \arg\min_{\{\boldsymbol{D},\boldsymbol{X},\boldsymbol{\Theta}_\mathbf{e},\boldsymbol{b},\boldsymbol{G}\}} \sum_{i=1}^{T} \Big[\|\boldsymbol{s}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 + \lambda\|\boldsymbol{x}_i\|_1 + \alpha\|\boldsymbol{x}_i - f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}_\mathbf{e}, \boldsymbol{b}, \boldsymbol{G})\|_2^2 \Big], \quad (3.16)$$

which allows the mapping function f_e to approximate the sparse code. Clearly, both approaches follow the synthesis framework where the encoder replaces the sparse coding algorithm, usually used to determine the sparse representation, and the decoder serves as a generative model to reconstruct the signals. This circumstance also motivates the term Predictive Sparse Decomposition (PSD).

A closely related approach inspired by Wavelet-based thresholding [44] is presented by Rubinstein et al. [139] termed analysis-synthesis thresholding. Instead of relying on a fixed wavelet basis, the analysis and the synthesis is decoupled by means of two distinct matrices that are learned simultaneously. Given the original signals $\boldsymbol{S} \in \mathbb{R}^{n\times T}$ and the corresponding corrupted versions $\boldsymbol{Y} \in \mathbb{R}^{n\times T}$, the analysis-synthesis thresholding process reads

$$\{\boldsymbol{D}^\star, \boldsymbol{\Theta}_\mathbf{e}^\star, \boldsymbol{\lambda}^\star\} \in \arg\min_{\{\boldsymbol{D},\boldsymbol{\Theta}_\mathbf{e},\boldsymbol{\lambda}\}} \sum_{i=1}^{T} \|\boldsymbol{s}_i - \boldsymbol{D} f_e(\boldsymbol{\Theta}_\mathbf{e}^\top \boldsymbol{y}_i)_{\boldsymbol{\lambda}}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}_{e,j}\|_2 = 1 \,\forall j, \quad (3.17)$$

where $f_e(\boldsymbol{\Theta}_\mathbf{e}^\top \boldsymbol{y}_i)_{\boldsymbol{\lambda}}$ is a thresholding function with the thresholding values for each weight vector defined by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$. To account for the scaling ambiguity, the norm of the encoding weight vectors $\boldsymbol{\theta}_{e,j}$, $j = 1, \ldots, k$ is constrained to be one. Although looking similar to a combination of the analysis and synthesis model, the introduced thresholding function ensures a sparse representation which in turn does not imply $\boldsymbol{\Theta}_\mathbf{e}^\top \boldsymbol{s}_i$ to be co-sparse as required in the analysis model. Especially the hard thresholding operator that nullifies every entry $\boldsymbol{\theta}_{e,j}^\top \boldsymbol{s}_i < \lambda_j$ and that is used in the proposed work neglects the true co-sparsity assumption $\boldsymbol{\theta}_{e,j}^\top \boldsymbol{s}_i = 0$. A similar idea is presented in [93] where the sparse code is obtained via hard thresholding the hidden representations.

Another strategy to approximate the sparse code via a non-linear deep feed forward predictor is presented in [60]. Analogously to the method introduced in [75], the proposed scheme directly considers the sparse representations, however without any reconstruction step that allows to measure the reconstruction error in the image space. To be precise, given a fixed dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times k}$, the optimal sparse codes $\boldsymbol{X}^\star \in \mathbb{R}^{k\times T}$ with respect to the input signals $\boldsymbol{S} \in \mathbb{R}^{n\times T}$ are obtained first based on existing sparse code inference algorithms, namely Iterative Shrinking and Thresholding Algorithm (FISTA) and Coordinate Descent.

Afterwards, the approximator $f_e(\boldsymbol{\Theta}_{\mathbf{e}}^{\top}\boldsymbol{s}_i)$ is learned via minimizing

$$\boldsymbol{\Theta}_{\mathbf{e}}^{\star} \in \arg\min_{\boldsymbol{\Theta}_{\mathbf{e}}\in\mathbb{R}^{n\times k}} \quad \sum_{i=1}^{T} \|\boldsymbol{x}_i^{\star} - f_e(\boldsymbol{\Theta}_{\mathbf{e}}^{\top}\boldsymbol{s}_i)\|_2^2, \tag{3.18}$$

which measures the deviation between the optimal sparse code $\boldsymbol{x}_i^{\star}$ and its approximation. Instead of using the Tanh non-linearity like in [75], the authors choose a sparsity promoting function like the shrinkage (soft-thresholding) function, i.e., $f_e(z_i) = \text{sign}(z_i)(|z_i| - \lambda)_+$. Since the pre-activation coefficients $a_{i,j} = \boldsymbol{\theta}_{e,j}^{\top}\boldsymbol{s}_i$ are mapped to zero depending on the threshold parameter λ, the co-sparse analysis model assumption is relaxed due to the fact that a weight vector $\boldsymbol{\theta}_{e,j}$ does not have to be strictly orthogonal to the signal to achieve co-sparsity.

Chapter 4.

Separable Analysis Operator Learning

Sparse signal representations have been proven very useful to extract and recover the underlying structure of the signal of interest. In contrary to the well-known and extensively studied synthesis model, which relies on a computationally expensive sparse coding step to decompose the signal into its sparse coefficient sequence, the co-sparse analysis model provides a much simpler approach. A set of filters is correlated with the signal yielding sparse filter responses for signals that belong to the model. As such, it may serve as a prior in inverse problems or as an efficient model for structural analysis of the signal content. Evidently, the more the model is adapted to the signal class, the more reliable it is for these purposes which renders the analysis operator learning task a crucial problem. The previous chapters clearly indicate the recent progress in analysis operator learning that has led to state-of-the-art results in many image processing applications.

However, the steady increase in image resolution or the wide range of possible application scenarios where the signals exhibit multiple dimensions impose another important demand on the model, namely to obtain the filter responses in a timely manner. This can be efficiently achieved by filters with a separable structure.

In this chapter, I will present an algorithm to learn analysis operators with separable structures. The following contributions are addressed in the subsequent sections:

- After discussing the computational complexity related to separable operators, a suitable smooth objective function that allows to easily incorporate the separability constraint is established. To minimize the cost, a geometric stochastic gradient descent scheme with a new variable step size selection that is based on the Armijo condition is introduced afterwards. Various numerical experiments concerning the parameter selection are carried out, that on the one hand illustrate the beneficial impact of the adaptive step size selection and on the other hand, demonstrate the robustness of the learning approach against parameter changes.
- Although the separability constraint imposed on the operator severely restricts the set of possible solutions, an image denoising experiment is conducted that clearly

shows that analysis operators with separable structures are still very useful as a regularizer in inverse problems. Even more, the performance in comparison to various state-of-the-art analysis operators reveals that the observed significant decrease in training complexity does not impair the reconstruction quality, which strongly emphasizes the benefit of the presented approach.

4.1. Computational Complexity

First of all, this section discusses the benefit in terms of computational complexity entailed with the proposed separable approach. Suppose we are given some signal $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ representing a three dimensional tensor. Standard approaches to calculate the sparse code of $\mathcal{S}$ typically rely on a vectorization of the data where each column of $\mathcal{S}$ is stacked on one another to form the signal $\boldsymbol{s} = \mathrm{vec}(\mathcal{S}) \in \mathbb{R}^{N_1 N_2 N_3}$. Accordingly, the signal is multiplied with the analysis operator $\widetilde{\boldsymbol{\Omega}} \in \mathbb{R}^{K_1 K_2 K_3 \times N_1 N_2 N_3}$ in order to obtain the sparse code $\boldsymbol{a} = \widetilde{\boldsymbol{\Omega}}\boldsymbol{s}$. In contrary, if the analysis operator exhibits a separable structure, i.e., $\boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{\Omega}_3)$, the co-sparse representation can be obtained by means of the n-mode product $\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{\Omega}_1 \times_2 \boldsymbol{\Omega}_2 \times_3 \boldsymbol{\Omega}_3$. Hence, instead of one large matrix $\widetilde{\boldsymbol{\Omega}}$, this approach relies on a the set of small operators $\{\boldsymbol{\Omega}_i \in \mathbb{R}^{K_i \times N_i}\}_{i=1}^3$.

If we now aim to learn the model from training samples, it can be readily seen that the additional structure directly influences the complexity of the learning task. To be precise, the number of filter coefficients that have to be estimated during learning drastically reduces from $\prod_i (K_i \cdot N_i)$ in the case of learning the unstructured operator $\widetilde{\boldsymbol{\Omega}}$, to $\sum_i (K_i \cdot N_i)$ in favor of the separable approach represented by $\boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{\Omega}_3)$.

Besides the reduced number of free parameters, the differences in the computational complexity between both approaches can be additionally assessed by means of the number of floating-point operations (FLOPs) necessary to calculate the matrix-vector or the n-mode product. One FLOP unit represents a multiplication or a summation, i.e., the inner product $\boldsymbol{u}^\top \boldsymbol{v}$, with $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$, requires N multiplications and $N - 1$ summations. Accordingly, the matrix-matrix product $\boldsymbol{U}\boldsymbol{V}$, with $\boldsymbol{U} \in \mathbb{R}^{K \times N}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times M}$ can be calculated with KNM multiplications and $KM(N-1)$ summations, resulting in $2KNM - KM$ FLOPs.

Suppose we are now given a 3D tensor $\mathcal{S}$, the matrix $\widetilde{\boldsymbol{\Omega}}$ and the operators $\{\boldsymbol{\Omega}_i\}_{i=1}^3$ as defined above. For simplicity let us assume that $\mathcal{S}$ is a cubic tensor with equal side length across all three dimensions, i.e., $N_1 = N_2 = N_3$. Furthermore, the number of filters K_i in $\boldsymbol{\Omega}_i$ equals the signal dimension N_i, thus we have $K = N$. The number of FLOPs required to calculate $\widetilde{\boldsymbol{\Omega}}\boldsymbol{s}$ amounts to $2 \cdot (N^3) \cdot N^3 - N^3$ while the n-mode product $\mathcal{S} \times_1 \boldsymbol{\Omega}_1 \times_2 \boldsymbol{\Omega}_2 \times_3 \boldsymbol{\Omega}_3$ only needs $2 \cdot (3N) \cdot N^3 - 3 \cdot N^3$ operations.

Consequently, the computational complexity is reduced from $\mathcal{O}(N^V)$ to $\mathcal{O}(V \cdot N)$ with

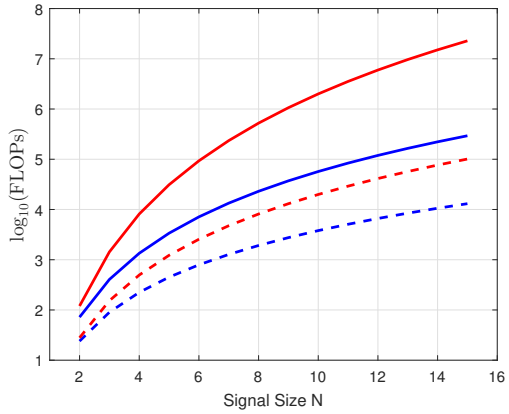


Figure 4.1.: Number of FLOPs required to calculate the n-mode product (blue lines) compared to the standard vectorization approach (red lines). The abscissa denotes the dimension N_i of the signal in each mode i which at the same time equals the number of filters K. The FLOP count is given in logarithmic scale. The solid lines represents the FLOP count for the 3D case, while the dotted lines indicates the number of FLOPs in a 2D signal setting.

V denoting the number of modes. Figure 4.1 illustrates the reduction in complexity for different choices of N. The number of FLOPs in logarithmic scale is plotted against the dimension N for a 2D image patch and a 3D tensor.

4.2. Algorithm Design

Separable analysis operator learning aims at finding a set of operators $\{\boldsymbol{\Omega}_i \in \mathbb{R}^{k_i \times n_i}\}_{i=1}^{V}$ that provide co-sparse representations when applied to signals from a particular signal class. In image processing, different types of image data, e.g. natural, medical or astronomical images, can be considered individual signal classes. Let $\{\mathcal{S}_j \in \mathbb{R}^{n_1 \times \cdots \times n_V}\}_{j=1}^{T}$ denote a set of T representative training signals extracted from some arbitrary class. Typically, these training sets comprise patches or tensors extracted from the image data. Eventually, the optimal set of analysis operators with regard to the training data is obtained via solving the optimization problem

$$\{\boldsymbol{\Omega}_i^{\star}\}_{i=1}^{V} \in \arg \min_{\boldsymbol{\Omega}_i^{\top} \in \mathsf{OB}(n_i,k_i)} \frac{1}{T} \sum_{j=1}^{T} g(\mathcal{S}_j \times_1 \boldsymbol{\Omega}_1 \cdots \times_V \boldsymbol{\Omega}_V), \tag{4.1}$$

where $g(\cdot)$ denotes a function that measures the co-sparsity and restricting $\boldsymbol{\Omega}_i^{\top}$ to $\mathsf{OB}(n_i, k_i)$ as defined in Section 2.2 avoids the trivial solution $\boldsymbol{\Omega}_i = \mathbf{0}_{k \times n}$ which is a global but useless

minimizer of problem (4.1). In the following, the individual components of the learning approach are further motivated and specified.

4.2.1. Sparsity Measure

Ideally, the co-sparsity is measured by means of the ℓ_0-norm that counts the number of non-zero entries in $\mathcal{A}_j = \mathcal{S}_j \times_1 \boldsymbol{\Omega}_1 \cdots \times_V \boldsymbol{\Omega}_V$. Minimizing the ℓ_0-norm, however, results in a combinatorial problem that is NP hard to solve [3]. That is why usually the ℓ_0-norm is relaxed to the ℓ_1-norm, which represents the closest convex surrogate function to the ℓ_0-norm. A convex problem is often desired since it ensures that a local minimizer is also the global minimizer. However, the property of the ℓ_1-norm to be a convex function comes at the cost of only roughly approximating the ℓ_0-norm.

Regarding the given setting, restricting the operators $\boldsymbol{\Omega}_i^\top$ to the oblique manifold renders (4.1) a non-convex problem anyways. Fortunately, the pursued geometric optimization framework allows to use any sparsity measure that is smooth. The function

$$g(\mathcal{A}) = \sum_l \frac{1}{\log(1+\nu)} \log(1 + \nu \cdot a_l^2) \tag{4.2}$$

with l denoting the index over all elements in $\mathcal{A}$, complies with these conditions while simultaneously being a good approximation of the ideal ℓ_0-norm due to the additional parameter ν that controls the slope of the function. Figure 4.2 illustrates this behavior in comparison to standard sparsity measures like the ℓ_0-norm and the ℓ_1-norm. The good performance as a sparsity promoting function has also been shown in the literature, see e.g. [104, 28].

The geometric optimization on the product of unit spheres naturally avoids the trivial solutions $\boldsymbol{\Omega}_i = \mathbf{0}_{k \times n}$. However, solely controlling the norm of the filters does not suffice to obtain a meaningful model. To motivate the necessary constraints along with their resulting impacts on the learned analysis operator I will follow the explanations of Yaghoobi et al. as presented in [175].

4.2.2. Full Rank Constraint

Although the trivial solution is already excluded, minimizing (4.1) will result in operators $\boldsymbol{\Omega}_i$ that exhibit repeated rows. The reason for this phenomenon is straightforward. Let us focus on $\boldsymbol{\Omega}_1$ that acts on the first mode of all the signals $\mathcal{S}_j$. If we denote $\boldsymbol{V} \in \mathbb{R}^{n_1 \times \tilde{T}}$ as the matrix that holds in its columns the n_1-dimensional first modes of all signals, we can easily find a filter $\boldsymbol{\omega}_1^\star$ that minimizes $g(\boldsymbol{\omega}_1^\top \boldsymbol{V})$. The optimum for $\boldsymbol{\Omega}_1$ is obtained by

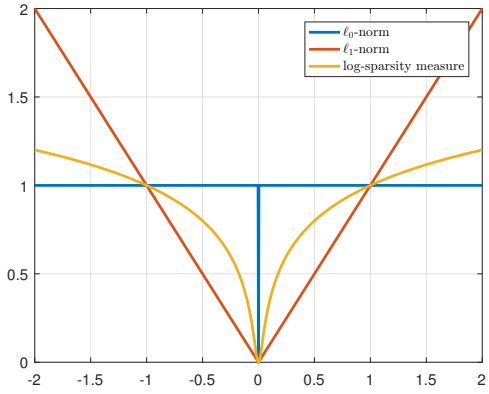


Figure 4.2.: Comparison of different sparsity measures. ℓ_0-norm, ℓ_1-norm, and proposed sparsity measure (4.2) with parameter $\nu = 1000$.

simply repeating $\boldsymbol{\omega}_1^\star$ exactly k_1 times, resulting in a rank-1 matrix. Obviously, utilizing rank deficient analysis operators $\boldsymbol{\Omega}_i$ as a regularizer in an inverse problem is useless because infinitely many solutions exist that fulfill the model assumption. Thus, in order to find a meaningful solution, the operators $\boldsymbol{\Omega}_i$ should have full rank.

Solely enforcing a full rank, however, will cause the rows in $\boldsymbol{\Omega}_i$ to only slightly deviate from its optimum $\boldsymbol{\omega}_j^\star$ which does not alleviate the problem of $\boldsymbol{\Omega}_i$ being close to singular and thus useless as a regularizer. That is why an additional penalty $r(\boldsymbol{\Omega}_i)$ is added to problem (4.1) that on the one hand enforces full rank of $\boldsymbol{\Omega}_i$ and simultaneously controls the condition number of $\boldsymbol{\Omega}_i$. The smooth log-barrier function

$$r(\boldsymbol{\Omega}_i) = -\frac{1}{n_i \log(n_i)} \log\det(\tfrac{1}{k_i}\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_i) \tag{4.3}$$

fulfills these requirements. In [64], Hawe et al. show that the condition $0 < \det(\frac{1}{k_i}\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_i) \leq (1/n_i)^{n_i}$ holds for matrices $\boldsymbol{\Omega}_i$ that have full rank. Equality on the right side is achieved when all eigenvalues of $\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_i$ are equal which is tantamount of having $\mathrm{cond}(\boldsymbol{\Omega}_i) = 1$. As a result, the regularizer defined in Eq. (4.3) can be added to the learning objective as a penalty to control the condition number of the analysis operator, while the minimizer of the function constitutes a tight frame.

The desired property of an analysis operator to have a moderate condition number has also been exploited in related works. Besides [64], especially the Transform Operator Learning literature, e.g. [126, 131, 168], points out the importance of controlling the condition number to attain a suitable model. These approaches, however, all focus on the

non-separable case. The question that arises is how does the penalty on the individual operators $\boldsymbol{\Omega}_i$ relate to enforcing a low condition number on $\iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)$? To answer the question, I make use of the following relation

$$\operatorname{cond}\left(\iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)\right) = \prod_{i=1}^{V} \operatorname{cond}(\boldsymbol{\Omega}_i). \tag{4.4}$$

Thus, enforcing the condition number property on all the $\boldsymbol{\Omega}_i$ individually has a direct impact on the condition number of the analysis operator in standard notation and thus to the ones learned in [64] or [126]. Regarding the cost function, the following property of the penalty is exploited. Let $\boldsymbol{\Omega}_i \in \mathbb{R}^{k_i \times n}\ \forall i = 1, \ldots, V$ be operators with possibly varying number of filters that are applied to cubical tensors, i.e., the size n is constant across all the V modes that can be considered the standard setting. In this case, the following relation holds true

$$\begin{aligned} & -\log\det\left(\tfrac{1}{\prod_i^V k_i}\iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)^\top \iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)\right) \\ =& -n^{(V-1)} \cdot \left(\log\det(\tfrac{1}{k_1}\boldsymbol{\Omega}_1^\top \boldsymbol{\Omega}_1) + \ldots + \log\det(\tfrac{1}{k_V}\boldsymbol{\Omega}_V^\top \boldsymbol{\Omega}_V)\right) \\ =& \; n^{(V-1)} \cdot \sum_{i=1}^{V} -\log\det(\tfrac{1}{k_i}\boldsymbol{\Omega}_i^\top \boldsymbol{\Omega}_i). \end{aligned} \tag{4.5}$$

Consequently, to control the condition number of the separable analysis operator $\iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V)$, the sum of the penalties applied to each of the individual operators is added to the objective function.

4.2.3. Coherence Penalty

Imposing the penalty (4.3) on $\boldsymbol{\Omega}_i$ is sufficient if $k_i = n_i$. For overcomplete operators that have more rows than columns, however, we still face the problem of possibly obtaining repeated rows $\boldsymbol{\omega}_j$ with $j > n_i$. The similarity between the normalized filters can be measured in terms of the mutual coherence, which reads

$$\mu(\boldsymbol{\Omega}_i) = \max_{j<l} |\boldsymbol{\omega}_j^\top \boldsymbol{\omega}_l| \tag{4.6}$$

with $\boldsymbol{\omega}_j$ and $\boldsymbol{\omega}_l$ denoting the j-th and l-th normalized row of $\boldsymbol{\Omega}_i$. From a practical point of view, the max operator in (4.6) is unsuited for gradient based optimization strategies. That is why another smooth coherence penalty is used in the presented learning algorithm that naturally avoids trivially linear dependent filters. The proposed log-barrier function

added to the cost introduced in (4.1) reads

$$h(\boldsymbol{\Omega}_i) = -\tfrac{1}{2}\sum_{j\neq l}\log(1-(\boldsymbol{\omega}_j^\top\boldsymbol{\omega}_l)^2). \tag{4.7}$$

The same penalty has been already successfully used in [64] and [65] to control the mutual coherence of the learned analysis and synthesis model, respectively.

It is shown in [65] that

$$h(\boldsymbol{\Omega}_i) \geq -\log(1-\mu(\boldsymbol{\Omega}_i)^2) \geq \tfrac{1}{N}h(\boldsymbol{\Omega}_i), \tag{4.8}$$

with N denoting the number of individual summands in (4.7). Analogously to the rank constraint introduced in the preceding subsection, the coherence of a separable operator $\iota(\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_V)$ is related to the coherence of the individual operators $\boldsymbol{\Omega}_i$ via

$$\mu\left(\iota(\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_V)\right) = \max\{\mu(\boldsymbol{\Omega}_1),\mu(\boldsymbol{\Omega}_2),\ldots,\mu(\boldsymbol{\Omega}_V)\}. \tag{4.9}$$

Combining (4.8) and (4.9) finally results in

$$\begin{aligned}\max\{h(\boldsymbol{\Omega}_1),\ldots,h(\boldsymbol{\Omega}_V)\} &\geq -\log(1-\mu(\iota(\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_V))^2)\\ &\geq \max\{\tfrac{1}{N_1}h(\boldsymbol{\Omega}_1),\ldots,\tfrac{1}{N_V}h(\boldsymbol{\Omega}_V)\}.\end{aligned} \tag{4.10}$$

A more thorough investigation of these properties can be found in [65].

As a result, keeping $\max\{h(\boldsymbol{\Omega}_1),\ldots,h(\boldsymbol{\Omega}_V)\}$ small implicitly controls the mutual coherence of $\iota(\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_V)$. On the other hand, the expression $\max\{h(\boldsymbol{\Omega}_1),\ldots,h(\boldsymbol{\Omega}_V)\}$ is small if all the elements of the set are small which motivates to apply the coherence penalty to all the individual operators. Eventually, the individual penalty function values are added up analogously to the condition number penalty introduced above.

4.2.4. Derivation of the Cost Function

Finally, an overall cost function for separable analysis operators learning involving the introduced penalties reads

$$\{\boldsymbol{\Omega}_1^\star, \ldots, \boldsymbol{\Omega}_V^\star\} \in \arg\min_{\boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i)} \frac{1}{T} \sum_{j=1}^{T} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \mathcal{S}_j)$$
$$\text{with} \quad f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \mathcal{S}_j) = g(\mathcal{S}_j \times_1 \boldsymbol{\Omega}_1 \cdots \times_V \boldsymbol{\Omega}_V) + \kappa \sum_{i=1}^{V} r(\boldsymbol{\Omega}_i) + \gamma \sum_{i=1}^{V} h(\boldsymbol{\Omega}_i), \tag{4.11}$$

where κ and γ are two additional parameters that weight the condition number penalty and the coherence penalty against the sparsity promoting function.

In order to enable a fair comparison between the separable and the non-separable approach, a slightly modified objective is additionally used. For two dimensional signals $\{\boldsymbol{S}_j \in \mathbb{R}^{n_1 \times n_2}\}_{j=1}^{T}$ the objective reads

$$\{\boldsymbol{\Omega}_1^\star, \boldsymbol{\Omega}_2^\star\} \in \arg\min_{\boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i)} \frac{1}{T} \sum_{j=1}^{T} f(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{S}_j)$$
$$\text{with} \quad f(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{S}_j) = g(\boldsymbol{S}_j \times_1 \boldsymbol{\Omega}_2 \times_2 \boldsymbol{\Omega}_2) + \kappa\, r(\iota(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)) + \gamma\, h(\iota(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)). \tag{4.12}$$

This formulation allows to use the same weighting parameters for structured as well as unstructured analysis operator learning, since the size of the considered operators is equal. In particular the evaluation of the sample complexity that will be presented in Chapter 5 benefits from this approach.

In the following, the geometric stochastic gradient descent algorithm that is used to optimize the objective function is introduced.

4.3. Stochastic Gradient Descent

SGD type optimization methods have attracted attention to solve large-scale machine learning problems [13, 92], where the earliest contributions in this direction made by Robbins & Monro [135] date back to the early 1950's. In contrast to *full* gradient methods that in each iteration require the computation of the gradient with respect to all the T training samples $\{\mathcal{S}_j\}_{j=1}^{T}$, in SGD the gradient computation at each iteration only involves a ran-

domly drawn small *mini-batch* or even a single sample to find a suitable search direction[1]. The motivation for this approach is rather intuitive. Frequently, the training set $\{\mathcal{S}_j\}_{j=1}^{T}$ contains samples that show approximately the same structures. Especially in natural image processing, this property is very pronounced due to the repetitive patterns that can be found in many real world images. Computing the gradient with respect to these redundant samples quickly becomes inefficient, since they do not provide further information about the structure of the data. The random selection of a small amount of samples, however, drastically reduces the computational effort at each iteration while the drawn batch of samples still provides a good approximation of the whole training set. Accordingly, the cost of each iteration is independent of T (assuming the cost of accessing each sample is independent of T).

In the following, I will use the notation $\mathcal{S}_{\{b(t)\}}$ to denote a mini-batch of cardinality $|b(t)|$, where $b(t)$ represents an index set randomly drawn from $\{1, 2, \ldots, T\}$ at iteration t. Accordingly, the function value corresponding to the mini-batch $\mathcal{S}_{\{b(t)\}}$ is denoted as $f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \mathcal{S}_{\{b(t)\}})$.

In order to account for the constraint set $\boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i)$, a geometric SGD optimization scheme is proposed that relies on the concepts introduced in Chapter 2. The Riemannian gradient for the i-th operator $\boldsymbol{\Omega}_i$ computed based on the mini-batch $\mathcal{S}_{\{b(t)\}}$ is denoted as

$$\mathtt{G}(\boldsymbol{\Omega}_i^\top)[\mathcal{S}_{\{b(t)\}}] = \Pi_{\mathtt{T}_{\boldsymbol{\Omega}_i^\top}\mathsf{OB}(n,k)} \left(\nabla_{\boldsymbol{\Omega}_i^\top} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \mathcal{S}_{\{b(t)\}}) \right). \tag{4.13}$$

With (4.13) and the concept of geodesics at hand, an update step at the t-th iteration of the geometric SGD reads

$$\boldsymbol{\Omega}_i^{\top(t+1)} = \Gamma(\boldsymbol{\Omega}_i^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha^{(t)}). \tag{4.14}$$

The choice of an adequate step size α is investigated in Section 4.3.2. Eventually, the whole SGD algorithm to learn separable analysis operators with a fixed step size α is summarized in Algorithm 4.1.

4.3.1. Stopping Criterion

To terminate the optimization, a suitable stopping criterion has to be defined. In standard Gradient Descent where at each iteration the gradient with respect to the full training set is

[1] Note that in the SGD related literature and contrary to the presented work, computing the gradient with respect to all samples is sometimes also referred to as batch gradient descent. To avoid confusions, I will use the term "full" gradient to indicate the gradient with respect to all samples, while a (mini-)batch of samples is only considered in the SGD setting.

Algorithm 4.1 SGD Fixed Step Size

Require: $\alpha^{(0)} > 0$, $\boldsymbol{\Omega}_i^{(0)}$ $i = 1, \ldots, V$
Set: $\alpha \leftarrow \alpha^0$, $t \leftarrow 1$
  while Stopping criterion not reached **do**
    choose $\{b(t)\}$
    calculate $\mathtt{G}(\boldsymbol{\Omega}_i^\top)[\mathcal{S}_{\{b(t)\}}]$ $i = 1, \ldots, V$
    update $\boldsymbol{\Omega}_i^{\top(t+1)} = \Gamma(\boldsymbol{\Omega}_i^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha)$.
    update $t \leftarrow t + 1$
  end while
Output: $\boldsymbol{\Omega}_i^\star$ $i = 1, \ldots, V$

available, the execution can be stopped for instance when the gradient vanishes, i.e., when a local minimum is reached. Even simpler, a predefined maximum number of iterations can cause the termination of the algorithm. Since in the SGD framework a small mini-batch of varying training signals is processed, the gradient fluctuates at each iteration which renders the vanishing gradient criterion unsuitable. The determination of a fixed number of iterations is difficult since too few iterations prevent the algorithm to capture the entire structure of the samples. On the other hand, too many iterations can result in an overfitting to the training set, i.e., although the learned model is well suited to explain the training samples, it is less descriptive with respect to the underlying signal class.

A common strategy to avoid the aforementioned problems is to use a validation dataset that contains a fixed set of samples from the same signal class. Monitoring some performance measure on the validation set can reveal useful information about when to stop the training process. To be precise, the proposed separable operator learning algorithm stops when the relative change of the average sparsity measure over previous iterations is below some predefined threshold. First, let $\mathcal{S}_{\text{validate}}$ represent a fixed set of T_{validate} training signals sampled from the same distribution as the samples $\mathcal{S}_{\{b(t)\}}$ present at iteration t. Accordingly, the overall sparsity attained on the validation set reads

$$z^{(t)} = \frac{1}{T_{\text{validate}}} \sum_{T_{\text{validate}}} g(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \mathcal{S}_{\text{validate}}), \tag{4.15}$$

which can be evaluated at each iteration or in periodical intervals. With this measure at hand, the average over the previous l iterations can be calculated via $\bar{z}^{(t)} = \frac{1}{l} \sum_{j=1}^{l} z^{(t-j)}$.

The optimization terminates if the relative variation of $z^{(t)}$, determined via

$$v = \frac{|z^{(t)} - \bar{z}^{(t)}|}{\bar{z}^{(t)}}, \tag{4.16}$$

falls below a certain threshold δ. The actual parameters for l and δ are determined in the experiment section.

4.3.2. Step Size Selection

A crucial factor that influences the convergence rate in SGD optimization is the selection of a suitable step size $\alpha^{(t)}$ (often also referred to as learning rate). For convex problems, the step size is typically based on the Lipschitz continuity property. If the Lipschitz constant is not known in advance, an appropriate learning rate is often chosen by using approximation techniques. Bottou [12] suggests to define a sequence of step sizes that decrease monotonically. However, in the same reference the author mentions that decreasing the learning rate too slowly will also cause the variance of the sought parameter $\boldsymbol{\Omega}_i$ to decrease equally slowly. On the other hand, when the step size decreases too quickly, the algorithm takes a very long time or even fails to reach the optimum. To alleviate these weaknesses, Bottou propose a predefined heuristic to iteratively shrink the step size. The update of the learning rate reads $\alpha^{(t)} = \alpha^{(0)}(1 + \alpha^{(0)}\lambda t)^{-1}$ which has the disadvantage of requiring the estimation of the additional hyper-parameter λ and the initial step $\alpha^{(0)}$. A straightforward strategy consists in determining a good constant step size empirically. According to [36], the frequent use of a constant step size in practice is due to the following advantages. First, only a single parameter has to be determined, and second, the performance is usually sufficient in practice.

In [81], the authors propose a basic line search that sequentially halves the step size if the current estimate does not minimize the cost. More precisely, $\alpha^{(t)}$ is halfed whenever the inequality

$$f(\boldsymbol{\Omega}^{(t+1)}, \mathcal{S}_{\{b(t)\}}) \leq f(\boldsymbol{\Omega}^{(t)}, \mathcal{S}_{\{b(t)\}}) - \frac{1}{2}\alpha^{(t)} \|\mathtt{G}(\boldsymbol{\Omega}^{\top(t)})[\mathcal{S}_{\{b(t)\}}]\|_F^2 \tag{4.17}$$

is not satisfied. The results reported in [81] clearly indicate the good performance of the simple line search approach. In this thesis, a more variable approach based on a variation of the backtracking line search algorithm adapted to SGD optimization is proposed.

Recall that instead of computing the gradient with respect to the full training set, the SGD framework only approximates the true gradient by means of a small signal batch or even a single signal sample. However, it is assumed that on average the SGD updates ap-

proach the minimum of the optimization problem stated in (4.11), i.e., the empirical mean over all training samples. This proposition is utilized to automatically find an appropriate step size such that for the next iterate an averaging Armijo condition is fulfilled. Let $\boldsymbol{H}_i^{(t)}$ denote the search direction for the i-th operator at iteration t as defined in Section 2.2. Furthermore, let us assume that $\boldsymbol{H}_i^{(t)}$ is estimated with respect to the full training set available to find an optimizer of the problem given in (4.11). The classical Armijo condition that ensures sufficient decrease in the cost function reads

$$f(\Gamma(\boldsymbol{\Omega}_i^{\top(t)}, \boldsymbol{H}_i^{(t)}, \alpha^{(t)})^\top, \ldots, \Gamma(\boldsymbol{\Omega}_V^{\top(t)}, \boldsymbol{H}_V^{(t)}, \alpha^{(t)})^\top, \mathcal{S}) \leq f(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \mathcal{S}) + c_1 \alpha^{(t)} \sum_{i=1}^{V} \langle \mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}], \boldsymbol{H}_i^{(t)} \rangle \tag{4.18}$$

which resembles the inequality given in (4.17) for the case $\boldsymbol{H}_i^{(t)} = -\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}]$ and $c_1 = 0.5$. Backtracking line search now consists of successively shrinking the current step size $\alpha^{(t)}$ with the factor $c_2 \in (0, 1)$ until the next iterate fulfills the Armijo condition.

When transferring the backtracking line search with Armijo condition to the SGD setting, one has to face the circumstance that at every iteration t only a noisy estimate of the true cost function is available. As already outlined above, SGD optimization is based on the assumption that the cost decreases on average. That is why in the proposed line search, the function value in (4.18) is replaced with the average over previous iterations. This can be easily achieved via a recursive sliding window implementation that reads

$$\bar{f}(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \mathcal{S}_{\{b(t)\}}) = \frac{1}{w} \sum_{j=1}^{w} f(\boldsymbol{\Omega}_1^{(t-j+1)}, \ldots, \boldsymbol{\Omega}_V^{(t-j+1)}, \mathcal{S}_{\{b(t-j)\}}) \tag{4.19}$$

where the summand on the right hand side denotes the function values obtained after applying the updated operators to the signal mini-batch $\mathcal{S}_{\{b(t-j)\}}$ and w corresponds to the window length. With this notation at hand, the new condition reads

$$f(\Gamma(\boldsymbol{\Omega}_1^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_1^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha^{(t)})^\top, \ldots, \Gamma(\boldsymbol{\Omega}_V^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_V^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha^{(t)})^\top, \mathcal{S}_{\{b(t)\}}) \leq \bar{f}(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \mathcal{S}_{\{b(t)\}}) - c_1 \alpha^{(t)} \sum_{i=1}^{V} \|\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}]\|_F^2. \tag{4.20}$$

If (4.20) is not fulfilled, i.e., if the function value for the updated operators is not at least as low as the previous average, the step size $\alpha^{(t)}$ goes to zero. To avoid needless line search iterations the execution is stopped after a predefined number of trials t_{ls}^{max} and proceeds with the next sample with resetting $\alpha^{(t+1)}$ to its initial value $\alpha^{(0)}$. Furthermore, after each

Algorithm 4.2 SGD Backtracking Line Search

Require: $\alpha^{(0)} > 0, 0 < c_1 < 1, 0 < c_2 < 1, t_{ls}^{max} = 20, \boldsymbol{\Omega}_i^{(0)}\ i = 1, \ldots, V$
Set: $\alpha \leftarrow \alpha^0, t \leftarrow 1$
  while Stopping criterion not reached **do**
    choose $\{b(t)\}$
    calculate $\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}]\ i = 1, \ldots, V$
    set $\alpha_{ls} \leftarrow \alpha^{(t)}, t_{ls} \leftarrow 1$
    while $f(\Gamma(\boldsymbol{\Omega}_1^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_1^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha_{ls}^{(t)})^\top, \ldots, \Gamma(\boldsymbol{\Omega}_V^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_V^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha_{ls}^{(t)})^\top, \mathcal{S}_{\{b(t)\}}) >$
      $\bar{f}(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \mathcal{S}_{\{b(t)\}}) - c_1 \alpha_{ls}^{(t)} \sum_{i=1}^{V} \|\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}]\|_F^2 \wedge$
      $t_{ls} < t_{ls}^{max}$ **do**
      $\alpha_{ls} \leftarrow \alpha_{ls} \cdot c_2$
      $t_{ls} \leftarrow t_{ls} + 1$
    end while
    update $\boldsymbol{\Omega}_i^{\top(t+1)} = \Gamma(\boldsymbol{\Omega}_i^{\top(t)}, -\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})[\mathcal{S}_{\{b(t)\}}], \alpha_{ls})$
    if $t_{ls} = t_{ls}^{max}$ **then**
      $\alpha^{(t+1)} \leftarrow \alpha^{(0)}$
    else
      $\alpha^{(t+1)} \leftarrow \alpha_{ls} \cdot c_2^{-2}$
    end if
    $t \leftarrow t + 1$
  end while
Output: $\boldsymbol{\Omega}_i^\star\ i = 1, \ldots, V$

iteration t, the step size $\alpha^{(t)}$ is multiplied by the factor c_2^{-2} which slightly increases the step size for the next iteration and thus enables a potential faster decay of the cost function at suitable regions. The complete step size selection approach is summarized in Algorithm 4.2.

4.4. Parameter Selection

In this section, numerical experiments to evaluate the separable analysis operator learning algorithm are provided. First, the proposed step size selection for the SGD approach is considered. In the following, experiments regarding the robustness against parameter changes are discussed, while at the end, the performance is compared to various learning algorithms from the literature.

Figure 4.3.: Training Images.

Figure 4.4.: Validation Images.

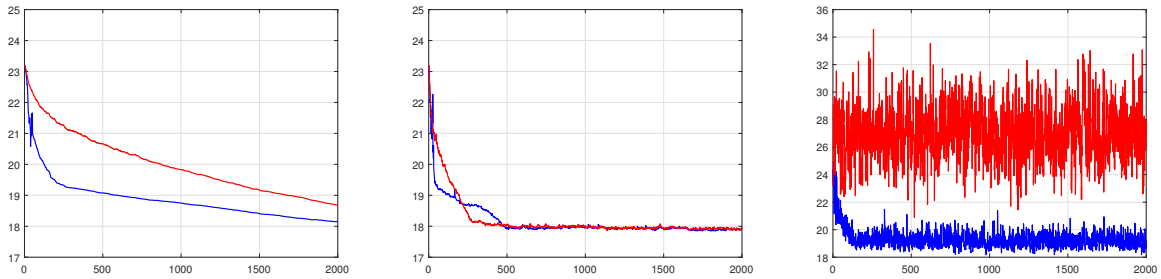
4.4.1. Impact of the Line Search

In the following, the impact of the proposed line search method that takes into account the average over previous iterations is discussed. For this purpose, a set of 20 000 structured non-flat 2D image patches, each of size 7×7, is extracted from five natural images shown in Figure 4.3. All the patches are centered and normalized to unit length. These signals serve as the training set. Analogously, another equally sized and preprocessed set is extracted from the images shown in Figure 4.4. These patches constitute the validation set. Now, the separable analysis operator learning algorithm including the proposed line search approach is compared to a setting with a fixed step size. The performance is evaluated by means of the average co-sparsity, measured via $g(\cdot)$ as introduced in (4.2), that the separable operator $\boldsymbol{\Omega}^{(t)} = \iota(\boldsymbol{\Omega}_1^{(t)}, \boldsymbol{\Omega}_2^{(t)}) \in \mathbb{R}^{81\times 49}$ achieves on the validation data set at each iteration. The parameters in the backtracking line search algorithm 4.2 are set to $c_1 = 10^{-3}$, $c_2 = 0.9$ and $t_{ls}^{max} = 20$. Both algorithms start with the same initial step size.

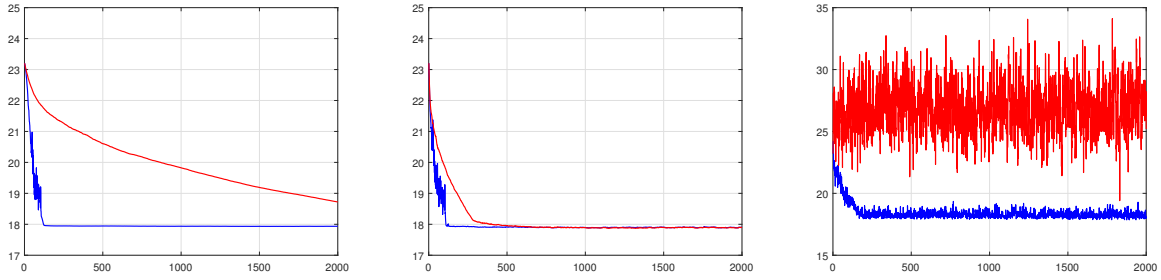
Figure 4.5 shows the progress of the validation set co-sparsity for both methods. The blue curve denotes the backtracking line search as introduced in Section 4.3.2 while the red one indicates the sparsity achieved with a fixed step size. In both settings the mini-batch size used to calculate the gradient is set to $|b(t)| \in \{5, 50\}$. From left to right, the initial step sizes read $\alpha^{(0)} \in \{0.001, 0.01, 0.5\}$.

The results from Figure 4.5 show that a fixed step size is a reasonable strategy to find a min-

(a) Initial step size $\alpha^{(0)} = 0.001$ **(b)** Initial step size $\alpha^{(0)} = 0.01$ **(c)** Initial step size $\alpha^{(0)} = 0.5$



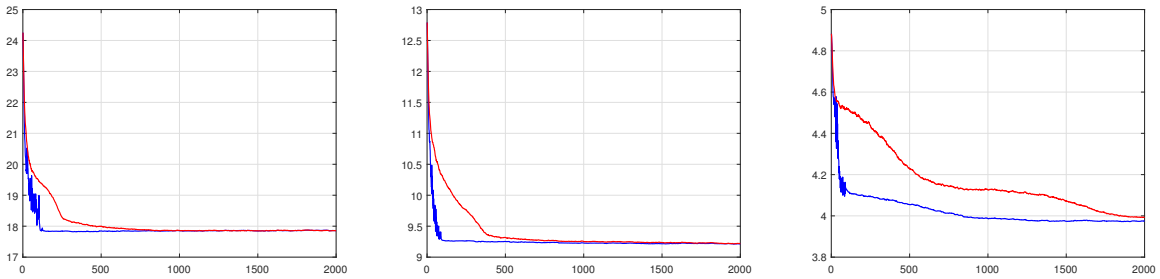
(d) Initial step size $\alpha^{(0)} = 0.001$ **(e)** Initial step size $\alpha^{(0)} = 0.01$ **(f)** Initial step size $\alpha^{(0)} = 0.5$



(g) Initial step size $\alpha^{(0)} = 0.001$ **(h)** Initial step size $\alpha^{(0)} = 0.01$ **(i)** Initial step size $\alpha^{(0)} = 0.5$

Figure 4.5.: Progress of the average co-sparsity achieved with the current separable operator that is applied to the validation set. First row: Mini-batch size $|b(t)| = 5$, Average window size $w = 1$; Second row: Mini-batch size $|b(t)| = 5$, Average window size $w = 25$; Third row: Mini-batch size $|b(t)| = 50$, Average window size $w = 25$. Backtracking line search in blue, fixed step size in red.

imizer of the operator learning problem (cf. Second column, Figures 4.5b, 4.5e, and 4.5h). However, this strategy necessitates a tedious and careful search for the best initialization of $\alpha^{(0)}$. A change in the parameter setting could require another choice of $\alpha^{(0)}$ that enables a



(a) Sparsity measure parameter $\nu = 1000$

(b) Sparsity measure parameter $\nu = 100$

(c) Sparsity measure parameter $\nu = 10$

Figure 4.6.: Progress of the average co-sparsity achieved with the current separable operator that is applied to the validation set. Initial step size $\alpha^{(0)} = 10^{-2}$, Mini-batch size $|b(t)| = 50$, Average window size $w = 25$; Backtracking line search in blue, fixed step size in red.

suitable decrease in the cost function. In contrast, the proposed line search technique offers several beneficial properties summarized below.

a) Compared to a fixed step size, the backtracking line search approach is able to adjust the step size and thus to handle initial step sizes that do not perfectly match the given setting. If $\alpha^{(0)}$ is set too small (cf. First column, Figures 4.5a, 4.5d, and 4.5g), the lifting factor c_2^{-2} in conjunction with the average over previous iterations allows for a faster convergence. Especially the comparison of Figures 4.5a and 4.5d, where the mini-batch size is kept fixed, reveals the benefit in considering the average cost over some small window w.

b) If $\alpha^{(0)}$ is chosen too big (cf. Figures 4.5c, 4.5f, and 4.5i), algorithm 4.2 decreases the learning rate until the function value decreases. In this scenario, a fixed step size leads to oscillating values for the average co-sparsity on the validation set.

c) A small initial step size in conjunction with a moderate batch size leads to a smooth progress of the validation sparsity (cf. Figures 4.5a, 4.5d, and 4.5g). Especially in this setting, the proposed approach results in a faster convergence of the sparsity.

d) For iterations where the averaged Armijo condition is not fulfilled, the current operator is only marginally updated since the step size α_{ls} is small after t_{ls}^{max} iterations. Thus, for moderate step sizes, strong oscillations in the validation sparsity are avoided.

Figure 4.6 additionally illustrates that although a fixed step size might be suitable for a

Figure 4.7.: Test Images. From left to right: Piecewise-Constant (PWC), Barbara, Boats, Lena, Peppers.

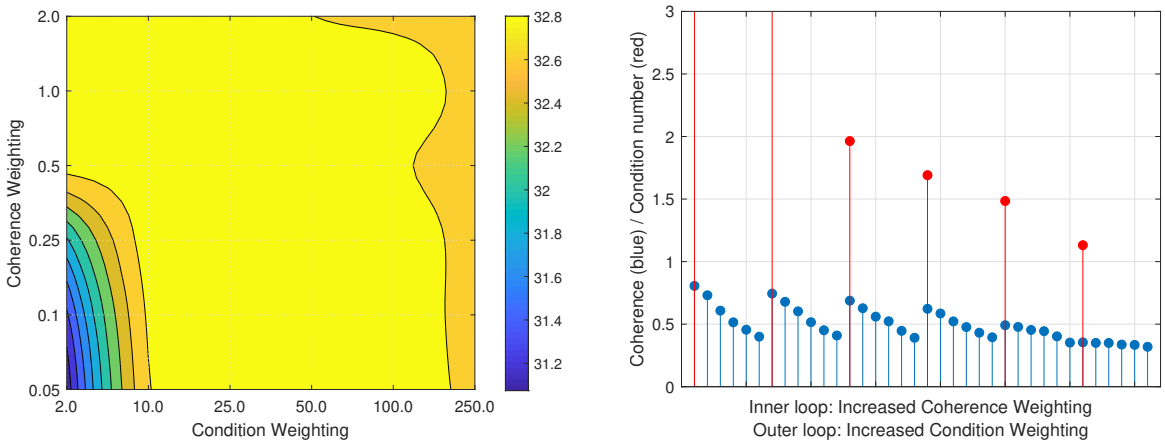
particular choice of parameters, the non-adaptivity comes at the cost of potentially suboptimal convergence if a different set of parameters is used. While the figure on the left indicates approximately equivalent progress in the average co-sparsity for both approaches, the behavior has changed in the middle and right figures, where the parameter ν in the sparsity measure (4.2) has been changed while the initial step size is kept fixed.

4.4.2. Robustness to Parameter Changes

The next experiment is intended to evaluate the robustness of the learning algorithm to the different parameters that have to be determined. Besides the size of the filters, especially the weighting coefficients κ and γ that control the condition number and the coherence of the operator play an important role in finding a suitable model. Also the proposed stopping criterion will be justified in the following. Since the regularization of inverse problems represents the primary field of application, the performance with different parameter settings is evaluated by means of an image denoising problem.

The training set is composed of 50 000 centered and normalized 2D image patches extracted from the images in Figure 4.3. The patch size is set to 7×7, while the parameter of the sparsity measure is set to $\nu = 1000$. According to the results from the previous section, the line search parameters are set to $c_1 = 10^{-3}$, $c_2 = 0.9$ and $t_{ls}^{max} = 20$ with an initial step size of $\alpha^{(0)} = 10^{-3}$, a batch size of $|b(t)| = 50$ and a window size $w = 25$ to calculate the average over the previous iterations. This setting corresponds to the validation error shown in Figure 4.5g. Unless otherwise stated, these parameters are used in all of the following experiments.

In the first experiment, the influence of the weighting parameters κ and γ is considered. In this setting, the learning algorithm runs a predefined number of iterations, which is set to $t_{max} = 2\,000$. The penalty $r(\cdot)$ that avoids a rank deficiency of the operator while simultaneously enforcing a low condition number is weighted with $\kappa \in \{2.0, 10.0, 25.0, 50.0, 100.0, 250.0\}$. The weights of the incoherence penalty $h(\cdot)$ read



(a) Average Denoising Performance in decibels (dB)

(b) Mutual coherence and Condition number

Figure 4.8.: Performance and properties of 36 learned separable operators. **(a)** Average Denoising Performance in decibels (dB) with respect to the weighting parameters. **(b)** Coherence and condition number of the learned separable operators.

$\gamma \in \{0.05, 0.1, 0.25, 0.5, 1.0, 2.0\}$. After the training on noise free samples, the learned separable operator $\boldsymbol{\Omega}^\star = \iota(\boldsymbol{\Omega}_1^\star, \boldsymbol{\Omega}_2^\star) \in \mathbb{R}^{81\times 49}$ serves as a regularizer in the Denoising problem. For this purpose, the NESTA algorithm [7] is utilized, which solves the analysis-based unconstrained inverse problem

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}\in\mathbb{R}^N} \ \tau\|\boldsymbol{\Omega}^\star(\boldsymbol{s})\|_1 + \tfrac{1}{2}\|\boldsymbol{y}-\boldsymbol{s}\|_2^2, \tag{4.21}$$

where $\boldsymbol{s}$ represents a vectorized image, $\boldsymbol{y}$ are the noisy measurements, and τ is a weighting factor. $\boldsymbol{\Omega}^\star(\boldsymbol{s})$ denotes the operation of applying the learned operator $\boldsymbol{\Omega}^\star$ to all overlapping patches of the image $\boldsymbol{s}$ via convolving each of the learned filters with the image. For all different operators, the sparsity weighting factor is set to $\tau = 0.125$. The AWGN added to the original signal has a standard deviation of $\sigma_{\text{noise}} = 10$ (assuming the image range between [0,255]). To evaluate the denoising ability, the performance on five different images shown in Figure 4.7 is tested in terms of the recovery Peak Signal to Noise Ratio (PSNR). The PSNR is calculated via

$$\text{PSNR } = 20\cdot\log_{10}\left(\frac{I_{max}^{orig}}{\frac{1}{\sqrt{N}}\|I^{rec}-I^{orig}\|_F}\right), \tag{4.22}$$

where I^{orig} denotes the original image, I^{rec} depicts the processed image, I_{max}^{orig} is the maxi-

Table 4.1.: Average Denoising performance on 5 different test images. The average PSNR in (dB) as well as the standard deviation is given for ten different trials, where the operators have been learned from different initializations.

	PWC	Barbara	Boats	Lena	Peppers
Avg. PSNR (dB)	32.01	32.27	33.10	34.56	33.54
Std. Dev.	0.06	0.03	0.02	0.03	0.02

mum value in the noise-free image and N denotes the number of pixels in the image. Figure 4.8a illustrates the average PSNR in decibels achieved over the five test images with regard to the weighting parameters. Figure 4.8b additionally shows the properties of the learned operators in terms of the mutual coherence as defined in (4.6) and the condition number. It can be clearly seen that a stronger weighting of the coherence penalty leads to a decrease of the mutual coherence of the operator. Analogously, increasing the weighting parameter κ decreases the condition number of $\boldsymbol{\Omega}$. Thus, the proposed penalties indeed allow to control the coherence as well as the condition number. Furthermore, the results in Figure 4.8a reveal that a moderate condition number is beneficial for the Denoising task at hand. It can be also observed that within a large range of parameter choices, the performance remains almost constant. This behavior clearly indicates that the algorithm is robust against changes in the weighting parameters which is extremely helpful with regard to the applicability. In the following, the weighting parameters are set to $\kappa = 25.0$ and $\gamma = 0.5$.

4.4.3. Robustness to Model Initializations

The robustness of the learning algorithm with respect to different operator initializations is evaluated by means of the same denoising problem as given in (4.21). For this purpose, ten different realizations of random analysis operator initializations are used as an input of the learning algorithm. After the separable operators have been successfully learned, the average denoising performance on the five different test images shown in Figure 4.7 is utilized to assess the robustness. AWGN with $\sigma_{\text{noise}} = 10$ has been added to the images, hence the sparsity weighting factor in (4.21) is set to $\tau = 0.125$ again. Table 4.1 presents the average PSNR as well as the standard deviation with regard to the regularization with the ten different separable operators previously learned from the training data. The low standard deviation clearly shows that the performance is consistent over different analysis operators which highlights the robustness of the presented learning algorithm against the initialization of the model.

4.4.4. Operator Size and Stopping Criterion

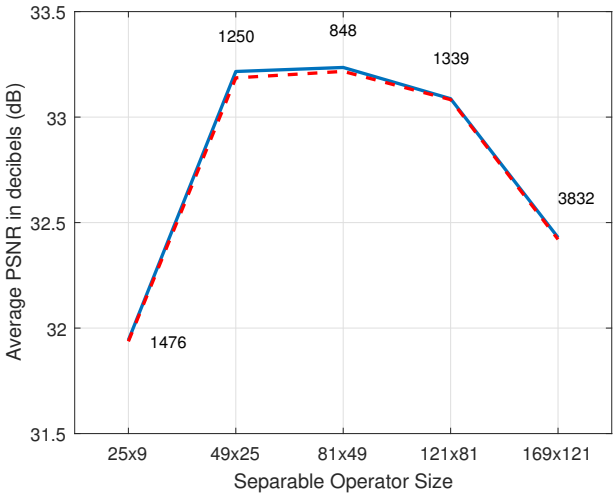
The following experiment concerns the size of the filters and the stopping criterion. For this purpose, the weighting parameters are fixed to the values stated above while the size of the separable operator is varied. The filter sizes read $n_1 = n_2 \in \{3,5,7,9,11\}$, while the number of filters k_i scales with $k_1 = k_2 \in \{5,7,9,11,13\}$. The operators are learned from noise free 2D patches extracted from the same training set as before. The number of iterations is fixed to $t_{max} = 5\,000$. Afterwards, the five test images from Figure 4.7 are contaminated with AWGN with standard deviation $\sigma_{\text{noise}} = 10$. Figure 4.9 depicts the Denoising performance achieved with the NESTA algorithm. For each operator, the weighting parameter τ is chosen that leads to the best average PSNR over the five test images.

Regarding the stopping criterion introduced in 4.3.1, Figure 4.9 also shows the performance of the operators that are returned by the learning algorithm once the stopping criterion is fulfilled. In the proposed SGD implementation, the average in Eq. (4.16) is calculated over the last $l = 500$ iterations. The execution of the algorithm stops if the relative variation of the validation set sparsity falls below the threshold $\delta = 10^{-4}$. As can be seen from the results, even a moderately sized operator achieves good denoising performance. Further increasing the patch or operator size does not improve the recovery accuracy. Also the stopping criterion appears to be a reasonable choice since the current estimate of the operator achieves almost the same performance as the one obtained after a fixed number of t_{max} iterations.

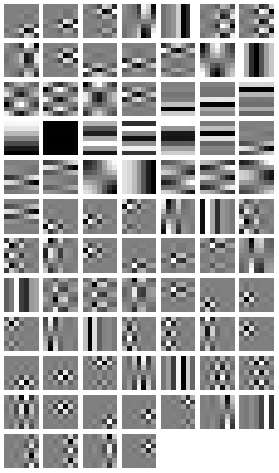
4.5. Performance Evaluation Compared to Related Work

In order to relate the performance of an analysis operator with separable structures to other approaches from the literature the same Denoising experiment as already described in Section 4.4.2 is conducted. All the learned operators are intended to serve as a sparsity prior to regularize the solution in the Denoising task. To allow for a fair comparison, the operator dimension has been adjusted to be twice overcomplete, i.e., $\boldsymbol{\Omega} \in \mathbb{R}^{100\times 49}$ for patches of size 7×7, which is a common choice in the sparse modeling literature. The weighting parameters regarding problem (4.11) have been slightly changed, namely they now read $\kappa = 35.0$ and $\gamma = 0.5$.

The evaluation involves the comparison to models that have been learned by means of various analysis operator learning algorithms presented in the literature. To be precise, the presented separable approach (*SEP*) is compared to the Analysis K-SVD method (*AKSVD*) [141], the Constrained Analysis Operator Learning framework (*CAOL*) [175], the Overcomplete Sparsifying Transform Operator Learning algorithm (*OTOL*) [125], the Geomet-



(a) Average Denoising Performance in decibels (dB) and iteration number where the stopping criterion is fulfilled.



(b) Learned Operator of size 81×49, shown as 2D filters

Figure 4.9.: (a) Average Denoising performance (PSNR in decibels) achieved with operators of various size. The size of the operator is shown on the abscissa. The solid line indicates the PSNR after t_{max} iterations, while the dotted line represents the performance of the operators returned after reaching the stopping criterion. The iteration count is given accordingly. **(b)** Learned separable operator shown as 2D filters.

ric Analysis Operator Learning approach (*GOAL*) [64], and the Analysis SimCO (*ASCO*) algorithm [38]. In order to directly assess the impact of the separability constraint, two additional analysis operators obtained with respect to Eq. (4.12) are considered. While the operator learned with the proposed SGD framework but without separable structures is denoted as (*NSEP*), the separable operator is labeled as (SEP_{kron}). In both scenarios, the weighting parameters in Eq. (4.12) are set to $\kappa = 100.0$ and $\gamma = 0.02$. All other parameters concerning the SGD optimization remain unchanged.

The training data for each learning method constitute a set of 50 000 noise free patches extracted from the images shown in Figure 4.3. According to the respective algorithm design, the patches have been centered and/or normalized. The number of iterations in the learning algorithms is set to the value suggested by the authors. Table 4.2 shows the execution time of the algorithms measured on a standard desktop computer.

After the learning phase, the operators are utilized in the sparsity regularizer of the Denoising problem given in (4.21), where the co-sparsity is measured with respect to centered patches. Again, the NESTA algorithm [7] is used to optimize the objective and to find a denoised version of the input signal. The five test images already shown in Figure 4.7 have

Table 4.2.: Algorithm execution time in seconds

Algorithm	SEP	NSEP	AKSVD	CAOL	OTOL	GOAL	ASCO
Time (sec)	37,2	561,1	21.038,4	7.744,6	99,2	141,2	2.333,9

been artificially corrupted via adding AWGN with standard deviation $\sigma_n \in \{10, 20, 30\}$. For each noise level, different choices of the parameter τ in (4.21), that weights the sparsity penalty against the data fidelity term, have been tested. Table 4.3 summarizes the Denoising performance in terms of PSNR (in decibels) as defined in (4.22). For each analysis operator, the parameter τ that leads to the best average PSNR has been chosen.

In the image processing literature it is also common to assess the reconstruction quality in terms of the Structural Similarity (SSIM) introduced in [165], which is also a full reference metric, i.e., the performance is evaluated with regard to the original clean image. The SSIM takes into account the degradation of structural information, a property that the human visual perception is highly adapted to. Contrary to the PSNR measure, the calculation of the structural similarity is performed locally within small windows, which are moved over the entire image. For each window w_r in the reconstructed image I^{rec}, the similarity to its corresponding window w_o in the original image I^{orig} is computed via

$$\text{SSIM}\,(w_r, w_o) = \frac{(2\mu_r\mu_o + C_1)(2\sigma_{ro} + C_2)}{(\mu_r^2 + \mu_o^2 + C_1)(\sigma_r^2 + \sigma_o^2 + C_2)}, \tag{4.23}$$

where μ, σ^2 and σ_{ro} denote the mean, the variance and the covariance for the particular windows, and C_1, C_2 represent constants to avoid instabilities. Eventually, the Mean Structural Similarity (MSSIM) between the two images I^{rec} and I^{orig} reads

$$\text{MSSIM}\,(I^{rec}, I^{orig}) = \frac{1}{M}\sum_{i=1}^{M}\text{SSIM}\,(w_{r,i}, w_{o,i}), \tag{4.24}$$

which allows to assess the overall image quality. Table 4.3 also lists the MSSIM quality metric.

The presented results in table 4.3 indicate that using separable filters does not reduce the image restoration performance and that separable filters are competitive with non-separable ones. On the one hand, the performance of the three approaches *SEP*, SEP_{kron} and *NSEP* is almost identical. It can be concluded that first, the proposed penalty weightings for Eq. (4.11) and Eq. (4.12) lead to similar operators. Second, the direct comparison between SEP_{kron} and *NSEP* reveals that the impact of the structural constraint on the achieved denoising results is negligible. On the other hand, the competitive performance

of separable operators is highlighted by means of the last column of the table that presents the average PSNR and MSSIM across all images. While the bold face number corresponds to the best performance, the underlined value shows the reconstruction quality achieved with a separable operator. For all different noise realizations, the deviation in PSNR is at most only 0.07dB. Regarding the MSSIM, the performance is slightly worse compared to the best competitor, but still on par to the quality achieved with the operator without the separability constraint (*NSEP*).

Having these results in mind, the benefit of separable operators gets even more pronounced with regard to the computational effort during the learning phase. Table 4.2 illustrates that the SGD implementation with separability constraint (*SEP*) requires much less time, and therefore less iterations and fewer samples, to reach the dropout criterion compared to the SGD algorithm without enforcing a separable structure of the learned operator (*NSEP*). Moreover, the SGD approach for separable analysis operator learning is the fastest learning scheme among the listed algorithms.

4.6. Summary

In this chapter, an analysis operator learning algorithm that imposes an additional separability constraint onto the model is introduced. Separable operators are especially useful for multidimensional data since the computational burden in learning and applying the filters is significantly reduced. The derivation of the cost function revealed that the separability constraint is easy to integrate into the objective while the properties of the operator can be still adjusted flexibly via the used penalty functions. In accordance to the results from the literature, a well-conditioned separable analysis operator turned out to be very useful.

In order to address an online learning scenario where the samples are acquired sequentially in time, a SGD algorithm on manifolds is presented to tackle the optimization problem. Empirical results indicate that the proposed variable step size selection makes the algorithm self-adapting to different parameter settings without a tedious search for an optimal learning rate. Furthermore, extensive numerical experiments have shown that the learning algorithm is robust against parameter changes and different initializations.

Last but not least, the separable structure of the analysis operator only marginally influences the performance of the model when it is used as a regularizer in inverse problems. Even more, almost the same accuracy is achieved although the effort in the learning phase is significantly reduced compared to various state-of-the-art algorithms.

In sum, separability has been proven a very valuable property, which can be easily incorporated into the objective function, significantly reduces the training complexity, and still leads to competitive performance in image processing tasks.

Table 4.3.: Denoising experiment for five different test images corrupted by three noise levels. For each model the achieved PSNR in decibels (dB) is shown on the left, while the MSSIM is given on the right.

σ_n	Algo.	PWC		Barbara		Boats		Lena		Peppers		Avg.	
10	SEP	31.78	0.929	32.12	0.914	33.18	0.878	35.04	0.908	33.51	0.914	33.13	0.909
	SEP_{kron}	31.85	0.930	32.12	0.914	33.16	0.877	35.04	0.908	33.53	0.914	33.13	0.909
	NSEP	31.68	0.929	32.15	0.915	33.12	0.878	35.16	0.911	33.50	0.916	33.12	0.909
	AKSVD	30.99	0.869	30.49	0.860	30.78	0.819	32.08	0.829	31.17	0.849	31.10	0.845
	CAOL	31.85	0.933	32.08	0.915	32.82	0.874	34.89	0.911	33.32	0.915	32.99	**0.910**
	OTOL	30.86	0.900	32.23	0.907	32.95	0.868	34.75	0.893	33.15	0.899	32.79	0.893
	GOAL	31.76	0.927	32.32	0.916	33.20	0.878	35.16	0.908	33.56	0.914	**33.20**	0.909
	ASCO	29.86	0.869	31.25	0.884	32.46	0.853	33.99	0.866	32.58	0.881	32.03	0.871
20	SEP	27.66	0.837	28.13	0.823	29.92	0.798	31.54	0.835	29.88	0.840	29.43	0.827
	SEP_{kron}	27.71	0.838	28.13	0.823	29.91	0.796	31.56	0.835	29.89	0.839	29.44	0.826
	NSEP	27.61	0.838	28.19	0.828	29.89	0.798	31.70	0.840	29.94	0.844	**29.47**	0.830
	AKSVD	26.09	0.767	26.02	0.733	26.59	0.675	28.24	0.736	26.77	0.740	26.74	0.730
	CAOL	27.66	0.849	28.18	0.832	29.55	0.792	31.54	0.850	29.72	0.847	29.33	0.834
	OTOL	26.94	0.846	28.07	0.831	29.76	0.798	31.85	0.855	29.83	0.851	29.29	0.836
	GOAL	27.33	0.865	28.02	0.832	29.92	0.805	32.03	0.863	30.04	0.862	**29.47**	**0.845**
	ASCO	26.20	0.747	26.78	0.756	28.91	0.747	30.11	0.748	28.86	0.776	28.17	0.755
30	SEP	25.55	0.778	25.98	0.749	28.10	0.741	29.70	0.789	27.94	0.790	27.45	0.769
	SEP_{kron}	25.59	0.780	25.98	0.749	28.10	0.740	29.74	0.790	27.95	0.789	27.47	0.769
	NSEP	25.52	0.781	26.03	0.756	28.08	0.742	29.88	0.797	28.02	0.796	**27.50**	0.774
	AKSVD	23.51	0.678	23.82	0.636	24.53	0.586	26.25	0.669	24.41	0.658	24.50	0.645
	CAOL	25.32	0.825	25.88	0.766	27.59	0.734	29.76	0.829	27.74	0.821	27.27	0.795
	OTOL	25.23	0.778	26.07	0.759	27.99	0.739	29.93	0.804	27.91	0.793	27.43	0.775
	GOAL	25.20	0.819	25.66	0.753	28.06	0.751	30.21	0.831	28.13	0.824	27.45	**0.795**
	ASCO	24.46	0.694	24.39	0.661	27.17	0.689	28.47	0.704	27.13	0.729	26.33	0.695

Chapter 5.

Empirical Investigation of the Sample Complexity and the Model Generalization

The task of learning the underlying structure from training samples immediately raises one of the fundamental questions in machine learning theory, namely does the model generalize to unseen data? Ideally, after learning, the model is representative for all signals from the same signal class. Simultaneously, the model should have the capability to exclude unwanted samples that do not belong to the same distribution, e.g. noisy samples. These two properties are highly desirable in signal reconstruction tasks where the learned model serves as a prior.

Intuitively, generalization will be achieved if all possible variations of the signal class are presented to the learning algorithm. Full knowledge about the data distribution however, renders the learning task meaningless. Since in real world scenarios the exact distribution of the samples is generally unknown, we are interested in a uniform generalization bound that enables to answer the following question: How many samples do we need for a reliable estimate of the model? This analysis is also referred to as the sample complexity with examples given in [97, 161, 61]. Specifically, [61] provides a broad overview of sample complexity results for various matrix factorizations. In our work [147], we investigate the sample complexity of the separable analysis operator learning problem whose main results are outlined in this section. The following contributions are addressed.

- The denoising results from the last chapter indicate that the separability constraint, that is imposed on the analysis operator, significantly reduces the necessary training time. Nevertheless, the algorithm still provides a reliable estimate of the model. Besides this task oriented evaluation, in this chapter, both theoretical and empirical results are given confirming that in the separable case less training signals are required in the learning process compared to an unstructured operator learning approach.
- In order to provide a task independent criterion to assess the generalization ability of the learned model, the Estimated Kullback-Leibler divergence between the distri-

bution of the training signals and the distribution of signals that strictly follow the analysis UoS model is considered. As expected, compared to analytically given sparsifying transforms, the presented learning approach clearly provides a better adaptability of the model to the underlying signal distribution. Moreover, the same evaluation framework additionally encourages the choice of the previously determined weighting parameters.

5.1. Sample Complexity

Let $\boldsymbol{S} = [\mathrm{vec}(\mathcal{S}_1), \ldots, \mathrm{vec}(\mathcal{S}_T)] = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T]$, $\boldsymbol{s}_j \in \mathfrak{X}$ denote a set of training samples, with each sample drawn according to an underlying distribution $\mathbb{P}$ over $\mathfrak{X}$[1]. To avoid trivial solutions, e.g. the zero matrix or rank deficient matrices, the sought operator $\boldsymbol{\Omega} \in \mathbb{R}^{K \times N}$ is an element of the constraint set $\mathfrak{C}$. A separable structure is enforced by further restricting the constraint set to the subset $\{\boldsymbol{\Omega}^\top \in \mathsf{OB}(N, K) \,:\, \boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V),\ \boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i)\}$ with the appropriate dimensions $N = \prod_i n_i$ and $K = \prod_i k_i$. Now, let $f \colon \mathfrak{C} \times \mathfrak{X} \to \mathbb{R}$ denote the learning objective as defined in (4.11). All the possible realizations of $\boldsymbol{\Omega}$ can be summarized as the family of functions $\mathcal{F} = \{f(\boldsymbol{\Omega}, \cdot) : \boldsymbol{\Omega}^\top \in \mathfrak{C}\}$ that map the sample space $\mathfrak{X}$ to $\mathbb{R}$. The ultimate goal of the learning process consists in finding the function $f \in \mathcal{F}$ for which the *expected value*

$$\mathbb{E}[f] := \mathbb{E}_{\boldsymbol{s} \sim \mathbb{P}}[f(\boldsymbol{\Omega}, \boldsymbol{s})] \tag{5.1}$$

over all possible inputs $\boldsymbol{s} \sim \mathbb{P}$ is minimal. The minimizer $f^\star = \arg\min_{f \in \mathcal{F}} \mathbb{E}[f]$ thus can be considered as the ideal function, i.e., the function that generalizes best to the underlying distribution $\mathbb{P}$. Unfortunately, since the true distribution is unknown, the best we can do is to find a minimizer of the *empirical mean* over some representative training set $\boldsymbol{S}$, which is defined as

$$\widehat{\mathbb{E}}_{\boldsymbol{S}}[f] := \tfrac{1}{T} \sum_{j=1}^{T} f(\boldsymbol{\Omega}, \boldsymbol{s}_j). \tag{5.2}$$

It is hoped that (5.2) behaves like minimizing the expectation (5.1) despite the noise introduced by the simplified procedure. Analogously, let $f_{\boldsymbol{S}}^\star = \arg\min_{f \in \mathcal{F}} \widehat{\mathbb{E}}_{\boldsymbol{S}}[f]$ denote the best function for the training set $\boldsymbol{S}$.

[1] Note that for ease of notation, vectorized samples are used in the subsequent part of this chapter. Nevertheless, the separable operators are sill learned on non-vectorized two-dimensional signals.

The goal of the sample complexity analysis is now to upper bound the difference between the expected error $\mathbb{E}[f]$ and the approximate solution $\widehat{\mathbb{E}}_{\boldsymbol{S}}[f]$ over all possible functions $f \in \mathcal{F}$ and all training sets $\boldsymbol{S}$ with samples independently drawn from the distribution $\mathbb{P}$. Following the analysis outlined in [147], this upper bound reads as follows.

Theorem 1. *(Theorem 9 in [147]). Let $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_T]$ be a set of samples independently drawn according to a distribution within the unit ℓ_2-ball in $\mathbb{R}^N$. Let $f(\boldsymbol{\Omega}, \boldsymbol{s}) = g(\boldsymbol{\Omega}, \boldsymbol{s}) + r(\boldsymbol{\Omega}) + h(\boldsymbol{\Omega})$ as previously defined where the sparsity promoting function g is λ-Lipschitz. Finally, let the function family $\mathcal{F}$ be defined as $\mathcal{F} = \{f(\boldsymbol{\Omega}, \cdot) : \boldsymbol{\Omega}^\top \in \mathfrak{C}\}$, where $\mathfrak{C}$ is either $\mathsf{OB}(N, K)$ for the non-separable case or the subset $\{\boldsymbol{\Omega}^\top \in \mathsf{OB}(N, K) : \boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \dots, \boldsymbol{\Omega}_V),\ \boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i)\}$ for the separable case. Then we have*

$$\mathbb{E}[f] - \widehat{\mathbb{E}}_{\boldsymbol{S}}[f] \leq \sqrt{2\pi}\,\frac{\lambda C_{\mathfrak{C}}}{\sqrt{T}} + 3\sqrt{\frac{2\lambda^2 K \ln(2/\delta)}{T}} \tag{5.3}$$

with probability at least $1 - \delta$, where $C_{\mathfrak{C}}$ is a constant that depends on the constraint set. In the non-separable case the constant is defined as $C_{\mathfrak{C}} = K\sqrt{N}$, whereas in the separable case it is given as $C_{\mathfrak{C}} = \sum_i k_i \sqrt{n_i}$.

In order to link the sample complexity results to the proposed SGD optimization approach, Bottou [11, 12] provides some helpful insights that are discussed in the following. Additional to $f^\star$ and $f_{\boldsymbol{S}}^\star$ defined above, let $\widetilde{f}_{\boldsymbol{S}}$ denote the solution found by the employed optimization algorithm. Clearly, $\widetilde{f}_{\boldsymbol{S}}$ depends on the chosen parameters like for example the step size or the initialization. Now in his work, Bottou considers the excess error $\varepsilon = \mathbb{E}[\widetilde{f}_{\boldsymbol{S}}] - \mathbb{E}[f^\star]$, which describes the difference between the ideal solution and the solution found by the optimization algorithm. The excess error can be further split into the sum $\varepsilon = \varepsilon_{\text{opt}} + \varepsilon_{\text{est}}$. The optimization error $\varepsilon_{\text{opt}} = \mathbb{E}[\widetilde{f}_{\boldsymbol{S}}] - \mathbb{E}[f_{\boldsymbol{S}}^\star]$ indicates how well the found solution resembles the ideal solution that could be found by minimizing the empirical mean. It clearly depends on the design of the optimization algorithm. The second term, the estimation error $\varepsilon_{\text{est}} = \mathbb{E}[f_{\boldsymbol{S}}^\star] - \mathbb{E}[f^\star]$, measures the deviation between the ideal solution obtained through minimizing the expected value and the optimal solution with respect to the empirical average. Figure 5.1 schematically illustrates these concepts.

Having in mind the result from Theorem 1, the estimation error ε_{est} is closely connected to the sample complexity result from above. Based on the proof given in [147], the estimation error can be upper bounded via

$$\varepsilon_{\text{est}} \leq 2\sqrt{2\pi}\,\frac{\lambda C_{\mathfrak{C}}}{\sqrt{T}} + 6\sqrt{\frac{2\lambda^2 J \ln(2/\delta)}{T}} \tag{5.4}$$

with probability at least $1 - \delta$.

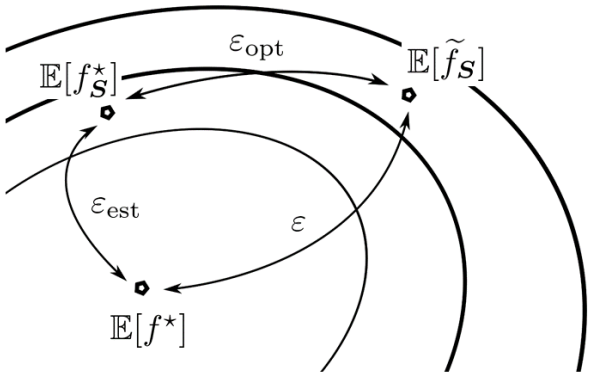


Figure 5.1.: Schematic illustration of the excess error.

From the observations above one can draw two important conclusions. First, do not waste energy on optimization. Even with an optimal optimization algorithm, the best one can reach is $\mathbb{E}[f^{\star}_{\boldsymbol{S}}]$ which might be still far away from the ideal solution. Second, to make $\mathbb{E}[f^{\star}_{\boldsymbol{S}}]$ approach $\mathbb{E}[f^{\star}]$ one has to keep the estimation error ε_{est} small. On the one hand, this can be achieved via increasing the size of the training set. Since the number of samples T occurs in the denominator, Eq. (5.4) confirms this intuition. On the other hand, the estimation error bound depends on the constant $C_{\mathfrak{C}}$. Note that for the non-separable case we have $C_{\mathfrak{C}} = K\sqrt{N}$, while for the separable case it reads $C_{\mathfrak{C}} = \sum_i k_i \sqrt{n_i}$. Consequently, the estimation error for learning separable operators has a lower bound compared to learning without this structural constraint. These theoretical results will be evaluated empirically in the remainder of this chapter.

5.2. Model Generalization

After the learning phase, we say that a model generalizes if it performs equally well on previously unseen data from the same distribution like the training data. In this work, the generalizability is evaluated in two different ways. Both approaches are intended to overcome the problem of not knowing the true distribution of the training signals. Since in the introduced learning framework we aim at minimizing the sparsity, the utilized sparsity measure is considered first to assess the generalization behavior of the learned model. The second attempt is based on a divergence criterion which is intended to show that the learned model is descriptive for the distribution of the training data.

As already outlined above, generalization implies that the difference between the empirical loss evaluated on the training set $\boldsymbol{S}$ and the expectation over the true distribution $\mathbb{P}$ is minimal. Since in general we do not have access to the expectation with respect to the distribution, the generalization is often assessed based on the difference between the training and validation error. For this purpose, the whole set of samples $\boldsymbol{S}$ is split into the two sets $\boldsymbol{S}_{\text{train}} \sim \mathbb{P}$ and the 'held out' validation set $\boldsymbol{S}_{\text{val}} \sim \mathbb{P}$. If the model generalizes well, the following relation should be fulfilled

$$\left| \widehat{\mathbb{E}}_{\boldsymbol{S}_{\text{train}}}[\widetilde{f}_{\boldsymbol{S}_{\text{train}}}] - \widehat{\mathbb{E}}_{\boldsymbol{S}_{\text{val}}}[\widetilde{f}_{\boldsymbol{S}_{\text{train}}}] \right| \approx 0, \tag{5.5}$$

where $\widetilde{f}_{\boldsymbol{S}_{\text{train}}}$ is the output of the optimization algorithm evaluated on $\boldsymbol{S}_{\text{train}}$.

Besides generalization, another important aspect is the discrimination ability, i.e., the learned model should only generalize to signals from the same distribution the training signals originate from. Signals from any other distribution should not be well described by the learned model. Given the signal set $\overline{\boldsymbol{S}}_{\text{val}}$ with signals sampled from a different distribution $\overline{\mathbb{P}}$, for a model that only generalizes to signals from $\mathbb{P}$ we additionally expect

$$\left| \widehat{\mathbb{E}}_{\boldsymbol{S}_{\text{train}}}[\widetilde{f}_{\boldsymbol{S}_{\text{train}}}] - \widehat{\mathbb{E}}_{\overline{\boldsymbol{S}}_{\text{val}}}[\widetilde{f}_{\boldsymbol{S}_{\text{train}}}] \right| > \delta \geq 0, \tag{5.6}$$

to ensure that the learned model prefers signals from $\mathbb{P}$ over $\overline{\mathbb{P}}$.

Divergence Criterion

During the learning process, the model captures the underlying structure of the training signals $\boldsymbol{S}_{\text{train}}$ in the sense that these signals exhibit an (approximately) co-sparse representation. Consequently, the distribution of some signals $\boldsymbol{S}_{\text{UoS}}$ that strictly follow the analysis Union-of-Subspace model described by the learned analysis operator, should be as close as possible to the distribution of the training signals. Formally, this assumption can be summarized as follows

Assumption 1. *A suitable analysis operator that adequately captures the structure of the training signals can be considered a generative model such that the divergence between the distribution of the generated signals and the distribution of the given training signals is as low as possible.*

As such the learned analysis operator may serves as a prior in inverse problems.

Since in real world scenarios the true distribution of both signal sets is unknown, the divergence is estimated by means of the Estimated Kullback-Leibler (EKL) divergence as introduced in [164]. Formally, let us assume that we are given two sets $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ that

contain n-dimensional signals drawn *i.i.d.* from $\mathbb{P}$ and $\mathbb{Q}$, respectively. Furthermore, let $\rho_k(i)$ denote the Euclidean distance between $\boldsymbol{s}_{1,i}$ and its k-Nearest Neighbor (k-NN) in $\{\boldsymbol{s}_{1,j}\}_{j\neq i}$ and let $\nu_k(i)$ represent the distance to the k-NN in $\boldsymbol{S}_2$. The divergence according to [164] is estimated via

$$D_{EKL}(\mathbb{P} \parallel \mathbb{Q}) = \frac{n}{T_1}\sum_{i=1}^{T_1}\log\frac{\nu_k(i)}{\rho_k(i)} + \log\frac{T_2}{T_1-1}, \tag{5.7}$$

with T denoting the number of samples in the respective set.

5.3. Evaluation

In the subsequent part of this chapter, numerical results concerning the superior sample complexity in favor of the separable approach are given. Furthermore, the suitability of the Estimated Kullback-Leibler divergence as a task independent criterion to assess the generalization ability is analyzed. As already mentioned in Section 4.2.4, in this chapter the objective given in Eq. (4.12) is used in order to evaluate the problem of learning structured and non-structured analysis operators.

5.3.1. Sample Complexity Evaluation

The sample complexity bound presented in Section 5.1 indicates that a reliable estimate of a separable operator can be learned from less training samples compared to a non-structured one. The following numerical experiments are intended to confirm these results by means of empirical verifications. For this purpose, synthetically generated sampled are used to control the experimental setup and to enable the assessment of the reliability of the current model. Given the ground truth separable operator $\boldsymbol{\Omega}_{\mathrm{GT}}$ the synthetic data set is generated via projecting signals onto the orthogonal complement of the subspace identified by selected rows of the operator. Let Λ_j denote the set of indices as defined in Section 1.1.2, the projection is realized via $\boldsymbol{s}_j^{\mathrm{proj}} = (\boldsymbol{I}_n - \boldsymbol{\Omega}_{\Lambda_j}^\top(\boldsymbol{\Omega}_{\Lambda_j}\boldsymbol{\Omega}_{\Lambda_j}^\top)^{-1}\boldsymbol{\Omega}_{\Lambda_j})\boldsymbol{s}_j$. After the projection step, the signals strictly follow the analysis Union-of-Subspace model introduced in Section 1.1.2, i.e., for each signal, $|\Lambda_j|$ rows of $\boldsymbol{\Omega}$ are orthogonal to the signal such that $\boldsymbol{\Omega}_{\Lambda_j}\boldsymbol{s}_j = \boldsymbol{0}$. The ground truth analysis operator $\boldsymbol{\Omega}_{\mathrm{GT}} \in \mathbb{R}^{81\times 49}$ has been learned from normalized and centered patches $\boldsymbol{S}_j \in \mathbb{R}^{7\times 7}$ extracted from natural images. The full set of samples has the size $T = 100\,000$. The co-sparsity, i.e., the number of zero filter responses, is fixed to 25. Given the artificially generated training data, separable as well as non-separable analysis operators are learned. A reliable estimate of the model is assumed if the current iterate $\boldsymbol{\Omega}_{\mathrm{learned}}$ is

close to the generative operator $\boldsymbol{\Omega}_{\text{GT}}$. More precisely, the deviation of the individual filters to the ground truth is utilized as a measure. However, one has to bear in mind that there is an inherent sign and permutation ambiguity in the learned filters. Hence, the absolute values of the correlation of the filters over all possible permutations is considered.

Let $\tilde{\boldsymbol{\omega}}_i$ denote the i-th row of $\boldsymbol{\Omega}_{\text{learned}}$ and let $\boldsymbol{\omega}_j$ correspond to the j-th row of $\boldsymbol{\Omega}_{\text{GT}}$. Both filters are represented as column vectors. Now, the deviation of these filters from each other is defined as $c_{ij} = 1 - |\tilde{\boldsymbol{\omega}}_i^\top \boldsymbol{\omega}_j|$. A standard procedure to measure the overall distance of $\boldsymbol{\Omega}_{\text{learned}}$ to $\boldsymbol{\Omega}_{\text{GT}}$ is to correlate one ground truth filter by a time to all the rows given in $\boldsymbol{\Omega}_{\text{learned}}$ and subsequently summing up the minimum deviations obtained at each iteration. This strategy has the drawback that each filter in $\boldsymbol{\Omega}_{\text{learned}}$ may be selected several times which can lead to false detections. For example if $\boldsymbol{\Omega}_{\text{GT}}$ exhibits highly correlated rows, these filters might be all mapped to only one filter in $\boldsymbol{\Omega}_{\text{learned}}$ resulting in a close overall distance. To alleviate this problem, the following approach is pursued. First the deviations c_{ij} for all possible combinations of i and j are computed which results in the confusion matrix $\boldsymbol{C}$, where the i,j-entry $\boldsymbol{C}_{ij}$ is 0 if $\tilde{\boldsymbol{\omega}}_i$ is equal to $\boldsymbol{\omega}_j$. Building the confusion matrix accounts for the permutation ambiguity between $\boldsymbol{\Omega}_{\text{GT}}$ and $\boldsymbol{\Omega}_{\text{learned}}$. Next, the Hungarian-method [78] is utilized to determine the path through the confusion matrix $\boldsymbol{C}$ with the lowest accumulated cost under the constraint that each row and each column is visited only once. In the end, the coefficients along the path are accumulated and this sum serves as the error measure denoted as $H(\boldsymbol{C})$. This strategy has the advantage that different retrieved filters $\tilde{\boldsymbol{\omega}}_i$ are not matched to the same filter $\boldsymbol{\omega}_j$.

In order to evaluate the sample complexity, i.e., the number of training samples necessary to recover the original model, the mini-batch size in the SGD implementation is varied. The employed mini-batch sizes read $|b(k)| \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$, while the performance of the operator retrieval is evaluated over five trials, i.e., five different synthetic sets are generated in advance. Figure 5.2 summarizes the results for the sample complexity experiment. For each mini-batch size, the error over all five trials is illustrated. The left box corresponds to the non-separable approach and accordingly the right box denotes the error for separable filters. While the horizontal dash inside the boxes indicates the median over all five trials the boxes themselves represent the mid-50%. The dotted dashes above and below the boxes indicate the maximum and minimum error obtained, where crosses belong to outliers. The distance between the estimated operator $\boldsymbol{\Omega}_{\text{learned}}$ and $\boldsymbol{\Omega}_{\text{GT}}$ obtained after executing $t \in \{100, 1000, 10000\}$ iterations is given. In the first experiment shown in the left column of Figure 5.2, the recovery of the ground truth operator starts from a fixed initialization of the analysis operator, where the entries are drawn from the normal distribution with a subsequent normalization of the rows to unit length. In the second experiment, the setting is slightly changed. In each trial the recovery starts from

different initializations of the separable operator. Furthermore, AWGN with standard deviation 0.05 is added to each synthetically generated signal sample. The results are shown in the right column of Figure 5.2.

From both experiments, it is evident that the error decreases significantly faster in the case when an analysis operator with separable structures is learned. For example, in the noise-free scenario, the deviation between the current iterate of the separable operator and the ground truth operator approaches zero even for small mini-batch sizes. Since in the SGD framework, the number of iterations reflects to the number of samples visited so far, the deviation error is directly connected to the sample complexity. The same general trend can be observed in the second experiment, where random operator initializations and noise contaminated samples are used.

To conclude, the presented results confirm the theoretical sample complexity results and indicate that imposing a separable structure on the analysis operator significantly reduces the amount of samples needed in order to provide a reliable estimate of the model.

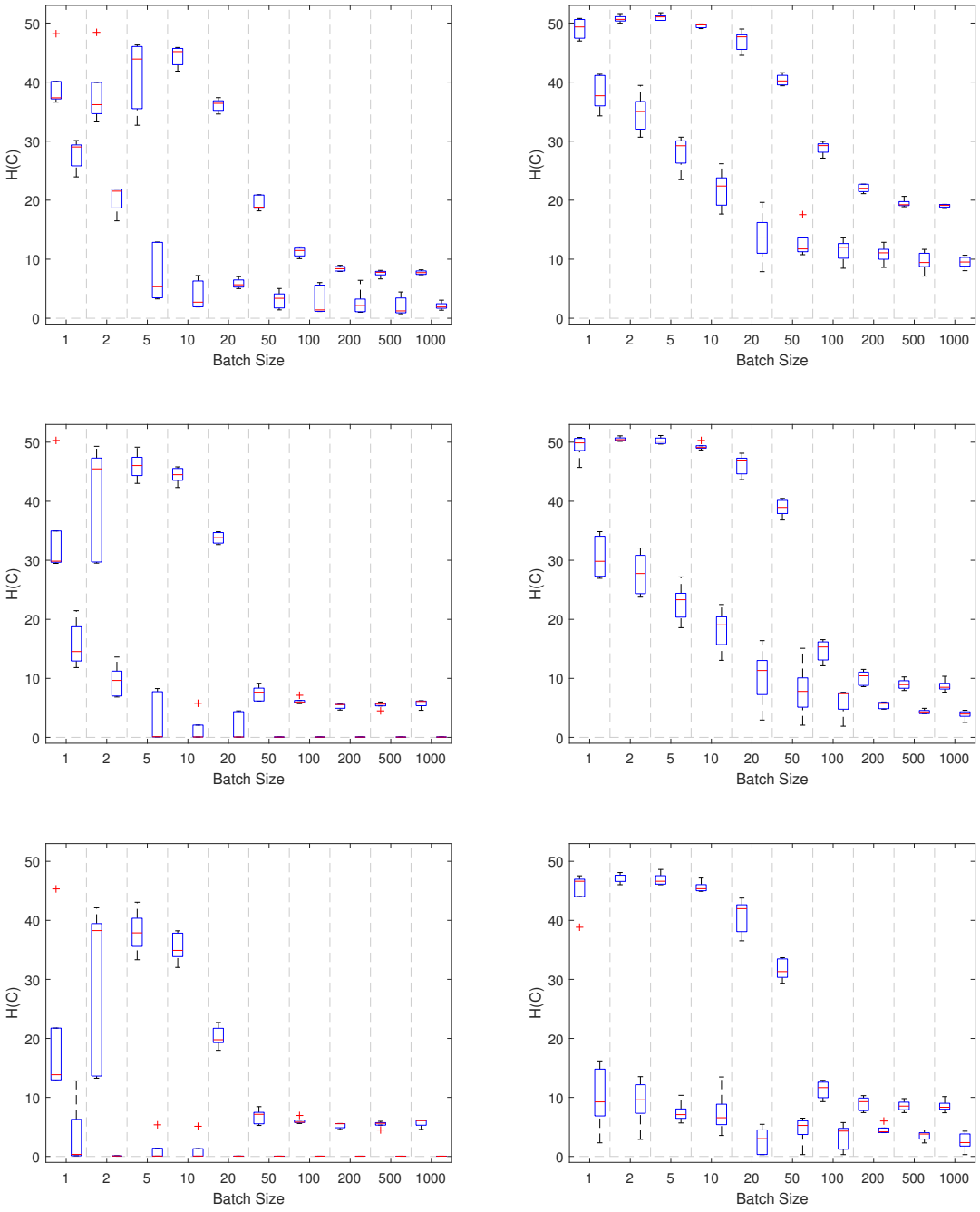


Figure 5.2.: Distance between the estimated operator $\boldsymbol{\Omega}_{\text{learned}}$ and $\boldsymbol{\Omega}_{\text{GT}}$ obtained after executing different numbers of iterations. From first to third row: Error after $t = 100, 1000$, and 10000 iterations. $H(\boldsymbol{C})$ indicates the distance error obtained after applying the Hungarian method on the confusion matrix. First column: In each trial, a fixed initialization of $\boldsymbol{\Omega}_{\text{learned}}$ is used. The artificially generated signals are noise free. Second column: In each trial, $\boldsymbol{\Omega}_{\text{learned}}$ is initialized randomly and learned from noise corrupted samples.

5.3.2. Generalization Evaluation

As already pointed out in Section 5.2, the resulting analysis operators should generalize well on previously unseen data from the same signal class. To evaluate this behavior, let $\boldsymbol{S}_{\text{train}} \sim \mathbb{P}$ denote a set of T n-dimensional samples, i.e., $\boldsymbol{S}_{\text{train}} \in \mathbb{R}^{n \times T}$, used for training. Analogous to the previous experiments, the signals are patches extracted from the images shown in Figure 4.3. The test set $\boldsymbol{S}_{\text{test}} \sim \mathbb{P}$ is comprised of samples from the same distribution, i.e., they are sampled from the same images.

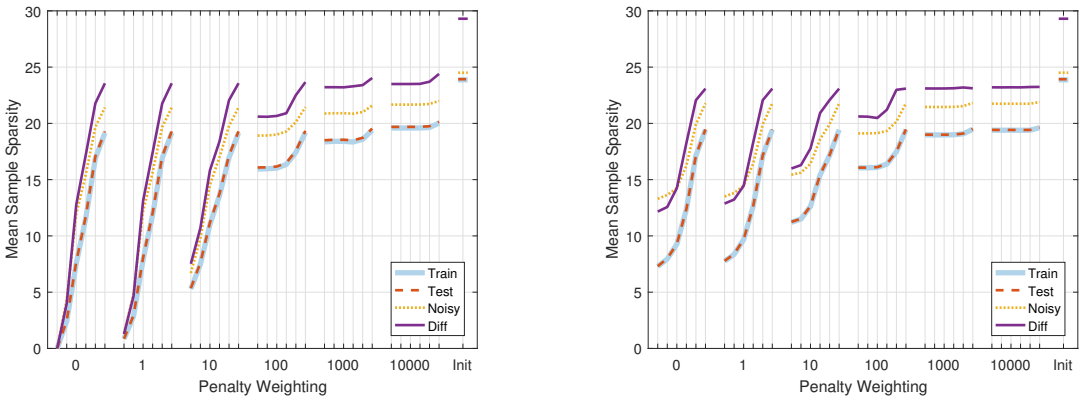
Besides generalization, the learned analysis operator also has to be discriminative for samples from the distribution $\mathbb{P}$, i.e., the model should prefer signals from $\mathbb{P}$ over signals from some other distribution $\overline{\mathbb{P}}$. In the context of co-sparsity, this means that the signals from $\mathbb{P}$ should exhibit a sparser representation compared to samples from $\overline{\mathbb{P}}$. For the numerical analysis, these signals are generated in two different ways. On the one hand, the test samples are artificially corrupted with AWGN resulting in $\boldsymbol{S}_{\text{noisy}} \sim \overline{\mathbb{P}}$. On the other hand, $\boldsymbol{S}_{\text{diff}} \sim \overline{\mathbb{P}}$ is composed of patches extracted from the MNIST database[2], i.e., the samples represent sections of short line segments.

To evaluate the generalization as well as discrimination behavior, both type of operators (separable and non-separable) are trained on $T = 10\,000$ samples $\boldsymbol{S}_{\text{train}}$. All the samples in the aforementioned four sets are centered and normalized to unit length. The size of the operators read $\boldsymbol{\Omega} \in \mathbb{R}^{81 \times 49}$, i.e., square patches of size 7×7 are extracted from the respective images. In order to evaluate the impact of the penalties, their weighting parameters are set to $\kappa \in \{0.0, 1.0, 10.0, 100.0, 1000.0, 10000.0\}$, and $\gamma \in \{0.0, 0.0002, 0.002, 0.02, 0.2, 2.0\}$, respectively. To guarantee convergence for all parameter choices, the batch size in the SGD framework is set to $|b(t)| = 500$, while the algorithm terminates after a fixed number of $t = 5\,000$ iterations.

After the learning phase, where the structured and non-structured analysis operators are trained on $\boldsymbol{S}_{\text{train}}$ with varying weighting parameters, the resulting models are applied sample wise to the above mentioned sets. Figure 5.3 presents the average sparsity per sample achieved for all four signal sets. While the outer loop indicates the weightings κ of the condition number penalty $r(\cdot)$, the inner loop of ticks depicts the variation in the parameter γ that controls the influence of the incoherence penalty $h(\cdot)$. The performance of the initial random operator on the signal sets is illustrated rightmost.

First of all, it can be seen that independent of the choice of parameters, all the learned models generalize well, i.e., the average sparsity for the training set equals that of the test set. Furthermore, Figure 5.3a shows that without any additional penalty, zero sparsity is

[2]The Modified National Institute of Standards and Technology (MNIST) database contains small binary images of handwritten digits



(a) Generalization Non-Separable Operators

(b) Generalization Separable Operators

Figure 5.3.: Generalization behavior of separable and non-separable analysis operators that have been learned with varying penalty weightings. The average sparsity per sample is calculated with regard to four different signal sets.

achieved. This is rather intuitive since geometrically, subtracting the mean of the signals is equivalent to projecting the signals into the subspace orthogonal to $\mathbb{1}_n = [1, \ldots, 1]^\top$. Consequently, without any penalties, the learned operator is composed of repeated rows of the form $\pm\mathbb{1}_n / \sqrt{n}$ which provide the lowest sparsity possible for centered signals. Regarding the discrimination ability, especially the results obtained for the separable operator indicate that regardless of the weighting, all operators exhibit higher sparsity when they are applied to signals sampled from $\overline{\mathbb{P}}$. This observation further emphasizes the ability of the learning algorithm to capture the structure from the training samples and to provide a suitable prior for data sampled from the distribution $\mathbb{P}$.

However, since all weightings lead to operators that generalize well while being discriminative simultaneously, it remains unclear whether these results allow conclusions to be drawn about the actual performance of the model in inverse problem regularization. To answer this question, two simple inverse problems are considered. In the first experiment, the normalized signals $\boldsymbol{S}_{\text{test}}$ are corrupted with AWGN with $\sigma_{\text{noise}} = 0.1$. In the second experiment, undersampled measurements are generated via applying an undersampled DCT sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ with $m < n$ to the signals. The Noise-aware Greedy Analysis Pursuit (GAPn) algorithm as proposed in [100] is used to reconstruct the original signals in both the Denoising and Compressed Sensing setting. Comparable to Matching Pursuit, the GAPn algorithm iteratively projects the signal estimates onto the orthogonal complement of rows from the learned operator. The pseudocode is given in Algorithm 5.1

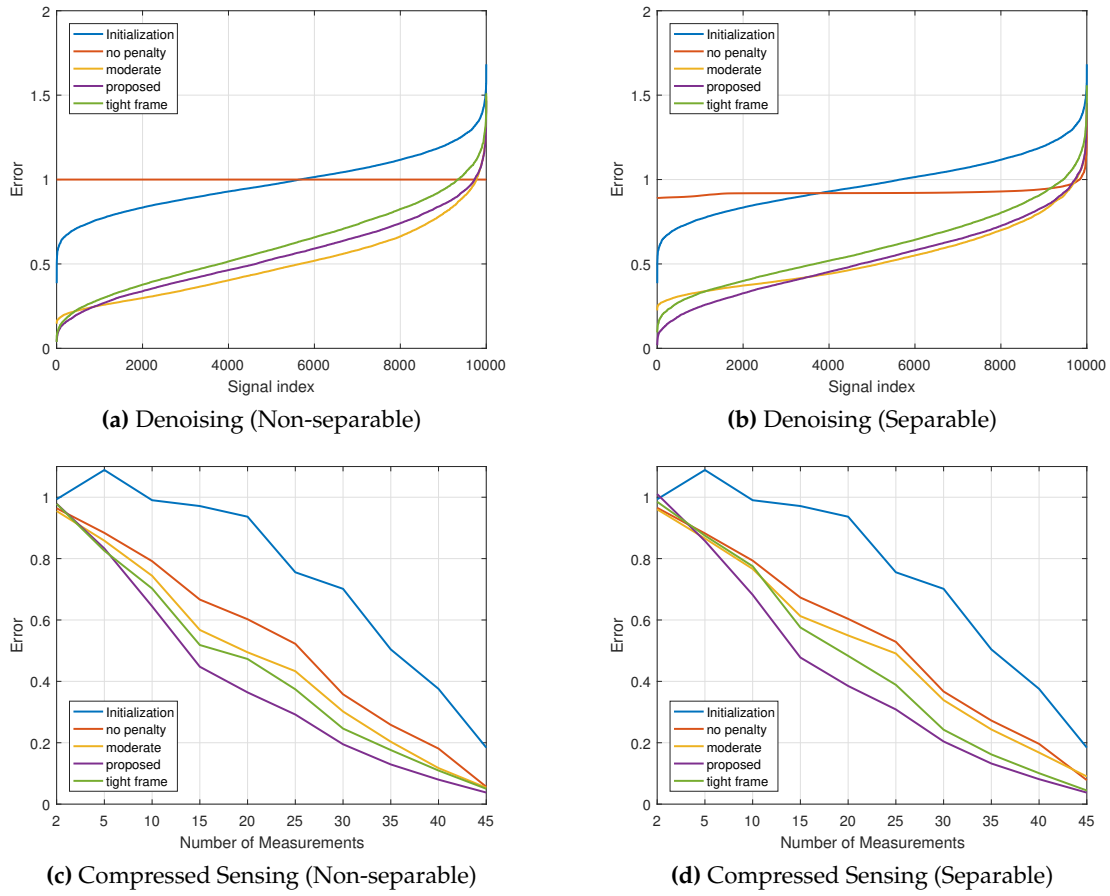
Algorithm 5.1 Noise-aware Greedy Analysis Pursuit (GAPn) as proposed in [100]

Require: $\boldsymbol{y}$, $\boldsymbol{\Phi}$, $\boldsymbol{\Omega}$, ϵ
Set: $t \leftarrow 0$, $\Lambda^{(0)} \leftarrow [1, \dots, K]$
  while Stopping criterion not reached **do**
    calculate $\tilde{\boldsymbol{s}}^{(t)} := \arg\min_{\boldsymbol{s}} \|\boldsymbol{\Omega}_{\Lambda^{(t)}} \boldsymbol{s}\|_2 \quad \text{s.t} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{s}\|_2 \leq \epsilon$
    update $\Lambda^{(t+1)} := \Lambda^{(t)} \backslash \{\arg\max\{|\boldsymbol{\omega}_j^\top \tilde{\boldsymbol{s}}^{(t)}| : j \in \Lambda^{(t)}\}\}$
    update $t \leftarrow t + 1$
  end while
Output: $\tilde{s}$

The recovery results for four different choices of the penalty weightings are summarized in Figure 5.4. These weightings are **a)** $\kappa = 0.0, \gamma = 0.0$ which corresponds to the *no penalty* setting; **b)** $\kappa = 1.0, \gamma_{\text{sep}} = 0.0002$ / $\gamma_{\text{non-sep}} = 0.002$ (*moderate* weights) that result in penalized separable/non-separable operators with a maximum absolute difference between the average sparsity on $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{S}_{\text{noisy}}$ (maximum according to (5.6)); **c)** $\kappa = 100.0, \gamma = 0.02$ that is in accordance to the *proposed* weights used in the image denoising experiment of Chapter 4; and lastly **d)** $\kappa = 10000.0, \gamma = 2.0$ which leads to operators with condition number close to one, i.e., the analysis operator closely resembles a *tight frame*. In all plots, the performance of the initial random operator is also given. Regarding the first experiment, given the reconstructed signal $\tilde{\boldsymbol{s}}_j \in \mathbb{R}^n$, the ground truth original signal $\boldsymbol{s}_j \in \mathbb{R}^n$ and the noise vector $\boldsymbol{e}_j \in \mathbb{R}^n$, the individual denoising error as shown in Figure 5.4a and 5.4b is measured via $\frac{\|\tilde{\boldsymbol{s}}_j - \boldsymbol{s}_j\|_2^2}{\|\boldsymbol{e}_j\|_2^2}$. That is, a value below one indicates an effective noise reduction. For the compressed sensing experiment in Figure 5.4c and 5.4d, the error is calculated via $\sum_j \|\tilde{\boldsymbol{s}}_j - \boldsymbol{s}_j\|_2^2$. Note that the dimension of the signal is $n = 49$ so the undersampling rate decreases along the x-axis.

Regarding the Denoising setting, the presented plot shows that the initial random operator coincidentally improves or deteriorates the signals. Furthermore, the operators that have been learned without additional penalties are not able to remove the noise from the signals since they are insensitive to all mean free signals, including zero mean Gaussian noise. Adding the penalties to the learning process clearly improves the performance while the *proposed* weightings achieve the best results with superior performance compared to the approximately *tight frame* weighting. The performance behavior of the different weightings are confirmed by the results observed for the second Compressed Sensing experiment.

To conclude, although all the learned models generalize well and almost all of them are sensitive to the presented signal class of natural images, i.e., in terms of average sparsity they perform worse on signals from $\overline{\mathbb{P}}$, it still remains difficult to establish a relation



(a) Denoising (Non-separable)

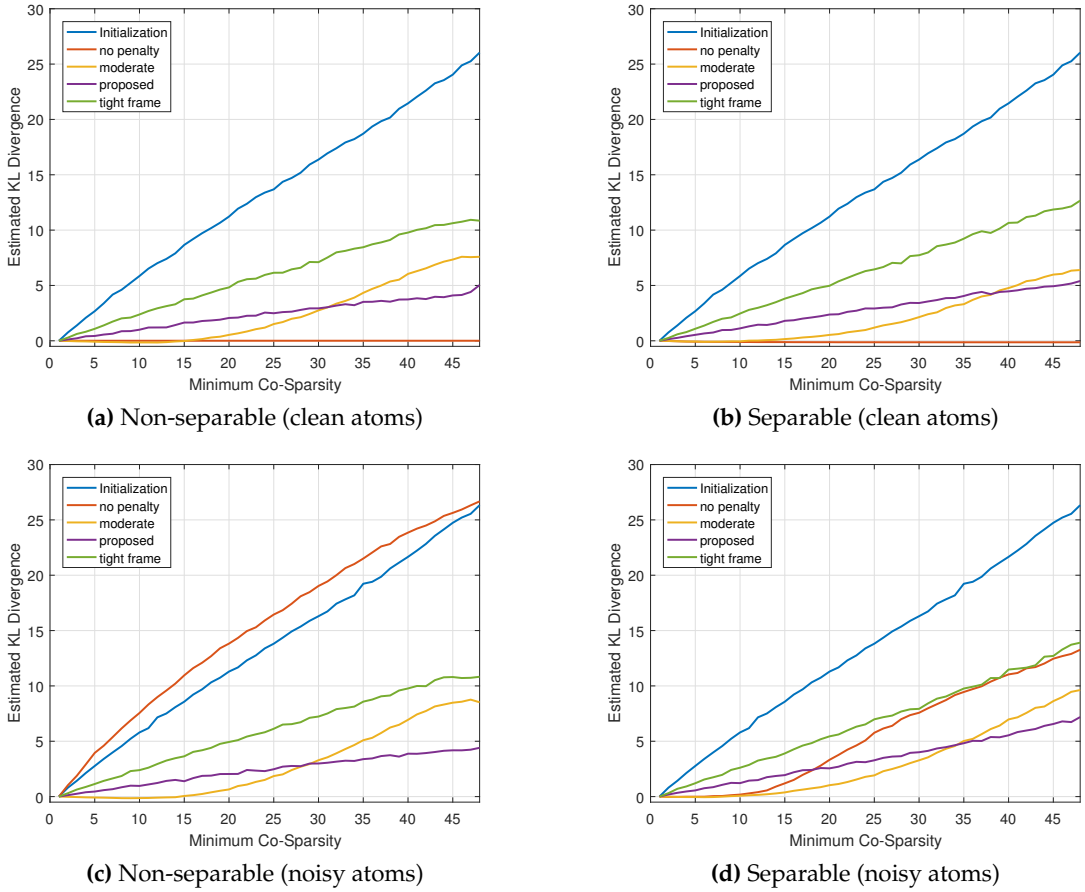
(b) Denoising (Separable)

(c) Compressed Sensing (Non-separable)

(d) Compressed Sensing (Separable)

Figure 5.4.: Denoising / Compressed Sensing Experiment

between the results from Figure 5.3 and Figure 5.4. First, lower average sparsity does not imply better reconstruction quality. While the average sparsity on the test set $\boldsymbol{S}_{\text{test}}$ increases with stronger penalty weightings, the recovery ability of the resulting operator improves as shown in Figure 5.4. Second, a large gap between the average sparsity evaluated on $\boldsymbol{S}_{\text{test}}$ and $\boldsymbol{S}_{\text{noisy}}$ does not necessarily result in a better reconstruction performance. While the difference in the average sparsity for the *moderate* setting is bigger than the gap observed for the *proposed* weighting, the overall performance of the prior obtained with the *proposed* weights is superior to the *moderate* one.



(a) Non-separable (clean atoms)

(b) Separable (clean atoms)

(c) Non-separable (noisy atoms)

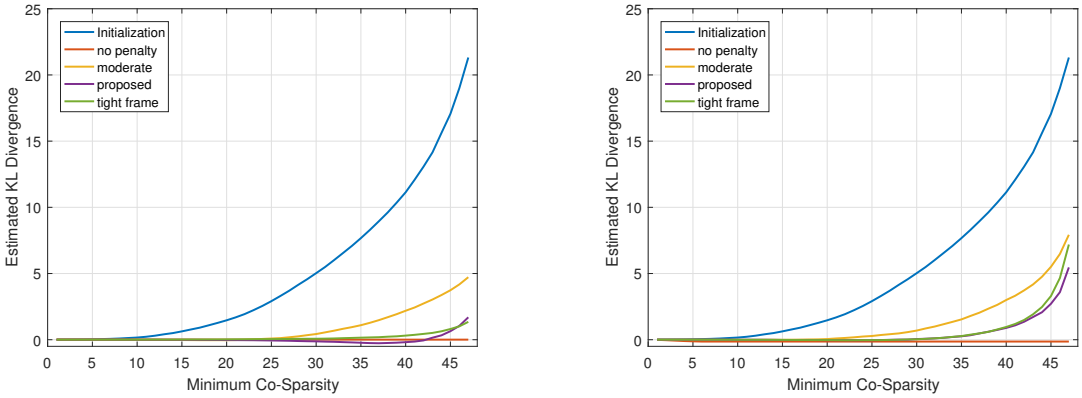
(d) Separable (noisy atoms)

Figure 5.5.: Estimated KL Divergence (random indices)

5.3.3. Estimated Divergence of Distributions

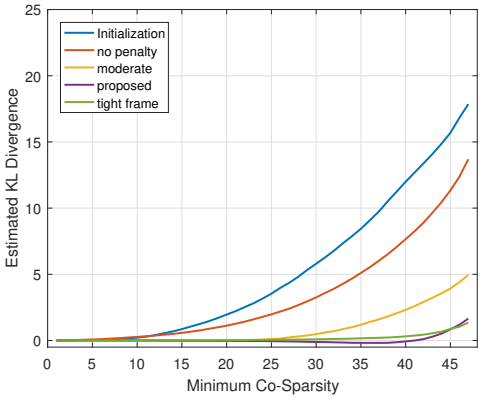
In order to establish a suitable measure that correlates with the performance of the learned model as a prior in inverse problems, the generalization ability of the learned operators is also assessed based on the divergence criterion introduced in Section 5.2. Following the same procedure introduced in Section 5.3.1, the signals from $\boldsymbol{S}_{\text{test}}$ are individually projected into different subspaces. After this procedure, the distribution of the projected signals $\boldsymbol{S}_{\text{test}}^{\text{proj}}$ is compared to the distribution of the training signals $\boldsymbol{S}_{\text{train}}$ in terms of the EKL divergence outlined in Eq. (5.7).

First of all, the estimated initial divergence between the training set $\boldsymbol{S}_{\text{train}}$ and the test set $\boldsymbol{S}_{\text{test}}$ with nearest neighbor $k = 1$ reads $D_{EKL} = 0.0052$, which indicates that the measure

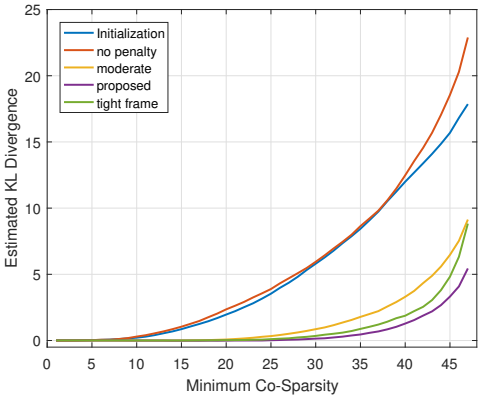


(a) Non-separable (clean atoms)

(b) Separable (clean atoms)



(c) Non-separable (noisy atoms)

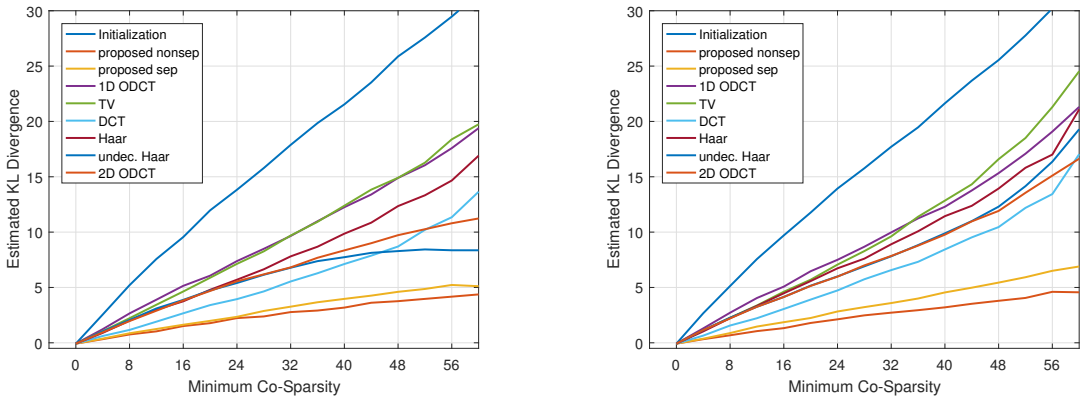


(d) Separable (noisy atoms)

Figure 5.6.: Estimated KL Divergence (Backward Greedy indices)

returns reasonable results, as we expect both signals sets to originate from the same distribution. Furthermore, the low divergence result also justifies the procedure of splitting the dataset into the training and 'held out' validation set as proposed in Eq. (5.5) to determine the generalization error.

In order to assess the divergence, two different scenarios are considered. First, the signals from $\boldsymbol{S}_{\text{test}}$ are projected onto the orthogonal complement of rows, *randomly* selected from $\boldsymbol{\Omega}$. In this case, the chosen indices are equally distributed. In this scenario, the estimated divergence reveals information about how good the entire set of filters describes the underlying distribution of the training signals. While a low divergence shows that the representational power is balanced across all the filters, a high value indicates that the model contains filters that rarely lead to a co-sparse response on the training data and thus



(a) Random Indices (clean atoms)

(b) Random Indices (noisy atoms)

Figure 5.7.: Estimated KL Divergence compared to analytically given transforms (random indices)

do not encode structural information with regard to the distribution. In order to avoid the assumption of equally distributed indices of the co-support, in the second scenario, the columns of $\boldsymbol{S}_{\text{test}}$ are projected such that the rank of $\boldsymbol{\Omega}_{\Lambda_j}$ exceeds a predefined target value while simultaneously the distance $\|\boldsymbol{s}_j - \boldsymbol{s}_j^{\text{proj}}\|_2$ is minimized. For this purpose, the Rank-BGP (Backward Greedy Pursuit) as introduced in [141] is used. Eventually, the set $\boldsymbol{S}_{\text{test}}^{\text{proj}}$ can be considered as the set of samples that strictly follow the Analysis Union-of-Subspace model while being closest to the original test set. A low divergence indicates that the filters sufficiently capture the structure of the training data without being biased by the weighting of the sparsity measure or the penalties. In other words, the penalties and objectives do not overrule the adaptation of the filters to the structure.

Figures 5.5 and 5.6 present the obtained results for the different penalty weightings introduced in the preceding section. The divergence is plotted against the number $|\Lambda|$ of rows that are selected to identify the orthogonal complement to the subspace the signals should reside in. Note that due to linear dependencies of the filters, only the minimum co-sparsity is indicated in the abscissa. For example the rank-1 operator composed of rows $\pm\mathbb{1}_n/\sqrt{n}$, i.e., the non-separable operator learned without any regularization penalties, exhibits maximal co-sparsity as long as the projected signals are elements of $\mathbb{S}^{n-1} \cap \mathbb{1}_n^{\perp}$. Since for this operator, the minimum co-sparsity criterion is fulfilled for different choices of $|\Lambda|$ anyway, the signals are left untouched without any projection into a union of lower dimensional subspaces. As a consequence, the estimated divergence does not increase which can be observed in the upper part of Figures 5.5 and 5.6. To avoid this phenomenon, a small amount of noise is added to the learned filters to reduce the effect of linear dependencies. The

lower part of Figures 5.5 and 5.6 depict the divergence results for this setting. Again, the signals are projected according to the two scenarios presented above. It can be clearly seen that this small perturbation severely deteriorates the performance of the rank-1 operator resulting in an estimated divergence that is even worse than the random initialization. Interestingly, the performance of the operators learned with the penalties is consistent under these perturbations which indicates the robustness achieved due to the penalties. Especially the results corresponding to the random sampling strategy emphasize the benefit of the presented strategy of incorporating penalty functions to relax the tight frame constraint. Furthermore, for both scenarios it can be observed that the operator learned with the *proposed* penalty weightings leads to the slightest increase of the divergence.

In order to show the general capability of a learned model to capture the structure of the training signals, the performance of the operators learned with the *proposed* weights are compared to analytically given transforms. More precisely, Figures 5.7 and 5.8 depict the estimated divergence of six common sparsity inducing transforms. Assuming a patch size of 8×8, these transforms are: (1) the orthogonal *DCT* transform , (2) the orthogonal *Haar* Wavelet transform , (3) and (4) the overcomplete DCT (*ODCT*) transform, either in its non-separable form (*1D ODCT*) or realized as a separable transform (*2D ODCT*) , (5) the operator that calculates the pairwise differences of pixels in the horizontal and vertical direction *TV* , and (6) the *undecimated Haar* Wavelet transform that is translation invariant. Again, both filter selection scenarios as well as the performance achieved with either clean or noise contaminated filters are considered. Throughout all experiments, the learned separable and non-separable operators perform best which further highlights the pursued approach of learning sparse data models.

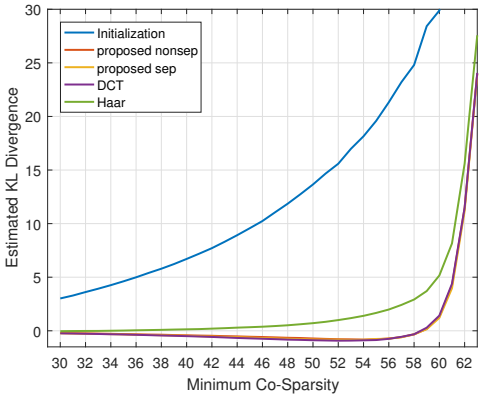
In sum, the presented divergence criterion reflects the generalization ability of the learned models in a great extent. As expected, learning the model from training data leads to a better adaptation to the distribution than analytically given transformation. Regarding the experiments with different penalty weightings, the generalization behavior of the operators is in accordance to the performance achieved in the Denoising and Compressed Sensing tasks from Section 5.3. Hence, the Estimated Kullback-Leibler Divergence can be considered a task independent measure to assess the suitability of the learned model to serve as a prior in inverse problems.

5.4. Summary

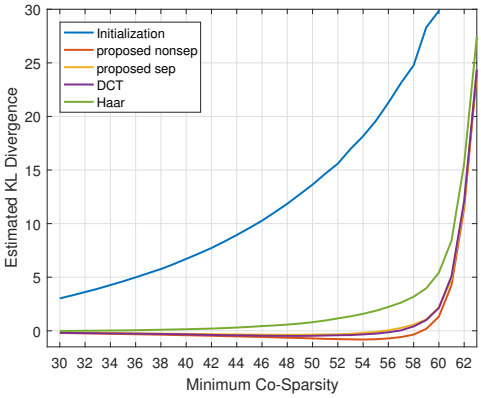
This chapter deals with the sample complexity of analysis operator learning and the generalization ability of a learned model. The theoretical sample complexity results are confirmed empirically, specifically the recovery of a ground truth operator from artificially

generated samples can be attained from less samples if a separability constraint is imposed on the operator.

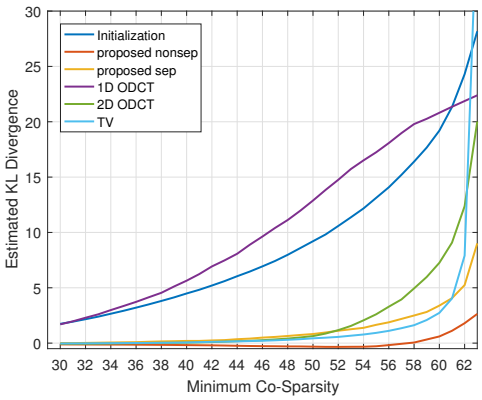
The assessment of the generalization solely based on a training and validation set does not indicate, which operator to chose for the subsequent task. Neither the overall sparsity achieved on the training set, nor the difference in the sparsity between signals from the true distribution and their noisy observations will lead to a precise criterion. As a consequence, the quality of the learned model to serve as a prior is usually determined in a task oriented way. The numerical experiments regarding the proposed divergence criterion however indicate that this measure correlates with the reliability of the model to regularize inverse problems in a very high extent. As such, it can be seen as a suitable task independent measure to at least distinguish useful models which will in turn reduce the necessary evaluation effort considerably.



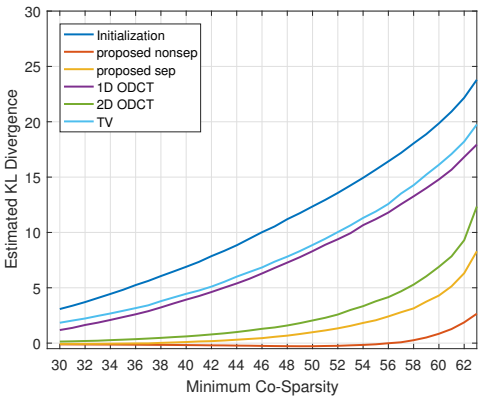
(a) Backward Greedy 8x8 (clean atoms)



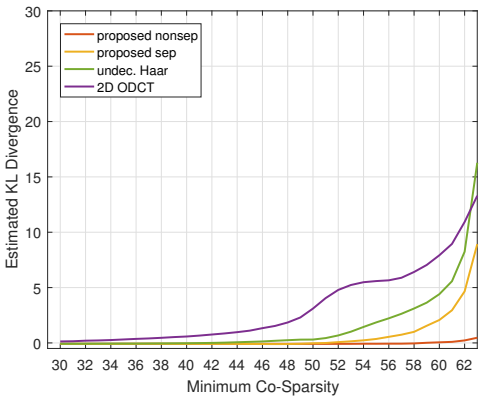
(b) Backward Greedy 8x8 (noisy atoms)



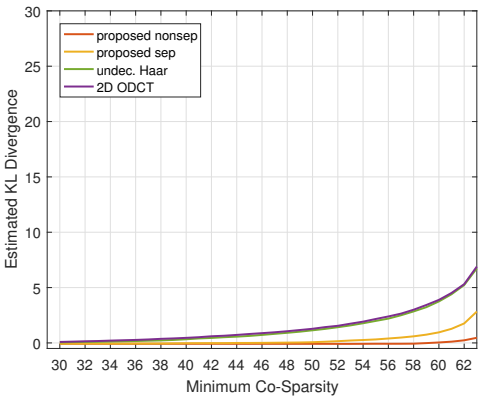
(c) Backward Greedy 11x8 (clean atoms)



(d) Backward Greedy 11x8 (noisy atoms)



(e) Backward Greedy 21x8 (clean atoms)



(f) Backward Greedy 21x8 (noisy atoms)

Figure 5.8.: Estimated KL Divergence compared to analytically given transforms (Backward Greedy indices)

Chapter 6.

Blind Analysis Operator Learning

The algorithm presented and analyzed in the previous chapters has been proven very useful to learn separable analysis operators from noise free samples. After the learning process, these operators are universally applicable to various problems in image processing, e.g. structural analysis or the regularization of inverse problems. This chapter extends the learning framework while focusing on the following contributions:

- To handle noise contaminated signals, an additional data fidelity term is added to the learning objective presented in (4.11). This formulation allows to compensate for various sampling noise models, e.g. Gaussian or impulsive noise, by simply exchanging the data fidelity term. Various experiments demonstrate that this simple strategy renders the proposed approach a very versatile method to cope with miscellaneous problems occurring in image processing.

- In contrast to the SGD optimization intended to find a suitable operator for the whole signal class of interest, in this chapter the operator will be adapted to the image at hand. Moreover, the image recovery and the model learning task is tackled simultaneously. A geometric CG method is used to solve the resulting blind reconstruction problem. Analogous to the SGD setting, this approach takes into account the product of spheres manifold structure.

- In many inverse problems, the provided measurements are a function of the whole image rather than local patches. Especially the reconstruction from undersampled measurements has attracted great attention in the sparse modeling literature. In order to handle these cases, the presented framework provides global support during image reconstruction. One of the major advantages of the analysis model consists in the fact that the co-sparse representation can be easily obtained via convolving the learned filters with the image - a strategy that furthermore strongly benefits from the separable structure of the filters. In this way, an image dependent regularization of the inverse problem is obtained. It is demonstrated numerically and visually that

the learned separable models successfully adapts to the image content which significantly improves the reconstruction performance.

6.1. Simultaneous Model Learning and Image Reconstruction

A widespread assumption in the sparse data model learning literature is the availability of clean training signals. In the image processing community, these signals are typically represented by image patches, which are extracted from diverse natural images. Due to the very high availability of images from this particular signal class, arbitrary amounts of training samples are easily accessible. Things change, however, if the acquisition of clean signals is expensive or even not possible. The recording of infrared images with high resolution for example is more expensive in comparison to the acquisition of images in the range of the visible light, due to the higher manufacturing costs for infrared sensors. In such a scenario, it is advantageous to adaptively learn the model based on the possibly corrupted or undersampled signals. Eventually, this approach results in a simultaneous model learning and image reconstruction framework.

Many of the adaptive learning approaches are straightforward extensions of the original model learning formulation. Especially in image denoising, the following simple and intuitive strategy is frequently pursued. First, the noisy image is decomposed into patches. After that, a sparse data model is adaptively learned based on the noise contaminated training samples. Afterwards, the individually reconstructed patches are placed back at their initial position in the image with an additional averaging step in the case overlapping patches are used. While this procedure usually leads to improved denoising results compared to the reconstruction with a fixed pre-learned model, this strategy is only applicable if the measurement process is patch-based itself rather than globally. In general, however, the measurement procedure is modeled by a system matrix that requires global image support. Formally, let $\boldsymbol{s} \in \mathbb{R}^N$ denote a vectorized image of size $N = wh$, with w being the width and h being the height of the image, respectively, obtained by stacking the columns of the image above each other. Given the system matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$, the acquisition process can be modeled via

$$\boldsymbol{y} = (\boldsymbol{\Phi}\boldsymbol{s}, \boldsymbol{e}). \tag{6.1}$$

This notation accounts for measurement noise $\boldsymbol{e} \in \mathbb{R}^M$ that can be additive, multiplicative or impulsive for example. Clearly, in the presence of AWGN (6.1) reduces to (1.2). In this work, the adaptive learning strategy is employed in various scenarios ranging from the classical denoising setting with $\boldsymbol{\Phi} = \boldsymbol{I}_N$ to Compressed Sensing problems where the number of measurements is smaller than the dimension of the signal, i.e., $M < N$.

6.1.1. Noise Dependent Data Term Formulation

Natural images are typically acquired with a standard digital Charge-Coupled Device (CCD) or Complementary Metal–Oxide–Semiconductor (CMOS) sensor that is sensitive to the visual light. The measurement process naturally involves degradations of the original image signal which are usually modeled via noise that follows a Gaussian distribution. Consequently, the data fidelity term that measures the fidelity of the recovered signal $\boldsymbol{s} \in \mathbb{R}^N$ to the measurements $\boldsymbol{y} \in \mathbb{R}^M$ can be modeled via the ℓ_2-norm as

$$d_{\text{additive}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M} \|\boldsymbol{y} - \boldsymbol{\Phi s}\|_2^2, \tag{6.2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ models the linear measurement process.

However, there exists a vast amount of various image modalities, e.g. medical or radar images, whose noise distributions do not follow this AWGN assumption. Thus, to cope with these differences, the data term has to take the respective noise distribution into account. In the following, other data fidelity terms that can be used interchangeably in the blind learning objective are introduced. Note that these fidelity measures are not necessarily functions of the residual of $\boldsymbol{\Phi s} - \boldsymbol{y}$.

To reconstruct signals in the presence of sparse outliers in the measurements a promising approach is to measure the data fidelity via

$$d_{\text{impulsive}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M} \sum_i \frac{1}{\log(1+c)} \log\left(1 + c \cdot (\boldsymbol{\Phi s} - \boldsymbol{y})_i^2\right), \tag{6.3}$$

where c is a positive constant. This function offers the advantage of being equally sensitive to both small and large deviations, which is highly desirable in the scenario of observations degraded by sparse outliers. As already pointed out in Section 4.2.1, the log-sparsity measure approximates the ideal ℓ_0-norm while still being continuously differentiable.

Mathematically, the degradation of the measurements of an image $\boldsymbol{s}$ with some multiplicative noise $\boldsymbol{e}_{\text{mult}} \in \mathbb{R}^M$ can be stated as $y_i = (\boldsymbol{\Phi s})_i \cdot e_i$. In the following experiments, the noise is assumed to be Gamma distributed, i.e., the i-th element in the noise vector $\boldsymbol{e}_{\text{mult}}$ is assumed to follow a Gamma distribution with probability density function [40, 8]

$$\text{pdf}(\boldsymbol{e}_i) = \frac{K^K}{\Gamma(K)} \boldsymbol{e}_i^{K-1} \exp^{(-K\boldsymbol{e}_i)}, \tag{6.4}$$

where $K \in \mathbb{N}^+$ and $\Gamma(K) = (K-1)!$. A smaller value of K indicates a higher noise level.

Accordingly, the data term

$$d_{\text{mult}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M} \sum_i \left(\log\left((\boldsymbol{\Phi s})_i \right) + \frac{y_i}{(\boldsymbol{\Phi s})_i} \right), \tag{6.5}$$

is utilized to reconstruct the clean signal $\boldsymbol{s}$, as it is done in [5] for example. Note that this fidelity term is restricted to inverse problems where $(\boldsymbol{\Phi s})_i$ is positive, as it is the case in problems like the classical denoising problem as well as Inpainting, Deblurring or Super-Resolution. In the case $\boldsymbol{\Phi} = \boldsymbol{I}_N$, i.e., if solely denoising is considered, one could directly optimize for $\boldsymbol{u}_i = \log(\boldsymbol{s}_i)$ [40, 8] which results in the data term

$$d_{\text{mult}}(\boldsymbol{u}, \boldsymbol{y}) = \frac{1}{M} \sum_i \left(\boldsymbol{u}_i + y_i \exp(-\boldsymbol{u}_i) \right). \tag{6.6}$$

This formulation offers the advantage of being strictly convex in all $\boldsymbol{u}_i$ [71]. Note that this approach amounts to optimizing over the log-image.

6.1.2. Algorithm Design

The goal of the blind learning and reconstruction approach is to find a separable analysis operator $\boldsymbol{\Omega} \in \mathbb{R}^{k \times n}$ simultaneously to the signal $\boldsymbol{s} \in \mathbb{R}^N$ that has to be recovered from the possibly compressed measurements. Note that the analysis operator has to be applied to local image patches rather than to the entire image at once. The proposed analysis model offers the advantage that in the case of fully overlapping patches, applying the operator to all of these patches is equal to simply convolve the learned filters with the current image estimate. The separable structure of the filters additionally encourages this strategy.

Eventually, the extension of the cost function (4.11) to the blind separable analysis operator learning scenario reads

$$\begin{aligned} \{\boldsymbol{\Omega}_1^\star, \dots, \boldsymbol{\Omega}_V^\star, \boldsymbol{s}^\star\} \in \arg \min_{\boldsymbol{\Omega}_i^\top \in \mathsf{OB}(n_i, k_i),\ \boldsymbol{s} \in \mathbb{R}^N} & \quad f(\boldsymbol{\Omega}_1, \dots, \boldsymbol{\Omega}_V, \boldsymbol{s}) \\ \text{with} \quad f(\boldsymbol{\Omega}_1, \dots, \boldsymbol{\Omega}_V, \boldsymbol{s}) = \tfrac{1}{N}\, & g(\boldsymbol{\Omega}(\boldsymbol{s})) + \lambda d(\boldsymbol{s}, \boldsymbol{y}) + \kappa \sum_{i=1}^{V} r(\boldsymbol{\Omega}_i) + \gamma \sum_{i=1}^{V} h(\boldsymbol{\Omega}_i), \end{aligned} \tag{6.7}$$

with the additional noise dependent data term $d(\boldsymbol{s}, \boldsymbol{y})$ being weighted against the sparsity measure via the parameter λ. The notation $\boldsymbol{\Omega}(\boldsymbol{s})$ accounts for the operation of convolving the whole image with every filter present in $\boldsymbol{\Omega}$. The reader is referred to the Appendix A for a derivation of the respective gradients.

6.2. Conjugate Gradient Optimization

In contrast to the SGD optimization framework, that has its strength in the online learning scenario where the samples from some particular signal class may be acquired successively, the blind learning and reconstruction approach focuses on a setting with a concrete and fixed set of samples. These samples usually fully cover the image that has to be inferred from the measurements. For that reason, the optimization problem is tackled via a CG approach that at each iteration takes the full set of samples into account. The CG approach is scalable and converges fast in practice. It is thus well-suited to handle the high dimensional problem of simultaneous image reconstruction and operator learning. A simultaneous update is achieved via employing the product manifold structure of $\mathsf{OB}(n_1, k_1) \times \cdots \times \mathsf{OB}(n_V, k_V) \times \mathbb{R}^N$ considered as a Riemannian submanifold of $\mathbb{R}^{n_1 \times k_1} \times \cdots \times \mathbb{R}^{n_V \times k_V} \times \mathbb{R}^N$. To enhance legibility, in the remainder of this section the oblique manifold is simply denoted by OB.

Recall from Section 2.2.1 that the Riemannian gradient at $\boldsymbol{\Omega}_i^\top$ is given by the orthogonal projection of the Euclidean gradient onto the tangent space $\mathtt{T}_{\boldsymbol{\Omega}_i^\top}\mathsf{OB}$. Using the product structure and denoting the partial derivatives of $f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})$ by $\nabla_{\boldsymbol{\Omega}_i^\top} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})$ and $\nabla_{\boldsymbol{s}} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})$, respectively, the Riemannian gradient of the cost function is denoted as

$$\begin{aligned}\mathtt{G}(\boldsymbol{\Omega}_1^\top, \ldots, \boldsymbol{\Omega}_V^\top, \boldsymbol{s}) = \Big(&\Pi_{\mathtt{T}_{\boldsymbol{\Omega}_1^\top}\mathsf{OB}}(\nabla_{\boldsymbol{\Omega}_1^\top} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})), \ldots, \\ &\Pi_{\mathtt{T}_{\boldsymbol{\Omega}_V^\top}\mathsf{OB}}(\nabla_{\boldsymbol{\Omega}_V^\top} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})), \nabla_{\boldsymbol{s}} f(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s})\Big). \end{aligned} \tag{6.8}$$

Accordingly, with the notation of the search directions $(\boldsymbol{H}_1^{(t)}, \ldots, \boldsymbol{H}_V^{(t)}, \boldsymbol{h}^{(t)}) \in \mathtt{T}_{\boldsymbol{\Omega}_1^\top}\mathsf{OB} \times \ldots \times \mathtt{T}_{\boldsymbol{\Omega}_V^\top}\mathsf{OB} \times \mathbb{R}^N$ at hand, the new iterates regarding the product manifold are determined via

$$(\boldsymbol{\Omega}_1^{\top(t+1)}, \ldots, \boldsymbol{\Omega}_V^{\top(t+1)}, \boldsymbol{s}^{(t+1)}) = \left(\Gamma(\boldsymbol{\Omega}_1^{\top(t)}, \boldsymbol{H}_1^{(t)}, \alpha^{(t)}), \ldots, \Gamma(\boldsymbol{\Omega}_V^{\top(t)}, \boldsymbol{H}_V^{(t)}, \alpha^{(t)}), \boldsymbol{s}^{(t)} + \alpha^{(t)}\boldsymbol{h}^{(t)}\right), \tag{6.9}$$

where $\alpha^{(t)}$ denotes the step size that leads to a sufficient decrease of the cost function and $\Gamma(\boldsymbol{\Omega}_j^{\top(t)}, \boldsymbol{H}_j^{(t)}, \alpha^{(t)})$ represents the geodesic emanating from $\boldsymbol{\Omega}_j^{\top(t)}$ along the direction $\boldsymbol{H}_j^{(t)}$.

In order to calculate the new search direction for the next iterate, the parallel transport involving the geodesics in the product manifold is denoted as

$$\mathtt{P}^{(t)}_{\boldsymbol{H}_1^{(t-1)}, \ldots, \boldsymbol{H}_V^{(t-1)}, \boldsymbol{h}^{(t-1)}} = \left(\mathtt{P}^{(t)}_{\boldsymbol{H}_1^{(t-1)}}, \ldots, \mathtt{P}^{(t)}_{\boldsymbol{H}_V^{(t-1)}}, \boldsymbol{h}^{(t-1)}\right). \tag{6.10}$$

In this work, a hybridization of the Hestenes-Stiefel (HS) and the Dai Yuan (DY) formula as motivated in [33] is used to determine the update of the search direction. Let $\mathtt{G}_j^{(t)} := \mathtt{G}(\boldsymbol{\Omega}_j^{\top(t)})$ and $\boldsymbol{g}^{(t)} := \mathtt{G}(\boldsymbol{s}^{(t)})$, as well as $\mathtt{U}_j^{(t)} = \mathtt{G}_j^{(t)} - \mathtt{P}^{(t)}_{\mathtt{G}_j^{(t-1)}}$ and $\boldsymbol{u}^{(t)} = \boldsymbol{g}^{(t)} - \boldsymbol{g}^{(t-1)}$, the CG-update parameters in the product of manifolds structure read

$$\beta_{\mathrm{HS}}^{(t)} = \frac{\langle \mathtt{G}_1^{(t)}, \mathtt{U}_1^{(t)} \rangle + \cdots + \langle \mathtt{G}_V^{(t)}, \mathtt{U}_V^{(t)} \rangle + \langle \boldsymbol{g}^{(t)}, \boldsymbol{u}^{(t)} \rangle}{\langle \mathtt{P}^{(t)}_{\boldsymbol{H}_1^{(t-1)}}, \mathtt{U}_1^{(t)} \rangle + \cdots + \langle \mathtt{P}^{(t)}_{\boldsymbol{H}_V^{(t-1)}}, \mathtt{U}_V^{(t)} \rangle + \langle \boldsymbol{h}^{(t-1)}, \boldsymbol{u}^{(t)} \rangle}, \tag{6.11}$$

$$\beta_{\mathrm{DY}}^{(t)} = \frac{\langle \mathtt{G}_1^{(t)}, \mathtt{G}_1^{(t)} \rangle + \cdots + \langle \mathtt{G}_V^{(t)}, \mathtt{G}_V^{(t)} \rangle + \langle \boldsymbol{g}^{(t)}, \boldsymbol{g}^{(t)} \rangle}{\langle \mathtt{P}^{(t)}_{\boldsymbol{H}_1^{(t-1)}}, \mathtt{U}_1^{(t)} \rangle + \cdots + \langle \mathtt{P}^{(t)}_{\boldsymbol{H}_V^{(t-1)}}, \mathtt{U}_V^{(t)} \rangle + \langle \boldsymbol{h}^{(t-1)}, \boldsymbol{u}^{(t)} \rangle}, \tag{6.12}$$

where $\langle \cdot, \cdot \rangle$ denotes the Riemannian metric. With the hybrid update formula

$$\beta_{\mathrm{hyb}}^{(t)} = \max\left(0, \min(\beta_{\mathrm{DY}}^{(t)}, \beta_{\mathrm{HS}}^{(t)})\right), \tag{6.13}$$

the new search directions are given by

$$(\boldsymbol{H}_1^{(t)}, \ldots, \boldsymbol{H}_V^{(t)}, \boldsymbol{h}^{(t)}) = \left(-\mathtt{G}(\boldsymbol{\Omega}_1^{\top(t)}, \ldots, \boldsymbol{\Omega}_V^{\top(t)}, \boldsymbol{s}^{(t)}) + \beta_{\mathrm{hyb}}^{(t)} \mathtt{P}^{(t)}_{\boldsymbol{H}_1^{(t-1)}, \ldots, \boldsymbol{H}_V^{(t-1)}, \boldsymbol{h}^{(t-1)}}\right). \tag{6.14}$$

Finally, the new iterate for the separable operator as well as the image signal can be calculated via Eq. (6.9). The complete pseudocode for the blind separable analysis operator learning algorithm with simultaneous image reconstruction is given in Algorithm 6.1. Similar to the SGD implementation, a suitable step size is determined via backtracking.

6.3. Numerical Experiments

In this section, numerical results of various blind image reconstruction experiments are provided. After defining the stopping criterion, an empirical convergence analysis is presented, where the achieved reconstruction performance from different random initializations emphasizes the robustness of the blind learning framework. The presented approach is further motivated by the ability of the algorithm to cope with different noise models. Image reconstruction results where the observations are corrupted with noise that follows various distributions are shown subsequently. These experiments highlight the versatility of the algorithm, whose general optimization procedure is not limited to selected scenarios like the AWGN assumption or additional constraints on the system matrix. Finally, the

Algorithm 6.1 CG Backtracking Line Search

Require: $\alpha^{(0)} > 0$, $0 < c_1 < 1$, $0 < c_2 < 1$, $t_{ls}^{max} = 200$, $\boldsymbol{\Omega}_i^{(0)}$ $i = 1, \ldots, V$, $\boldsymbol{s}^{(0)}$
Set: $\alpha \leftarrow \alpha^0$, $t \leftarrow 1$
  while Stopping criterion not reached **do**
    calculate $\mathtt{G}(\boldsymbol{\Omega}_1^{\top(t)}, \ldots, \boldsymbol{\Omega}_V^{\top(t)}, \boldsymbol{s}^{(t)})$
    if $t = 1$ **then**
      $\boldsymbol{H}_i^{(t)} = -\mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)})$, for all $i = 1, \ldots, V$
      $\boldsymbol{h}^{(t)} = -\mathtt{G}(\boldsymbol{s}^{(t)})$
    else
      calculate search direction according to (6.14)
    end if
    set $\alpha_{ls} \leftarrow \alpha^{(t)}$, $t_{ls} \leftarrow 1$
    while $f(\Gamma(\boldsymbol{\Omega}_1^{\top(t)}, \boldsymbol{H}_1^{(t)}, \alpha_{ls}^{(t)})^\top, \ldots, \Gamma(\boldsymbol{\Omega}_V^{\top(t)}, \boldsymbol{H}_V^{(t)}, \alpha_{ls}^{(t)})^\top, \boldsymbol{s}^{(t)} + \alpha_{ls}^{(t)} \boldsymbol{h}^{(t)}) >$
      $f(\boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_V^{(t)}, \boldsymbol{s}^{(t)}) + c_1 \alpha_{ls}^{(t)} \sum_{i=1}^{V} \langle \mathtt{G}(\boldsymbol{\Omega}_i^{\top(t)}), \boldsymbol{H}_i^{(t)} \rangle \wedge$
      $t_{ls} < t_{ls}^{max}$ **do**
      $\alpha_{ls} \leftarrow \alpha_{ls} \cdot c_2$
      $t_{ls} \leftarrow t_{ls} + 1$
    end while
    update $\boldsymbol{\Omega}_i^{\top(t+1)} = \Gamma(\boldsymbol{\Omega}_i^{\top(t)}, \boldsymbol{H}_i^{(t)}, \alpha_{ls})$
    update $\boldsymbol{s}^{(t+1)} = \boldsymbol{s}^{(t)} + \alpha_{ls} \boldsymbol{h}^{(t)}$
    $\alpha^{(t+1)} \leftarrow \alpha_{ls} \cdot c_2^{-2}$
    $t \leftarrow t + 1$
  end while
Output: $\boldsymbol{\Omega}_i^\star$ $i = 1, \ldots, V$ and $\boldsymbol{s}^\star$

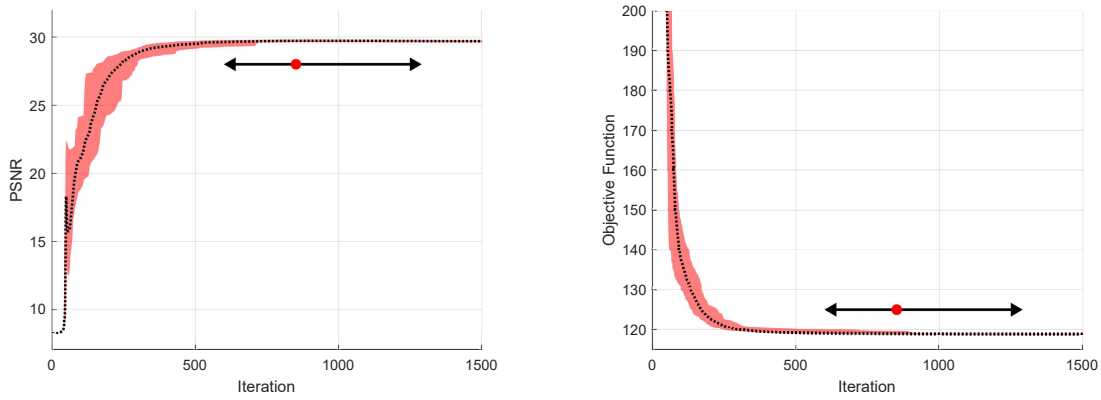
recovery performance in a Compressed Sensing problem where the image data is three-dimensional is investigated.

Stopping Criterion

Since in the following experiments we are also interested in the reconstructed image, the algorithm terminates when the image update saturates. To be precise, at each iteration the norm of the image gradient $\mathtt{G}(\boldsymbol{s}^{(t)})$ is evaluated. The image recovery stops when the Euclidean norm of the current gradient is below some threshold δ_{adaptive}, i.e., when the condition

$$\tfrac{1}{N} \|\mathtt{G}(\boldsymbol{s}^{(t)})\|_2 < \delta_{\text{adaptive}} \tag{6.15}$$

is fulfilled for $\delta_{\text{adaptive}} \in [10^{-7}, 10^{-6}]$.



(a) Convergence of PSNR

(b) Convergence of objective function value

Figure 6.1.: Empirical Convergence based on a Denoising experiment.

6.3.1. Empirical Convergence Analysis

To assess the empirical convergence of the blind learning and reconstruction approach, a denoising experiment is conducted. For this purpose, the *Cameraman* image (second image in Figure 4.4) has been corrupted with different realizations of AWGN noise with $\sigma_{\text{noise}} = 20$. Analogously, the separable analysis operators are initialized as random matrices where the entries follow a normal distribution. Different realizations of samples from the uniform distribution constitute the initial images. The same experiment with different initializations is repeated 20 times. To evaluate the convergence, the algorithm runs a predefined number of iterations. For each run, the iteration count is recorded where the stopping criterion with the threshold $\delta_{\text{adaptive}} = 10^{-6}$ is fulfilled. The dotted lines in Figure 6.1 depict the progress of the PSNR (Fig. 6.1a) and the overall function value (Fig. 6.1b) averaged over all the 20 trials. In both plots, the shaded area indicates the difference between the minimum and maximum value at each iteration. The average number of iterations required to reach the stopping criterion is illustrated by the red dot, while the interval boundaries correspond to the minimum and maximum number of iterations across all trials.

Figure 6.1 clearly shows that the proposed blind separable learning algorithm converges to almost the same solution from various random initializations. The average PSNR across all trials at the iteration where the stopping criterion is met reads 29.72 dB with a standard deviation of just 0.05 dB.

Figure 6.2.: Test Images (cropped to 256×256). From left to right: Piecewise-Constant (PWC), Barbara, Boats, Lena, Peppers.

6.3.2. 2D Blind Learning

In this section, two dimensional image signals are considered. The blind reconstruction involves the recovery of the original image signal from noise contaminated or undersampled observations while simultaneously learning a separable analysis operator. The images for testing are illustrated in Figure 6.2 which are all of size 256×256. For all the experiments, the accuracy of the recovery is measured in terms of the PSNR (4.22) and the MSSIM (4.24). Since the learned filters are only applied to valid pixel positions (there is no artificially generated border added to the images during filtering), the quality measures are also only evaluated at pixel positions that are equally involved in the sparsity regularizer. In all the blind reconstruction experiments, the separable analysis operators are initialized with entries randomly drawn from a normal distribution. Since in most of the experiments the images are normalized to the range $[0, 1]$, they are initialized as matrices with entries from a uniform distribution.

Additive White Gaussian Noise

The first image recovery experiment deals with observations that are corrupted with AWGN. For this reason, the data fidelity term in the blind learning framework (6.7) is set to the ℓ_2-norm which is given in (6.2). The original image is normalized to the range $[0, 1]$, thus the standard deviation of the AWGN reads $\sigma_n = 0.0784$ which corresponds to a deviation of $\sigma_n = 20$ in the common pixel range of $[0, 255]$. For the penalties $r(\cdot)$ and $h(\cdot)$ the same weighting parameters $\kappa = 35.0$ and $\gamma = 0.5$ as in the non-blind setting are used to learn a separable operator $\boldsymbol{\Omega} = \iota(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \in \mathbb{R}^{100 \times 49}$. The operator is applied to all overlapping patches via filtering while the sparsity is measured via the log sparsity measure introduced in (4.2) with the parameter $\nu = 2000$. The weighting of the data term is set to $\lambda = 1800$. The blind reconstruction algorithm terminates when the stopping criterion falls below the threshold $\delta_{\text{adaptive}} = 10^{-6}$.

Table 6.1.: Adaptive Denoising experiment for five different test images (256×256) corrupted by AWGN with $\sigma_n = 20$. Achieved PSNR in decibels (dB) and MSSIM.

σ_n / PSNR	Algorithm	PWC	Barbara	Boats	Lena	Peppers
20 / 22.11	adaptive	**30.02 0.895**	**28.55 0.861**	**29.58 0.842**	**29.47 0.829**	**30.46 0.861**
	fixed	27.60 0.796	27.79 0.824	29.28 0.824	28.94 0.788	29.80 0.812

For comparison, the PSNR and MSSIM achieved by running the same algorithm, however, with a fixed operator is given in Table 6.1. The used operator is the same as in the denoising experiment stated in Section 4.5 and that was trained on patches from the images given in Figure 4.3.

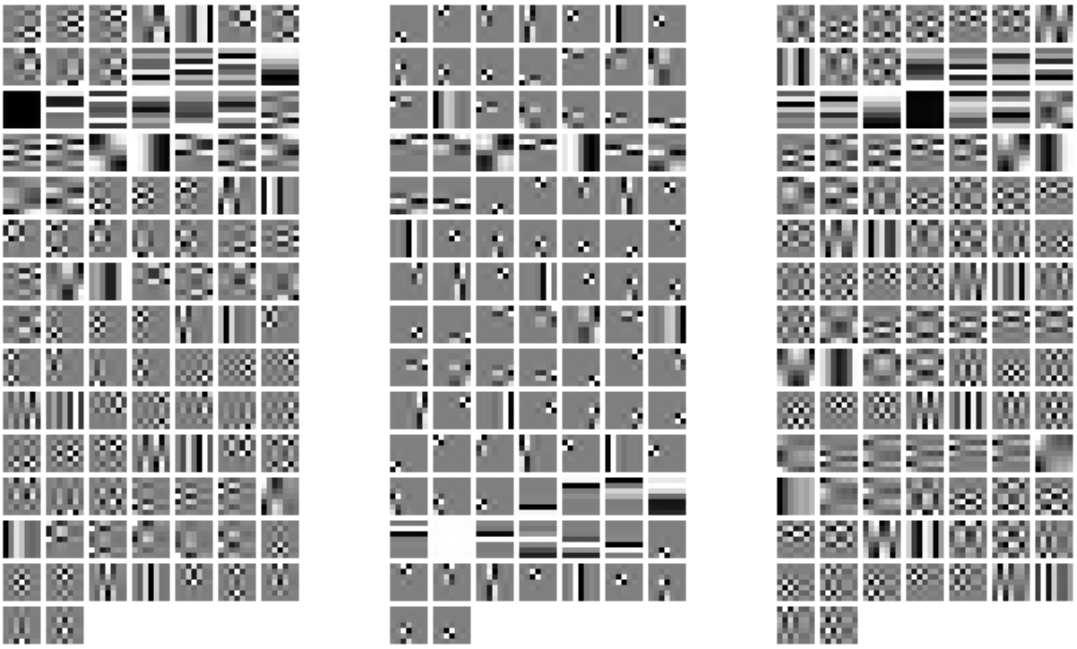
From the achieved results it is evident that the blind reconstruction framework is able to provide higher recovery accuracy than compared to a globally learned operator. This is due to the ability of the regularizer to adapt the image content at hand. Especially for the artificially generated *Piecewise-Constant* (PWC) image, the recovery quality is significantly higher. This can be attributed to the circumstance that the underlying structure of such an image is not adequately captured by the fixed operator since these kind of images are underrepresented in the training set. The adaptability can be also visually seen in Figure 6.3 where the filters of the fixed operator are compared to the adaptively learned ones. Each of the small patches represents a 2D filter kernel. As expected, the filters adaptively learned on the *PWC* image resemble the filters from the Total Difference Operator[1]. Similarly, the high frequency parts from the *Barbara* image are represented by the operator shown on the right panel in Figure 6.3.

Impulsive Noise

If the noise in the measurements is no longer Gaussian distributed, the ℓ_2-error term introduced in (6.2) might be less effective. Erroneous pixels of an image sensor or missing pixel information can be modeled as impulsive noise, where the images are severely degraded by sporadic large amplitude samples. Since these samples typically have values from the lower or upper limit of the image intensity range, e.g. 0 or 1 in the normalized case, impulsive noise is also called salt-and-pepper noise. The left panel of Figure 6.4 shows the image *Boats* with 20% of its pixels corrupted by impulsive noise.

To cope with this type of noise, the data fidelity term in (6.7) is set to the function given in (6.3). The parameter that controls the slope of the function is set to $c = 500$. Identical

[1]The Total Difference Operator calculates the differences between adjacent pixels in horizontal and vertical direction. Utilizing the Total Difference Operator in the sparsity prior is closely related to minimizing the Total Variation of the image signal.



(a) Operator learned from training images.

(b) Operator adaptively learned while reconstructing the *Piecewise-Constant* (PWC) image.

(c) Operator adaptively learned while reconstructing the *Barbara* image.

Figure 6.3.: Adaptively learned separable analysis operators from images corrupted with AWGN.

to the previous experiment, the parameters for the regularizers are again set to $\kappa = 35.0$ and $\gamma = 0.5$. The parameter in the sparsity measure reads $\nu = 2000$, while the pixel intensities are again normalized to the range $[0, 1]$. To evaluate the performance of the blind recovery approach in the presence of impulsive noise, a certain number of pixels in the test images is artificially set to the extreme values 0 and 1. The ratio of corrupted pixels reads $P_{\text{corrupted}} = 0.2$. The weighting λ of the fidelity term is set to $\lambda = 80.0$. The reconstruction terminates with the threshold set to $\delta_{\text{adaptive}} = 10^{-6}$.

Table 6.2 lists the PSNR and MSSIM of the blind separable framework compared to the accuracy achieved after median filtering with different filter sizes. The Median filter is a nonlinear filtering operation that replaces the current pixel value with the median value of its neighboring entries. In the first column, the average PSNR over all input images is given, which indicates the severe degradation with only a small amount of corrupted

Table 6.2.: Adaptive Denoising experiment for five different test images (256 × 256) corrupted by impulsive noise. Achieved PSNR in decibels (dB) and MSSIM.

σ_n / PSNR	Algorithm	PWC	Barbara	Boats	Lena	Peppers
20 % / 11.88	adaptive	20.78 0.789	**27.38 0.850**	**28.48 0.860**	**29.84 0.875**	**28.55 0.897**
	MED 7x7	21.44 0.824	21.79 0.601	23.41 0.657	26.06 0.746	26.20 0.824
	MED 5x5	22.75 0.865	21.10 0.602	25.05 0.739	27.79 0.818	27.84 0.864
	MED 3x3	**23.74 0.879**	23.93 0.793	26.22 0.813	28.23 0.863	27.73 0.867

pixels. Figure 6.4 shows the reconstruction of the *Boats* image, which has been corrupted by 20% impulsive noise. It can be clearly seen, that the proposed approach recovers visually pleasant results while the median filtered image suffers from the typical blocking artifacts.

While the blind recovery algorithm is able to accurately recover all of the natural images with superior performance compared to the median filtering, it performs less accurate on the synthetic *Piecewise-Constant* image. This can be attributed to the fact that separable operators primarily exhibit filters, which are more sensitive to vertical and horizontal edges. Since the *Piecewise-Constant* image is composed of rotated homogeneous rectangles, most of the edges are aligned diagonally. During the blind reconstruction, the regularizer prefers horizontal and vertical edges while the data term treats the small deviations from the ideal diagonal edges as sparse outliers. Figure 6.5 shows the disturbed input as well as the reconstruction with the proposed approach and the median filtered image. Note that while the blind approach results in fringed edges the median filter operation severely smoothes the corners of the rectangles.

Multiplicative Noise

The phenomenon of measurement noise that is multiplicative, rather than additive, can be observed in imaging modalities like synthetic aperture radar (SAR), ultrasound, sonar, and laser imaging [143, 8]. In this context it is often referred to as speckle noise. The multiplicative and non-Gaussian nature of the noise on the one hand severely degrades the original image, while on the other hand renders standard denoising frameworks, which rely on an AWGN assumptions, ineffective. Hence, specialized algorithms are needed to cope with this type of Gamma distributed noise.

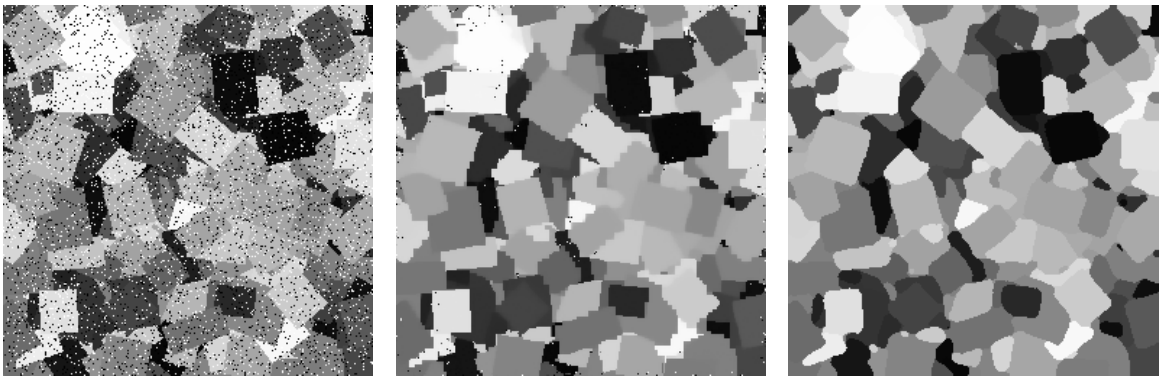
Analogous to the impulsive noise setting, the proposed blind reconstruction framework is able to handle multiplicative noise by simply exchanging the data fidelity term. For this purpose, the data term given in Eq. (6.6) is used to reconstruct the original signal. In this setting, the log-image is recovered, thus during reconstruction the analysis operator is simultaneously learned on $\log(\boldsymbol{s})$, where the natural logarithm is applied elementwise.

(a) Noisy input image.

(b) Reconstructed image with adaptively learned separable analysis operator.

(c) Median filtered image with same filter size (7×7).

Figure 6.4.: Reconstruction of the *Boats* image, which has been corrupted by 20% impulsive noise.



(a) Noisy input image.

(b) Reconstructed image with adaptively learned separable analysis operator.

(c) Median filtered image with same filter size (7×7).

Figure 6.5.: Reconstruction of the *Piecewise-Constant* image, which has been corrupted by 20% impulsive noise.

To account for the modified dynamic range of the image, the parameters for the penalties are set to $\kappa = 10.0$ and $\gamma = 0.1$, while the parameter that controls the slope of the sparsity measure reads $\nu = 100$. The test images have been corrupted with multiplicative noise, i.e., $\boldsymbol{y} = \boldsymbol{s} \odot \boldsymbol{e}_{\text{mult}}$, where $\boldsymbol{e}_{\text{mult}}$ follows the Gamma distribution introduced in (6.4) with $K = 10$. The threshold in the stopping criterion is set to $\delta_{\text{adaptive}} = 10^{-6.2}$.

Table 6.3 summarizes the reconstruction quality results of the proposed approach in

Table 6.3.: Adaptive Denoising experiment for five different test images (256 × 256) corrupted by multiplicative noise. Achieved PSNR in decibels (dB) and MSSIM .

K / PSNR	Algorithm	PWC	Barbara	Boats	Lena	Peppers
10 / 14.96	adaptive	24.25 0.749	**24.42 0.719**	25.32 0.736	24.82 0.666	26.26 0.757
	AASCO	25.09 0.817	23.16 0.658	**25.58 0.737**	**26.15 0.733**	**26.80 0.791**
	MIDAL	**25.78 0.833**	22.88 0.637	25.20 0.719	26.04 0.721	26.70 0.781

terms of PSNR and MSSIM. To assess the performance, the adaptive learning framework is compared to the *MIDAL* (Multiplicative Image Denoising by Augmented Lagrangian) method introduced in [8] and the *AASCO* algorithm presented in [40]. The objective in the *MIDAL* algorithm is composed of the same convex data term (6.6) with an additional standard isotropic discrete TV-norm regularizer. The authors propose variable splitting and the application of the ADMM (Alternating Direction Method of Multipliers) method to solve the optimization problem. Besides the TV regularizer, Dong et al. [40] add another analysis operator based regularizer to the objective that is learned adaptively on the image that has to be reconstructed. Again, variable splitting and the ADMM algorithm is used to tackle the constrained optimization problem, while the subproblem of analysis operator learning is based on the Analysis SimCO algorithm presented by partially the same authors in [39]. Opposed to *MIDAL* and the proposed method, the *AASCO* (Adaptive Analysis SimCO) algorithm is purely patch based and thus cannot directly handle any inverse problem formulation with $\boldsymbol{\Phi} \neq \boldsymbol{I}_N$.

The reported performance results indicate that the approaches that utilize a TV-norm regularizer are already well suited to handle the severe degradation due to the multiplicative noise characteristics. The proposed blind framework shows its strengths in the case of a more structured image content. Figure 6.6 depicts the reconstruction of the *Barbara* image. It can be observed that the high frequency parts of both the scarf and the chair in the background are better restored by using the presented blind recovery algorithm.

Inpainting

If the indices of the disturbed pixels in the impulsive noise setting are known, the problem can also be modeled as an Inpainting problem. In this setting, the number M of measurements is significantly smaller than the size of the original signal. The sampling process can be stated as $\boldsymbol{y} = \boldsymbol{\Phi s}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ is a matrix composed of an undersampled set of canonical basis vectors for $\mathbb{R}^N$ in its rows. A typical application of this type of inverse problem is to remove text or scratches from images.

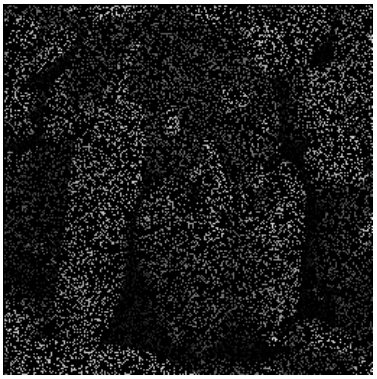
Since in this experiment no sampling noise is assumed, the blind reconstruction objective

(a) Noisy input image.

(b) Reconstructed image with adaptively learned separable analysis operator.

(c) *AASCO* reconstruction.

Figure 6.6.: Reconstruction of the *Barbara* image, which has been corrupted by multiplicative noise with $K = 10$. The dynamic range has been set to $[0, 255]$ for visual purposes.



(a) Masked input image.

(b) Reconstructed image with adaptively learned separable analysis operator.

(c) Bicubic interpolated image.

Figure 6.7.: Inpainting mising pixels of the *Peppers* image, where 80% of the pixels are missing.

(6.7) involves the standard ℓ_2-norm to measure the fidelity to the measurements. As the size of the operator is not changed, and the image is normalized to have intensities in the range $[0, 1]$, the parameters for the full-rank and coherence penalties are reset to $\kappa = 35.0$ and $\gamma = 0.5$. Accordingly, the parameter in the sparsity measure reads $\nu = 2000$ again. The measurements are artificially generated by randomly sampling 20% of the pixels, i.e.,

Table 6.4.: Adaptive Inpainting experiment for five different test images (256 × 256) where 80% of the pixels are masked out. Achieved PSNR in decibels (dB) and MSSIM.

Miss. pix.	Algorithm	PWC		Barbara		Boats		Lena		Peppers	
80 %	adaptive	20.31	0.773	**24.45**	**0.791**	**27.26**	**0.837**	**27.90**	0.838	**27.73**	**0.877**
	Nearest	20.25	0.777	21.19	0.686	23.02	0.710	24.92	0.766	24.25	0.785
	Cubic	**22.18**	**0.821**	22.36	0.749	25.24	0.785	27.55	**0.840**	26.95	0.859

80% of the image content is missing. Since the measurements are noise free, the fidelity is weighted with $\lambda = 2 \cdot 10^4$. Results are achieved after stopping the execution of the algorithm with the threshold $\delta_{\text{adaptive}} = 10^{-6.5}$.

Table 6.4 summarizes the recovery accuracy achieved on the five test images in terms of PSNR and MSSIM. Except for the *Piecewise-Constant* image, the adaptive learning approach outperforms standard reconstruction methods like Nearest-Neighbor or Cubic interpolation. The worse performance on the *Piecewise-Constant* image can be attributed to the same effect as observed in the experiment shown in Section 6.3.2. As a reference, Figure 6.7 shows the recovery of the *Peppers* image from 20% of the pixels. The proposed adaptive method leads to visually more pleasant results.

6.3.3. 3D Blind Learning

This section focuses on the recovery of volumetric data from undersampled observations with an adaptively learned separable analysis operator. Besides two dimensional data like images, many vision based signals are inherently three or even multidimensional. Typical examples for volumetric data are videos, where multiple sill images are acquired over time. Hyperspectral volumes are composed of several images that additionally encode the spectral components of the scene. In medical imaging, volumetric MRI or CT scans allow to assess the three dimensional structure of body parts. Because of the high correlation of voxels across all dimensions, processing these multidimensional data at once is highly desired. However, the exponential growth of the sample points limits the applicability of standard vectorization approaches. On the contrary, the separable structure of the proposed analysis operator is very well suited to exploit the volumetric structure of the data. Consequently, instead of reconstructing each slice separately like in standard image reconstruction, the information along the additional third dimension can be easily incorporated in the recovery framework.

3D MRI Compressed Sensing

In recent years, the CS theory has been applied especially in the field of medical imaging where the reduction of measurements has a significant and valuable impact. Ongoing from the seminal work of Lustig et al. [87] MRI has become a cornerstone to motivate CS based image reconstruction [25, 117]. MRI is an imaging modality often used in clinical diagnostics due to its ability to visualize fine anatomical structures. However, the image acquisition process suffers from the sequential sampling of spatial Fourier coefficients in k-space, making MRI a rather slow modality. The k-space constitutes the 2D Fourier coefficients of the image, i.e., once the k-space is fully sampled, the resulting MR image can be easily obtained via an inverse Fourier Transformation. The acquisition of 3D MRI volumes is usually performed by collecting samples of different 2D slices.

The sequential sampling of k-space entries is a severe limitation that on the one hand reduces the throughput of patients in the clinical environment and on the other hand makes it hard to capture moving body parts like in cardiac MRI for example. Even respiration of the patient under investigation can cause artifacts in the image. Besides artifacts observable in a single slice, another issue is the acquisition of temporal/volumetric data. In the case of dynamic MRI, where each slice represents another time instance of a moving body part, lowering the scanning time that is needed to sample a single slice will automatically increase the time resolution. Furthermore, volumetric MRI is also used to image whole body parts like the head, where a plethora of successive slices are mandatory to achieve reasonable resolution. In this case, reducing the number of measurements via undersampling during acquisition reduces the time the patient has to spend in the scanner. For this reason, CS based image reconstruction has been proven very useful to alleviate these drawbacks while providing accurate image quality [122, 127].

Since reducing the number of measurements has a direct impact on the acquisition time, many different acquisition strategies have been proposed in the literature so far. If the 2D k-space is sampled line by line on a Cartesian grid, possibly the easiest way of scan time reduction is to leave out scanning lines. However, the imbalanced undersampling either along the vertical or horizontal dimension leads to severe artifacts that manifest themselves in ghost images. That is why in the presented experiments, the sampling is performed along radial lines that intersect in the origin as shown in Figure 6.8 (The DC component of the Fourier transform is assumed to be located at the center). This strategy offers the advantage of consistently distributed samples along all spatial directions. Furthermore, because of the higher sampling density around the origin, this pattern puts an emphasis on the important low frequency parts of the image that carry most of the image information.

Two different data sets are considered in the numerical evaluation of the blind reconstruction framework. The first set consists of synthetically generated Head-MRI at-

lases. The data generation process is described in [31, 32, 79] with the volumes being publicly available at `http://www.bic.mni.mcgill.ca/brainweb/`. For the experiments, each of the 2D slices has been cropped to 128×128. The second dataset from `http://www.mridata.org/` consists of real Knee-MRI volumes, where each slice has a resolution of 200×200. In all the experiments, five successive slices are considered resulting in two datasets of dimensions $128 \times 128 \times 5$ and $200 \times 200 \times 5$. As already pointed out, in the CS-MRI framework the measurements $\boldsymbol{y}$ constitute undersampled Fourier coefficients, i.e., we have $\boldsymbol{y} = \boldsymbol{\Phi s}$, with $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$, $M < N$, being the undersampled Fourier transform matrix with M Fourier basis vectors in its rows that correspond to the radial sampling positions identified by the sampling mask. Here, the approach of successively sampled 2D slices is followed, i.e., each slice is sampled independently according to a two dimensional Fourier transform. Thus, the data term can be expressed as a sum over the different slices $\boldsymbol{s}_i$, resulting in

$$d_{\text{additive}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M} \sum_{i=1}^{L} (\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{s}_i)^{\mathsf{H}} (\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{s}_i), \tag{6.16}$$

where in this case $L = 5$ is the number of slices and M denotes the total amount of measurement samples. In order to account for complex measurements, the Hermitian transpose $(\cdot)^{\mathsf{H}}$ is used. To evaluate the benefit of the multidimensional reconstruction, two different sampling schemes are considered. First, the number of radial lines is varied along the additional third dimension. To be precise, the number of lines per slice read $\{25, 19, 17, 23, 21\}$ as illustrated in the upper row of Figure 6.8. In the second scheme, the number of lines has been fixed to 21 but the pattern is rotated around the origin such that after five iterations the pattern periodically repeats itself (cf. Figure 6.9). In both settings, the middle image out of the five slices is the image of interest that is used to assess the recovery performance in terms of PSNR and MSSIM. While the size of the operators has been set to $\boldsymbol{\Omega}_i \in \mathbb{R}^{7 \times 5}, i = 1, 2, 3$, which for a vectorization approach would already result in a matrix $\iota(\boldsymbol{\Omega}^{(1)}, \boldsymbol{\Omega}^{(2)}, \boldsymbol{\Omega}^{(3)}) \in \mathbb{R}^{343 \times 125}$, the parameters for the penalties still read $\kappa = 35.0$ and $\gamma = 0.5$. The dynamic range of the slices has been set to $[0, 1]$, thus the parameter in the sparsity measure reads $\nu = 2000$ and the weighting of the fidelity term is set to $\lambda = 1000$ assuming noise free samples. The algorithm terminates when the threshold $\delta_{\text{adaptive}} = 10^{-7}$ is reached.

Figures 6.8, 6.9, and 6.10 show the recovered slices along with the utilized sampling patterns. For comparison, the Dictionary Learning MRI (*DLMRI*) method, introduced in [122], and the Transform Learning MRI (*TLMRI*) approach proposed in [132] is used. The DLMRI algorithm recovers the image from undersampled measurements while adaptively learn-

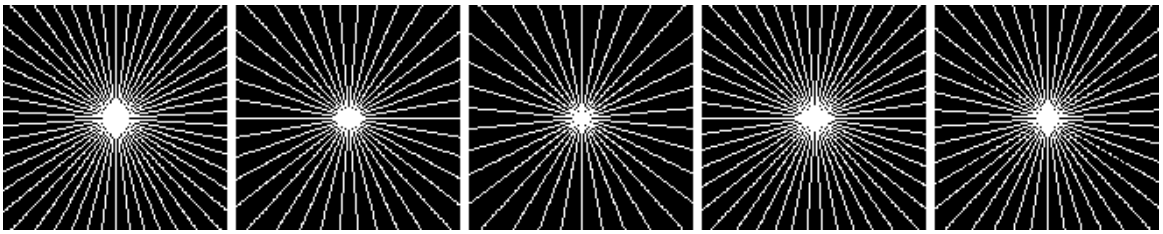
ing a dictionary based on the sparse synthesis model. Since this algorithm is intended to reconstruct single images, only the middle slice is considered. Assuming a patch size of 5×5, the resulting dictionary has the dimension $\boldsymbol{D} \in \mathbb{R}^{25\times 49}$. Various parameter settings have been tested, while the results are reported for the following setup (DLMRI parameters: lambda $= 300$, threshold $= 0.025$, sparsity $= 5$, iterations = 150). The TLMRI algorithm is based on the sparse transform model as introduced Section 1.2.3. Analogous to the DLMRI method, only the middle slice is considered during the blind reconstruction with the transform matrix $\boldsymbol{W} \in \mathbb{R}^{49\times 25}$. Again, after testing different parameter settings, the experiments are conducted with the setup (TLMRI parameters: lambda $= 0.2$, nu $= 10^6/128^2$, threshold $= 0.1$, iterations $= 100$). For both methods and all experiments, the sampling mask is set to 22 radial lines which results in a point symmetric pattern around the origin (cf. Figure 6.8). Note that for the 3D case, the average amount of measurements per image is still below the number of samples used in the 2D based reconstruction of *DLMRI* and *TLMRI*.

As can be seen visually and numerically, incorporating the additional dimension into the reconstruction framework significantly increases the recovery quality. In particular the first experiment, presented in Figure 6.8, shows an increase of around $3-4$dB for the 3D approach compared to its competitors *DLMRI* and *TLMRI*, although the image of interest is reconstructed from only 80% of the measurements (17 radial lines compared to 22 radial lines). Note that in the 3D case, the Fourier transform is applied to each slice separately, thus, the performance increase can be attributed to the analysis co-sparsity regularizer that leverages the additional information from adjacent layers. Also the real Knee-MRI images exhibit finer anatomical structures in the reconstruction which is highly desired for exact treatment planning.

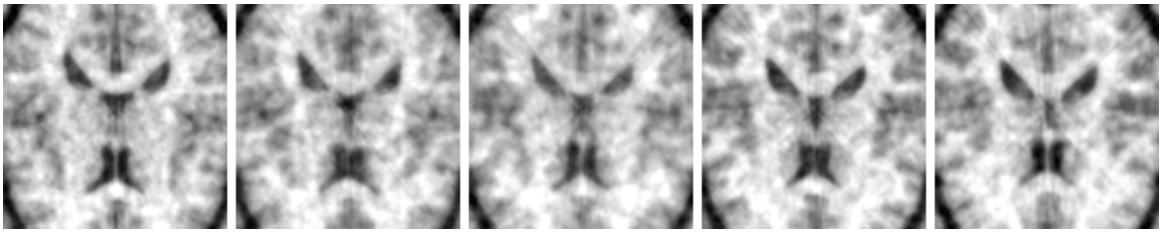
6.4. Summary

In this chapter a blind separable analysis operator learning scheme that simultaneously recovers the original image from possibly corrupted or compressed measurements is considered. The formulation as a smooth optimization problem allows to easily integrate a data fidelity term that on the one hand accounts for the assumed noise distribution. On the other hand, in contrast to purely patch based approaches, the presented framework is able to handle measurements that require global image support. The resulting product of manifolds structure of the problem is efficiently tackled by a conjugate gradient on manifolds approach. In the numerical experiments it is shown that the blind learning strategy is able to to adaptively capture the underlying structure of the images, which leads to a superior reconstruction performance in various inverse problems. Furthermore, the separable structure of the operator allows to efficiently handle multidimensional signals like

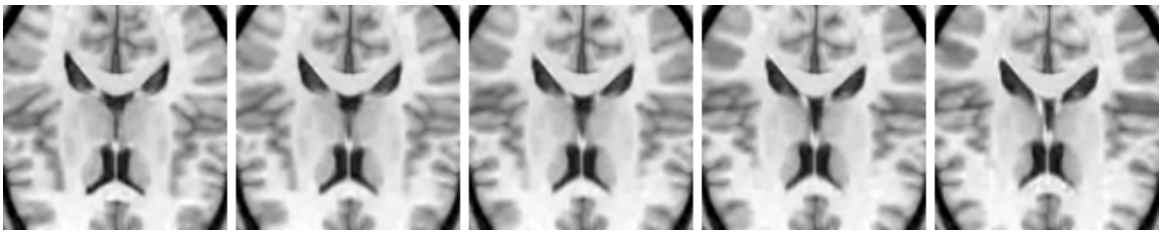
volumetric MRI data, whereas the standard vectorization approach becomes impractical because of the exponentially increasing complexity. Finally, the CS results indicate that exploiting the correlation of the additional dimension of the data via the separable co-sparse analysis model significantly improves the reconstruction performance.



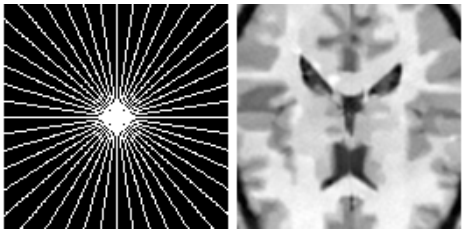
(a) Sampling Masks.



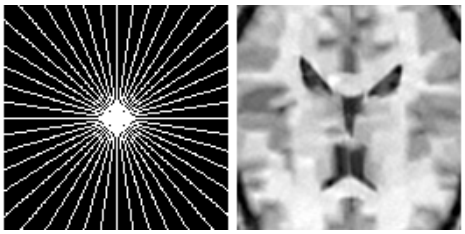
(b) Zero-filled reconstructed images.



(c) Reconstructed image with adaptively learned 3D separable analysis operator (Middle slice, PSNR: 28.86, MSSIM: 0.892).

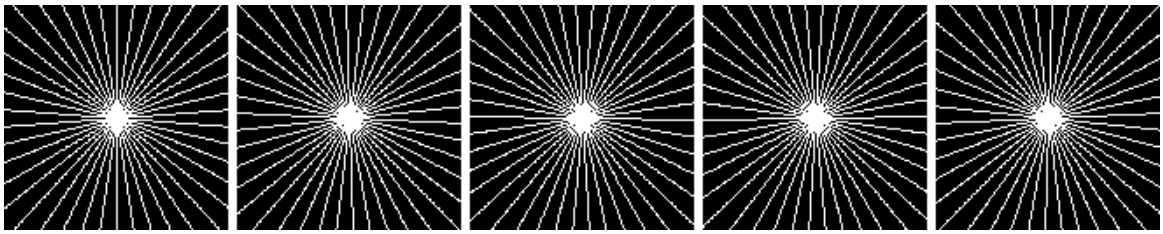


(d) Reconstructed image with DLMRI (PSNR: 25.66, MSSIM: 0.790).

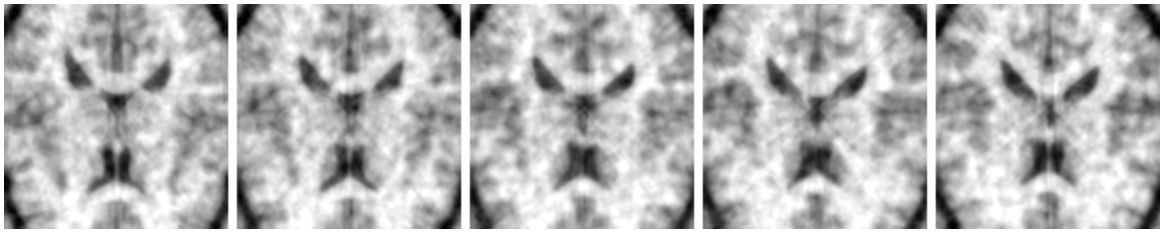


(e) Reconstructed image with TLMRI (PSNR: 24.77, MSSIM: 0.757).

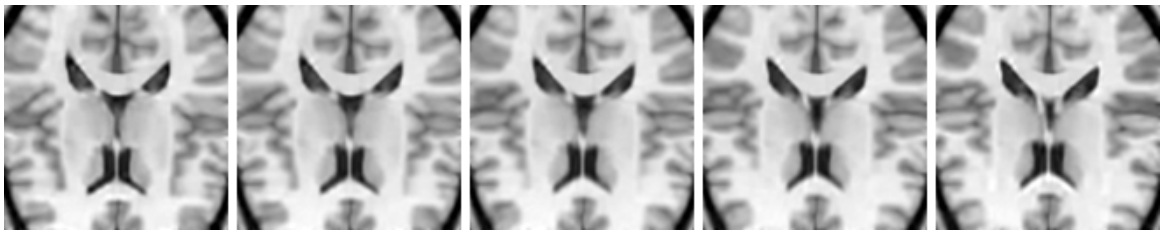
Figure 6.8.: Reconstruction of the MRI volume from radial samples with varying density (first sampling scheme). The percentage of samples for each slice reads {19.6%, 15.2%, 13.7%, 18.4%, 16.6%}



(a) Sampling Masks.

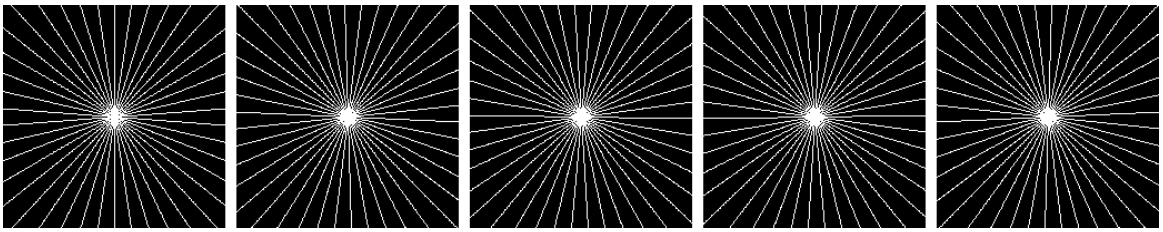


(b) Zero-filled reconstructed images.

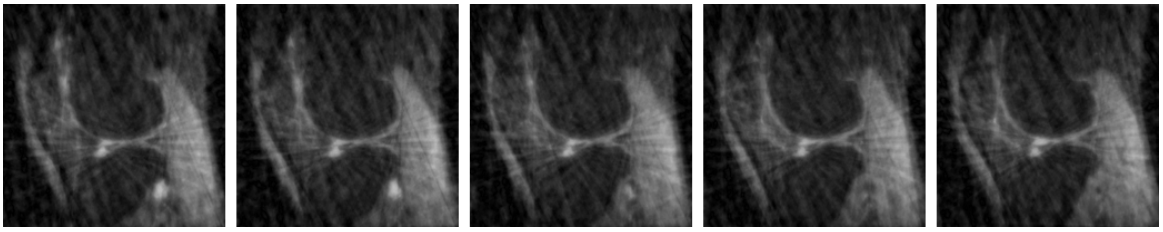


(c) Reconstructed image with adaptively learned 3D separable analysis operator (Middle slice, PSNR: 29.83, MSSIM: 0.901).

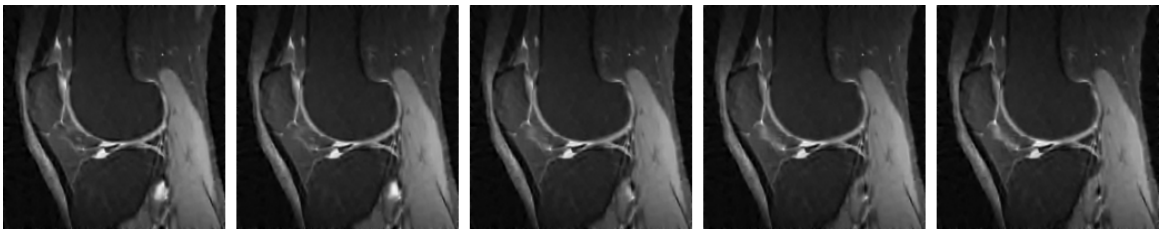
Figure 6.9.: Reconstruction of the MRI volume from rotated radial samples (second sampling scheme). The percentage of samples for each slice reads 16.6%



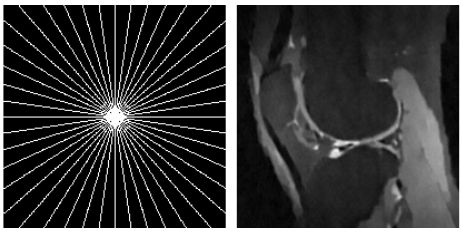
(a) Sampling Masks.



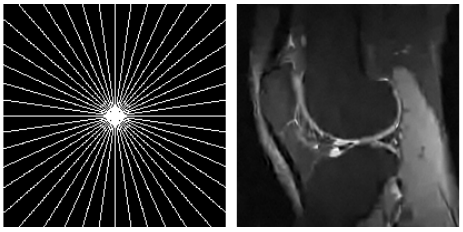
(b) Zero-filled reconstructed images.



(c) Reconstructed image with adaptively learned 3D separable analysis operator (Middle slice, PSNR: 30.82, SSIM: 0.820).



(d) Reconstructed image with DLMRI (PSNR: 29.26, MSSIM: 0.777).



(e) Reconstructed image with TLMRI (PSNR: 29.62, MSSIM: 0.793).

Figure 6.10.: Reconstruction of the MRI volume from rotated radial samples (second sampling scheme). The percentage of samples for each image reads 10.9%

Chapter 7.

Learning Separable Analysis Operators as Co-sparse Auto-Encoders

In recent years, Artificial Neural Networks (ANNs) have attracted lots of researchers in the machine learning community. The term 'neural' stems from the fact that earliest approaches were inspired by the functionality of the biological brain [45]. This strategy is rather intuitive since transferring these functionalities to artificial systems is motivated by the inherent capability of the human brain to show intelligent behavior. Hence, the building blocks of these networks are referred to as neurons that process input information from their connections to several other neurons analogous to the dendrites in biological nerve cells.

From another perspective, these networks can be treated as function approximators. In general, the network maps samples $\boldsymbol{s}$ from input space $\mathfrak{S}$ to the output space $\mathfrak{Z}$ via the function $f\colon \mathfrak{S} \to \mathfrak{Z}$. Assuming some target function $f^\star$ with $\boldsymbol{z}^\star = f^\star(\boldsymbol{s})$ and $\boldsymbol{z} \in \mathfrak{Z}$, learning the model amounts to finding an approximate function $f(\boldsymbol{s}, \boldsymbol{\Theta})$, parametrized by the learnable parameters $\boldsymbol{\Theta}$, such that $f^\star(\boldsymbol{s}) \approx f(\boldsymbol{s}, \boldsymbol{\Theta})$. The strength of the networks in various applications originates from the strategy to connect several functions in series, i.e., $f(\boldsymbol{s}, \boldsymbol{\Theta}) = f_L(\dots f_2(f_1(\boldsymbol{s}, \boldsymbol{\Theta}_1), \boldsymbol{\Theta}_2), \dots), \boldsymbol{\Theta}_L)$ which gives rise to name them Deep Neural Networks (DNNs) or deep learning models. In its basic form, the network resembles a directed graph where the samples are simply fed through the network to generate the output, resulting in the terminology of Feedforward Neural Networks (FNNs) or Multilayer Perceptrons (MLPs). In the literature, various modified architectures specialized for certain domains like images and speech have been presented. Among others, these architectures range from Convolutional Neural Networks (CNNs) [82] for image processing, Recurrent Neural Networks (RNNs) [145] and Long Short-Term Memory Networks (LSTMs) [68] for sequence modeling, up to Residual Networks (ResNets) [66] that lead to state of the art results in image classification and object detection.

The parameters $\boldsymbol{\Theta}$ of the network are estimated via minimizing a loss function $\mathcal{L}_{\boldsymbol{\Theta}}(\cdot)$ that typically measures the deviation between the output of the network to some target

value. Suppose a classical supervised learning problem, where we are given the labeled dataset $\{\boldsymbol{s}_i, \boldsymbol{z}_i\}_{i=1}^{T} \subset \mathfrak{S} \times \mathfrak{Z}$. A popular choice of the loss function is the Euclidean distance that reads $\mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{s}_i, \boldsymbol{z}_i) = \|f(\boldsymbol{s}_i, \boldsymbol{\Theta}) - \boldsymbol{z}_i\|_2^2$ and which is commonly used in a regression task. However, depending on the task at hand, several other choices of the loss function exist, e.g. in classification tasks where $\boldsymbol{z}_i$ represent class labels, the hinge loss or cross entropy loss is used to measure the prediction accuracy.

A common strategy to tackle the learning problem is the backpropagation algorithm [169]. For this purpose, after each forward pass where the actual loss based on the current weightings is calculated, the error compared to the target value is recursively backpropagated through the network to determine the partial derivatives of each parameter matrix in the respective layer. A simple gradient descent step is then performed to update the weights such that a new forward pass can be initialized which eventually constitutes the iterative optimization scheme.

Unsupervised Learning

Many machine learning tasks can be considered an unsupervised learning problem. Instead of finding a nonlinear mapping between the inputs and some given labels, as it is done in the classification scenario, one of the key ideas behind unsupervised learning is to find a signal representation that reveals the underlying structure of the data [74]. In general, it is assumed that the obtained representation is more suitable for the desired task of interest, e.g. signal compression. In the last decade, Auto-Encoders have been extensively used to learn patterns from the input data in an unsupervised fashion.

The ultimate goal of an Auto-Encoder is to automatically learn representations from unlabeled data by means of forcing the target output values to resemble the inputs. This can be achieved via learning the model $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_{\mathbf{d}}, \boldsymbol{b}_{\mathbf{d}}, \boldsymbol{\Theta}_{\mathbf{e}}, \boldsymbol{b}_{\mathbf{e}}\}$ that minimizes

$$\begin{aligned} \boldsymbol{\Theta}^{\star} \in \arg \min_{\boldsymbol{\Theta}=\{\boldsymbol{\Theta}_{\mathbf{d}}, \boldsymbol{b}_{\mathbf{d}}, \boldsymbol{\Theta}_{\mathbf{e}}, \boldsymbol{b}_{\mathbf{e}}\}} \quad \widehat{\mathbb{E}}_{\boldsymbol{S}}[\mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{S})] &= \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{s}_i) \\ &= \frac{1}{T} \sum_{i=1}^{T} \|f_d(f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}_{\mathbf{e}}, \boldsymbol{b}_{\mathbf{e}}), \boldsymbol{\Theta}_{\mathbf{d}}, \boldsymbol{b}_{\mathbf{d}}) - \boldsymbol{s}_i\|_2^2, \end{aligned} \tag{7.1}$$

with $f_d(\cdot, \boldsymbol{\Theta}_{\mathbf{d}}, \boldsymbol{b}_{\mathbf{d}})$ denoting the decoder and $f_e(\cdot, \boldsymbol{\Theta}_{\mathbf{e}}, \boldsymbol{b}_{\mathbf{e}})$ being the encoder function. The matrices $\boldsymbol{\Theta}_{\mathbf{d}} \in \mathbb{R}^{k \times n}$ and $\boldsymbol{\Theta}_{\mathbf{e}} \in \mathbb{R}^{n \times k}$ represent the weights, while $\boldsymbol{b}_{\mathbf{e}} \in \mathbb{R}^k$ and $\boldsymbol{b}_{\mathbf{d}} \in \mathbb{R}^n$ denote the bias vectors. The intermediate or hidden representation $\boldsymbol{h}_i \in \mathbb{R}^k$ is obtained via

$$\boldsymbol{h}_i = f_e(\boldsymbol{s}_i, \boldsymbol{\Theta}_{\mathbf{e}}, \boldsymbol{b}_{\mathbf{e}}) = \sigma(\boldsymbol{a}_i) = \sigma(\boldsymbol{\Theta}_{\mathbf{e}}^{\top} \boldsymbol{s}_i + \boldsymbol{b}_{\mathbf{e}}), \tag{7.2}$$

with $\sigma(\cdot)$ denoting an elementwise possibly non-linear activation function. In general, the activation function of the decoder is set to be linear, i.e., $f_d(\boldsymbol{h}_i, \boldsymbol{\Theta}_\mathbf{d}, \boldsymbol{b}_\mathbf{d}) = \boldsymbol{\Theta}_\mathbf{d}^\top \boldsymbol{h}_i + \boldsymbol{b}_\mathbf{d}$. During learning, the Auto-Encoder tries to obtain parameters such that $f_d \circ f_e = \mathrm{id}$, i.e., it tries to achieve zero reconstruction error. At this point, it is important to note that we are not interested in the composite function $f(\boldsymbol{s}, \boldsymbol{\Theta}) = f_d \circ f_e$ since this function will simply approximate the input without providing any further information about its structure. It is the *encoder* that allows to map the signal to a meaningful representation that conveys the underlying structural information. That is why Auto-Encoders have also been used as a layerwise pre-learning of deep neural networks [80]. After learning an Auto-Encoder on the input data, the decoder has been discarded and the hidden representation is used as the input for the next Auto-Encoder, thus sequentially building the deep architecture that is eventually fine-tuned in a subsequent learning step.

There exist different strategies to learn useful representations. The most common one is to impose a structural constraint on the hidden representation, i.e., reducing the dimension in the hidden layer with $k < n$. In this setting, the Auto-Encoder tries to find a compressed version of the signal that retains as much information as possible to reconstruct a good approximation of the signal in the decoder step. It has been shown that with a linear activation function and a squared ℓ_2-error loss term, the Auto-Encoder learns a low-dimensional embedding similar to PCA [6]. A generalization to non-linear PCA is shown in [77], however depending on the chosen non-linearity it can happen that a simple scaling cause the weights to work in the linear regime of the activation function which resembles the linear PCA approach [14].

Instead of imposing any constraints on the representations, Vincent et al. [162] propose the concept of Denoising Auto-Encoders. In this framework, the Auto-Encoder tries to recover the original signal from its noise corrupted input. The goal is to learn a robust representation of the signal that contains the required information to reconstruct the clean input while those components that describe the noise are discarded. As already pointed out above, in [162] the authors also emphasize that they leverage this strategy as a training criterion to obtain meaningful representations rather than aiming to learn a state-of-the-art Denoising algorithm.

Another approach that encourages robustness of the hidden representations to small changes in the input is presented by Rifai et al. [133], which proposes to add a penalty term to the learning objective comprised of the Frobenius norm of the Jacobian of the encoder output. For example assuming a Sigmoid activation function, a small value of the Jacobian is achieved if the activation output lies in the saturated part of the non-linearity. Thus, the penalty enforces the encoder mapping to be contractive.

In the following, the focus is on another option, that is imposing a sparsity constraint on the hidden representation to extract the underlying structure of the data. As already

discussed in the related work, this strategy immediately reminds one of the analysis and synthesis framework with the encoder weights representing a co-sparse analysis operator. In contrast to the penalty based learning approach outlined in the previous chapters, I aim to explore the suitability of the Sparse Auto-Encoder framework to regularize the analysis operator learning problem. The following contributions are addressed in this chapter:

- Sparsity in the hidden representation can be achieved in different ways. Among others, rescaling the encoder weights or employing adequate activation functions are the simplest approaches to achieve sparsity. However, they do not permit to interpret the encoder as a co-sparse analysis model. For this reason, the encoder matrix is subjected to the same product of spheres manifold as done in the preceding chapters. Furthermore, a suitable activation function together with a simple ℓ_1-sparsity penalty is utilized to learn the Auto-Encoder mapping. This setting will enforce the encoder to follow the co-sparse analysis model, whose performance and reliability can be straightforwardly compared to the standard learning approach.
- In order to exclude trivial solutions like rank-1 operators, different penalty functions have been proposed regrading the conventional operator learning paradigm. The Auto-Encoder architecture naturally prevents the learning algorithm to reach such a solution, since identical hidden representations do not allow to recover the original input signals. Other than additional penalties, a norm constraint on the decoder weights is used to control the condition number of the encoder matrix. The subsequent numerical experiments indicate that the separable encoder matrix estimated via the proposed Sparse Auto-Encoder framework is on par with the conventionally learned models.

7.1. Co-sparse Auto-Encoder

In the literature, several approaches are proposed to realize an Auto-Encoder that exhibits a sparse hidden representation. An investigation of the principles that encourage common Auto-Encoder architectures like DAEs and CAEs to learn sparse representations can be found in [4]. One straightforward approach to achieve sparsity in the hidden representation is to use an appropriate activation function. In [119] the authors use the Rectified Linear Unit (ReLU) to obtain sparsity in the hidden layer. Other choices include the Sigmoid function $\sigma(a_{i,j}) = 1/(1 + \exp(-a_{i,j}))$ or the SoftPlus function $\sigma(a_{i,j}) = \ln(1 + \exp(a_{i,j}))$ that both asymptotically converge to zero for negative pre-activation coefficients.

The introduction of an additional penalty function is another commonly used technique in the Sparse Auto-Encoder literature. Nair & Hinton [99] propose to use a cross entropy

penalty that measures the deviation between the average activation $\overline{h_j}$ of *binary* hidden units and a predefined probability p of being active. The average activation associated to the j-th weight vector $\boldsymbol{\theta}_j$ is calculated via $\overline{h_j} = \frac{1}{T}\sum_{i=1}^{T} h_{i,j}$. The applied penalty reads $g(\boldsymbol{h}) = -\sum_{j=1}^{k}[p\log(\overline{h_j}) + (1-p)\log(1-\overline{h_j})]$. Lee et al. [83] follow a similar idea, however they use the standard squared ℓ_2-loss to measure the error between the average activation of the binary hidden units and the predetermined target activation p. A close variant of the cross entropy penalty is the Kullback-Leibler (KL) divergence which is investigated in [102, 4] and which is given by:

$$g(\boldsymbol{h}) = \sum_{j=1}^{k}\left[p\log\left(\frac{p}{\overline{h_j}}\right) + (1-p)\log\left(\frac{1-p}{1-\overline{h_j}}\right)\right]. \tag{7.3}$$

The KL penalty has its minimum at zero for $p = \overline{h_j}$ and can be added to the Auto-Encoder objective stated in Eq. (7.1) together with a hyperparameter γ that weights the reconstruction error against the sparsity penalty. The authors in [4] point out the close relationship of Eq. (7.3) to the classical sparse coding problem with ℓ_1-penalty. For $p \to 0$ and small, positive values of $\overline{h_j}$, as it is the case in the aforementioned scenarios, the weighted KL term reduces to $-\gamma\sum_{j=1}^{k}\log(1-\overline{h_j}) \approx \gamma\sum_{j=1}^{k}\overline{h_j} = \gamma\|\overline{\boldsymbol{h}}\|_1 = \frac{\gamma}{T}\sum_{i=1}^{T}\|\boldsymbol{h}_i\|_1$. Note that due to the binary features, a small value of $\overline{h_j}$ directly implies a sparse activation of the hidden unit. However, depending on the employed activation function and the encoding bias, the encoder function does not necessarily follow the co-sparse analysis model, i.e., the weight vector does not need to be orthogonal to the signal to achieve zero activation. That is why for a Sparse Auto-Encoder to follow the co-sparse analysis model, we restrict ourselves to the following constraint

$$\begin{cases} h_{i,j} = 0 & \text{iff } a_{i,j} = \boldsymbol{\theta}_j^\top \boldsymbol{s}_i = 0 \\ h_{i,j} \neq 0 & \text{otherwise.} \end{cases} \tag{7.4}$$

Thus, the following sections consider a Sparse Auto-Encoder of the form

$$\boldsymbol{\Theta}^\star \in \arg\min_{\boldsymbol{\Theta}=\{\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{b}_\mathbf{d},\boldsymbol{\Theta}_\mathbf{e}\}} \frac{1}{T}\sum_{i=1}^{T}\Big[\|(\boldsymbol{\Theta}_\mathbf{d}^\top f_e(\boldsymbol{s}_i,\boldsymbol{\Theta}_\mathbf{e}) + \boldsymbol{b}_\mathbf{d}) - \boldsymbol{s}_i\|_2^2 + \gamma\|f_e(\boldsymbol{s}_i,\boldsymbol{\Theta}_\mathbf{e})\|_1\Big], \tag{7.5}$$

that utilizes an activation function $\sigma(\cdot)$ that fulfills the requirements stated in (7.4) like the Tanh, Softsign, Bent-Identity, or even Linear activation function.

7.1.1. Model Constraints

While encouraging a sparse hidden representation of the input signals, the objective function proposed in Eq. (7.5) can still be prone to deliver meaningless or trivial results. On the one hand, sparsity can be easily achieved if the norm of some of the weights in $\boldsymbol{\Theta}_{\mathbf{e}}$ is shrinked to zero, while the remaining weights are used to reconstruct the signals. On the other hand, the linear regime of the activation function around zero will cause a scaling of the encoder and decoder weights. If we consider a constant $c > 1$ and scale the encoding weights via $\widehat{\boldsymbol{\Theta}_{\mathbf{e}}} = \frac{1}{c}\boldsymbol{\Theta}_{\mathbf{e}}$, the reconstruction error is not changed as long as we use the decoding weights $\widehat{\boldsymbol{\Theta}_{\mathbf{d}}} = c\boldsymbol{\Theta}_{\mathbf{d}}$. However, the scaling of the encoder will reduce the sparsity if it is measured in terms of the ℓ_1-norm as in Eq. (7.5). Thus, to learn meaningful representations in the hidden layer, we have to further restrict the set of admissible solutions. More precisely, to account for the scaling ambiguity, the norm of the encoder weights is restricted to $\|\boldsymbol{\theta}_{\mathbf{e},j}\|_2 = 1 \,\text{for}\, j = 1, \ldots, k$, i.e., the encoder weight matrix is an element of $\mathsf{OB}(n, k)$.

Restricting the norm of the weights in a neural network is a common strategy to avoid overfitting. Typically, an additional regularizer that penalizes the ℓ_2 norm of each weight vector is added to the cost function to prevent the network from fitting the sampling error. This particular method is referred to as weight decay, which has been shown to improve the generalization ability [69]. Instead of penalizing the norm, another promising approach consists of constraining the maximum norm of the weights. Whenever the norm exceeds some predefined constant, the weight vector is projected back to fulfill the requirements. Srivastava et al. [150] point out, that while this method is not only very well suited for their Dropout training scheme, it typically improves the performance of stochastic gradient descent training in general. Also the fixed unit norm weight constraint as proposed in this work has been utilized in the literature. Huang et al. [70] propose the projection based weight normalization (PBWN) scheme. In their work, they show that the additional unit norm constraint improves the classification accuracy of standard deep learning architectures. They attribute this behavior to the reduction of the ill-conditioning of the Hessian-matrix caused by the scaling ambiguity introduced above. A normalization scheme that decouples the lengths of the weight vectors from their directions is presented in [146]. This behavior is achieved via a reparameterization into a uniform, normalized weight vector together with an additional scaling parameter. Hence, the optimization is performed with respect to both the weight vector and its corresponding scaling parameter. Due to the reparameterization, the authors observe an improved conditioning of the optimization problem and a speed up of the SGD convergence.

Another related and commonly used normalization approach is Batch Normalization (BN) [73]. In BN, the preactivation is normalized based on the mean and the standard

deviation of the mini-batch output. Although BN became a very popular regularization method in the recent literature about deep learning, it has the drawback of being data dependent. That is, the statistics have to be evaluated for each mini-batch separately, which constitutes an extra overhead in computation [69].

7.1.2. Implicit Condition Number Regularization

After motivating the choice of the nonlinear activation function as well as the unit norm constraint on the encoder matrix to avoid the scaling ambiguity problem, in the following the impact of the decoder is considered. Although the issue that a vanishing norm of the weights trivially implies sparsity in the hidden representation has been resolved by the unit norm constraint, the structure of the encoder weights still pose another problem. Recall the discussion from Section 4.2.2 that minimum sparsity with regard to the ℓ_1-measure is achieved if we simply repeat the encoder weight $\boldsymbol{\theta}_{\mathbf{e}}^{\star}$ that minimizes $\|\boldsymbol{\theta}_{\mathbf{e}}^{\top}\boldsymbol{S}\|_1$. Now, in order to allow the Auto-Encoder to reconstruct the signals, the weights of the optimal rank-1 encoder matrix have to be slightly changed such that $\mathrm{rk}(\boldsymbol{\Theta}_{\mathbf{e}}) = n$. Since many activation functions are nearly linear for small values around the origin, the Pseudo-Inverse of the encoder serves as a suitable decoder matrix in this case, i.e., $\boldsymbol{\Theta}_{\mathbf{d}} = \boldsymbol{\Theta}_{\mathbf{e}}^{\dagger}$. However, this causes the norm of $\boldsymbol{\Theta}_{\mathbf{d}}$ to explode. This observation, in turn, motivates to also restrict the norm of the decoder to implicitly regularize the condition number of the encoder. In the remainder of this section, this strategy will be discussed in detail.

In the following, the data fidelity term that measures the deviation between the original samples and their approximations based on the hidden representations is considered. Let $\boldsymbol{S} \in \mathbb{R}^{n \times T}$ denote a set of normalized samples with $\mathrm{rk}(\boldsymbol{S}) = n$. Furthermore, the decoder bias $\boldsymbol{b}_{\mathbf{d}}$ is set to zero, and the activation function is chosen to be the identity map, i.e., $\sigma(a_{i,j}) = a_{i,j}$ in order to better illustrate the effect. In this scenario, the loss simply reads

$$\mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{S}) = \frac{1}{T}\sum_{i=1}^{T} \|\boldsymbol{\Theta}_{\mathbf{d}}^{\top}\boldsymbol{\Theta}_{\mathbf{e}}^{\top}\boldsymbol{s}_i - \boldsymbol{s}_i\|_2^2. \tag{7.6}$$

Clearly, the optimal solution is achieved for $\boldsymbol{\Theta}_{\mathbf{d}} = \boldsymbol{\Theta}_{\mathbf{e}}^{\dagger}$. Let $\boldsymbol{\Theta}_{\mathbf{e}} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\top}$ denote the SVD of the encoder weights, by construction we have

$$\det(\boldsymbol{\Theta}_{\mathbf{e}}^{\dagger\top}\boldsymbol{\Theta}_{\mathbf{e}}^{\dagger}) \cdot \det(\boldsymbol{\Theta}_{\mathbf{e}}\boldsymbol{\Theta}_{\mathbf{e}}^{\top}) = \prod_{i=1}^{n}\frac{1}{\sigma_i^2} \cdot \prod_{i=1}^{n}\sigma_i^2 = 1, \tag{7.7}$$

where σ_i denotes the i-th singular value. Note that $\boldsymbol{\Theta}_{\mathbf{e}}$ is assumed to be a full rank matrix with normalized columns, i.e., we have $\sigma_n > 0$ and $\sum_{i=1}^{n}\sigma_i^2 = k$. Consequently, for $\boldsymbol{\Theta}_{\mathbf{e}}$ to

approach a rank-1 matrix composed of nearly identical filters, the product of the singular values becomes smaller, while the reciprocal becomes bigger. From the relation between the arithmetic and the geometric mean, which reads $\sqrt[n]{\prod a_i} \leq \frac{1}{n}\sum a_i$, it follows that

$$\sqrt[n]{\prod_{i=1}^{n}\frac{1}{\sigma_i^2}} \leq \tfrac{1}{n}\sum\frac{1}{\sigma_i^2} = \tfrac{1}{n}\|\boldsymbol{\Theta}_{\mathbf{e}}^{\dagger}\|_F^2. \tag{7.8}$$

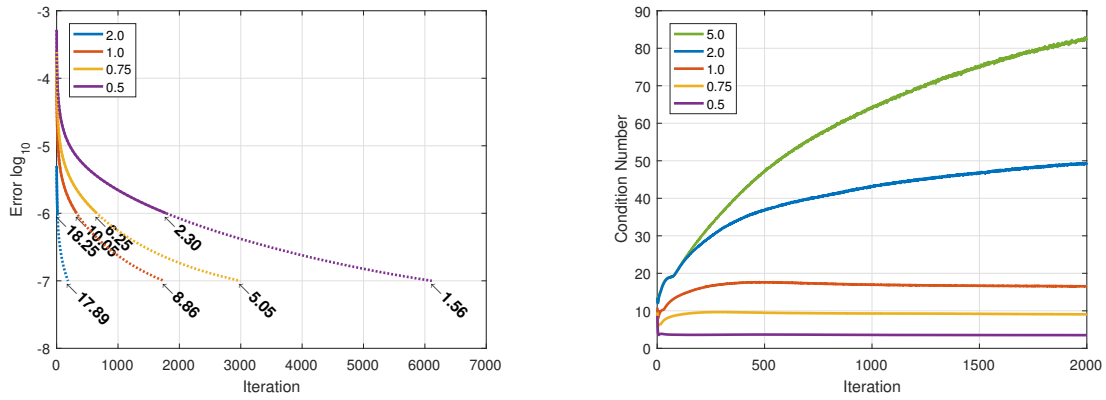
Thus, although the norm of the encoder matrix remains constant, the structure of the encoder weights will directly influence the norm of the decoder matrix, i.e., the Frobenius norm will increase.

Based on this observation, the opposite strategy of constraining the norm of the decoder to affect the behavior of the singular values and eventually, to implicitly control the condition number of the sought encoder matrix $\boldsymbol{\Theta}_{\mathbf{e}}$ is pursued. To this end, the norm of the individual decoder weights is constrained to be smaller than some predefined constant c, i.e., $\|\boldsymbol{\theta}_{\mathbf{d},i}\|_2 < c,\ i = 1,\dots,k$, where $\boldsymbol{\theta}_{\mathbf{d},i}$ denotes the i-th row of the decoder matrix $\boldsymbol{\Theta}_{\mathbf{d}}$. The impact of the additional constraint is empirically verified based on the problem introduced in (7.6). That is, we are given the set $\boldsymbol{S} \in \mathbb{R}^{49\times 10000}$ of normalized samples and a random initialization of $\boldsymbol{\Theta}_{\mathbf{e}} \in \mathbb{R}^{49\times 100}$, with each column normalized to unit length. Furthermore, the decoder matrix is initialized as $\boldsymbol{\Theta}_{\mathbf{d}} = \boldsymbol{\Theta}_{\mathbf{e}}^{\dagger} \in \mathbb{R}^{100\times 49}$, where all rows, whose norm exceeds the constant c, are rescaled to fulfill the constraint. With both initializations at hand, the loss in Eq. (7.6) is minimized via SGD.

Figure 7.1a illustrates the progress in the loss for different choices of $c \in \{2.0, 1.0, 0.75, 0.5\}$. That is, a different amount of rows has to be rescaled to fulfill the constraint right from the beginning. During optimization, the algorithm tries to reach a solution that offers both a minimal reconstruction error and compliance to the constraints. At two distinct error levels, the condition number of the encoder matrix is given. Clearly, a lower value of c forces the encoder to exhibit well-balanced singular values, which finally results in a low condition number. The condition number of the initialization $\boldsymbol{\Theta}_{\mathbf{e}}^{(0)}$ reads 18.73, while the maximum norm of the rows in $\boldsymbol{\Theta}_{\mathbf{d}}^{(0)}$ is 2.33.

In view of these results, the final objective to learn a separable analysis operator within a Co-sparse Auto-Encoder framework reads

$$\begin{aligned}\boldsymbol{\Theta}^{\star} \in \arg\min_{\boldsymbol{\Theta}=\{\boldsymbol{\Theta}_{\mathbf{d}},\boldsymbol{b}_{\mathbf{d}},\boldsymbol{\Theta}_{\mathbf{e}}\}} \frac{1}{T}\sum_{i=1}^{T}\Big[\|(\boldsymbol{\Theta}_{\mathbf{d}}^{\top} f_e(\boldsymbol{s}_i,\boldsymbol{\Theta}_{\mathbf{e}}) + \boldsymbol{b}_{\mathbf{d}}) - \boldsymbol{s}_i\|_2^2 + \gamma\|f_e(\boldsymbol{s}_i,\boldsymbol{\Theta}_{\mathbf{e}})\|_1\Big],\\ \text{s.t.}\quad \boldsymbol{\Theta}_{\mathbf{e}} \in \mathsf{OB}(n,k),\quad \boldsymbol{\Theta}_{\mathbf{e}} = \iota(\boldsymbol{\Theta}_{\mathbf{e},1},\boldsymbol{\Theta}_{\mathbf{e},2}),\quad \|\boldsymbol{\theta}_{\mathbf{d},i}\|_2 < c,\ i = 1,\dots,k.\end{aligned} \tag{7.9}$$



(a) Impact of max-norm constraint on the Condition number of the encoder matrix.

(b) Progress of the Condition number of the encoder matrix for different choices of c.

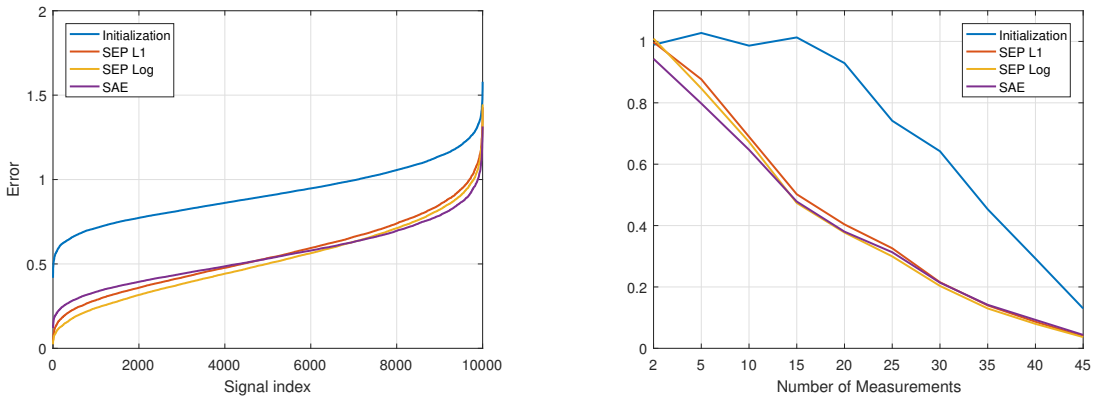
Figure 7.1.: Implicit Condition number regularization. **(a)** The progress of the loss (logarithmic scale) for different choices of the max-norm constant c is depicted. At two distinct error levels, the Condition number of the encoder matrix is given. **(b)** Condition number of the encoder matrix, while minimizing problem (7.9) that includes a sparsity penalty.

In order to demonstrate the usefulness of the learned encoder matrix, some numerical experiments are conducted in the subsequent part of this chapter.

7.2. Numerical Experiments

The aim of the proposed learning framework is to find a representation of the input signals that is useful for some particular task. In order to capture structural information of the samples, the encoder follows the co-sparse analysis model assumption. As a consequence, the suitability of the learned mapping is assessed by means of the same numerical experiments as conducted in the preceding chapters. For this purpose, $T = 10\,000$ centered and normalized patches from the images shown in Figure 4.3 are used for training. The patch size reads 7×7, while the encoders $\{\boldsymbol{\Theta}_{\mathbf{e},j}\}_{j=1}^{2}$, with $\boldsymbol{\Theta}_{\mathbf{e},j} \in \mathbb{R}^{7\times 10}$ map the input signals $\boldsymbol{S}_i$ to the (vectorized) hidden representation $\boldsymbol{h}_i \in \mathbb{R}^{100}$. The non-linearity is chosen to be the Softsign function that reads $\sigma(a_{i,j}) = \frac{a_{i,j}}{1+|a_{i,j}|}$. To optimize the objective, the Adadelta method [178] is used. This SGD variant utilizes a running average over gradients that is simultaneously used to determine a suitable step. In all experiments the decoder bias is initialized as a random vector, the batch size is set to $|b(t)| = 50$, and the sparsity penalty weight reads $\gamma = 0.05$.

For this setting, the impact of the max-norm constraint imposed on the decoder is eval-



(a) Denoising **(b)** Compressed Sensing

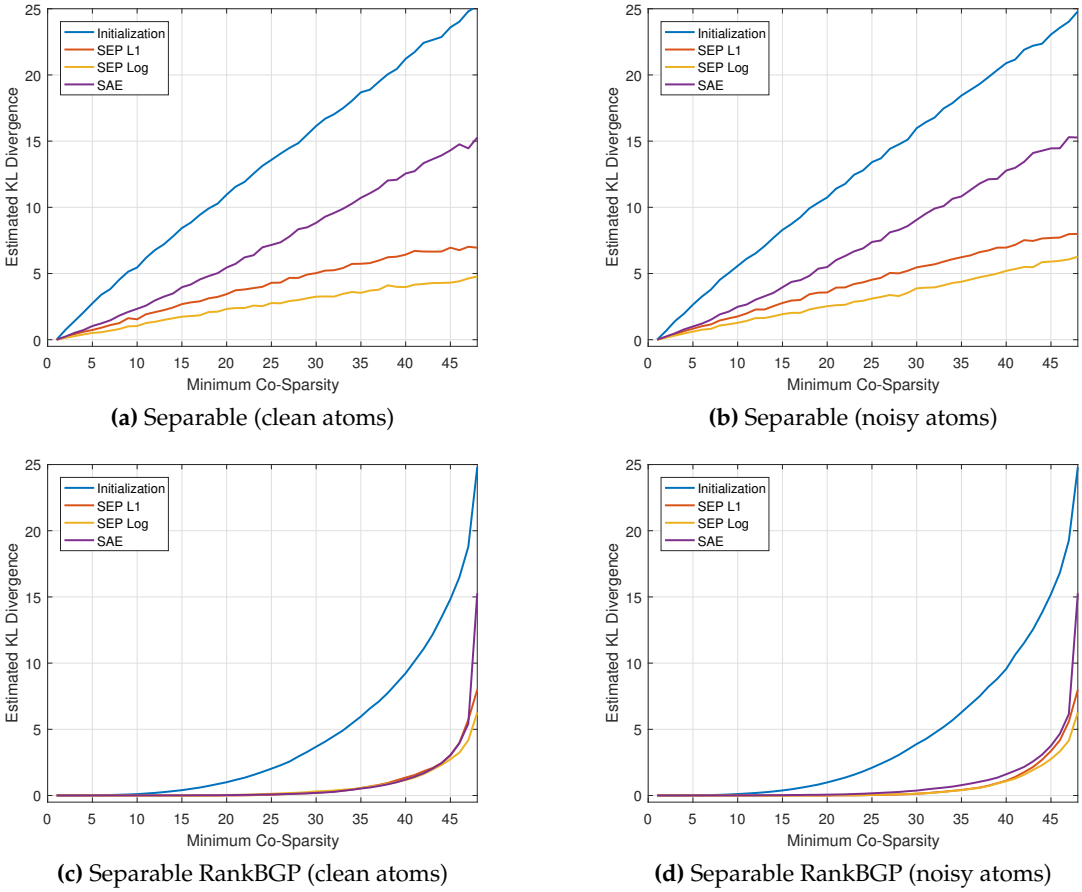
Figure 7.2.: Performance of the separable analysis operator that has been learned in the SAE framework.

uated first. As can be seen from Figure 7.1b, the additional max-norm constraint prevents the condition number of the encoder matrix $\boldsymbol{\Theta}_{\mathbf{e}} = \iota(\boldsymbol{\Theta}_{\mathbf{e},1}, \boldsymbol{\Theta}_{\mathbf{e},2})$ to explode. Analogous to the results shown in Figure 7.1a, the regularization can be adjusted depending on the parameter c. In the following, a fixed value of $c = 0.75$ is used throughout all experiments.

7.2.1. Inverse Problem Regularization

In the first experiment, the learned encoder matrix is used in a sparsity prior to regularize the solution of the same inverse problems as already introduced in Section 5.3. That is, on the one hand, clean signals have to be restored from noisy observations while on the other hand, the original signals have to inferred from undersampled measurements. For comparison, two additional separable analysis operators that have been learned on the same training signals but with the learning framework outlined in Chapter 4 are considered. While in the first case, Eq. (4.2) is used as the sparsity measure, resulting in the operator *SEP Log*, the second operator *SEP L1* is learned based on the standard ℓ_1-norm to enable a better comparison to the proposed *SAE* setting.

Figures 7.2a and 7.2b show the performance of all three approaches. Regarding both problems, the reconstruction quality based on the *SAE*-learned operator is on par to the recovery achieved with the operators that have been learned with the conventional SGD framework.



(a) Separable (clean atoms)

(b) Separable (noisy atoms)

(c) Separable RankBGP (clean atoms)

(d) Separable RankBGP (noisy atoms)

Figure 7.3.: Estimated KL Divergence in the Co-sparse Auto-Encoder setup

7.2.2. Model Generalization

Besides a task oriented evaluation, the reliability of the learned model is also assessed by means of its generalization behavior. For this purpose, the same experimental setup as introduced in Section 5.3.3 is considered. Accordingly, the general ability of the model to capture the distribution of the signals is measured via estimating the divergence between the distributions of the true training signals and signals that have been projected into the orthogonal complement of randomly selected weight vectors from $\boldsymbol{\Theta}_{\mathbf{e}}$. Figures 7.3a and 7.3b illustrate the results for clean and noisy filters. Analogous to the experiments presented in Section 5.3.3, the Rank-BGP algorithm is used to generate the second set of samples. This set can be considered as the closest approximation to the original signals, while at the same

Table 7.1.: Denoising performance evaluated on the test images from Figure 4.7.

	τ	PWC	Barbara	Boats	Lena	Peppers	Avg.
SEP Log	0.25	27.61 0.836	28.09 0.822	29.88 0.797	31.54 0.836	29.86 0.840	29.40 0.826
SEP L1	0.25	27.80 0.858	27.89 0.824	29.65 0.795	31.45 0.847	29.76 0.848	29.31 0.834
SAE	0.35	28.02 0.872	27.36 0.812	29.54 0.792	31.36 0.846	29.71 0.848	29.19 0.834
SAE (linear)	0.35	28.01 0.851	27.52 0.803	29.54 0.785	31.15 0.825	29.62 0.832	29.17 0.820

time, all the samples from the set exhibit a co-sparse representation with respect to the generative operator. The estimated divergence between both signal distributions is shown in Figures 7.3c and 7.3d.

Interestingly, in the second scenario, where the signals are projected into a UoS that is closest to the original training signals, the divergence is almost identical to the results achieved with the conventional learning approach. Thus, the encoder adequately adapts to the signals at hand even within the Auto-Encoder framework. In contrast to this, the estimated divergence of the SAE approach, plotted in Figures 7.3a and 7.3b, significantly deviates from the other results.

7.2.3. Image Denoising

Finally, the encoder matrix as well as the operators are plugged into the sparsity regularizer of the denoising problem as given in Eq. (4.21). Again, the performance is evaluated by means of the PSNR ans MSSIM achieved on the five different test images from Figure 4.7. For the conventional SGD learning method, the robustness to different initializations has been already verified (cf. Section 4.4.3). The same strategy is applied to the *SAE*-based setting. To that end, ten operators whose learning process starts from different random initializations of $\{\boldsymbol{\Theta}_{\mathbf{e},j}\}_{j=1}^{2}$ and $\boldsymbol{\Theta}_{\mathbf{d}}$ are determined. Afterwards, the average denoising performance achieved with these ten different operators as regularizers is considered. For comparison, the same experiment but with a linear activation function is performed. Table 7.1 summarizes the results. The parameter τ that weights the sparsity prior against the data fidelity term in Eq. (4.21), is set to the value that gives the best results with respect to applied analysis operator.

First of all, while being equivalent with respect to the PSNR, it can be seen that in terms of MSSIM, the encoder learned with the Softsign function slightly outperforms the one

Figure 7.4.: Additional set of Test Images. From left to right: Butterfly, Girl, Parrot, Parthenon, Raccoon.

Table 7.2.: Denoising performance evaluated on 5 additional test images, shown in Figure 7.4.

	τ	Butterfly	Girl	Parrot	Parthenon	Raccoon	Avg.
SEP Log	0.25	27.09 0.866	30.39 0.735	30.41 0.846	27.08 0.821	28.53 0.766	28.70 0.807
SEP L1	0.25	26.97 0.874	30.39 0.740	30.36 0.860	26.80 0.813	28.32 0.752	28.57 0.808
SAE	0.35	27.24 0.883	30.36 0.740	30.34 0.859	26.87 0.818	28.38 0.757	28.64 0.812
SAE (linear)	0.35	27.34 0.871	30.20 0.731	30.25 0.838	26.97 0.816	28.48 0.766	28.65 0.805

obtained with a linear activation function. Referring to the same MSSIM measure, the *SAE*-learned separable operator achieves a quality that is equivalent to the performance of conventionally learned operators. However, the average PSNR performance is slightly worse. On closer inspection, the *SAE*-learned model performs better on the synthetically generated *PWC* image, while the recovery of highly textured images like *Barbara* is less accurate compared to the conventional learning approach.

In order to exclude a potential overfitting of the weighting parameter τ to the image set at hand, the same operators as well as the same algorithm with exactly the same parameter set is applied to another collection of images depicted in Figure 7.4. The recovery performance is presented in Table 7.2. Again, the same behavior can be observed. While in terms of PSNR, the linear and Softsign activation function perform equivalently, while being slightly worse compared to the other two approaches, the *SAE*-learned model is competitive with regard to the MSSIM quality.

7.3. Discussion

In this chapter, the applicability of the Sparse Auto-Encoder framework to regularize the co-sparse analysis operator learning problem is explored. First, in order to fulfill the analysis model assumption, the activation function must meet certain requirements. Secondly, trivial solutions due to the scaling ambiguity have to be excluded by means of a weight norm constraint which is in accordance to the conventional learning approach. In the previous chapters it is confirmed that the condition number of the operator plays a crucial role with regard to its suitability to serve as a reliable signal prior. For that reason, another max-norm penalty imposed on the decoder matrix is proposed. Numerical experiments confirm that this strategy actually prevents the condition number to grow unbounded.

The evaluation with respect to the generalization as well as the performance in inverse problem regularization emphasizes that the *SAE* framework indeed allows to learn useful analysis operators. This is confirmed by the results presented in Figures 7.2a and 7.2b as well as based on the model generalization shown in Figures 7.3c and 7.3d. The average image denoising results are on par, especially in terms of the MSSIM measure. However, the results presented in Figures 7.3a and 7.3b illustrate that an explicit penalty on the singular values is beneficial with regard to the generalization of the model. Recall that in this scenario, the chosen indices to identify the orthogonal complement are equally distributed. Consequently, the learned operator in the *SAE* framework exhibits filters which provide responses that are less sparse compared to operators learned with the conventional approach. However, this is not surprising since the original goal of the Auto-Encoder is to find representations that allow to reconstruct the signals. Interestingly, the results concerning the inverse problems and the divergence shown in Figures 7.3c and 7.3d indicate that the filters still provide enough representational power to ensure a reconstruction performance which is close to the one obtained with operators that are learned with explicit penalty functions.

To conclude, solely based on the reconstruction performance there is no clear evidence which one of the approaches performs better. At least the estimated divergence with regard to a randomly selected co-support indicates that the penalty based learning algorithm provides filters whose contributions to the sparse representation are more balanced. Also the straightforward extension to the blind scenario renders the concept introduced in the previous chapters a more versatile approach. On that basis, the conventional learning scheme is slightly more suited to determine a reliable co-sparse analysis model.

However, the *SAE*-based learning approach also offers some other benefits. On the one hand, it was assumed that the Auto-Encoder framework is originally intended to learn signal representations in an unsupervised way. The proposed architecture essentially helps to improve the interpretability of the hidden representation, due to the compliance to the

co-sparse model assumption. During learning, the co-sparse representation could be also readily used for other purposes like object classification and detection. This approach is in line with a task-driven perspective, where one is primarily interested in a reasonable solution for the task at hand, rather than perfect signal reconstruction. Nevertheless, the interpretability is still guaranteed by the model assumption. On the other hand, utilizing a max-norm constraint to control the condition number of the encoder matrix avoids the search for appropriate weighting parameters for additionally required penalty functions. This might be especially useful for deeper architectures where different layers of representations are stacked together. Note that multilayer or hierarchical sparse data models have been already discussed in literature as mentioned in Section 3.1.4.

Chapter 8.

Conclusion

In this thesis, the problem of learning co-sparse analysis operators with separable structures is explored. The benefit of the additional structural constraint is twofold. On the one hand, the lower number of free parameters significantly reduces the complexity of the learning problem. On the other hand, separable filters can be efficiently applied to the signal at hand, which renders the presented approach especially useful for multidimensional data. Regarding the multidimensional scenario, conventional learning algorithms that are based on a vectorization approach are usually hardly applicable due to the exponential increase in parameters. These observations motivate to analyze structured sparse data models, and with that the presented work.

It has been shown that the proposed Stochastic Gradient Descent on manifolds is very well suited to learn separable analysis operators. The new variable step size selection leads to a fast convergence while being robust to parameter changes as well. Furthermore, the used penalty functions allow to flexibly control the properties of the operator. Both theoretical and numerical experiments confirmed the reduced sample complexity in favor of the separable approach. In order to evaluate the suitability of the model, a divergence criterion was used that allows to assess the adaptability to the underlying signal distribution in a task independent manner. It was shown that this measure correlates with the achieved performance of the model in inverse problem regularization.

The applicability of the proposed approach has been further improved by formulating the objective as a blind learning problem. In this scenario, the model is learned from noise corrupted and/or undersampled measurements. This strategy is particularly beneficial if clean training signals are costly to acquire or even not available. The adaptation of the model to the simultaneously reconstructed signal further improves the recovery quality. Furthermore, various noise distributions are easy to handle by simply exchanging the data term, which further emphasizes the universal applicability of the proposed approach.

Although being a general data model, in this thesis, applications in image processing are addressed. In particular the reconstruction of image data from noise corrupted and undersampled measurements is examined. The numerical results indicate that the sepa-

rable structure does not impair the regularization performance of the analysis model in a great extent. Regarding multidimensional signals, the separable model allows to easily incorporate the structural information from all different signal dimensions. Based on a Compressed Sensing problem, where volumetric Magnetic Resonance images are recovered from severely undersampled measurements, the benefit compared to the conventional approach of processing each slice individually is clearly demonstrated.

Last but not least, the concept of co-sparse signal representations has been connected to deep learning approaches, more specifically Sparse Auto-Encoders. To that end, the necessary ingredients and modifications in order to enable the layer-wise mapping to follow the co-sparse analysis model are investigated. In addition to the required constraints on the encoder matrix that prevent the optimization algorithm to obtain trivial solutions, the structure of the decoder matrix is utilized in order to learn a meaningful model. It is shown that separable co-sparse analysis operators can be successfully learned within the Sparse Auto-Encoder framework. This conclusion has been derived by means of several signal recovery experiments as well as based on the model generalization behavior. Besides the achieved interpretability of the representations, the presented approach without additional regularization penalties might be especially useful for hierarchical sparse representations, which constitute one of the key factors that made deep learning successful.

Appendix A.

A.1. Derivation of the Euclidean Gradient

Let $\mathbb{R}^{k\times n}$ be endowed with the scalar product $\langle \boldsymbol{U}, \boldsymbol{V}\rangle = \operatorname{tr}(\boldsymbol{U}^\top \boldsymbol{V})$, and let $f : \mathbb{R}^{k\times n} \to \mathbb{R}$ be a differential function. Denoting $\boldsymbol{H} \in \mathbb{R}^{k\times n}$ as an arbitrary direction, the following relation between the directional derivative and the gradient of the function holds

$$\left.\frac{\partial}{\partial t}\right|_{t=0} f(\boldsymbol{X} + t\boldsymbol{H}) = \langle\, \boldsymbol{H}, \nabla f(\boldsymbol{X})\,\rangle = \operatorname{tr}\Big(\boldsymbol{H}^\top \nabla f(\boldsymbol{X})\Big), \tag{A.1}$$

where $\nabla f(\boldsymbol{X}) \in \mathbb{R}^{k\times n}$ denotes the Euclidean gradient with respect to the matrix $\boldsymbol{X}$.

Sparsity Measure

First, to derive the gradient of the sparsity measure in Eq. (4.2) with respect to each operator separately, the unfolding introduced in Eq. (2.3) is utilized. Given the tensor data $\mathcal{S} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_V}$ the 1-mode unfolding $\boldsymbol{S}_1 \in \mathbb{R}^{I_1\times(\prod_{j\neq V} I_j)}$ is used to derive the gradient with respect to $\boldsymbol{\Omega}_1$. Let $\boldsymbol{e}_j$ denote the canonical basis vector of appropriate size with its 1 entry at the j-th position, with $c = \frac{1}{\log(1+\nu)}$ and $\boldsymbol{\Omega}_U = (\boldsymbol{\Omega}_2 \otimes \cdots \otimes \boldsymbol{\Omega}_{V-1})$ the sparsity measure reads

$$g(\boldsymbol{\Omega}_1) = \sum_i \sum_j c \cdot \log\left(1 + \nu \cdot (\boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i)^2\right).$$

By computing the directional derivative of $g(\boldsymbol{\Omega}_1)$ in the direction of $\boldsymbol{H}$ as

$$\begin{aligned}\left.\frac{\partial}{\partial t}\right|_{t=0} g(\boldsymbol{\Omega}_1 + t\boldsymbol{H}) &= \sum_i \sum_j c \cdot \frac{2\nu \cdot \boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i}{1 + \nu \cdot (\boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i)^2} \cdot \boldsymbol{e}_j^\top \boldsymbol{H} \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i \\ &= \operatorname{tr}\left(\boldsymbol{H} \cdot \sum_i \sum_j c \cdot \frac{2\nu \cdot \boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i}{1 + \nu \cdot (\boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i)^2} \cdot \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i \boldsymbol{e}_j^\top\right),\end{aligned} \tag{A.2}$$

we have the gradient of $g(\boldsymbol{\Omega}_1)$ with respect to the Euclidean metric as

$$\nabla g(\boldsymbol{\Omega}_1) = \sum_i \sum_j c \cdot \frac{2\nu \cdot \boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i}{1 + \nu \cdot (\boldsymbol{e}_j^\top \boldsymbol{\Omega}_1 \boldsymbol{S}_1 \boldsymbol{\Omega}_U^\top \boldsymbol{e}_i)^2} \cdot \boldsymbol{e}_j \boldsymbol{e}_i^\top \boldsymbol{\Omega}_U \boldsymbol{S}_1^\top. \tag{A.3}$$

The Euclidean gradient with respect to the remaining operators $\{\boldsymbol{\Omega}_i\}_{i=2}^V$ can be derived analogously.

Regarding the blind learning approach, the gradient of the sparsity measure with respect to the signal $\boldsymbol{s} \in \mathbb{R}^N$ is required. Although being implemented in a much more efficient way, the gradient $\nabla_{\boldsymbol{s}} g(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_V, \boldsymbol{s}) := \nabla g(\boldsymbol{s})$ can be easily derived by considering the single matrix $\widetilde{\boldsymbol{\Omega}} \in \mathbb{R}^{Q \times N}$ that represents the application of all filters to all overlapping patches in $\boldsymbol{s}$. Hence, the gradient reads

$$\nabla g(\boldsymbol{s}) = \sum_i c \cdot \frac{2\nu \cdot \boldsymbol{e}_i^\top \widetilde{\boldsymbol{\Omega}} \boldsymbol{s}}{1 + \nu \cdot (\boldsymbol{e}_i^\top \widetilde{\boldsymbol{\Omega}} \boldsymbol{s})^2} \cdot \widetilde{\boldsymbol{\Omega}}^\top \boldsymbol{e}_i. \tag{A.4}$$

Full Rank Constraint

The function introduced in Section 4.2.2 is applied to each operator separately, which significantly simplifies the derivation of the gradient. The penalty for the matrix $\boldsymbol{\Omega} \in \mathbb{R}^{k \times n}$ with $k \geq n$ is of the form

$$r(\boldsymbol{\Omega}) = -\log\det(\boldsymbol{\Omega}^\top \boldsymbol{\Omega}).$$

Again, the directional derivative of $r(\boldsymbol{\Omega})$ in the direction $\boldsymbol{H}$ reads

$$\begin{aligned} \left.\frac{\partial}{\partial t}\right|_{t=0} r(\boldsymbol{\Omega} + t\boldsymbol{H}) &= -\operatorname{tr}\left((\boldsymbol{\Omega}^\top \boldsymbol{\Omega})^{-1} \boldsymbol{\Omega}^\top \boldsymbol{H} + (\boldsymbol{\Omega}^\top \boldsymbol{\Omega})^{-1} \boldsymbol{H}^\top \boldsymbol{\Omega}\right) \\ &= -\operatorname{tr}(2\boldsymbol{\Omega}(\boldsymbol{\Omega}^\top \boldsymbol{\Omega})^{-1} \boldsymbol{H}^\top), \end{aligned} \tag{A.5}$$

which eventually results in the gradient of $r(\boldsymbol{\Omega})$ given as

$$\nabla r(\boldsymbol{\Omega}) = -2\boldsymbol{\Omega}(\boldsymbol{\Omega}^\top \boldsymbol{\Omega})^{-1} \tag{A.6}$$

Coherence Penalty

Analogous to the rank penalty, the coherence penalty given in (4.7) is also applied individually to each operator. With the penalty function

$$h(\boldsymbol{\Omega}) = -\tfrac{1}{2}\sum_{j\neq l}\log(1-(\boldsymbol{\omega}_j^\top\boldsymbol{\omega}_l)^2),$$

at hand, the directional derivative is computed as

$$\begin{aligned}\frac{\partial}{\partial t}\bigg|_{t=0} h(\boldsymbol{\Omega}+t\boldsymbol{H}) &= \sum_{j\neq l}\frac{\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l}{1-(\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l)^2}\cdot\Big(\operatorname{tr}(\boldsymbol{e}_j^\top\boldsymbol{H}\boldsymbol{\Omega}^\top\boldsymbol{e}_l)+\operatorname{tr}(\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{H}^\top\boldsymbol{e}_l)\Big)\\ &= \sum_{j\neq l}\frac{\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l}{1-(\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l)^2}\cdot\operatorname{tr}\Big((\boldsymbol{e}_j\boldsymbol{e}_l^\top+\boldsymbol{e}_l\boldsymbol{e}_j^\top)\boldsymbol{\Omega}\boldsymbol{H}^\top\Big).\end{aligned} \tag{A.7}$$

The gradient of $h(\boldsymbol{\Omega})$ with respect to the Euclidean metric reads

$$\nabla h(\boldsymbol{\Omega}) = \sum_{j\neq l}\frac{\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l}{1-(\boldsymbol{e}_j^\top\boldsymbol{\Omega}\boldsymbol{\Omega}^\top\boldsymbol{e}_l)^2}\cdot(\boldsymbol{e}_j\boldsymbol{e}_l^\top+\boldsymbol{e}_l\boldsymbol{e}_j^\top)\boldsymbol{\Omega}. \tag{A.8}$$

Gradient of the Data Fidelity Terms

Lastly, the blind reconstruction approach that includes different data terms requires access to the image gradient $\nabla_{\boldsymbol{s}} d(\boldsymbol{s}, \boldsymbol{y}) := \nabla d(\boldsymbol{s})$. In the following, the Euclidean gradients with regard to the fidelity terms that are used to account for the various noise distributions are given.

Additive White Gaussian Noise: Given the function $d_{\text{additive}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M}\|\boldsymbol{y} - \boldsymbol{\Phi s}\|_2^2$, the gradient reads

$$\nabla d(\boldsymbol{s}) = \frac{2}{M} \boldsymbol{\Phi}^\top (\boldsymbol{\Phi s} - \boldsymbol{y}). \tag{A.9}$$

Impulsive Noise: In the presence of Salt-and-Pepper noise, the following data term it used $d_{\text{impulsive}}(\boldsymbol{s}, \boldsymbol{y}) = \frac{1}{M} \sum_i \frac{1}{\log(1+c)} \log\left(1 + c \cdot (\boldsymbol{e}_i^\top (\boldsymbol{\Phi s} - \boldsymbol{y}))^2\right)$. Hence, the gradient reads

$$\nabla d(\boldsymbol{s}) = \frac{1}{M} \sum_i \frac{1}{\log(1+c)} \cdot \frac{2c \cdot \boldsymbol{e}_i^\top (\boldsymbol{\Phi s} - \boldsymbol{y})}{1 + c \cdot (\boldsymbol{e}_i^\top (\boldsymbol{\Phi s} - \boldsymbol{y}))^2} \cdot \boldsymbol{\Phi}^\top \boldsymbol{e}_i. \tag{A.10}$$

Multiplicative Noise: The data term that accounts for multiplicative noise reads $d_{\text{mult}}(\boldsymbol{u}, \boldsymbol{y}) = \frac{1}{M} \sum_i \left(\boldsymbol{e}_i^\top \boldsymbol{u} + \boldsymbol{e}_i^\top \boldsymbol{y} \cdot \exp(-\boldsymbol{e}_i^\top \boldsymbol{u})\right)$, where the objective is optimized with respect to the log-image, i.e., $\boldsymbol{e}_i^\top \boldsymbol{u} = \log(\boldsymbol{e}_i^\top \boldsymbol{s})$. Consequently, the Euclidean gradient is given as

$$\nabla d(\boldsymbol{u}) = \frac{1}{M} \sum_i \left(1 - \boldsymbol{e}_i^\top \boldsymbol{y} \cdot \exp(-\boldsymbol{e}_i^\top \boldsymbol{u})\right) \cdot \boldsymbol{e}_i. \tag{A.11}$$

Bibliography

1. P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

2. M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. In *IEEE Transactions on Signal Processing*, 54(11), pp. 4311–4322, Nov 2006. ISSN 1053-587X. doi:10.1109/TSP.2006.881199.

3. M. Aharon and M. Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. In *SIAM J. Img. Sci.*, 1(3), pp. 228–247, July 2008. ISSN 1936-4954. doi:10.1137/07070156X.

4. D. Arpit, Y. Zhou, H. Ngo, and V. Govindaraju. Why regularized auto-encoders learn sparse representation? In M.F. Balcan and K.Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 136–144. PMLR, New York, New York, USA, 20–22 Jun 2016.

5. G. Aubert and J.F. Aujol. A variational approach to removing multiplicative noise. In *SIAM Journal on Applied Mathematics*, 68(4), pp. 925–946, 2008. doi:10.1137/060671814.

6. P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. In *Neural Netw.*, 2(1), pp. 53–58, January 1989. ISSN 0893-6080. doi:10.1016/0893-6080(89)90014-2.

7. S. Becker, J. Bobin, and E.J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. In *SIAM J. Imaging Sci.*, 4(1), pp. 1–39, 2011. doi:10.1137/090756855.

8. J.M. Bioucas-Dias and M.A.T. Figueiredo. Multiplicative noise removal using variable splitting and constrained optimization. In *IEEE Transactions on Image Processing*, 19(7), pp. 1720–1730, July 2010. ISSN 1057-7149. doi:10.1109/TIP.2010.2045029.

9. J. Bobin, J.L. Starck, Y. Moudden, and J.M. Fadili. Blind Source Separation: the Sparsity Revolution. In e. Peter Hawkes (ed.), *Advances in Imaging and Electron Physics*, volume 152, pp. 221–306. Academic Press, Elsevier, 2008.

10. S. Bonnabel. Stochastic gradient descent on riemannian manifolds. In *IEEE Trans. Autom. Control*, 58(9), pp. 2217–2229, 2013. ISSN 0018-9286. doi:10.1109/TAC.2013.2254619.

11. L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proc. Int. Conf. Computational Statistics*, pp. 177–187. 2010.

12. L. Bottou. Stochastic gradient tricks. In *Neural Networks, Tricks of the Trade, Reloaded*, Lecture Notes in Computer Science (LNCS 7700), pp. 430–445. Springer, 2012.

13. L. Bottou and Y. LeCun. Large scale online learning. In *Adv. Neural Information Processing Systems*, pp. 217–224. 2004.

14. H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. In *Biol. Cybern.*, 59(4-5), pp. 291–294, September 1988. ISSN 0340-1200. doi:10.1007/BF00332918.

15. A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. In *SIAM Rev.*, 51(1), pp. 34–81, February 2009. ISSN 0036-1445. doi:10.1137/060657704.

16. J.F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 104–111. June 2009. ISSN 1063-6919. doi:10.1109/CVPR.2009.5206743.

17. C.F. Caiafa and A. Cichocki. Computing sparse representations of multidimensional signals using kronecker bases. In *Neural Computation*, 25(1), pp. 186–220, 2013. doi:10.1162/NECO_a_00385. PMID: 23020110.

18. E.J. Candès and Y. Plan. Near-ideal model selection by l1 minimization. In *Annals of Statistics*, 37, pp. 2145–2177, 2009.

19. E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. In *IEEE Transactions on Information Theory*, 52(2), pp. 489–509, Feb 2006. ISSN 0018-9448. doi:10.1109/TIT.2005.862083.

20. E.J. Candès and M.B. Wakin. An introduction to compressive sampling. In *IEEE Signal Processing Magazine*, 25(2), pp. 21–30, March 2008. ISSN 1053-5888. doi:10.1109/MSP.2007.914731.

21. E.J. Candès and D.L. Donoho. *Curvelets - A Surprisingly Effective Nonadaptive Representation For Objects with Edges*. 1999.

22. E.J. Candès, J.K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. In *Comm. Pure Appl. Math.*, 59(8), pp. 1207–1223, August 2006. ISSN 0010-3640. doi:10.1002/cpa.20124.

23. R.E. Carrillo, K.E. Barner, and T.C. Aysal. Robust sampling and reconstruction methods for sparse signals in the presence of impulsive noise. In *IEEE Journal of Selected Topics in Signal Processing*, 4(2), pp. 392–408, April 2010. ISSN 1932-4553. doi: 10.1109/JSTSP.2009.2039177.

24. O. Chabiron, F. Malgouyres, J.Y. Tourneret, and N. Dobigeon. Toward fast transform learning. In *International Journal of Computer Vision*, 114(2), pp. 195–216, 2015. ISSN 1573-1405. doi:10.1007/s11263-014-0771-z.

25. R. Chartrand. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 262–265. June 2009. ISSN 1945-7928. doi: 10.1109/ISBI.2009.5193034.

26. G.H. Chen, J. Tang, and S. Leng. Prior image constrained compressed sensing (PICCS): A method to accurately reconstruct dynamic CT images from highly undersampled projection data sets. In *Medical Physics*, 35(2), pp. 660–663, 2008. doi: http://dx.doi.org/10.1118/1.2836423.

27. S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. In *SIAM Journal on Scientific Computing*, 20(1), pp. 33–61, 1998. doi:10.1137/S1064827596304010.

28. Y. Chen, R. Ranftl, and T. Pock. Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. In *IEEE Transactions on Image Processing*, 23(3), pp. 1060–1072, March 2014. ISSN 1057-7149. doi:10.1109/TIP.2014.2299065.

29. K. Choi, J. Wang, L. Zhu, T.S. Suh, S. Boyd, and L. Xing. Compressed sensing based cone-beam computed tomography reconstruction with a first-order method. In *Medical Physics*, 37(9), pp. 5113–5125, 2010.

30. A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H.A. PHAN. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. In *IEEE Signal Processing Magazine*, 32(2), pp. 145–163, March 2015. ISSN 1053-5888. doi:10.1109/MSP.2013.2297439.

31. C.A. Cocosco, V. Kollokian, R.K.S. Kwan, G.B. Pike, and A.C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*, 5, p. 425, 1997.

32. D.L. Collins, A.P. Zijdenbos, V. Kollokian, J.G. Sled, N.J. Kabani, C.J. Holmes, and A.C. Evans. Design and construction of a realistic digital brain phantom. In *IEEE Transactions on Medical Imaging*, 17(3), pp. 463–468, June 1998. ISSN 0278-0062. doi: 10.1109/42.712135.

33. Y. Dai and Y. Yuan. An efficient hybrid conjugate gradient method for unconstrained optimization. In *Annals of Operations Research*, 103(1), pp. 33–47, Mar 2001. ISSN 1572-9338. doi:10.1023/A:1012930416777.

34. C.F. Dantas, M.N. da Costa, and R. d. R. Lopes. Learning dictionaries as a sum of kronecker products. In *IEEE Signal Processing Letters*, 24(5), pp. 559–563, May 2017. ISSN 1070-9908. doi:10.1109/LSP.2017.2681159.

35. J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. In *Vision Research*, 20(10), pp. 847 – 856, 1980. ISSN 0042-6989. doi:http://dx.doi.org/10.1016/0042-6989(80)90065-6.

36. A. Dieuleveut, A. Durmus, and F. Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains. arXiv:1707.06386, 2017.

37. M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. In *IEEE Transactions on Image Processing*, 14(12), pp. 2091–2106, Dec 2005. ISSN 1057-7149. doi:10.1109/TIP.2005.859376.

38. J. Dong, W. Wang, and W. Dai. Analysis simco: A new algorithm for analysis dictionary learning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7193–7197. May 2014. ISSN 1520-6149. doi: 10.1109/ICASSP.2014.6854996.

39. J. Dong, W. Wang, W. Dai, M.D. Plumbley, Z.F. Han, and J. Chambers. Analysis simco algorithms for sparse analysis model based dictionary learning. In *IEEE Transactions on Signal Processing*, 64(2), pp. 417–431, Jan 2016. ISSN 1053-587X. doi:10.1109/TSP.2015.2483480.

40. J. Dong, Z. Han, Y. Zhao, W. Wang, A. Prochazka, and J. Chambers. Sparse analysis model based multiplicative noise removal with enhanced regularization. In *Signal Processing*, 137, pp. 160 – 176, 2017. ISSN 0165-1684. doi:https://doi.org/10.1016/j.sigpro.2017.01.032.

41. D.L. Donoho. De-noising by soft-thresholding. In *IEEE Transactions on Information Theory*, 41(3), pp. 613–627, May 1995. ISSN 0018-9448. doi:10.1109/18.382009.

42. D.L. Donoho. Compressed sensing. In *IEEE Transactions on Information Theory*, 52(4), pp. 1289–1306, April 2006. ISSN 0018-9448. doi:10.1109/TIT.2006.871582.

43. D.L. Donoho. Superresolution via sparsity constraints. In *SIAM J. Math. Anal.*, 23(5), pp. 1309–1331, September 1992. ISSN 0036-1410. doi:10.1137/0523074.

44. D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. In *Biometrika*, 81, pp. 425–455, 1994.

45. G. E. Hinton and T. Shallice. Lesioning an attractor network: Investigations of acquired dyslexia. In *Psychological review*, 98, pp. 74–95, 02 1991.

46. M. Elad. Sparse and redundant representation modeling, what next? In *IEEE Signal Processing Letters*, 19(12), pp. 922–928, Dec 2012. ISSN 1070-9908. doi:10.1109/LSP.2012.2224655.

47. M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. In *IEEE Transactions on Image Processing*, 15(12), pp. 3736–3745, 2006.

48. M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. In *Inverse Problems*, 23(3), p. 947, 2007.

49. K. Engan, S.O. Aase, and J.H. Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 5, pp. 2443–2446 vol.5. 1999. ISSN 1520-6149. doi:10.1109/ICASSP.1999.760624.

50. M. Fadili, J.L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. In *Comput. J.*, 52(1), pp. 64–79, January 2009. ISSN 0010-4620. doi:10.1093/comjnl/bxm055.

51. D.J. Field. *Scale-invariance and Self-similar 'Wavelet' Transforms: an Analysis of Natural Scenes and Mammalian Visual Systems*. Oxford University Press, 1993.

52. D.J. Field. What is the goal of sensory coding? In *Neural Comput.*, 6(4), pp. 559–601, July 1994. ISSN 0899-7667. doi:10.1162/neco.1994.6.4.559.

53. M.A.T. Figueiredo and R.D. Nowak. Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior. In *IEEE Transactions on Image Processing*, 10(9), pp. 1322–1331, Sep 2001. ISSN 1057-7149. doi:10.1109/83.941856.

54. J.J. Fuchs. On sparse representations in arbitrary redundant bases. In *IEEE Transactions on Information Theory*, 50(6), pp. 1341–1344, June 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.828141.

55. R. Giryes and M. Elad. Cosamp and SP for the cosparse analysis model. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 964–968. Aug 2012. ISSN 2219-5491.

56. R. Giryes and M. Elad. Sparsity-based poisson denoising with dictionary learning. In *IEEE Transactions on Image Processing*, 23(12), pp. 5057–5069, Dec 2014. ISSN 1057-7149. doi:10.1109/TIP.2014.2362057.

57. R. Giryes, S. Nam, M. Elad, R. Gribonval, and M. Davies. Greedy-like algorithms for the cosparse analysis model. In *Linear Algebra and its Applications*, 441, pp. 22 – 60, 2014. ISSN 0024-3795. doi:http://dx.doi.org/10.1016/j.laa.2013.03.004.

58. R. Giryes, S. Nam, R. Gribonval, and M.E. Davies. Iterative cosparse projection algorithms for the recovery of cosparse vectors. In *2011 19th European Signal Processing Conference*, pp. 1460–1464. Aug 2011. ISSN 2076-1465.

59. S. Gleichman and Y.C. Eldar. Blind compressed sensing. In *IEEE Transactions on Information Theory*, 57(10), pp. 6958–6975, 2011.

60. K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 399–406. Omnipress, USA, 2010. ISBN 978-1-60558-907-7.

61. R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert. Sample complexity of dictionary learning and other matrix factorizations. In *IEEE Trans. Inf. Theory*, 61(6), pp. 3469–3486, 2015. ISSN 0018-9448. doi:10.1109/TIT.2015.2424238.

62. R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst. Atoms of all channels, unite! Average case analysis of multi-channel sparse recovery using greedy algorithms. In *The Journal of Fourier Analysis and Applications*, 14(5), pp. 655–687, 2008. doi:10.1007/s00041-008-9044-y.

63. T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015. ISBN 1498712169, 9781498712163.

64. S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis operator learning and its application to image reconstruction. In *IEEE Transactions on Image Processing*, 22(6), pp. 2138–2150, June 2013. ISSN 1057-7149. doi:10.1109/TIP.2013.2246175.

65. S. Hawe, M. Seibert, and M. Kleinsteuber. Separable dictionary learning. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438–445. June 2013. ISSN 1063-6919. doi:10.1109/CVPR.2013.63.

66. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. June 2016. doi:10.1109/CVPR.2016.90.

67. G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science (New York, N.Y.)*, 313, pp. 504–7, 08 2006. doi:10.1126/science.1127647.

68. S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural Comput.*, 9(8), pp. 1735–1780, November 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

69. E. Hoffer, R. Banner, I. Golan, and D. Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. In *CoRR*, abs/1803.01814, 2018.

70. L. Huang, X. Liu, B. Lang, and B. Li. Projection based weight normalization for deep neural networks. In *CoRR*, abs/1710.02338, 2017.

71. Y.M. Huang, M.K. Ng, and Y.W. Wen. A new total variation method for multiplicative noise removal. In *SIAM Journal on Imaging Sciences*, 2(1), pp. 20–40, 2009. doi:10.1137/080712593.

72. P.J. Huber. Robust estimation of a location parameter. In *Annals of Statistics*, 53(1), pp. 73–101, 1964.

73. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015.

74. J. Karhunen, T. Raiko, and K. Cho. Chapter 7 - unsupervised deep learning: A short review. In E. Bingham, S. Kaski, J. Laaksonen, and J. Lampinen (eds.), *Advances in Independent Component Analysis and Learning Machines*, pp. 125 – 142. Academic Press, 2015. ISBN 978-0-12-802806-3. doi:https://doi.org/10.1016/B978-0-12-802806-3.00007-5.

75. K. Kavukcuoglu, M. Ranzato, and Y. LeCun. *Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition*. Technical Report CBLL-TR-2008-12-01, Computational and Biological Learning Lab, Courant Institute, NYU, 2008.

76. T.G. Kolda and B.W. Bader. Tensor decompositions and applications. In *SIAM Rev.*, 51(3), pp. 455–500, August 2009. ISSN 0036-1445. doi:10.1137/07070111X.

77. M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. In *AIChE Journal*, 37(2), pp. 233–243. doi:10.1002/aic.690370209.

78. H.W. Kuhn. The Hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, 2(1–2), pp. 83–97, 1955. ISSN 1931-9193. doi:10.1002/nav.3800020109.

79. R.K.S. Kwan, A.C. Evans, and G.B. Pike. MRI simulation-based evaluation of image-processing and classification methods. In *IEEE Transactions on Medical Imaging*, 18(11), pp. 1085–1097, Nov 1999. ISSN 0278-0062. doi:10.1109/42.816072.

80. H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. In *J. Mach. Learn. Res.*, 10, pp. 1–40, June 2009. ISSN 1532-4435.

81. N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Adv. Neural Information Processing Systems*, pp. 2663–2671. 2012.

82. Y. Lecun. *Generalization and network design strategies*. Elsevier, 1989.

83. H. Lee, C. Ekanadham, and A.Y. Ng. Sparse deep belief net model for visual area v2. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 873–880. Curran Associates, Inc., 2008.

84. T.S. Lee. Image representation using 2d gabor wavelets. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10), pp. 959–971, October 1996. ISSN 0162-8828. doi:10.1109/34.541406.

85. Y.M. Lu and M.N. Do. A theory for sampling signals from a union of subspaces. In *IEEE Transactions on Signal Processing*, 56(6), pp. 2334–2345, June 2008. ISSN 1053-587X. doi:10.1109/TSP.2007.914346.

86. M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing MRI. In *IEEE Signal Processing Magazine*, 25(2), pp. 72–82, March 2008. ISSN 1053-5888. doi:10.1109/MSP.2007.914728.

87. J.M.P. M. Lustig, D. Donoho. Sparse MRI: The application of compressed sensing for rapid mr imaging. In *Magnetic Resonance in Medicine*, 58(6), pp. 1182–1195, 2007.

88. L.L. Magoarou and R. Gribonval. Flexible multilayer sparse approximations of matrices and applications. In *IEEE Journal of Selected Topics in Signal Processing*, 10(4), pp. 688–700, June 2016. ISSN 1932-4553. doi:10.1109/JSTSP.2016.2543461.

89. J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), pp. 791–804, April 2012. ISSN 0162-8828. doi:10.1109/TPAMI.2011.156.

90. J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. In *IEEE Transactions on Image Processing*, 17(1), pp. 53–69, Jan 2008. ISSN 1057-7149. doi:10.1109/TIP.2007.911828.

91. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 689–696. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-516-1. doi:10.1145/1553374.1553463.

92. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. In *J. Mach. Learn. Res.*, 11, pp. 19–60, 2010.

93. A. Makhzani and B.J. Frey. k-Sparse autoencoders. In *CoRR*, abs/1312.5663, 2013.

94. S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), pp. 674–693, Jul 1989. ISSN 0162-8828. doi:10.1109/34.192463.

95. S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. In *IEEE Transactions on Signal Processing*, 41(12), pp. 3397–3415, Dec 1993. ISSN 1053-587X. doi:10.1109/78.258082.

96. S. Marĉelja. Mathematical description of the responses of simple cortical cells. In *J. Opt. Soc. Am.*, 70(11), pp. 1297–1300, Nov 1980. doi:10.1364/JOSA.70.001297.

97. A. Maurer and M. Pontil. K-dimensional coding schemes in Hilbert spaces. In *IEEE Trans. Inf. Theory*, 56(11), pp. 5839–5846, 2010.

98. M.K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. In *IEEE Signal Processing Letters*, 6(12), pp. 300–303, Dec 1999. ISSN 1070-9908. doi:10.1109/97.803428.

99. V. Nair and G.E. Hinton. 3D object recognition with deep belief nets. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems 22*, pp. 1339–1347. Curran Associates, Inc., 2009.

100. S. Nam, M.E. Davies, M. Elad, and R. Gribonval. Recovery of cosparse signals with greedy analysis pursuit in the presence of noise. In *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 361–364. Dec 2011. doi:10.1109/CAMSAP.2011.6136026.

101. S. Nam, M.E. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. In *Appl. Comput. Harmon. Anal.*, 34(1), pp. 30–56, 2013.

102. A. Ng. Sparse autoencoder. In *CS294A Lecture notes*, 72, 2011.

103. J. Nocedal and S.J. Wright. *Numerical Optimization*. 2nd edition. Springer, New York, NY, USA, 2006.

104. B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. In *Nature*, 381, pp. 607–609, 1996.

105. B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? In *Vision Research*, 37(23), pp. 3311 – 3325, 1997. ISSN 0042-6989. doi:http://dx.doi.org/10.1016/S0042-6989(97)00169-7.

106. B. Ophir, M. Elad, N. Bertin, and M.D. Plumbley. Sequential minimal eigenvalues - an approach to analysis dictionary learning. In *2011 19th European Signal Processing Conference*, pp. 1465–1469. Aug 2011. ISSN 2076-1465.

107. B. Ophir, M. Lustig, and M. Elad. Multi-scale dictionary learning using wavelets. In *IEEE Journal of Selected Topics in Signal Processing*, 5(5), pp. 1014–1024, Sept 2011. ISSN 1932-4553. doi:10.1109/JSTSP.2011.2155032.

108. T. Peleg and M. Elad. Performance guarantees of the thresholding algorithm for the cosparse analysis model. In *IEEE Transactions on Information Theory*, 59(3), pp. 1832–1845, March 2013. ISSN 0018-9448. doi:10.1109/TIT.2012.2226924.

109. E.L. Pennec and S. Mallat. Sparse geometric image representations with bandelets. In *IEEE Transactions on Image Processing*, 14(4), pp. 423–438, April 2005. ISSN 1057-7149. doi:10.1109/TIP.2005.843753.

110. G. Peyré and J.M. Fadili. Learning Analysis Sparsity Priors. In *Sampta'11*, p. 4 pp. Singapour, Singapore, 2011.

111. L. Pfister and Y. Bresler. Tomographic reconstruction with adaptive sparsifying transforms. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6914–6918. May 2014. ISSN 1520-6149. doi:10.1109/ICASSP.2014.6854940.

112. J. Provost and F. Lesage. The application of compressed sensing for photo-acoustic tomography. In *Medical Imaging, IEEE Transactions on*, 28(4), pp. 585–594, 2009.

113. N. Qi, y. Shi, X. Sun, J. Wang, B. Yin, and J. Gao. Multi-dimensional sparse models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), pp. 1–1, 2017. ISSN 0162-8828. doi:10.1109/TPAMI.2017.2663423.

114. N. Qi, Y. Shi, X. Sun, J. Wang, and W. Ding. Two dimensional analysis sparse model. In *2013 IEEE International Conference on Image Processing*, pp. 310–314. Sept 2013. ISSN 1522-4880. doi:10.1109/ICIP.2013.6738064.

115. N. Qi, Y. Shi, X. Sun, and B. Yin. TenSR: Multi-dimensional tensor sparse representation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5916–5925. June 2016. doi:10.1109/CVPR.2016.637.

116. N. Qi, Y. Shi, X. Sun, J. Wang, and B. Yin. Two dimensional synthesis sparse model. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. July 2013. ISSN 1945-7871. doi:10.1109/ICME.2013.6607508.

117. X. Qu, D. Guo, B. Ning, Y. Hou, Y. Lin, S. Cai, and Z. Chen. Undersampled MRI reconstruction with patch-based directional wavelets. In *Magnetic Resonance Imaging*, 30(7), pp. 964 – 977, 2012. ISSN 0730-725X. doi:https://doi.org/10.1016/j.mri.2012.02.019.

118. C. Quinsac, A. Basarab, D. Kouame, and J.M. Gregoire. 3D compressed sensing ultrasound imaging. In *Ultrasonics Symposium (IUS), 2010 IEEE*, pp. 363–366. 2010.

119. A. Rangamani, A. Mukherjee, A. Arora, T. Ganapathy, A. Basu, S.P. Chin, and T.D. Tran. Critical points of an autoencoder can provably recover sparsely used overcomplete dictionaries. In *CoRR*, abs/1708.03735, 2017.

120. M.A. Ranzato, C. Poultney, S. Chopra, and Y.L. Cun. Efficient learning of sparse representations with an energy-based model. In B. Schölkopf, J.C. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems 19*, pp. 1137–1144. MIT Press, 2007.

121. H. Rauhut, K. Schnass, and P. Vandergheynst. Compressed sensing and redundant dictionaries. In *IEEE Transactions on Information Theory*, 54(5), pp. 2210–2219, May 2008. ISSN 0018-9448. doi:10.1109/TIT.2008.920190.

122. S. Ravishankar and Y. Bresler. MR image reconstruction from highly undersampled k-space data by dictionary learning. In *IEEE Transactions on Medical Imaging*, 30(5), pp. 1028–1041, May 2011. ISSN 0278-0062. doi:10.1109/TMI.2010.2090538.

123. S. Ravishankar and Y. Bresler. Learning sparsifying transforms for image processing. In *2012 19th IEEE International Conference on Image Processing*, pp. 681–684. Sept 2012. ISSN 1522-4880. doi:10.1109/ICIP.2012.6466951.

124. S. Ravishankar and Y. Bresler. Learning doubly sparse transforms for images. In *IEEE Transactions on Image Processing*, 22(12), pp. 4598–4612, Dec 2013. ISSN 1057-7149. doi:10.1109/TIP.2013.2274384.

125. S. Ravishankar and Y. Bresler. Learning overcomplete sparsifying transforms for signal processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3088–3092. May 2013. ISSN 1520-6149. doi:10.1109/ICASSP.2013.6638226.

126. S. Ravishankar and Y. Bresler. Learning sparsifying transforms. In *IEEE Transactions on Signal Processing*, 61(5), pp. 1072–1086, March 2013. ISSN 1053-587X. doi:10.1109/TSP.2012.2226449.

127. S. Ravishankar and Y. Bresler. Sparsifying transform learning for compressed sensing MRI. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pp. 17–20. April 2013. ISSN 1945-7928. doi:10.1109/ISBI.2013.6556401.

128. S. Ravishankar and Y. Bresler. Online sparsifying transform learning Part II: Convergence analysis. In *IEEE Journal of Selected Topics in Signal Processing*, 9(4), pp. 637–646, June 2015. ISSN 1932-4553. doi:10.1109/JSTSP.2015.2407860.

129. S. Ravishankar and Y. Bresler. Data-driven learning of a union of sparsifying transforms model for blind compressed sensing. In *IEEE Transactions on Computational Imaging*, 2(3), pp. 294–309, Sept 2016. ISSN 2333-9403. doi:10.1109/TCI.2016.2567299.

130. S. Ravishankar, B. Wen, and Y. Bresler. Online sparsifying transform learning for signal processing. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 364–368. Dec 2014. doi:10.1109/GlobalSIP.2014.7032140.

131. S. Ravishankar, B. Wen, and Y. Bresler. Online sparsifying transform learning Part I: Algorithms. In *IEEE Journal of Selected Topics in Signal Processing*, 9(4), pp. 625–636, June 2015. ISSN 1932-4553. doi:10.1109/JSTSP.2015.2417131.

132. S. Ravishankar and Y. Bresler. Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging. In *SIAM Journal on Imaging Sciences*, 8(4), pp. 2519–2557, 2015. doi:10.1137/141002293.

133. S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 833–840. Omnipress, USA, 2011. ISBN 978-1-4503-0619-5.

134. R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua. Learning separable filters. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2754–2761. June 2013. ISSN 1063-6919. doi:10.1109/CVPR.2013.355.

135. H. Robbins and S. Monro. A stochastic approximation method. In *Ann. Math. Statist.*, 22(3), pp. 400–407, 1951. doi:10.1214/aoms/1177729586.

136. F. Roemer, G.D. Galdo, and M. Haardt. Tensor-based algorithms for learning multidimensional separable dictionaries. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3963–3967. May 2014. ISSN 1520-6149. doi: 10.1109/ICASSP.2014.6854345.

137. S. Roth and M.J. Black. Fields of experts: a framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 860–867 vol. 2. June 2005. ISSN 1063-6919. doi:10.1109/CVPR.2005.160.

138. R. Rubinstein, A.M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. In *Proceedings of the IEEE*, 98(6), pp. 1045–1057, June 2010. ISSN 0018-9219. doi:10.1109/JPROC.2010.2040551.

139. R. Rubinstein and M. Elad. Dictionary learning for analysis-synthesis thresholding. In *IEEE Transactions on Signal Processing*, 62(22), pp. 5962–5972, Nov 2014. ISSN 1053-587X. doi:10.1109/TSP.2014.2360157.

140. R. Rubinstein, T. Faktor, and M. Elad. K-SVD dictionary-learning for the analysis sparse model. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5405–5408. March 2012. ISSN 1520-6149. doi:10.1109/ICASSP.2012.6289143.

141. R. Rubinstein, T. Peleg, and M. Elad. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. In *IEEE Transactions on Signal Processing*, 61(3), pp. 661–677, Feb 2013. ISSN 1053-587X. doi:10.1109/TSP.2012.2226445.

142. R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. In *IEEE Transactions on Signal Processing*, 58(3), pp. 1553–1564, March 2010. ISSN 1053-587X. doi:10.1109/TSP.2009.2036477.

143. L. Rudin, P.L. Lions, and S. Osher. *Multiplicative Denoising and Deblurring: Theory and Algorithms*, pp. 103–119. Springer New York, New York, NY, 2003. ISBN 978-0-387-21810-6. doi:10.1007/0-387-21810-6_6.

144. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. In *Physica D: Nonlinear Phenomena*, 60(1), pp. 259 – 268, 1992. ISSN 0167-2789. doi:http://dx.doi.org/10.1016/0167-2789(92)90242-F.

145. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. In *Nature*, 323(6088), pp. 533–536, October 1986.

146. T. Salimans and D.P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 901–909. Curran Associates, Inc., 2016.

147. M. Seibert, J. Wörmann, R. Gribonval, and M. Kleinsteuber. Learning co-sparse analysis operators with separable structures. In *IEEE Transactions on Signal Processing*, 64(1), pp. 120–130, Jan 2016. ISSN 1053-587X. doi:10.1109/TSP.2015.2481875.

148. J. Silva, M. Chen, Y.C. Eldar, G. Sapiro, and L. Carin. Blind compressed sensing over a structured union of subspaces. arXiv:1103.2469v1, 2011.

149. A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua. Learning separable filters. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1), pp. 94–106, Jan 2015. ISSN 0162-8828. doi:10.1109/TPAMI.2014.2343229.

150. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, 15, pp. 1929–1958, 2014.

151. J.L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. In *IEEE Transactions on Image Processing*, 14(10), pp. 1570–1582, Oct 2005. ISSN 1057-7149. doi:10.1109/TIP.2005.852206.

152. J.L. Starck, E.J. Candès, and D.L. Donoho. The curvelet transform for image denoising. In *IEEE Transactions on Image Processing*, 11(6), pp. 670–684, Jun 2002. ISSN 1057-7149. doi:10.1109/TIP.2002.1014998.

153. J.L. Starck, F. Murtagii, and A. Bijaoui. Multiresolution support applied to image filtering and restoration. In *Graphical Models and Image Processing*, 57(5), pp. 420 – 431, 1995. ISSN 1077-3169. doi:http://dx.doi.org/10.1006/gmip.1995.1036.

154. C. Studer and R. Baraniuk. Dictionary learning from sparsely corrupted or compressed signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012.

155. J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad. Trainlets: Dictionary learning in high dimensions. In *IEEE Transactions on Signal Processing*, 64(12), pp. 3180–3193, June 2016. ISSN 1053-587X. doi:10.1109/TSP.2016.2540599.

156. D.S. Taubman and M.W. Marcellin. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Norwell, MA, USA, 2001. ISBN 079237519X.

157. R. Tibshirani. Regression shrinkage and selection via the lasso. In *Journal of the Royal Statistical Society, Series B*, 58, pp. 267–288, 1996.

158. A.N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, 4, pp. 1035–1038, 1963.

159. I. Tosic and P. Frossard. Dictionary learning. In *IEEE Signal Processing Magazine*, 28(2), pp. 27–38, March 2011. ISSN 1053-5888. doi:10.1109/MSP.2010.939537.

160. J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. In *Proceedings of the IEEE*, 98(6), pp. 948–958, June 2010. ISSN 0018-9219. doi:10.1109/JPROC.2010.2044010.

161. D. Vainsencher, S. Mannor, and A.M. Bruckstein. The sample complexity of dictionary learning. In *J. Mach. Learn. Res.*, 12, pp. 3259–3281, 2011.

162. P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1096–1103. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-205-4. doi:10.1145/1390156.1390294.

163. G.K. Wallace. The jpeg still picture compression standard. In *Commun. ACM*, 34(4), pp. 30–44, April 1991. ISSN 0001-0782. doi:10.1145/103085.103089.

164. Q. Wang, S.R. Kulkarni, and S. Verdu. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. In *IEEE Transactions on Information Theory*, 55(5), pp. 2392–2405, May 2009. ISSN 0018-9448. doi:10.1109/TIT.2009.2016060.

165. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. In *IEEE Transactions on Image Processing*, 13(4), pp. 600–612, April 2004. ISSN 1057-7149. doi:10.1109/TIP.2003.819861.

166. J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag, Stuttgart, Germany, 1998.

167. B. Wen, S. Ravishankar, and Y. Bresler. Video denoising by online 3D sparsifying transform learning. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 118–122. Sept 2015. doi:10.1109/ICIP.2015.7350771.

168. B. Wen, S. Ravishankar, and Y. Bresler. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. In *International Journal of Computer Vision*, 114(2), pp. 137–167, 2015. ISSN 1573-1405. doi:10.1007/s11263-014-0761-1.

169. B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. In *Proceedings of the IEEE*, 78(9), pp. 1415–1442, Sept 1990. ISSN 0018-9219. doi:10.1109/5.58323.

170. J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. In *Proceedings of the IEEE*, 98(6), pp. 1031–1044, June 2010. ISSN 0018-9219. doi:10.1109/JPROC.2010.2044470.

171. J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma. Robust face recognition via sparse representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), pp. 210–227, Feb 2009. ISSN 0162-8828. doi:10.1109/TPAMI.2008.79.

172. M. Yaghoobi and M.E. Davies. Compressible dictionary learning for fast sparse approximations. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pp. 662–665. Aug 2009. ISSN 2373-0803. doi:10.1109/SSP.2009.5278490.

173. M. Yaghoobi, S. Nam, R. Gribonval, and M.E. Davies. Analysis operator learning for overcomplete cosparse representations. In *2011 19th European Signal Processing Conference*, pp. 1470–1474. Aug 2011. ISSN 2076-1465.

174. M. Yaghoobi, S. Nam, R. Gribonval, and M.E. Davies. Noise aware analysis operator learning for approximately cosparse signals. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5409–5412. March 2012. ISSN 1520-6149. doi:10.1109/ICASSP.2012.6289144.

175. M. Yaghoobi, S. Nam, R. Gribonval, and M.E. Davies. Constrained overcomplete analysis operator learning for cosparse signal modelling. In *IEEE Transactions on Signal Processing*, 61(9), pp. 2341–2355, May 2013. ISSN 1053-587X. doi:10.1109/TSP.2013.2250968.

176. J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. June 2008. ISSN 1063-6919. doi:10.1109/CVPR.2008.4587647.

177. H. Yu and G. Wang. Compressed sensing based interior tomography. In *Physics in Medicine and Biology*, 54(9), p. 2791, 2009.

178. M.D. Zeiler. Adadelta: An adaptive learning rate method. arXiv:1212.5701, 2012.

179. F. Zhang, Y. Cen, R. Zhao, and H. Wang. Improved separable dictionary learning. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 884–889. Nov 2016. ISSN 2164-5221. doi:10.1109/ICSP.2016.7877957.

180. F. Zhang, Y. Cen, R. Zhao, H. Wang, Y. Cen, L. Cui, and S. Hu. Analytic separable dictionary learning based on oblique manifold. In *Neurocomputing*, 236, pp. 32 – 38, 2017. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2016.09.099. Good Practices in Multimedia Modeling.

181. M. Zibulevsky and B.A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. In *Neural Computation*, 13(4), pp. 863–882, April 2001. ISSN 0899-7667. doi:10.1162/089976601300014385.

182. H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. In *Journal of Computational and Graphical Statistics*, 15(2), pp. 265–286, 2006. ISSN 10618600.