Technische Universität München
Lehrstuhl für Datenverarbeitung

Sample Complexity of Representation Learning for Sparse and Related Data Models

Matthias Seibert

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.-Ing. Wolfgang Kellerer

Prüfer der Dissertation:

1. Priv.-Doz. Dr. rer. nat. Martin Kleinsteuber
2. Prof. Dr.-Ing. Wolfgang Utschick

Die Dissertation wurde am 17.09.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 05.04.2019 angenommen.

Matthias Seibert. *Sample Complexity of Representation Learning for Sparse and Related Data Models.* Dissertation, Technische Universität München, Munich, Germany, 2019.

Acknowledgments

I would like to take this opportunity to thank all the people that supported me during the past years and without whom writing this dissertation would not have been possible.
First of all, I would like to thank my supervisor and mentor Dr. Martin Kleinsteuber for giving me the opportunity to work in such a fascinating research field. His technical and moral support motivated me to stay on course and ultimately enabled me to write this thesis. Thanks to my fellow GOL members Clemens, Simon, Vince, Martin, Alex, and Peter for the many fruitful – sometimes even technical – discussions and the awesome time we had at the chair. Thanks to Dominik, Julian, Tim, Uli, Hao, Johannes, Martin, Marko, and all the other people at LDV for showing the math guy how to solder and for making the past couple of years such a fun experience. Thanks to Prof. Dr. Klaus Diepold for providing me the opportunity of continuing my work at his chair after the untimely end of the GOL research group. Thanks to Prof. Dr. Wolfgang Utschick for reviewing my thesis. Thanks to my family and friends for the moral support and encouragement.

Abstract

The representation of data is a central aspect of machine learning algorithms. Representation learning aims at finding a data representation that facilitates ensuing tasks without having to manually create relevant features. A backbone of many representation learning algorithms is the assumption that the information contained within the data can be represented via a dictionary in some way while additional general purpose priors further encourage representations that highlight explanatory factors of the data. A prior that has proven to be particularly suited for extracting relevant information is sparsity, and models featuring this prior such as sparse dictionary learning have become popular ways of representing data. After the general type of representation model has been established, the optimal model is determined by fine tuning its descriptive parameters such that a representative set of training samples can be accurately expressed via the proposed model. Hence, the data on which a representation model is trained is crucial to its performance. Due to this dependency, an inherently important question regarding the quality of the model is: How much data is required such that the learned model generalizes well to previously unseen data samples?

This thesis provides an insight into that question by analyzing the generalization error and sample complexity of representation learning algorithms. The sample complexity of a learning algorithm can be loosely stated as the number of samples required such that the trained model exhibits a small generalization error with high probability. Two frameworks that are capable of bounding the sample complexity of representation learning algorithms are thoroughly elaborated and presented as readily applicable step-by-step procedures. The first bounding scheme is based on the central aspects of Hoeffding's inequality and a covering argument and is referred to as (HC). The second framework coined (MR) employs McDiarmid's bounded differences inequality and Rademacher complexity to achieve generalization error bounds. Both of these methods highlight the importance of aspects of the model such as the signal dimension, the structure of the employed dictionaries, and other priors used in the respective representation model.

The provided bounding frameworks are applied to a variety of algorithms that constitute a large portion of representation learning models with sparsity constraints. In particular, we derive novel results for the sample complexity of two variants of co-sparse analysis operator learning. Aside from the standard co-sparse analysis model, we also investigate the generalization error of a co-sparse analysis model where a separable structure is enforced on the filters, and show that this additional constraint results in lower bounds on the sample complexity. Further, we reestablish bounds of other representation learning schemes such as sparse dictionary learning, and also discuss principal component analysis, an algorithm outside the scope of sparse representations that still fits into the proposed frameworks.

Finally, a stochastic optimization algorithm is proposed for co-sparse analysis operator learning that leverages the geometric properties of the learning task and features an averaging adaptive step size selection. The experiments conducted on synthetic data corroborate the theoretical findings on the generalization error and sample complexity.

Contents

List of Figures

List of Symbols

$\mathbf{A}$, $\mathbf{X}$, $\mathbf{\Omega}$	Matrices, written as capital boldface letters.
$\boldsymbol{\alpha}$, $\mathbf{s}$, $\mathbf{x}$	Vectors, written as lowercase boldface letters.
α, κ, t	Scalars, written as lowercase Arabic/Greek or capital letters.
$\mathcal{A}$, $\mathcal{S}, \mathcal{X}$	Higher-order tensors, written as capital calligraphic letters.
$\mathfrak{C}$, $\mathfrak{D}$	Constraint sets, written as capital Fraktur letters.
$\mathfrak{F}_{\mathfrak{D}}$	Set of hypotheses with parameterizing weights from the set $\mathfrak{D}$.
$\mathbf{x}_i$	i-the column of the matrix $\mathbf{X}$.
x_{ij}	Entry in the i-th row and j-th column of matrix $\mathbf{X}$.
$\|\cdot\|_2$	ℓ_2-norm.
$\|\cdot\|_1$	ℓ_1-norm.
$\|\cdot\|_F$	Frobenius norm.
$\|\mathfrak{A}\|$	Cardinality of the set $\mathfrak{A}$.
$r \propto t$	r is proportional to t means that $y = cx$ for some constant c.
$\otimes$	Kronecker product of matrices.
$\times_k$	k-mode product for tensors, see Definition 5.5.
p	Dimensionality of signal.
d	Dimensionality of feature.
n	Number of samples.
$N^{\text{cov}}(\mathfrak{C}, \varepsilon)$	Minimal ε covering of set $\mathfrak{C}$.
$\mathcal{O}(\cdot)$	Big O notation. Asymptotic upper bound.
$\mathbf{x}$, $\mathbf{X}$	Training data, denoted by bold variants of the letter x.
y	Training labels, denoted by the letter y.
$f_{\mathbf{x}}(\mathbf{W})$	A function parameterized by the weights $\mathbf{W}$ applied to data $\mathbf{x}$.
$g(\cdot)$	Penalty function.
$\mathbf{I}_p$	Identity matrix of dimension (p, p).
$\mathbf{1}_p$	Vector of all ones of dimension p.

$\mathbf{0}_p$	Vector of all zeros of dimension p.
$\mathrm{diag}(\mathbf{A})$	For a quadratic matrix $\mathbf{A}$ returns a diagonal matrix with same diagonal entries as $\mathbf{A}$.
$\mathrm{St}(p, d)$	The (p, d)-Stiefel manifold, which consists of orthogonal matrices in $\mathbb{R}^{p \times d}$.
$\mathrm{Ob}(p, d)$	The (p, d)-oblique manifold, which consists of matrices in $\mathbb{R}^{p \times d}$ with unit norm columns/rows (dependent on application).
$T_{\mathbf{W}}M$	Tangent space to manifold M at $\mathbf{W} \in M$.
$\gamma(\mathbf{W}, \mathbf{G}, \cdot)$	Geodesic emanating from $\mathbf{W}$ in direction $\mathbf{G} \in T_{\mathbf{W}}M$.
$\mathcal{E}(\cdot)$, $\mathcal{E}_n(\cdot)$	Estimated and empirical risk as defined in (2.1), (2.5).
$\mathfrak{R}_n, \widehat{\mathfrak{R}}_{\mathbf{X}}$	Rademacher complexity and empirical Rademacher complexity.
$\mathfrak{G}_n, \widehat{\mathfrak{G}}_{\mathbf{X}}$	Gaussian complexity and empirical Gaussian complexity.
$X \sim \mathbb{P}$	Random variable X distributed according to probability distribution $\mathbb{P}$.
$\mathcal{N}(0, 1)$	Gaussian distribution with mean 0 and standard deviation 1.
$\Pr[E]$	Probability of the event E.
$\mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[f(\mathbf{x})]$	Expectation of the random variable $\mathbf{x}$.

This dissertation is based in parts on the following peer-reviewed articles that were published during my work as a research assistant.

M. Seibert, J. Wörmann, R. Gribonval, and M. Kleinsteuber, "Separable cosparse analysis operator learning", *Proceedings of the 22nd European Signal Processing Conference*, pp. 770–774, 2014.

M. Seibert, M. Kleinsteuber, R. Gribonval, R. Jenatton, and F. Bach, "On the sample complexity of dictionary learning", *IEEE Statistical Signal Processing Workshop*, pp. 244–247, 2014.

R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and **M. Seibert**, "Sample complexity of dictionary learning and other matrix factorizations", *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3469–3486, 2015.

M. Seibert, J. Wörmann, R. Gribonval, and M. Kleinsteuber, "Learning co-sparse analysis operators with separable structures", *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 120–130, 2016.

Furthermore I was involved in the publication of the following articles which were published during my time as a doctoral student.

M. Seibert, M. Kleinsteuber, and K. Hüper, "Properties of the BFGS method on Riemannian manifolds", *Mathematical System Theory: Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday*, pp. 395–412, 2013.

S. Hawe, **M. Seibert**, and M. Kleinsteuber, "Separable dictionary learning", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438–445, 2013.

Chapter 1

Introduction

The performance of machine learning algorithms is inherently dependent on the choice of data representation to which they are applied. It comes at no surprise that an important aspect in designing machine learning algorithms is the development of preprocessing techniques and data transformations which generate data representations that facilitate the effectiveness of machine learning. When performed manually this process is called feature engineering. The data is processed according to the knowledge of a human operator as to which data features are necessary for creating an algorithm that performs the aspired assignment satisfactorily. This is a task that requires extensive knowledge of the fundamental properties of the applied machine learning algorithm, as well as a vast amount of time spent on fine-tuning the interface of the generated data features with the actual algorithm.

In order to simplify the application of machine learning algorithms and increase their scope, it is highly desirable to make the algorithms less dependent on manual feature engineering. Representation learning is a means to that end. While initially only a preprocessing step it has now become a research field of its own in the machine learning landscape. With the International Conference on Learning Representations which was established in 2013 it is even represented by its own conference. In general, representation learning is comprised of techniques that automatically discover features that are required for high level tasks such as classification. Robust, automated representation learning procedures allow for the rapid construction of new algorithms. Extracting these features is often not task specific, and general-purpose priors such as for example smoothness, manifold structure, or sparsity can be used to unravel the original high-dimensional data.

Representation learning algorithms can be roughly summarized into two categories, namely

supervised and unsupervised learning methods. Supervised learning methods take tuples (x, y) as inputs, which consist of the data point x and the corresponding label y. The task of representation learning algorithms then is to estimate a target hypothesis h that predicts the label y given input variable x as $h(x) = y$. Different types of representations make different assumptions about the structure of the function being learned, e.g., linear or non-linear, and how to best optimize a representation to approximate the optimal target function. Supervised learning methods are for example feed forward neural networks and supervised dictionary learning. On the other hand, unsupervised methods only handle the data x while no target label is supplied. Their goal is to find the function h that achieves the minimal value for $h(x)$ directly. Representatives for unsupervised learning algorithms are k-means clustering, principal components analysis, independent component analysis, and unsupervised dictionary learning. A branch of representation learning that has been particularly successful focuses on generating sparse representations of the data. One prime example in this category is sparse dictionary learning where signals are modeled as the synthesis of only a few elements of a dictionary. The task of sparse dictionary learning algorithms is to find a dictionary that is able to provide a representation that is as sparse as possible for a given set of training samples.

Whether a learning algorithm, be it supervised or unsupervised, is efficient can be measured with various benchmarks. On the one hand, there is the computational complexity that measures how much computational power has to be expended in order to manipulate the data such that a good approximation to the target is achieved. The other aspect is the sample complexity which, loosely phrased, is a measure of how large the set of training samples provided to the algorithm has to be in order to learn a model that generalizes well to previously unseen samples that belong to the same distribution as the training data. Sample complexity results fall in the general category of theoretical learning guarantees for algorithms that depend on the complexity of the considered model and the size of the set of training samples.

1.1 Representation Learning

What makes one representation better than another? Marr in [57] gives the example of Roman and Arabic numerals. While both are perfectly adequate ways to denote numbers,

with Arabic numerals it is straightforward to divide 270 by 6 by hand through using long division. Conversely, dividing CCLXX by VI cannot be achieved as easily and would typically involve translating the Roman to Arabic numerals. This serves as an example that a good representation is one that enables subsequent tasks to achieve the desired results in a more efficient manner. Inherently, all representation learning tasks face a trade-off between keeping as much information of the input as possible versus achieving desirable properties for the reduced data. A summary of these beneficial properties is provided by Bengio *et al.* in [7]. Principal component analysis, for example, forces the new variables to be independent from one another, while sparse synthesis models promote the property that for any given observation only a small fraction of possible factors are relevant.

A central assumption for many representation learning algorithms is that a data sample $\mathbf{x} \in \mathbb{R}^p$ can be represented over some dictionary $\mathbf{D} \in \mathbb{R}^{p \times d}$ that is dependent on the nature of the class, which the data originates from. A typical way of modeling the data is to approximate $\mathbf{x}$ as a linear combination of the columns of the dictionary $\mathbf{D}$, which is formally expressed by

$$\mathbf{x} \approx \mathbf{D}\boldsymbol{\alpha}. \tag{1.1}$$

The coefficients that describe the signal $\mathbf{x}$ with respect to the new features are stored in the coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^d$. Both, the coefficient vector $\boldsymbol{\alpha}$, as well as the dictionary $\mathbf{D}$ are subject to some constraints, which depend on the type of representation learning that we conduct. One example is sparse dictionary learning where the coefficient vector has to be sparse and the columns of the dictionary $\mathbf{D}$ are typically constrained to have unit norm.

In order to train a dictionary that is able to represent a certain class of signals, we need to supply some data that is exemplary for the signals we want to process with the model that the learning algorithm generates. This is commonly referred to as training data which is typically given in matrix form $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$. Each column of $\mathbf{X}$ represents a single training sample. The corresponding n coefficient vectors are also summarized in a matrix $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n] \in \mathbb{R}^{d \times n}$. The representation learning problem can then be expressed as the optimization problem

$$\underset{\mathbf{D},\mathbf{A}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \sum_{i=1}^{n} g(\boldsymbol{\alpha}_i) \qquad \text{subject to} \qquad \mathbf{D} \in \mathfrak{C},\ \mathbf{A} \in \mathbb{R}^{d \times n}. \tag{1.2}$$

Therein, the function $g\colon \mathbb{R}^d \to \mathbb{R}^+$ is a prior that promotes the constraints enforced on the coefficient vector, such as for example sparsity. By $\mathbb{R}^+$ we denote the positive real values including zero. The set $\mathfrak{C}$ is a constraint set that forces the dictionary to have a certain structure. In the case of sparse dictionary learning, the constraint set is the set of all (p, d)-matrices with unit norm columns.

After learning a dictionary, the coefficient that represents a data sample $\mathbf{x}$ is typically obtained by solving the minimization problem

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}). \tag{1.3}$$

This implies that given a pre-learned dictionary, obtaining the coefficient vector for a signal with respect to this dictionary requires solving an optimization problem. By definition, it is highly unlikely that a signal can be perfectly expressed as a combination of the dictionary columns. Especially regarding the constraints on both the dictionary, as well as the coefficient vector. At best we can hope for a good approximation of the original signal. The quality of how well a signal $\mathbf{x}$ can be represented with respect to a dictionary $\mathbf{D}$ is measured by

$$f_{\mathbf{x}}(\mathbf{D}) = \inf_{\boldsymbol{\alpha}\in\mathbb{R}^d} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}), \tag{1.4}$$

where we are taking the infimum over all vectors in the coefficient space.

Sample complexity

Machine learning algorithms critically rely on the data samples they are trained on, and while for classic computer science algorithms typical measures of quality are time and space complexity, for machine learning algorithms the additional concept of sample complexity becomes very relevant. The sample complexity of an algorithm is the size of the sample set that is required for the algorithm to learn a specific model up to a certain accuracy.

Its definition is based on the generalization error. We adopt the previously introduced notation where the function class is parameterized by the $\mathbf{D}$ and f is a smooth function in both $\mathbf{x}$ and $\mathbf{D}$. The generalization error of $f_{\mathbf{x}}(\mathbf{D})$ for a set of samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \sim \mathbb{P}$

is defined as

$$\left|\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[f_{\mathbf{x}}(\mathbf{D})]\right|. \tag{1.5}$$

Since we are dealing with dictionaries $\mathbf{D}$ that are elements of a constraint set $\mathfrak{C}$, we have to evaluate the worst case generalization error. Furthermore, since $\mathbf{x}_i$ are random samples, we can only guarantee the generalization error up to a certain probability. This can be expressed by bounding the probability that $\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D})$ is further away from its expectation $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[f_{\mathbf{x}}(\mathbf{D})]$ than a certain value for all possible dictionaries $\mathbf{D} \in \mathfrak{C}$. That is, for some $\eta, \delta > 0$ we have

$$\Pr\left[\sup_{\mathbf{D}\in\mathfrak{C}} \left|\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[f_{\mathbf{x}}(\mathbf{D})]\right| \geq \eta\right] < \delta. \tag{1.6}$$

On a side note, when Equation (1.6) holds for all $\eta, \delta > 0$, then the sequence f_n defined as $f_n \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D})$ is said to converge uniformly to its expectation.

The sample complexity then is the minimum number of samples N such that for $n \geq N$ Equation (1.6) is fulfilled for a chosen η and δ. The first results regarding generalization error bounds, and in conjunction sample complexity, are based on results on the complexity of hypothesis classes obtained by Vapnik and Chervonenkis in [93, 94], which are the publications that instantiated statistical learning theory. One of the key contributions of their work is the concept of VC dimension, which is used as a measure of expressiveness of (potentially) infinite dimensional function classes. While in the beginning statistical learning theory was purely a theoretical analysis of the problem of function estimation from a given collection of data, it later became the theoretical foundation for the development of the widely known support vector machine classification algorithm [22]. With this development, statistical learning theory evolved from being simply a tool for theoretical analysis to a tool for creating practical algorithms. The VC dimension can be used in order to obtain probabilistic bounds on the generalization error and a multitude of articles exist that discuss the complexity of various learning algorithms. One of the first VC-based sample complexity bounds is presented in [29]. The authors show that the sample complexity is a linear function of VC dimension of the hypothesis space. A detailed summary on VC theory and their application to generalization error bounds can be found in [86, 92] and we will also address

this topic in more detail in Chapter 2. An important property of the generalization error results based on the VC dimension is that they do not depend on the data distribution. This has the advantage that the bounds are very general. However, this aspect has also been the main source of criticism of the VC theory, as the obtained bounds are loose in general [20, 36]. Based on this criticism Vayatis and Azencott formulate a distribution-dependent variant of the VC bounds [95] that uses Cramér transforms to incorporate data distribution information into the bounds.

Since the original generalization error bounds based on VC theory, more sophisticated methods have been proposed. One branch that has achieved good results is based on using the structure of the constraint set of the parameterizing weights of the hypothesis class. The complexity of a family of functions is measured by covering numbers, which is the minimal number of ε-balls required to cover the constraint set. This method was first devised in [23] and is able to obtain tighter bounds for the generalization error. Finally, a more recent way for deriving generalization bounds is based on the Rademacher complexity, which was first advocated in [5, 47, 48]. We will discuss these concepts in more detail in Chapter 4.

These publications cover the sample complexity of a wide variety of representation learning algorithms. However, only few publications offer a thorough analysis of the sample complexity of sparse representation learning algorithms. The work published on this topic is limited to the publications [58, 90] which give sample complexity results for a variety of k-dimensional coding schemes such as k-means clustering, non-negative matrix factorization, and dictionary learning for signal distributions within the unit ball.

Stochastic gradient descent

Stochastic optimization methods are a cornerstone of machine learning algorithms. They date back to the 1950s and have proven to be the preferred optimization algorithm when it comes to tackling large scale problems. Gradient descent methods are optimization methods that determine the minimum of a cost function in an iterative manner by using first order information. They are typically initialized at a first estimate and then take steps along the negative gradient, where each iterate is a better approximation of the true minimum of the cost function. While standard (batch) gradient descent methods use the entire data set to determine the search direction, stochastic approximation methods only process a small

subset of randomly chosen samples in each iteration. More precisely, given a set of n samples in matrix form $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and a cost function $f_{\mathbf{x}}(\mathbf{W})$ that is determined by its weights $\mathbf{W}$. These weights are the dictionaries in the presented representation learning scenario. The goal is to find the minimum of $\frac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{W})$ with respect to $\mathbf{W}$. Instead of computing the gradient with respect to the entire $\mathbf{X}$, stochastic gradient descent methods use a noisy approximation of the gradient by only considering either a single sample $\mathbf{x}_i$ or a small batch of samples. This sample (or subset of samples) is picked at random and used to compute the associated gradient which is then employed as the next search direction.

Recently, these methods have witnessed an increase in popularity due to their ability of dealing with large data sets and the growing amount of data that is available in many machine learning settings. Stochastic gradient descent algorithms have become the standard optimization procedure for machine learning problems. A review of stochastic learning algorithms can be found in [12], where the stochastic gradient descent algorithm is applied to a variety of learning algorithms, such as perceptrons, k-means clustering, and multi-layer neural networks.

As we will see in a subsequent chapter, the notion of stochastic gradient descent and sample complexity are closely interwoven concepts.

1.2 Research Problem

With the continuously growing number of representation learning algorithms and the incorporation of machine learning techniques in a wide variety of commercial applications as well as their increasing presence in everyday life, it is important to gain a thorough theoretical understanding of the performance of these algorithms. Of particular note are sparse representations that have proven to be powerful methods for expressing the data effectively in many applications.

Modern machine learning algorithms rely on the vast amount of data that is available. Despite this abundance of available data, there are many application scenarios where it is important to work with as little data as possible in order to reduce the computational overhead. This is essential in scenarios where it is necessary to provide fast results, or to work with a computational infrastructure that is not potent enough to process large data sets. In addition to situations, where data is available in large quantities and the computation

time is the bottleneck, there are also circumstances where only a limited amount of data is available by design of the problem. For example in [99] Wörmann *et al.* propose a blind compressive sensing model learning scheme, where given a compressively sensed signal the algorithm reconstructs the original image and simultaneously learns the adequate model for the denoising task. Hence, the available data is inherently limited by the dimensions of the image.

It is crucial to understand how sparse representation learning algorithms depend on the number of training samples that are employed during the learning phase. A thorough analysis of their sample complexity is necessary to provide meaningful performance guarantees. The central aspect of this work is to investigate two strategies to derive the generalization error bounds of representation learning and detail the bounding process both thoroughly and comprehensibly. The main focus will be on learning algorithms that feature a sparsity prior.

As mentioned previously, apart from the sample complexity, another important performance indicator for representation learning algorithms is their computational complexity. This aspect will not be covered within this thesis. Furthermore, while more rigorous bounds can certainly be derived when considering very concrete algorithms under specific assumptions, this is not within the scope of this work. This thesis will cover the following topics.

Stochastic gradient descent and sample complexity

Stochastic gradient descent methods have become the commonly used optimization technique for modern machine learning algorithms which is arguably a result of their generalization performance. In addition to their ability to produce good results with only a limited amount of data and their ability to work with large sets of data, there is also a close connection to the sample complexity of representation learning algorithms. This thesis gives a review of how this connection is established and derives the actual bounding effect of a generalization error bound to the optimization error that occurs during training.

Sample complexity bounding frameworks

While there exist a variety of papers discussing the sample complexity and generalization error properties of many different representation learning algorithms, these publications are commonly algorithm specific and investigate the generalization performance for a setting

particular to the coupled learning algorithm. Furthermore, due to the specificity of the setting these works address, the results therein are highly technical and very hard to grasp unless one is already an expert in the field. The goal of this dissertation is to give a more readily understandable approach to the problem of deriving sample complexity bounds which can be applied to a wide variety of representation learning algorithms with a focus on sparse representation schemes. In order to achieve this goal, this dissertation presents two general frameworks for deriving upper bounds on the generalization error. Each framework is thoroughly introduced by first presenting the necessary basic principles, roughly outlining the bounding process, and then giving a detailed step-by-step explanation by means of a concrete example. The proposed frameworks are both data distribution-dependent and offer enough generality to be applicable to a wide variety of representation learning algorithms.

The central elements of the first approach are the application of Hoeffding's inequality and covering numbers of the constraint set that describes the hypothesis class. Hoeffding's inequality is a concentration inequality that bounds the probability of a random variable deviating from its mean. The covering number bound serves as a more sophisticated solution to express the complexity of the hypothesis class which replaces the VC dimension in the bounding process. The second approach that we take is based on McDiarmid's inequality and Rademacher complexity as a measure of complexity of the function class.

Results for sparse representation learning

After establishing the fundamentals and briefly reviewing the steps necessary to derive the final bounds, we apply the devised techniques to bound a variety of representation learning algorithms that cover the spectrum of sparse representations. The first examined model that serves as the introductory example for applying the bounding framework is sparse dictionary learning, which is a well known learning scheme for a sparse synthesis model. After this initial example, we discuss the co-sparse analysis operator model. This model is closely related to sparse dictionary learning, although less well known. Nonetheless, it has proven to be very potent for various image processing tasks. Aside from the standard co-sparse analysis operator learning, which requires vectorized signals, we also take a look at what influence enforcing a separability constraint on the parameterizing weights has on the generalization error. Separability in this case means that the linear operator can be

written as the Kronecker product of smaller operators. This has the added benefit that for higher dimensional signals, such as for example gray-scale image data which exhibits a two-dimensional structure, it is not necessary to vectorize the signal. Therefore, the inherent structure of the data is maintained. Finally, we investigate a supervised variant of dictionary learning models, also referred to as task-driven dictionary learning, where in addition to learning an encoding dictionary, a decoding operator is trained simultaneously. The signal is first sparsely coded via a dictionary and then the sparse code is mapped to the domain of a variable associated to the signal by the decoder. The associated variable can often be interpreted as the label corresponding to the signal. This case study illustrates that the devised frameworks for deriving sample complexity results are also applicable to supervised sparse representation learning.

1.3 Contribution

This thesis examines the sample complexity of representation learning algorithms for sparse and related models by means of two bounding schemes which are used to estimate the generalization error with high probability. The first approach uses the Lipschitz property of the cost function, Hoeffding's concentration inequality, and an ε-covering of the constraint set that parameterizes the class of admissible hypothesis functions as core elements. The second framework is based on more recent bounding techniques. The central estimation tools it utilizes are McDiarmid's bounded difference concentration inequality, Rademacher calculus as a complexity measure for the hypothesis class, and Slepian's Lemma. Both devised bounding schemes are formulated as step-by-step procedures that are readily applicable to a wide variety of representation learning algorithms. Each individual step is motivated and explained in detail offering the reader a comprehensible introduction to sample complexity analysis. The investigation of two different frameworks provides a better understanding of the key aspects of the estimation process and offers an insight into properties of representation learning models relevant for the error bounds such as the structure of the constraint set that occurs as a driving factor in different estimation steps for both methods.

The proposed frameworks are applied to derive new bounds for a variety of representation learning methods. These bounds highlight the role of the sparsity promoting penalty function, constraints on the dictionary, and distribution of the data samples. For representation

learning models that were previously investigated in literature the bounds devised in this thesis offer a new perspective on the parameters that drive the error bounds. As a first case study that serves as an introductory example this thesis examines the sparse dictionary learning model. The achieved bounds recover results from literature in terms of the behavior with respect to the number of samples. A specialized investigation of dictionary learning then extends the results to a wider variety of sample distributions. Next, this thesis studies principal component analysis and develops bounds that for the first technique are slightly more pessimistic than previous result with respect to the required samples. However, our results have the advantage of covering more diverse data distributions. The second bounding scheme then recovers the behavior from the literature with respect to the required training data samples for data distributions within the unit ball. In addition to unsupervised methods the devised bounding techniques are also applicable to supervised representation learning models. Generalization error bounds for the supervised dictionary learning model are developed which cover the elastic-net penalty. Finally, novel bounds are provided for the co-sparse analysis operator learning model that elucidate the importance of the role of the constraints on the operator. This result is further supported by investigating a second specification of the co-sparse analysis model. In addition to the standard formulation, we also examine a separable variant of the co-sparse model and show that the additional structure enforced on the operator results in more sophisticated generalization error bounds.

The bounds derived for the generalization error as defined in Equation (1.5), which we denote by $\Phi(\mathbf{X})$, with the technique based on Hoeffding's inequality and covering number bounds generally can be expressed as

$$\Phi(\mathbf{X}) \leq c_1 \sqrt{\frac{\log n}{n}} + c_2 \sqrt{\frac{1}{n}} \tag{1.7}$$

that holds with probability at least $1-\delta$ where n is the number of available training samples. The positive variables c_1, c_2 depend on the Lipschitz constant of the cost function, the size of the set of available hypotheses, and the dimensionality of the signals and coefficients. For the second bounding scheme that employs McDiarmid's inequality and Rademacher complexity, the generalization error bounds exhibit the general form

$$\Phi(\mathbf{X}) \leq c \sqrt{\frac{1}{n}} \tag{1.8}$$

that holds with probability greater than or equal to $1 - \delta$. The scalar parameter c depends again on the Lipschitz constant of the cost function, the signal dimension, and the structure of the constraint set that describes the parameters for the hypothesis class.

The final contribution is an analysis of the algorithmic performance of a co-sparse analysis operator learning framework with respect to its sample complexity. We provide the algorithmic framework for a stochastic geometric learning scheme featuring an adaptive averaging step size selection for both the non-separable as well as the separable variant of the model and conduct several experiments to evaluate their performance in recovering a ground truth operator. The experiments illustrate the previously obtained results regarding the sample complexity of the algorithms.

1.4 Thesis Outline

After giving a motivational introduction and overview in this first chapter, Chapter 2 provides a more detailed introduction to the related work and basic principles of sample complexity theory and the stochastic gradient descent algorithm. Chapter 3 reviews the connection of the sample complexity of an algorithm to the learning procedure when using stochastic gradient descent and points out the benefits of replacing the batch optimization with the more adaptive stochastic mini-batch optimization. The main chapters 4 & 5 focus on establishing the sample complexity results using two different bounding schemes for a variety of learning algorithms. Chapter 4 introduces the definitions and theorems that are necessary to provide the final bounds, gives a schematic of the steps taken to determine the upper generalization error bounds, and then demonstrates the entire bounding procedure by means of the first case study which is sparse dictionary learning. Furthermore, Chapter 4 also provides a more detailed investigation of the sample complexity of the specific dictionary learning case. By leveraging more refined bounding arguments specifically adapted to this problem the results are extended to more general data distributions and penalty functions. Chapter 5 then applies the bounding paradigms derived in the previous chapter to a variety of representation learning algorithms, such as PCA, co-sparse analysis operator learning in both non-separable as well as separable specifications, and supervised dictionary learning.

Finally, Chapter 6 provides an implementation of one of the previously discussed representation learning algorithms. This chapter presents a geometric stochastic gradient descent

optimization scheme with adaptive step size selection that tackles the co-sparse analysis operator learning problem in both the non-separable as well as the separable variant. The progress throughout the training procedure is evaluated with respect to a ground truth operator and both methods are compared with respect to their performance for different mini-batch sizes. The results are indicative of the sample complexity results derived in the previous chapter.

Chapter 2

Basic Principles & Related Work

In this chapter we give an introduction to generalization error and sample complexity by recalling the basic principles of devising bounds for these two concepts. Furthermore, this chapter reviews sample complexity results for the considered representation learning frameworks. Finally, we provide a brief recapitulation of stochastic gradient descent methods as a precursor to showing the relation of stochastic gradient descent and sample complexity.

2.1 Sample Complexity

Let $\mathfrak{X}$ be a space, which we refer to as input space, $\mathfrak{Y}$ a space we call output space. Each sample is actually a tuple from the direct product $\mathfrak{X} \times \mathfrak{Y}$. Fix a function space $\mathfrak{F}$ that consists of functions $f\colon \mathfrak{X} \times \mathfrak{Y} \to \mathbb{R}^+$ that map from the product space $\mathfrak{X} \times \mathfrak{Y}$ to the non-negative real numbers $\mathbb{R}^+ \coloneqq \{r \in \mathbb{R} : r \geq 0\}$. Each element $f \in \mathfrak{F}$ is the composition of a hypothesis $h\colon \mathfrak{X} \to \mathfrak{Y}$ and a loss function $\ell\colon \mathfrak{Y} \times \mathfrak{Y} \to \mathbb{R}^+$ that measures the performance of a hypothesis. For a sample data point $(\mathbf{x}, y)$ we have $f((\mathbf{x}, y)) = \ell(h(\mathbf{x}), y)$. The goal of a learning algorithm is to find a function h that minimizes the loss of the target output y and the predicted output $h(\mathbf{x})$, or in other words a function $f \in \mathfrak{F}$ that achieves the minimal value. For a given joint probability distribution $\mathbb{P}$ on $\mathfrak{X} \times \mathfrak{Y}$, the *expected risk* of $f \in \mathfrak{F}$ is defined as

$$\mathcal{E}(f) \coloneqq \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}}[f((\mathbf{x}, y))] = \int_{\mathfrak{X}\times\mathfrak{Y}} f((\mathbf{x}, y))\, d\mathbb{P}((\mathbf{x}, y)). \tag{2.1}$$

The optimal risk with respect to the set of hypothesis $\mathfrak{F}$ is the hypothesis that achieves

the smallest *achievable expected risk*. It is defined as

$$\mathcal{E}_{\mathfrak{F}}^{\star} \coloneqq \inf_{f \in \mathfrak{F}} \mathcal{E}(f). \tag{2.2}$$

The hypothesis that achieves the optimal risk is referred to as the optimal hypothesis

$$f_{\mathfrak{F}}^{\star} \coloneqq \arg\min_{f \in \mathfrak{F}} \mathcal{E}(f). \tag{2.3}$$

In a general learning setting the joint probability distribution $\mathbb{P}$ is unknown and the only available information is contained in the training set $S_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \sim \mathbb{P}^n$, where each element is drawn independently from $\mathfrak{X} \times \mathfrak{Y}$ according to the distribution $\mathbb{P}$. Let us for now denote the learning process as $\hat{f}_{\mathfrak{F},n} = \mathrm{ALG}(S_n)$. The algorithm ALG takes a set of samples S_n and produces a hypothesis in $\mathfrak{F}$ which it believes to be an adequate approximation of the optimal hypothesis. The hypothesis $\hat{f}_{\mathfrak{F},n}$ is a random variable that depends on the random variable S_n. ALG is called consistent if $\mathcal{E}(\hat{f}_{\mathfrak{F},n})$ probabilistically converges to $\mathcal{E}_{\mathfrak{F}}^{\star}$, i.e., for all $\eta, \delta > 0$ there exists a positive integer N such that for all $n \geq N$ we have

$$\Pr\left[\mathcal{E}(\hat{f}_{\mathfrak{F},n}) - \mathcal{E}_{\mathfrak{F}}^{\star} \geq \eta\right] < \delta. \tag{2.4}$$

The *sample complexity* of ALG is the minimum N for which this holds as a function of $\mathbb{P}, \eta$ and δ. If the algorithm is not consistent, then it is common practice to set the sample complexity to infinite.

2.1.1 Generalization bounds

The expected risk of a hypothesis f as defined in (2.1) can in practice not be computed since the joint probability distribution on $\mathfrak{X}$ and $\mathfrak{Y}$ is typically unknown. Instead, given the training set S_n we can replace the expected risk with the *empirical risk* which is defined as

$$\mathcal{E}_n(f) = \tfrac{1}{n} \sum_{i=1}^{n} f((\mathbf{x}_i, y_i)). \tag{2.5}$$

The general idea of any machine learning and representation learning algorithm is to approximate the function that minimizes (2.1) with the function that approximates (2.5). This

principle is called empirical risk minimization. ERM is bound to be inaccurate and in order to gauge the deviation from the optimal solution we need a way to measure its performance. The *generalization error* is the difference between the expected and the empirical risk for a hypothesis f, or in other words the difference between the error on the training set and the error on the underlying joint distribution. It is defined as

$$G = \mathcal{E}_n(f) - \mathcal{E}(f). \tag{2.6}$$

One thing to note in this definition is that it is based on a single, randomly chosen training set S_n, and therefore is a random variable itself.

An algorithm is said to generalize if $\lim_{n\to\infty} \mathcal{E}_n(f) - \mathcal{E}(f) = 0$. As stated earlier, in most cases it is impossible to compute the generalization error explicitly since the underlying joint distribution of the data points is unknown. Instead, the goal of statistical learning problems is often to bound the generalization error in probability. That is, characterize a probability that the generalization error is below a certain bound η

$$\Pr\left[\mathcal{E}_n(f) - \mathcal{E}(f) < \eta\right] \geq 1 - \delta_n. \tag{2.7}$$

This provides a measure of the generalization performance of a single hypothesis f. However, in learning problems the correct hypothesis is not known beforehand. In fact, it is this correct hypothesis which we are trying to recover. The goal of the learning algorithm is, after all, to pick the hypothesis best suited to the training data. Therefore, we need a generalization error that represents this challenge.

In order to measure this inaccuracy, we take the supremum over all hypotheses in $\mathfrak{F}$ with $\sup_{f\in\mathfrak{F}} \mathcal{E}_n(f) - \mathcal{E}(f)$. This difference can be bounded by using the concept of uniform convergence in probability, which as per [93] is defined as follows.

Definition 2.1 (Uniform convergence in probability)**.** The empirical risk $\mathcal{E}_n(f)$ converges uniformly in probability to the expected risk $\mathcal{E}(f)$ if for all $\eta > 0$ there exists $N_{\max} \in \mathbb{N}$ such that for all $n > N_{\max}$

$$\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)| < \eta \tag{2.8}$$

holds with high probability.

This definition is very similar to the definition of a consistent learning algorithm as proposed earlier. The main difference is that we take the supremum over the entire function class $\mathfrak{F}$. In the following, we investigate the probability of the complementary event to (2.8), i.e., the case that $\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f)-\mathcal{E}(f)| \geq \eta$. As a result of the supremum, a single hypothesis $f \in \mathfrak{F}$ having a generalization error larger than η is sufficient for the entire hypothesis space to exhibit a generalization gap larger than η. We can replace the supremum with a union over all hypotheses

$$\Pr\big[\sup_{f\in\mathfrak{F}} \big(\mathcal{E}_n(f) - \mathcal{E}(f)\big) \geq \eta\big] = \Pr\big[\bigcup_{f\in\mathfrak{F}} \big(\mathcal{E}_n(f) - \mathcal{E}(f)\big) \geq \eta\big]. \tag{2.9}$$

Next, by applying a union bound argument to the union of events we extract the summation outside of the probability and then, under the assumption that the hypothesis space is limited to functions that range between 0 and 1, use Hoeffding's inequality to further bound this by

$$\begin{aligned}\Pr\big[\sup_{f\in\mathfrak{F}} \big(\mathcal{E}_n(f) - \mathcal{E}(f)\big) \geq \eta\big] &\leq \sum_{f\in\mathfrak{F}} \Pr\big[\mathcal{E}_n(f) - \mathcal{E}(f) \geq \eta\big] \\ &\leq 2|\mathfrak{F}| \exp(-2n\eta^2),\end{aligned} \tag{2.10}$$

where $|\mathfrak{F}|$ is the cardinality of the hypothesis space. We will discuss both the union bound argument, as well as Hoeffding's inequality in detail in Chapter 4.

The problem with the generalization error bound in (2.10) is that the magnitude of even simple hypothesis spaces, such as for example the space of two-dimensional linear hypotheses, is infinite. For more complex hypothesis spaces this property holds as well. The reason that the term $|\mathfrak{F}|$ occurs in the first place stems from the fact that we used a union bound argument which is a worst case estimate since it assumes that all events (and therefore all hypotheses in $\mathfrak{F}$) are independent.

In order for the bound (2.10) to be reasonable, the two events $\mathcal{E}_n(f_1) - \mathcal{E}(f_1) \geq \eta$ and $\mathcal{E}_n(f_2) - \mathcal{E}(f_2) \geq \eta$ have to be independent for every pair of hypotheses $f_1, f_2 \in \mathfrak{F}$. As an example, imagine a binary classification problem as depicted in Figure 2.1. The blue and red dots represent the training data and the target hypothesis is portrayed by the solid black line. All dashed colored lines result in an identical classification of the available samples and

produce the same empirical risk $\mathcal{E}_n$. One could ask whether we could simply keep one and discard all others, resulting in a reduced hypothesis space. However, this is not feasible, as the generalization error is also dependent on the estimated risk as defined in Equation (2.1), and since we do not know the underlying distribution, there is no way of evaluating the dependence of the generalization error hypotheses.

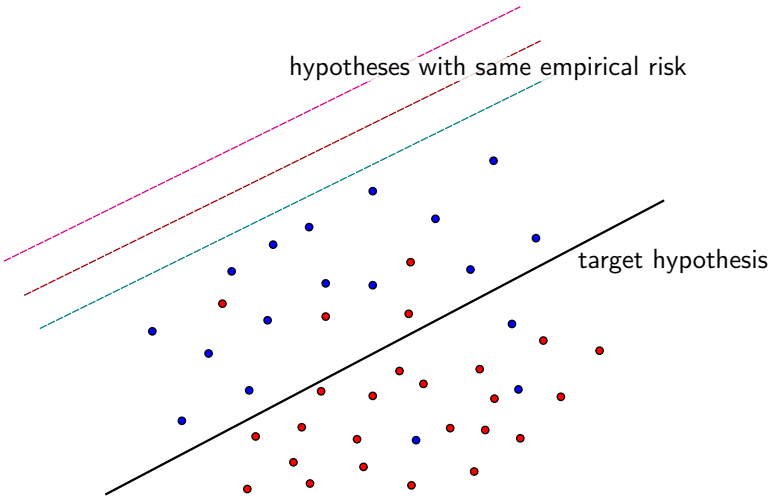


Figure 2.1: Two-dimensional data set consisting of two classes. For the given data set all dashed affine classifiers return the same value for the empirical risk.

In general, there is no way of accounting for these dependencies in an analytical fashion without risking to violate the claim of the supremum. Thus, we can summarize that the independence assumption on the hypothesis is the best assumption we can make, although it highly overestimates the probability that the generalization gap is large and makes the final bound very pessimistic.

While it is not possible to reduce the set of hypothesis without altering the model assumption, another approach is to reduce the definition space and instead of investigating the entire $\mathfrak{X} \times \mathfrak{Y}$ evaluate the function class on the subset restricted to the available samples. Vapnik and Chervonenkis show in their seminal work [93] that it is possible to estimate the generalization error by introducing a ghost data set. That is, by introducing a data set S'_n, which is drawn according to the same distribution as S_n. With this modification we are able

to remove the expected risk in the equality. This technique is similar to how algorithms are evaluated in practice. Typically, when assessing the performance of machine learning algorithms all available data is split into two sets, the training set and the test set. The algorithm is then trained using the training data and after the learning phase is concluded the model accuracy is evaluated using the test data. The ghost data set corresponds to this test data. Vapnik and Chervonenkis state that if $\mathcal{E}'_n(f)$ denotes the empirical risk over the data set S'_n, then the probability of the absolute generalization gap being greater than η is at most twice the probability that for two set of n samples the absolute difference between their empirical risks is greater than $\eta/2$. This is called the *symmetrization lemma* and is expressed as a formula by

$$\Pr\left[\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)| \geq \eta\right] \leq 2\Pr\left[\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}'_n(f)| \geq \tfrac{\eta}{2}\right]. \tag{2.11}$$

Since the right-hand side of (2.11) only depends on the empirical risk, we can restrict the function class $\mathfrak{F}$ to the hypotheses that operate on the union of samples $S_n \cup S'_n$ without loss of generality. This restriction is denoted by $\mathfrak{F}\big|_{S_n\cup S'_n}$. By applying the union bound argument we get as before

$$\Pr\left[\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}'_n(f)| \geq \tfrac{\eta}{2}\right] \leq \left|\mathfrak{F}\big|_{S_n\cup S'_n}\right| \cdot \Pr\left[|\mathcal{E}_n(f) - \mathcal{E}'_n(f)| \geq \tfrac{\eta}{2}\right]. \tag{2.12}$$

Instead of having to deal with the magnitude of the entire hypothesis space, we now only have to consider the restricted hypothesis space. Furthermore, we are now able to derive bounds without having to rely on the underlying data distribution, and instead are now dealing with the samples directly. The performance of the restricted hypothesis space is now measured by how well it separates the given samples. As an example, consider a binary classifier with $\mathfrak{Y} = \{-1, +1\}$. The maximum number of distinct labels for n samples is 2^n, and we have $|\mathfrak{F}\big|_{S_n\cup S'_n}| = 2^n$. In general, the maximum number of distinct labelings of a data set with n samples by a hypothesis space $\mathfrak{F}$ is called the growth function of $\mathfrak{F}$ given n, and is denoted by

$$m_{\mathfrak{F}}(n) \coloneqq \max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}} \left|\left\{\big(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n)\big) \,:\, f\in\mathfrak{F}\right\}\right|. \tag{2.13}$$

Thus, we can further pursue our estimation of the distance of empirical risks by

$$\left|\mathfrak{F}\big|_{S_n \cup S'_n}\right| \cdot \Pr\left[|\mathcal{E}_n(f) - \mathcal{E}'_n(f)| \geq \tfrac{\eta}{2}\right] \leq m_{\mathfrak{F}}(2n) \cdot \Pr\left[|\mathcal{E}_n(f) - \mathcal{E}'_n(f)| \geq \tfrac{\eta}{2}\right]. \tag{2.14}$$

The factor 2 in the growth function originates from the fact that we are considering two sets of n samples, namely S_n and S'_n.

For binary classification, the size of the restricted hypothesis class $|\mathfrak{F}\big|_{S_n \cup S'_n}|$ is given as 2^n and therefore exponential in n, which results in a very large estimate of the probability. Fortunately, this bound can be improved. The term 2^n is based on the assumption that the hypothesis space $\mathfrak{F}$ can produce all possible labelings for n samples. If a set of hypotheses can produce all possible labels on a set of data points, we say that the hypothesis space *shatters* the set. For example a linear classifier in two-dimensional space shatters any set of two data points. However, no linear two-dimensional classifier can shatter any set of more than three data points.

Sauer's Lemma as stated in [76] is a way to formalize this property. It says that if a hypothesis space $\mathfrak{F}$ cannot shatter any data set with size more than k, then we can bound the growth function by

$$m_{\mathfrak{F}}(n) \leq \sum_{i=0}^{k} \binom{n}{i} \leq \mathcal{O}(n^k). \tag{2.15}$$

The parameter k, i.e., the maximum number of points that can be shattered by a space hypotheses $\mathfrak{F}$, is the famous VC dimension of $\mathfrak{F}$. It is often denoted as $d_{\mathrm{VC}}(\mathfrak{F})$ and is arguably the most prominent result in [93]. As an example for VC dimensions, affine classifiers in $\mathbb{R}^p$ have VC dimension $d_{\mathrm{VC}}(\mathfrak{F}) = p + 1$.

In conclusion, the VC dimension can be used to upper bound the growth function and hence as a proxy for the size of the restricted space. The final VC inequality for binary classification with a 0-1-loss function

$$\Pr\left[\sup_{f \in \mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)| \geq \eta\right] \leq 8 m_{\mathfrak{F}}(n) \exp\left(-\frac{n\eta^2}{32}\right) \tag{2.16}$$

follows, see [25]. By introducing the auxiliary variable $\delta = 8 m_{\mathfrak{F}}(n) \exp\left(-\frac{n\eta^2}{32}\right)$ and solving

for η, we finally get

$$\sup_{f \in \mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)| \leq \sqrt{\frac{32 \log(m_{\mathfrak{F}}(n)) + 32 \log(8/\delta)}{n}} \tag{2.17}$$

Since the first introduction in [93], this has become the standard procedure of determining the generalization bounds for learning algorithms.

2.1.2 Results for sparse representation learning algorithms

The results presented in the previous section are based on the VC dimension for the class of hypotheses used to approximate the optimal hypothesis. Classical generalization error bounds that use VC theory date back to the publications [93, 94]. The first bounds for representation learning algorithms of this kind where derived in [29] and then improved upon in various articles [51, 68, 86, 92]. These bounds were further advanced by Lugosi and Pintér in [52] by using information about the cost function for a specific learning problem. They show that the generalization error depends on certain covering numbers of the class of possible cost functions, rather than the VC dimension of the corresponding class of Bayes classifiers. Horváth and Lugosi improved on these results in [41], where they showed that these covering numbers can be bounded using a scale-sensitive dimension if the true cost function is in the class of real valued functions used by the estimator.

Typically, generalization bounds that are based on VC dimension are distribution independent in the sense that they are not influenced by the distribution of the data. This property is to a certain extent beneficial as it guarantees that the bounds hold for any data distribution. But on the other hand, generality always comes at the cost of tightness since all worst case scenarios have to be covered [20, 36]. The first data-dependent variation of VC bounds was proposed in [95], and created a new type of generalization bounds that are now the norm. After this first introduction to data-dependent bounds, other authors offered different approaches to measure the complexity of the set of hypothesis as an alternative to VC-based bounds. A popular variant proposed in [23] uses coverings as a complexity measure. The minimal number of ε-balls (by some metric) required to cover the set of hypothesis functions, which is commonly referred to as covering number, plays a central role in this approach. Another popular variant is based on the Rademacher complexity of

the hypothesis space [5, 47, 48]. The most recognized of these publications is [5] which covers sample complexity bounds for a variety of representation learning algorithms, namely, decision trees, neural networks, and kernel methods.

The main focus of this work lies in the analysis of the sample complexity of sparse representation learning algorithms, but in addition to this we also briefly investigate principal component analysis with both devised bounding frameworks. Results of the sample complexity of PCA are presented in [9, 82]. The sample complexity result derived in [9] has the general form $\sqrt{Cd/n}$ for some variable C and coefficient dimension d. Both of these publications achieve the bounds by leveraging the fact that they only consider distributions within the unit ball.

Only few publications investigate the sample complexity of representation learning algorithms that feature a sparsity promoting prior. In [58] Maurer and Pontil provided results for k-dimensional coding schemes. They propose two bounding schemes. For the first one the results are based on using Rademacher averages as a direct bound of the loss class induced by the reconstruction error. The second approach employs a combination of covering numbers and union bound arguments and yields more complicated bounds. This work covers penalty functions g that are indicator functions for sets of k-sparse representations for some $k \geq 1$. The bounds in [58] exhibit a behavior proportional to $\sqrt{\log(n)/n}$ with respect to the number of samples n. The bound is driven by the term $d^{4-2/k}$, where d is the number of elements in the dictionary and k is the predefined sparsity. The results of this bounding technique cover machine learning methods such as k-means clustering, non-negative matrix factorization, and sparse coding methods.

Vainsencher *et al.* [90] provide an analysis of the sample complexity for sparse dictionary learning algorithms specialized to data distributed on the unit sphere. The key property that their bounding scheme employs is the cumulative coherence of the dictionary $\mathbf{D}$ (also referred to as the Babel function). The coherence of a dictionary $\mathbf{D}$ is a measure of orthogonality of its columns and is defined as $\max_{i \neq j} |\mathbf{d}_i^\top \mathbf{d}_j|$. The cumulative coherence as used in [90] is an extension of the standard coherence to a subsets of columns and is defined as $\max_{|\Lambda|=k} \max_{j \notin \Lambda} \sum_{i \in \Lambda} |\mathbf{d}_i^\top \mathbf{d}_j|$. For signals with unit norm the bounds in [90] show a behavior that is proportional to $\sqrt{pd \log(n)/n}$, where p is the dimension of the signal and d is the number of elements in the dictionary.

In certain settings it is possible to achieve fast rate results of order $1/n$ for sparse repre-

sentation learning under substantially stricter conditions as presented in [58, 90]. It is not clear whether it is a realistic objective to achieve fast rates in the general setting considered in this work.

Another machine learning algorithm that fits into the proposed framework is a supervised variant of dictionary learning [53] also referred to as task-driven dictionary learning. The sparse code is mapped to some target space via a decoding linear operator. Both the dictionary and the decoder are learned together such that the sparse representation with respect to the learned dictionary facilitates the subsequent prediction task. The sample complexity of supervised dictionary learning is investigated in [59], where the bounds are derived for the LASSO, i.e., minimizing the squared ℓ_2-distance between the prediction and the target with an ℓ_1-penalty on the coefficient vector. The result is derived using the covering number of the constraint set and achieves bounds of the order $\sqrt{pd\log(n)/n}$ for overcomplete dictionaries and univariate labels.

2.2 Stochastic Gradient Descent

In the modern age of signal processing, the amount of available data has increased dramatically. While previously the limiting factor for training a model was the number of available data samples, the bottleneck now often is the computing time and hardware limitations. In many cases it has become infeasible to deal with data sets in their entirety during training time as using all available data samples for a determining a parameter update is simply not possible due to memory constraints. This problem dates back to the early days of computing, when storage capacity was inherently limited. One of the first approaches called stochastic approximation method was first described by Robbins and Monroe in [74]. Subsequently Kiefer and Wolfowitz presented their algorithm in [45], which is widely recognized in the machine learning community as the origin of stochastic gradient descent methods. Today, stochastic gradient descent methods have become the method of choice to deal with large-scale optimization problems as typically encountered in machine learning. In the following, we first give a succinct introduction to gradient descent methods and then provide a definition for stochastic gradient methods that rely on single samples and mini-batches.

2.2.1 Gradient descent

In a supervised learning situation we are given samples $z = (\mathbf{x}, y)$ where $\mathbf{x} \in \mathfrak{X}$ is the data point while $y \in \mathfrak{Y}$ is the corresponding label. The data is distributed according to a joint probability distribution $(\mathbf{x}, y) \sim \mathbb{P}$. We consider a loss function $\ell(\hat{y}, y)$ that measures the deviation of the predicted label $\hat{y}$ to the ground truth y. We compute the predicted label via a function $h_{\mathbf{x}}(\mathbf{w}) \in \mathfrak{F}$, where $\mathbf{w}$ denotes the weights that determine the behavior of the function. These two functions are combined to the loss function $f_z(\mathbf{w}) = f_{(\mathbf{x},y)}(\mathbf{w}) = \ell(h_{\mathbf{x}}(\mathbf{w}), y)$. The goal of learning is to determine the weights that parameterize $h_{\mathbf{x}}(\mathbf{w})$ in such a way that the loss $f_z(\mathbf{w})$ for all $(\mathbf{x}, y) \sim \mathbb{P}$ is minimized. Here, we face the same problem as with the generalization error: While the goal is to minimize the expected risk $\mathcal{E}(f)$, this task is impossible to solve as the distribution $\mathbb{P}$ is, in general, unknown. Thus, we have to resort to computing the average on a number of samples $S_n = \{z_1, \ldots, z_n\}$. And as we have seen before, minimizing the empirical risk instead of the expected risk is viable when the function family $\mathfrak{F}$, over which is optimized, is sufficiently restrictive, see [93].

A commonly used technique for reducing the empirical risk is using gradient descent. Therein, each iteration updates the parameterizing weights $\mathbf{w}$ by taking a step in the direction of the negative gradient of $\mathcal{E}_n(f)$,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - a \left(\tfrac{1}{n} \sum_{i=1}^{n} \nabla_{\mathbf{w}} f_{z_i}(\mathbf{w}_t) \right). \tag{2.18}$$

The scalar parameter a is called step size which is typically either chosen as a fixed positive value, or can be adaptively set in each iteration.

It is a well known property of gradient descent algorithms that if the function f is convex and differentiable and the gradient is Lipschitz with constant $L > 0$, then if we run the gradient descent algorithm for t steps with fixed step size $a \leq 1/L$, then the function value at the t-th iteration satisfies

$$\tfrac{1}{n} \sum_{i=1}^{n} \left(f_{z_i}(\mathbf{w}_t) - f_{z_i}(\mathbf{w}^\star) \right) \leq \frac{\|\mathbf{w}_0 - \mathbf{w}^\star\|_2^2}{2at}, \tag{2.19}$$

where $\mathbf{w}^\star$ is the point that yields the minimal value of the cost function. This means that the gradient descent algorithm is guaranteed to converge with rate $\mathcal{O}(1/t)$.

For functions that are bounded from below with a Lipschitz continuous gradient the convergence of gradient descent methods for diminishing step sizes can also be proven. In [8] the authors show that if the variable step size a_t is chosen such that $a_t \to 0$ for $t \to \infty$, then gradient descent converges to a local optimum. This claim only holds under the additional requirement that the step size does not decrease too fast, which would hinder the algorithm from getting close enough to the minimum. In the extreme case setting a_t equal to 0 for all t would result in making no progress towards the minimum at all. Therefore, it is required that the step size satisfies the condition $\sum_{t=1}^{\infty} a_t = \infty$. If these conditions hold, then every limit point of $\{\mathbf{w}_t\}$ is a stationary point of f.

More general convergence properties can be shown for adaptive step sizes. If the step size selection fulfills the so-called Armijo and Wolfe conditions, then it can be shown that the algorithm converges to a local minimum in the sense that $\nabla_{\mathbf{w}} f_z(\mathbf{w}_t) \to 0$ as t goes to infinity as proven in [66]. This result also holds for other gradient-based descent algorithms as long as the step direction the algorithm chooses is bounded away from being orthogonal to the gradient. This convergence guarantee comes at the cost of additional function evaluations. In order to determine a step size that fulfills the conditions, typically several function evaluations are necessary, which adds to the run time of the algorithm.

In the experiment performed in Chapter 6 we will return to the aspect of adaptive step size selection when we introduce a variant to the Armijo condition that adds an averaging step to the rule imposed on the step length.

2.2.2 Stochastic gradient descent

When performing classic gradient descent, the entire data set is considered to update the weights. Stochastic gradient descent simplifies this process significantly. Instead of an exact computation of the gradient of $\mathcal{E}_n(f)$, in each iteration a sample $z_{i(t)}$, $i(t) \in \{1, \ldots, n\}$ is picked randomly, and the update is performed w.r.t. only this sample, i.e.,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - a \nabla_{\mathbf{w}} f_{z_{i(t)}}(\mathbf{w}_t). \tag{2.20}$$

For the next iteration, a new data sample $z_{i(t+1)}$ is picked. Optimization in this way is then a stochastic process that depends on the examples that are randomly picked in each iteration. Note that when averaging the update (2.20) over all available samples, we would

recover the previously introduced gradient descent step (2.18). This way large data sets do not have to be processed in a single update step, but can be split into incremental updates resulting in an optimization procedure that is much less restricted to the computing power available. Another advantage of this method is that it can be applied in systems where new data points continuously acquired and should be incorporated into the model. By being able to successively update the model with newly available data representation learning algorithms that incorporate stochastic gradient update methods can adapt to changes in signal distribution that occur over time while the model is already in place.

For stochastic gradient descent methods, the best convergence result one can hope for is almost sure convergence. This means that if the infinite sum of all function values for each iteration converges to some fixed value, the stochastic algorithm is said to converge. The general requirements for convergence of stochastic gradient descent methods are is a decreasing step size that fulfills the conditions $\sum_i a_i^2 < \infty$ and $\sum_i a_i = \infty$, see [11]. Under sufficient regularity conditions, the best convergence speed is achieved using the step size $a_t \sim t^{-1}$ as proposed in [64]. The proof of convergence follows the basic steps of proving convergence. First define a Lyapunov function that measures the distance of the current weights to the optimal weights $\|\mathbf{w}_t - \mathbf{w}^\star\|_2^2$, then show that the Lyapunov function converges as t goes to infinity, and finally conclude that the convergence of the Lyapunov function implies convergence of the algorithm.

In practice, the step size is commonly set to a fixed value which is used until the reduction in function value falls below a certain threshold for a predetermined number of iterations. Then it is successively reduced until the iteration is aborted by either reaching some convergence criterion, or simply if a certain number of steps has been surpassed.

The convergence of SGD methods to minimizers of strongly-convex functions and to stationary points of non-convex functions is shown in [13]. Additionally, they feature other beneficial properties such as saddle-point avoidance [31, 50] and robustness to noisy training data [34], which further explains their popularity.

In practical application it is often infeasible to process every data sample individually since this approach does not use the processing power to its full potential. When dealing with a large number of data points it is common practice to work on a subset of samples of appropriate size. These subsets are commonly referred to as mini-batches. For a mini-batch

size of k, the update then has the form

$$\mathbf{w}_{t+1} = \mathbf{w}_t - a \left(\frac{1}{k} \sum_{i \in \{t\}} \nabla_{\mathbf{w}} f_{z_i}(\mathbf{w}_t) \right), \tag{2.21}$$

where $\{t\}$ denotes an index set that is an index-subset of size k of all possible indices $\{t\} \subset \{1, \ldots, n\}$ and $|\{t\}| = k$. The mini-batches are chosen randomly, with the constraint that within an epoch, i.e., one entire pass over the available training data, samples are not included in batches twice.

Using mini-batches offers the best from both the batch gradient descent as well as the stochastic gradient descent side. It has the advantage that the model update frequency is higher than for classic batch gradient descent, which allows for optimization progress that is more robust against local minima. Furthermore, it is not necessary to keep all training samples in the memory at all time. Using multiple data samples at once leads to a computationally more efficient algorithm than naive single sample stochastic gradient descent. An aspect to keep in mind is that in order to obtain an accurate measure of the performance of the optimization procedure, it is necessary to sum the error information, i.e., the value of the cost function, across mini-batches. This property will become relevant in the discussion of a geometric stochastic gradient descent method in Chapter 6.

Chapter 3

Sample Complexity in Relation to Stochastic Gradient Descent

In the previous Section 2.1 we introduced the generalization error as a measure of how well a function f generalizes to previously unseen data. It measures the distance between the expected value of the function value evaluated over the underlying distribution of the data with the empirical average of the function value over a set of n data samples.

In addition to asking how well the function generalizes, we can also ask how optimal the hypothesis that the algorithm provides actually is. We have already seen that the hypothesis can at best minimize the empirical average. However, there are even more sacrifices in accuracy that have to be made. In order to provide a more detailed analysis, we introduce the following notations which all denote functions that are optimal in some sense.

$$f^\star \coloneqq \arg\min_f \mathcal{E}(f) \tag{3.1}$$

$$f^\star_{\mathfrak{F}} \coloneqq \arg\min_{f\in\mathfrak{F}} \mathcal{E}(f) \tag{3.2}$$

$$f^\star_{\mathfrak{F},n} \coloneqq \arg\min_{f\in\mathfrak{F}} \mathcal{E}_n(f) \tag{3.3}$$

$$\hat{f}_{\mathfrak{F},n} \coloneqq \mathrm{ALG}(\mathfrak{F}) \tag{3.4}$$

The function $f^\star$ is the overall minimizer of the problem and minimizes the expected risk. It denotes the hypothesis that perfectly fits to the data distribution of the considered type of signals. Since it is not realistic to find this overall best function, we restrict the search to a space of hypotheses $\mathfrak{F}$ that has a structure suited to the discussed signal class. This

hypothesis space is typically defined in such a way that it can be fully described by a set of parameterizing weights. This is beneficial as it allows an algorithmic adaptation of the function to optimally represent the given training data. The function in this hypothesis space that achieves the smallest expected risk is denoted by $f^\star_{\mathfrak{F}}$. Since in general the underlying distribution of the data is unknown, the entire optimization procedure is restricted to a set of n available training samples. Therefore, instead of the expected error, we actually minimize the empirical error during the learning procedure. The function that produces the smallest empirical error for a set of n data points is denoted by $f^\star_{\mathfrak{F},n}$. Finally, in practice not even this empirical minimum is reached. Typical optimization algorithms such as gradient descent can only be expected to approximate the optimal solution up to a certain accuracy as they are by design dependent on inexact line-searches which in general only approximate the empirical minimum. The best function found by the optimization algorithm is denoted by $\hat{f}_{\mathfrak{F},n}$.

The expected difference of the algorithm output $\hat{f}_{\mathfrak{F},n}$ and the optimal function $f^\star$ w.r.t. the sample distribution is called the excess error. Formally, it is defined as

$$E = \mathbb{E}_{\mathbf{X}}[\mathcal{E}_n(f) - \mathcal{E}(f)]. \tag{3.5}$$

To see the relation between the excess error and stochastic gradient descent, we follow the work of Bottou and Bousquet in [15]. As a first step, we can expand the excess error to include the previously defined minimizing functions as

$$\begin{aligned} E &= \mathbb{E}[\mathcal{E}_n(\hat{f}_{\mathfrak{F}}) - \mathcal{E}(f^\star)] \\ &= \underbrace{\mathbb{E}[\mathcal{E}(f^\star_{\mathfrak{F}}) - \mathcal{E}(f^\star)]}_{E_{\text{app}}} + \underbrace{\mathbb{E}[\mathcal{E}(f^\star_{\mathfrak{F},n}) - \mathcal{E}(f^\star_{\mathfrak{F}})]}_{E_{\text{est}}} + \underbrace{\mathbb{E}[\mathcal{E}_n(\hat{f}_{\mathfrak{F},n}) - \mathcal{E}(f^\star_{\mathfrak{F},n})]}_{E_{\text{opt}}}. \end{aligned} \tag{3.6}$$

The respective expectations are taken w.r.t. the random choice of n training samples. This decomposition consists of three error measures.
The *approximation error*

$$E_{\text{app}} \coloneqq \mathbb{E}[\mathcal{E}(f^\star) - \mathcal{E}(f^\star_{\mathfrak{F}})] \tag{3.7}$$

measures how well functions $f \in \mathfrak{F}$ can approximate the optimal hypothesis $f^\star$. This error can be controlled via the cardinality of the function class.

The *estimation error*

$$E_{\text{est}} \coloneqq \mathbb{E}[\mathcal{E}(f_{\mathfrak{F}}^{\star}) - \mathcal{E}_n(f_{\mathfrak{F},n}^{\star})] \tag{3.8}$$

states how accurately the expected risk $\mathcal{E}(f_{\mathfrak{F}}^{\star})$ is approximated by the empirical risk $\mathcal{E}_n(f)$. An increase in sample size typically reduces this error.
The *optimization error*

$$E_{\text{opt}} \coloneqq \mathbb{E}[\mathcal{E}_n(f_{\mathfrak{F},n}^{\star}) - \mathcal{E}_n(\hat{f}_{\mathfrak{F},n})] \tag{3.9}$$

describes the error that occurs when the optimization algorithm does not find the exact empirical minimizer. This error is a consequence of the iterative design of optimization algorithms. Choosing appropriate step sizes can reduce this error. Figure 3.1 provides an illustration of the relationship of the optimal functions $f^{\star}$, $f_{\mathfrak{F}}^{\star}$, $f_{\mathfrak{F},n}^{\star}$, $\hat{f}_{\mathfrak{F},n}$, and the respective errors.

With this decomposition it becomes obvious that optimization of the expected risk involves an inherent trade-off. The optimization of a learning problem involves three principal variables that have to be considered for each optimization problem, as well as two inherent constraints. The first variable is the magnitude of the hypothesis space $\mathfrak{F}$, the second is the optimization accuracy ρ, and the third is the number of training data n. The general constraints are the maximal number of available training samples $n_{\max}$ and the maximal computation time $T_{\max}$. As an optimization problem, this can be expressed as

$$\begin{aligned} &\min_{\mathfrak{F},\rho,n} E = E_{\text{app}} + E_{\text{est}} + E_{\text{opt}} \\ &\text{subject to } n \leq n_{\max},\ T(\mathfrak{F},\rho,n) \leq T_{\max}. \end{aligned} \tag{3.10}$$

There are effectively two problem scenarios that can occur. The first are small scale learning problems that are constrained by the maximal number of available examples. For small scale problems the computation time is not an issue. The optimization accuracy can be set arbitrarily low in order to minimize E_{opt}, and furthermore E_{est} can be minimized by setting $n = n_{\max}$. The remaining problem is an approximation-estimation trade-off, where by modifying the cardinality of the set of hypotheses $\mathfrak{F}$ the estimation error can be leveraged against the approximation error. This setting is a standard optimization problem and is extensively studied in optimization literature. The second case are learning scenarios where a

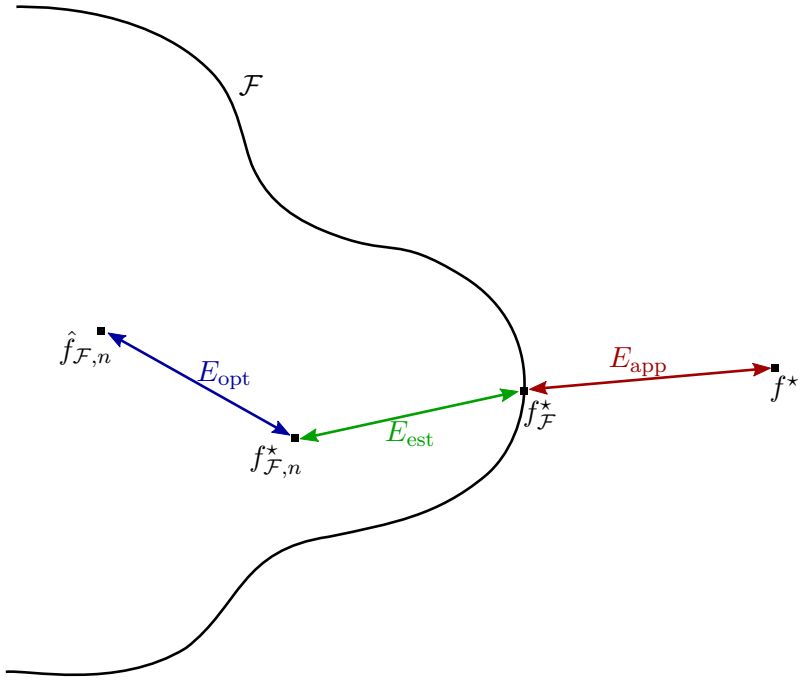


Figure 3.1: An illustration of the excess error decomposition. The function $f^\star$ minimizes the empirical risk. Due to modeling constraints, we can only find a solution within a certain function class $\mathfrak{F}$. The optimal (constrained) solution is $f^\star_{\mathfrak{F}}$. But not even this can be achieved since we do not know the underlying distribution and are limited to using a set of n training samples. The optimal solution within function class $\mathfrak{F}$ that can be achieved with n samples is denoted by $f^\star_{\mathfrak{F},n}$. Finally, due to inaccuracies of optimization algorithms the best achievable function in a real-world learning scenario is $\hat{f}_{\mathfrak{F},n}$.

large number of training data is available and using all of the available data becomes infeasible. This is the situation that occurs for large scale learning problems where the maximal computing time becomes the primary constraint. In this case approximate optimization methods, such as stochastic gradient descent, achieve better values for the expected risk since they are capable of processing more training examples in a predefined time frame.

When investigating the behavior in the limits of (3.10), the convergence rate of the excess error is equal to the convergence rate of its slowest term. Any computational effort made to ensure that a single term converges faster is wasted. Therefore, all the terms in (3.10) should converge at similar rates. It has been shown in [15] that SGD methods

perform significantly worse than their non-stochastic counterparts when it comes to general optimization performance, i.e., the asymptotic behavior of the time for SGD methods to achieve a certain accuracy behaves with $1/\rho$ in contrast to $n \log 1/\rho$ for classic gradient descent. However, when considering the asymptotic behavior of time required to achieve a certain excess error ε, the asymptotic behavior of stochastic methods is $1/\varepsilon$, whereas classic gradient descent acts as $\frac{1}{\varepsilon^{1/\alpha}} \log(1/\varepsilon)$. Thus, in a large scale setup, where the limiting factor is the computation time, stochastic learning methods perform asymptotically better.

In a real world learning scenario, the choice of function class $\mathfrak{F}$ is typically determined during model selection and thus the approximation error is a result of the choice of model and the considered features. In the following, we consider fixed classes of functions $\mathfrak{F}$, and therefore omit the discussion of the behavior of E_{app}. Instead, the focus is put on the discussion of the error that is related to the optimization algorithm.

In typical modern learning scenarios a huge amount of data is available and in many cases it is not uncommon that new data samples arrive after the learning algorithm has been initiated. This is especially common for unsupervised learning algorithms, where new, unlabeled data is continuously provided to the algorithm. Thus, it is a reasonable question to ask what happens to the specific errors E_{est} and E_{opt} when the number of available samples n increases. The influence on the estimation error, however, is considerable as more samples allow for a better approximation of the true minimum within the function class $\mathfrak{F}$. Therefore, when $n \to \infty$, we get $E_{\text{est}} \to 0$ under the assumption that the drawn samples represent the underlying probability distribution. This result tells us that if we want to achieve a certain accuracy of the result, i.e., we want the total excess error to be smaller than some predefined bound, then the more samples are included in the optimization process, the more tolerance we get for the optimization error E_{opt}. This is illustrated in Figure 3.2.

For standard gradient descent algorithms this implies that the more samples we use for the learning procedure, the sooner a certain excess error is reached and the algorithm can be terminated, a conclusion that follows from the law of large numbers. For stochastic gradient descent it has the additional consequence that it is possible to adaptively increase the set of training samples by injecting new data into the learning algorithm during training in order to effectively lower the estimation error. In essence, the amount of training samples that are made available to the optimization algorithm can be used to control the estimation error

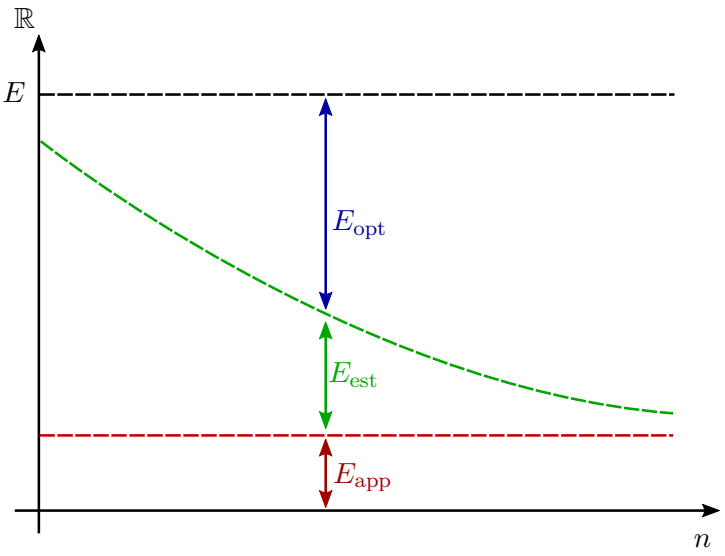


Figure 3.2: For an increasing number of training samples the approximation error remains the same since the available class of hypothesis functions remains unchanged. The estimation error decreases due to the property that the more samples are available, the better the expected error is approximated by the empirical error. For a fixed total error this allows for a larger optimization error, which can lead to a speed-up in convergence time of the optimization algorithm.

directly and used as a way to counteract any performance penalties that using a gradient approximation entails.

Apart from this informal observation there exists a more concrete relation between the excess error and the sample complexity. Assume that the class of hypotheses $\mathfrak{F}$ is fully parameterized by weights $\mathbf{w} \in \mathbb{R}^d$. Then according to Vapnik and Chervonenkis [93] and Bousquet [14] the generalization error for a sample size n can be bounded

$$\mathbb{E}\left[\sup_{f \in \mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)|\right] \leq \eta(n), \tag{3.11}$$

with high probability, where the upper bound $\eta(n)$ can be expressed in terms of the VC dimension as we showed in Equation (2.17). From this property it is possible to derive bounds for the estimation error as defined in (3.8). By expanding the estimation error with

the zero terms $\mathcal{E}_n(f^\star_{\mathfrak{F}}) - \mathcal{E}_n(f^\star_{\mathfrak{F}})$ and $\mathcal{E}_n(f^\star_{\mathfrak{F},n}) - \mathcal{E}_n(f^\star_{\mathfrak{F},n})$ we can reshape the estimation error

$$\begin{aligned} E_{\text{est}} &= \mathbb{E}\left[\mathcal{E}(f^\star_{\mathfrak{F},n}) - \mathcal{E}(f^\star_{\mathfrak{F}})\right] \\ &= \mathbb{E}\left[(\mathcal{E}(f^\star_{\mathfrak{F},n}) - \mathcal{E}_n(f^\star_{\mathfrak{F},n})) + (\mathcal{E}_n(f^\star_{\mathfrak{F},n}) - \mathcal{E}_n(f^\star_{\mathfrak{F}})) + (\mathcal{E}_n(f^\star_{\mathfrak{F}}) - \mathcal{E}(f^\star_{\mathfrak{F}}))\right] \\ &\leq 2\mathbb{E}\left[\sup_{f\in\mathfrak{F}} |\mathcal{E}_n(f) - \mathcal{E}(f)|\right] + \mathbb{E}\left[\mathcal{E}_n(f^\star_{\mathfrak{F},n}) - \mathcal{E}_n(f^\star_{\mathfrak{F}})\right] \\ &\leq 2\eta(n). \end{aligned} \tag{3.12}$$

The last inequality follows from the fact that $f^\star_{\mathfrak{F},n}$ minimizes $\mathcal{E}_n$, and thus the term $(\mathcal{E}_n(f^\star_{\mathfrak{F},n}) - \mathcal{E}_n(f^\star_{\mathfrak{F}}))$ is either negative or equal to zero. Due to its generality, this bound is pessimistic in several cases. More sophisticated generalization error based bounds can be derived under more rigorous assumptions on the hypothesis class and the underlying distribution. We will not cover these bounds in this work and refer the interested reader to [4, 6, 14], where faster rates are devised for function classes that are scaled convex hulls of some finite-dimensional base class.

In summary, we can derive a bound for the estimation error of a learning algorithm if we are able to restrict the generalization error. Thus, an analysis of the sample complexity also provides concrete insights on the optimization procedure.

Chapter 4

Establishing Sample Complexity Bounds

There are various ways to find upper bounds for the generalization error, and thus the sample complexity, of representation learning methods. The most prominent one is arguably the method proposed by Vapnik and Chervonenkis that invokes the VC dimension to bound the magnitude of the hypothesis space, which we discussed in Chapter 2. This work's focus is on two approaches for deriving generalization error bounds that have proven to be applicable to a multitude of different learning frameworks and are particularly suited for deriving bounds for sparse representation learning algorithms. In this section I will first provide the definitions that are necessary to formulate both bounding schemes, give a rough sketch of the steps that have to be performed in order to derive the generalization bounds, and finally give a concrete in depth example by investigating the error bounds of sparse dictionary learning. The first strategy is a generalization of the bounding scheme introduced in [33], which we discuss in detail in the conclusion of this chapter.

Before we start with the derivation of the bounds, a short word about the learning setting. In Section 2.1 we introduced the generalization error for supervised learning methods. In the following discussion, the main focus will be on unsupervised methods, i.e., instead of being given a tuple $(\mathbf{x}, y)$, where $\mathbf{x}$ is some data point and y is the corresponding label, the data is comprised solely of unlabeled data $\mathbf{x} \in \mathfrak{X}$ and the objective is to minimize some cost function that maps from $\mathfrak{X}$ to the non-negative real numbers $\mathbb{R}^+$. This cost function for a single data sample $\mathbf{x}$ can be typically written as

$$f_{\mathbf{x}}(\mathbf{W})\colon \mathfrak{X} \to \mathbb{R}^+. \tag{4.1}$$

The matrix $\mathbf{W}$ contains the weights that are adjusted during the learning process. They fully

describe the class of hypothesis functions and are commonly limited to some constraint set $\mathfrak{C}$. The general structure of f remains unchanged throughout the optimization procedure and determining the best hypothesis function becomes a matter of finding the optimal weights to minimize the function value for the given training data.

Based on the parameterization of the function class the generalization error of the function class w.r.t. a set of data samples $\mathbf{x}_i$ is then measured via

$$\sup_{\mathbf{W}\in\mathfrak{C}} \left| \frac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{W}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{W})] \right|, \tag{4.2}$$

as proposed in Section 2.1.1. Taking the supremum over the function class $\mathfrak{F}$ is replaced with taking the supremum over all parameterizing weights $\mathbf{W} \in \mathfrak{C}$.

4.1 Hoeffding's Inequality & Covering Numbers

The first results for bounding the generalization, and with that the sample complexity of learning algorithms, date back to results in [93], where the VC dimension was first introduced. Bounds of this manner were discussed in detail in Chapter 2. The first bounding scheme we propose is closely related to the original VC dimension bounding technique. It hinges on the application of Hoeffding's inequality and the covering number bounds for the set of parameters $\mathfrak{C}$ that define the model. In the following section we summarize all necessary concepts for this first generalization error bounding approach.

Union bound and Hoeffding's inequality

The first step in this approach is to use a union bound argument, which is a standard tool in probability theory. As its name suggests it provides a way to bound the probability that a union of events occurs. We use the definition as proposed in [21].

Theorem 4.1 (Union bound/Boole's inequality)**.** *Given the countable or finite set of events B_i, the probability that at least one event happens is less than or equal to the sum of all probabilities of the events taken individually. Formally, this can be expressed by*

$$\Pr\left[\bigcup_i B_i\right] \leq \sum_i \Pr[B_i].$$

After applying the union bound argument, we utilize Hoeffding's inequality, which is a technique to bound the probability that the sum of bounded independent random variables deviates too much from its expected value. The original derivation of this inequality can be found in [40, 70]. The following definition is a slight modification of the original inequality in that instead of measuring the deviation of the empirical average of a random variable from its expectation, we apply a smooth function ϕ to the random variable.

Theorem 4.2 (Hoeffding's inequality)**.** *Let $X_1, \dots, X_n$ be i.i.d. random vectors in $\mathfrak{X}$ following the distribution $\mathbb{P}$, and let $\phi\colon \mathfrak{X} \to \mathbb{R}$ be a smooth function. If for all $i \in \{1, \dots, n\}$*

$$a_i \leq \phi(X_i) \leq b_i,$$

then

$$\Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n}\left(\phi(X_i) - \mathbb{E}[\phi(X_i)]\right)\right| \geq t\right] \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Or equivalently,

$$\left|\tfrac{1}{n}\sum_{i=1}^{n}\left(\phi(X_i) - \mathbb{E}[\phi(X_i)]\right)\right| < \sqrt{\frac{\sum_{i=1}^{n}(b_i - a_i)^2 \log(2/\delta)}{2n^2}}$$

with probability at least $1 - \delta$.

Lipschitz Continuity

Lipschitz continuity of a function can be seen as a limit of how fast a function can change. It is a tightening of the continuity property. The standard definition states that a function f that maps from $\mathfrak{X} \subset \mathbb{R}^p$ to $\mathbb{R}$ is called Lipschitz continuous if there exists a constant C such that for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathfrak{X}$ and a norm $\|\cdot\|$ defined on $\mathfrak{X}$ we have

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq C\|\mathbf{x}_1 - \mathbf{x}_2\|,$$

see [27]. We say that the function f is Lipschitz on $\mathfrak{X}$ with Lipschitz constant C.

For this bounding method, we require the Lipschitz continuity of the cost function $f_{\mathbf{x}}(\mathbf{W})$ with respect to $\mathbf{W}$, where the Lipschitz constant is not dependent on $\mathbf{x}$.

Covering Numbers

In order to estimate the final bounds, we need a tool to measure the "size" of a set. While for a finite set, this is simply the cardinality of the set, it becomes more difficult to provide this number for sets with infinitely many elements.

We will be dealing with subsets of $\mathfrak{V} \subseteq \mathbb{R}^p$, endowed with a distance metric d. A distance metric is a mapping $\mathrm{d}\colon \mathfrak{V} \times \mathfrak{V} \to \mathbb{R}$ that is non-negative, symmetric, and fulfills the triangle inequality. With a given distance metric, we can define balls with certain radii. For a point $\mathbf{v} \in \mathfrak{V}$ and $\varepsilon > 0 \in \mathbb{R}$ the open ball centered at $\mathbf{v}$ with radius ε is denoted by $B(\mathbf{v}, \varepsilon)$. Given a distance metric, we are able to give the following definition.

Definition 4.3 (Covering number)**.** An ε-cover of a set $\Theta \subset \mathfrak{V}$ with respect to a metric d is a set $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\} \subset \Theta$ such that for each $\mathbf{v} \in \Theta$, there exists an $i \in \{1, \ldots, N\}$ such that $\mathrm{d}(\mathbf{v}, \mathbf{v}_i) \leq \varepsilon$.
The ε-covering number $N^{\mathrm{cov}}(\Theta, \varepsilon)$ is the cardinality of the smallest ε-cover.

Covering numbers provide a measure of complexity for a family of functions. The larger the sample complexity of a set that parameterizes a class of functions, the richer the class of functions is. The covering numbers of a variety of constraint sets are presented in Table 4.1, see [33].

Example 4.4 (Covering number of the unit ball)**.** As an example, we determine the covering number of the unit Euclidean ball in $\mathbb{R}^p$. In general, a ball around a point $\mathbf{x} \in \mathbb{R}^p$ with radius ε is defined as $B(\mathbf{s}, \varepsilon) = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{s} - \mathbf{x}\|_2 \leq 1\} \subset \mathbb{R}^p$. The unit ball is defined as the ball around the origin $B_1 = B(\mathbf{0}_p, 1)$. The covering number of the unit ball satisfies

$$N^{\mathrm{cov}}(B_1, \varepsilon) \leq \left(1 + \frac{2}{\varepsilon}\right)^p. \tag{4.3}$$

In order to see this, first, note that the covering number is smaller or equal to the maximal ε-packing number. That is,

$$M(B_1, \varepsilon) \coloneqq \max\{m \in \mathbb{N} : \{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \subset B_1, \min_{i \neq j} \|\mathbf{v}_i - \mathbf{v}_j\| > \varepsilon\}. \tag{4.4}$$

Let $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ with $\mathbf{v}_i \in B_1$ for $i = 1, \ldots, m$ be a maximal packing of B_1. Then for all $\mathbf{v} \in B_1 \setminus \mathcal{V}$ there exists an i such that $\|\mathbf{v} - \mathbf{v}_i\| \leq \varepsilon$. If this were not the case, we could

construct a bigger packing by adding $\mathbf{v}$ to $\mathcal{V}$. Thus, $\mathcal{V}$ is an ε-covering of B_1, and since $N^{\text{cov}}(B_1, \varepsilon)$ is a minimal ε-cover the inequality

$$N^{\text{cov}}(B_1, \varepsilon) \leq M(B_1, \varepsilon) \tag{4.5}$$

follows.

Next, notice that the packing of a set can be equivalently defined as $\bigcap_{i=1}^{m} B(\mathbf{v}_i, \varepsilon/2) = \emptyset$, i.e., the balls $B(\mathbf{v}_i, \varepsilon/2)$ are disjoint. Furthermore, the union over all balls around $\mathbf{v}_i$ with radius $\varepsilon/2$ for all $i = 1, \ldots, m$ is a subset of a ball of radius $(1+\varepsilon/2)$ which can be expressed as

$$\bigcup_{i=1}^{m} B\big(\mathbf{v}_i, \tfrac{\varepsilon}{2}\big) \subset B\big(\mathbf{0}_p, 1 + \tfrac{\varepsilon}{2}\big). \tag{4.6}$$

Note that the ball $B(\mathbf{0}_p, 1 + \varepsilon/2)$ can be equivalently written as $(1 + \varepsilon/2)B_1$. By taking the volume over both sides of inequality (4.6) we get

$$M(B_1, \varepsilon) \operatorname{vol}\big(\tfrac{\varepsilon}{2} B_1\big) = \operatorname{vol}\Big(\bigcup_{i=1}^{m} B(\mathbf{v}_i, \tfrac{\varepsilon}{2})\Big) \leq \operatorname{vol}\big((1 + \tfrac{\varepsilon}{2})B_1\big). \tag{4.7}$$

By solving this term for $M(B_1, \varepsilon)$ and since for a ball in $\mathbb{R}^p$ the volume has the property $\operatorname{vol}(\varepsilon B_1) = \varepsilon^p \operatorname{vol}(B_1)$ for $\varepsilon > 0$, we get

$$M(B_1, \varepsilon) \leq \left(1 + \frac{2}{\varepsilon}\right)^p, \tag{4.8}$$

which proves (4.3). On a side note, Vershynin shows in [96] that the same covering bound number also applies to the Euclidean unit sphere $S_{(p-1)}$.

The ε-cover of a set Θ can be visualized as a collection of balls with radius ε that cover the set Θ, as illustrated in Figure 4.1.

These are the main tools required for establishing the generalization error bound via the Hoeffding's inequality and covering number technique. Before we proceed in presenting the concrete strategy in Section 4.3, we first introduce the tools required for the second investigated bounding strategy.

	$\mathfrak{C}$	h	C
Unit sphere	$S_{(p-1)}$	p	3
Unit norm columns	$\mathrm{Ob}(p,d)$	pd	3
Separable unit norm columns	$\mathrm{Ob}_\times$	$\sum_i^q p_i d_i$	3
Stiefel	$\mathrm{St}(p,d)$	$pd - p(p+1)/2$	$3\pi e^\pi$

$$S_{(p-1)} \coloneqq \{\mathbf{x} \in \mathbb{R}^p \,:\, \|\mathbf{x}\|_2 = 1\}$$

$$\mathrm{Ob}(p,d) \coloneqq \{\mathbf{D} \in \mathbb{R}^{p\times d} \,:\, (\mathbf{D}^\top \mathbf{D})_{ii} = 1,\ i = 1,\ldots,d\}$$

$$\mathrm{Ob}_\times \coloneqq \mathrm{Ob}(p_1,d_1) \times \ldots \times \mathrm{Ob}(p_q,d_q)$$

$$\mathrm{St}(p,d) \coloneqq \{\mathbf{D} \in \mathbb{R}^{p\times d} \,:\, \mathbf{D}^\top \mathbf{D} = \mathbf{I}_d\}$$

Table 4.1: Covering number bounds in the shape $N^{\mathrm{cov}}(\mathfrak{C},\varepsilon) \leq (C/\varepsilon)^h$ for a variety of constraint sets.

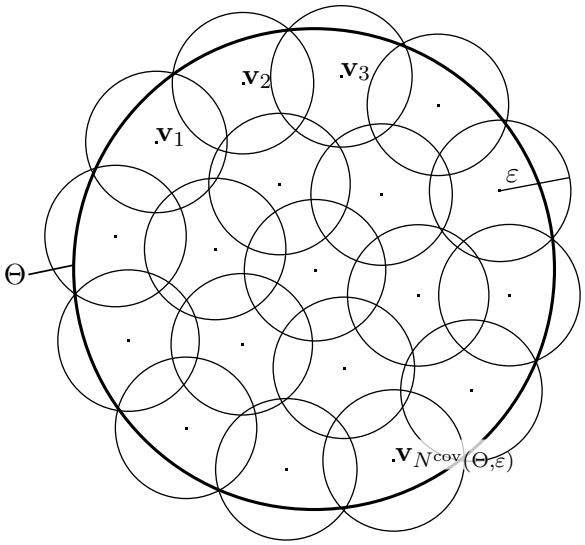


Figure 4.1: A minimal ε-covering of a set Θ is a collection of elements $\mathbf{v}_1,\ldots,\mathbf{v}_{N^{\mathrm{cov}}(\Theta,\varepsilon)}$ such that for each $\mathbf{v} \in \Theta$ there exists $i \in \{1,\ldots,N^{\mathrm{cov}}(\Theta,\varepsilon)\}$ such that $\mathrm{d}(\mathbf{v},\mathbf{v}_i) \leq \varepsilon$ and $N^{\mathrm{cov}}(\Theta,\varepsilon)$ is as small as possible. In other words, the union of balls with radius ε that are centered around $\mathbf{v}_i$ cover Θ.

4.2 McDiarmid's Inequality & Rademacher Complexity

The second method for bounding the sample complexity is focused around the use of the Rademacher and Gaussian complexity of the hypothesis class $\mathfrak{F}$. These complexities serve as a measure of the expressiveness of a function class. Approaches based on methods of this type were first used in [5, 47, 48, 61] for bounding purposes.

McDiarmid's Inequality

The first central step in this method is an application of McDiarmid's inequality. It plays a similar role to Hoeffding's inequality used in the previous method in that it provides an upper bound on the probability that a function that takes random variables $x_1, \ldots, x_m$ as its input deviates from its mean. In fact, it can be shown that McDiarmid's inequality implies Hoeffding's inequality [87].

Theorem 4.5 (McDiarmid's Inequality)**.** *Let $X_1, \ldots, X_m$ be independent random variables all taking values in the set $\mathfrak{X}$. Further, let $f\colon \mathfrak{X}^m \to \mathbb{R}$ be a function of $X_1, \ldots, X_m$ that satisfies the inequality*

$$|f(x_1, \ldots, x_i, \ldots, x_m) - f(x_1, \ldots, x_i', \ldots, x_m)| \leq b_i.$$

for all $i \in \{1, \ldots, m\}$ and all $x_1, \ldots, x_m, x_i' \in \mathfrak{X}$. Then for any $\varepsilon > 0$ the property

$$\Pr\left[f(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] \geq \varepsilon\right] \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{i=1}^{m} b_i^2}\right)$$

holds.

Rademacher & Gaussian Complexity

Definition 4.6 (Empirical Rademacher complexity)**.** Let $\mathfrak{F}$ be a family of real valued functions defined on the set $\mathfrak{X}$. Furthermore, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ be a matrix containing a set of samples $\mathbf{x}_i \in \mathfrak{X}$ for $i \in \{1, \ldots, n\}$. The *empirical Rademacher complexity* of $\mathfrak{F}$

w.r.t. the set of samples $\mathbf{X}$ is defined as

$$\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}) \coloneqq \mathbb{E}_{\sigma}\left[\sup_{f\in\mathfrak{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(\mathbf{x}_i)\right],$$

where $\sigma_1, \ldots, \sigma_n$ are independent Rademacher variables, i.e., random variables that fulfill $\Pr[\sigma_i = 1] = \Pr[\sigma_i = -1] = 1/2$ for $i = 1, \ldots, n$.

The empirical Rademacher complexity is a measure of how well the function class $\mathfrak{F}$ correlates with the random labels $\{\sigma_1, \ldots, \sigma_n\}$ over a set of n samples $\mathbf{X}$. Since σ_i are picked at random, the empirical Rademacher complexity quantifies how well the function class correlates with noise, giving an indication of the richness of the function family $\mathfrak{F}$.

Definition 4.7 (Rademacher complexity)**.** Let the samples $\mathbf{x}_i$ be independently drawn according to a probability distribution $\mathbb{P}$. Then the *Rademacher complexity of the function class* $\mathfrak{F}$ is defined as the expectation of the empirical Rademacher complexity over all sets of samples of size n and denoted by

$$\mathfrak{R}_n(\mathfrak{F}) \coloneqq \mathbb{E}_{\mathbf{X}}\left[\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F})\right].$$

Note that this definition slightly deviates from the original definition given in [5, 91] which includes an absolute value surrounding the sum, i.e., $\mathbb{E}_{\sigma}[\sup_{f\in\mathfrak{F}}(1/n)|\sum_{i=1}^{n}\sigma_i f(\mathbf{x}_i)|]$. The version stated here can be found in, e.g., [62, 81]. Both definitions are equivalent if the function class $\mathfrak{F}$ is closed under negation, i.e., if $f \in \mathfrak{F}$ implies that $-f \in \mathfrak{F}$. Definition 4.6 has the property that it vanishes for function classes consisting of a single constant function and is always dominated by the standard Rademacher complexity, see [60]. The derivation of the bounds does not depend significantly upon the choice of which definition is used and only slight modifications have to be made during the bounding process.

Analogously to the Rademacher complexity, we define the Gaussian complexity.

Definition 4.8 (Gaussian complexity)**.** Let $\mathfrak{F}$ be a family of real valued functions defined on the set $\mathfrak{X}$. Furthermore, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ be a matrix comprised of a set of samples $\mathbf{x}_i \in \mathfrak{X}$ for all $i \in \{1, \ldots, n\}$. The *empirical Gaussian complexity* of $\mathfrak{F}$ w.r.t. the set of

samples $\mathbf{X}$ is defined as

$$\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}) \coloneqq \mathbb{E}_{\gamma}\left[\sup_{f\in\mathfrak{F}} \tfrac{1}{n}\sum_{i=1}^{n} \gamma_i f(\mathbf{x}_i)\right],$$

where $\gamma_1, \ldots, \gamma_n$ are independent normal Gaussian variables, i.e., Gaussian random variables with expectation 0 and variance 1 which can also be expressed as $\gamma_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$.

Further assume that all $\mathbf{x}_i$ are independently drawn according to a probability distribution $\mathbb{P}$. Then the *Gaussian complexity of* $\mathfrak{F}$ is defined as

$$\mathfrak{G}_n(\mathfrak{F}) \coloneqq \mathbb{E}_{\mathbf{X}}\left[\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F})\right].$$

The definitions of Rademacher and Gaussian complexity are identical except for the different distributions of the included random variables. Therefore, it is not surprising that there exist a direct relation between them as shown in [32].

Lemma 4.9. *Let $\mathfrak{F}$ be a class of function mappings from $\mathfrak{X}$ to $\mathbb{R}$. For any set of samples $\mathbf{X}$ the empirical Rademacher complexity can be upper bounded by*

$$\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}) \leq \sqrt{\pi/2}\,\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}).$$

Lemma 4.9 only provides a way to upper bound the Rademacher complexity by the Gaussian complexity. It is in fact possible to also give a lower bound using the Gaussian complexity but we only need the upper bound for the following discussion. A proof for the upper and lower bound can be found in [49]. Both complexity numbers give an indication of the expressiveness of the investigated function class. As an intuitive example, consider a binary classification problem. We can model this learning problem with a function class $\mathfrak{F}$ that maps the data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ to $\{-1, 1\}$. If $\mathfrak{F}$ consists of a single hypothesis f, then the empirical Rademacher complexity would be 0 as $f(\mathbf{x}_i) = \sigma_i$ with probability $1/2$ for any randomly chosen $\boldsymbol{\sigma}$. On the other hand, if $\mathfrak{F}$ shatters the training set $\mathbf{X}$, i.e., for every possible distribution of $\boldsymbol{\sigma}$ there exists a hypothesis f that reproduces the labels that correspond to $\boldsymbol{\sigma}$, then the empirical Rademacher complexity of $\mathfrak{F}$ would be 1, which is indicative of the property that a higher empirical Rademacher complexity corresponds to a more expressive set of hypotheses. There actually is a close connection between the covering number and Rademacher complexity. Specifically, the empirical Rademacher complexity can

be bounded in terms of the covering number, as proposed by Dudley in [26]. The covering number, in turn, can be bounded in terms of the VC dimension as proven in [35].

Slepian's Lemma

Slepian's Lemma is a Gaussian comparison inequality named after David Slepian, who proved it in 1962. Its original definition is given in [83].

Lemma 4.10 (Slepian's Lemma)**.** *Given two Gaussian random variables* $X = (X_1, \ldots, X_n)$ *and* $Y = (Y_1, \ldots, Y_n)$ *in* $\mathbb{R}^n$ *satisfying* $\mathbb{E}[X] = \mathbb{E}[Y] = 0$*,* $\mathbb{E}[X_i^2] = \mathbb{E}[Y_i^2]$*,* $i = 1, \ldots, n$ *and* $\mathbb{E}[Y_i Y_j] \leq \mathbb{E}[X_i X_j]$ *for* $i \neq j$*, then for all real numbers* $u_1, \ldots, u_n$ *the inequality*

$$\Pr[Y_1 \leq u_1, \ldots, Y_n \leq u_n] \leq \Pr[X_1 \leq u_1, \ldots, X_n \leq u_n]$$

holds.

Following [49] this Lemma can be reformulated as subsequently shown.

Corollary 4.11. *Let* X *and* Y *be two centered Gaussian random vectors in* $\mathbb{R}^n$ *such that* $\mathbb{E}[(Y_i - Y_j)^2] \leq \mathbb{E}[(X_i - X_j)^2]$ *for* $i \neq j$*. Then*

$$\mathbb{E}\left[\sup_{1 \leq i \leq n} Y_i\right] \leq \mathbb{E}\left[\sup_{1 \leq i \leq n} X_i\right].$$

This is the form of Slepian's Lemma that we will use in the following bounding process. These are all the prerequisites for the second strategy. In the next section we first present the setting for the first worked example before proposing both step-by-step bounding procedures.

4.3 Worked Example: Dictionary Learning

After having established all necessary basic principles for the two techniques that we want to examine, we introduce a step-by-step approach to deriving sample complexity bounds with both frameworks by means of a worked example.

Sparse dictionaries are a well known representative of synthesis models. There are two types of dictionaries, analytic dictionaries and learned dictionaries. Analytic dictionaries are constructed on mathematical models for a specific type of signals that they represent. Typical

examples are wavelets [56] and curvelets [85]. It is well known that learned dictionaries, i.e., dictionaries that are trained on a representative set of training signals such that they offer a maximally sparse representation, produce sparser representations than analytic dictionaries [3, 89]. Dictionary learning was first proposed by Olshausen and Field in [67] based on inspiration taken from the behavior of cells in the visual cortex, which they refer to as sparse coding. There is a multitude of signal processing problems that can be tackled with sparse dictionaries. Most notably, dictionaries were used to denoise signals corrupted by Gaussian noise [3, 19]. The k-SVD dictionary learning scheme proposed by Aharon *et al.* in [3] in particular has become widely known and delivered state-of-the-art denoising performance at the time. The key step in the learning procedure of the k-SVD algorithms relies on a singular value decomposition. Dictionary learning is also a key component of compressed sensing [18] and even finds application in higher level tasks such as classification. In [72] the authors train a dictionary using unlabeled data and then use the sparse code of labeled data as input for an SVM classifier, resulting in improved classification performance. Additionally, sparse dictionaries can be used for compression in the sense that if a signal $\mathbf{x}$ has an approximate sparse representation in some dictionary $\mathbf{D}$, then it can be efficiently stored and transmitted. While finding a sparse representation can be challenging, Bruckstein *et al.* proved in [16] that if $\mathbf{D}$ fulfills certain geometric conditions, then the sparse representation of a signal is unique and can be found efficiently.

As previously stated, dictionary representations are a synthesis model where a signal $\mathbf{x} \in \mathbb{R}^p$ is synthesized as a linear combination of a few columns (also referred to as atoms) of a dictionary $\mathbf{D} \in \mathbb{R}^{p \times d}$. This formally reads as

$$\mathbf{x} \approx \mathbf{D}\boldsymbol{\alpha}, \qquad \boldsymbol{\alpha} \text{ is sparse.} \tag{4.9}$$

Finding the optimal dictionary $\mathbf{D}$ for a given set of examples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ can be formulated as the minimization problem

$$\underset{\mathbf{D},\mathbf{A}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \sum_{i=1}^{n} g(\boldsymbol{\alpha}_i) \quad \text{subject to } \mathbf{D} \in \mathfrak{D}. \tag{4.10}$$

The sparse coefficient vectors are stored in the columns of the matrix $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$. The function g acts as a penalty function that encourages vectors to be sparse. While the ℓ_0-

quasi-norm, which for a vector is defined as the number of its non-zero entries, would be the ideal penalty to quantify the sparsity of the signals, it is not feasible for use in optimization algorithms since it is not differentiable and does not have a slope that allows gradient-based optimization algorithms to find a search direction that leads to a sparser solution. It is therefore common practice to use approximations to the ℓ_0-norm, such as ℓ_1- or ℓ_p-norms with $0 < p < 1$. Furthermore, the constraint set $\mathfrak{D}$ ensures that only viable dictionaries are generated. Without any regularization, training could produce ill-suited dictionaries. For example, the atom norm could decrease towards zero while the absolute value of the sparse coefficients increases towards infinity. While this would technically solve the optimization problem, using such a dictionary is infeasible in practice.

In order to avoid this situation, it is common practice to normalize the columns of $\mathbf{D}$. The set $\mathfrak{D}$ that realizes this constraint is the oblique manifold $\mathrm{Ob}(p, d)$, which is defined as the set of matrices of size (p, d) with unit norm columns

$$\mathrm{Ob}(p, d) = \{\mathbf{D} \in \mathbb{R}^{p \times d} \,:\, (\mathbf{D}^\top \mathbf{D})_{ii} = 1,\ i = 1, \ldots, d\}. \tag{4.11}$$

The oblique manifold is a Riemannian sub-manifold of the embedding Euclidean space $\mathbb{R}^{p \times d}$, see [1]. While the more advanced geometric properties of manifolds are not relevant for the discussion of the generalization error bounds, they do become relevant for implementation purposes and will be discussed in Chapter 6 in more detail.

Recall from the introduction that the quality of how good a signal $\mathbf{x}$ can be encoded by a given dictionary $\mathbf{D}$ is measured via the function

$$f_{\mathbf{x}}(\mathbf{D}) \coloneqq \inf_{\boldsymbol{\alpha}} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}), \tag{4.12}$$

which is also the function that plays the pivotal role in establishing the generalization error bounds for the dictionary learning algorithm.

In the subsequent discussion we assume that the considered signals are within the unit ball, i.e., they have norm $\|\mathbf{x}\|_2 \leq 1$. Since it is common practice to normalize signals prior to submitting them to a representation learning algorithm, this assumption is reasonable. Furthermore, for both methods we have to ensure that the function value $f_{\mathbf{x}}(\mathbf{D})$ is bounded in order to be able to apply the concentration inequalities. Under the unit norm assumption

in combination with the unit column norm of the dictionary the value of $f_{\mathbf{x}}(\mathbf{D})$ can be bounded by some variable

$$0 \leq f_{\mathbf{x}}(\mathbf{D}) \leq b, \tag{4.13}$$

as long as the penalty function g is continuous and its growth rate is limited.

4.3.1 Bounding framework using Hoeffding & Covering

In this section we discuss the general strategy when bounding the generalization error employing a union bound argument in conjunction with an ε-covering of a class of discrete, Lipschitz continuous hypothesis functions. Concretely, the steps are as follows.

- **(HC1) – Union bound and concentration inequality:** Expand the generalization error into three separate terms by introducing dictionary sampling points that are elements of a minimal ε-covering of the constraint set. Upper bound the term that is only dependent of elements of this ε-covering using Hoeffding's inequality and a union bound argument.
- **(HC2) – Lipschitz property:** Establish the Lipschitz continuity of the cost function w.r.t. the weight parameter and use the Lipschitz property to bound the remaining terms in the decomposition of the original generalization error.
- **(HC3) – Final covering and Lipschitz bound:** Use the structural properties of the minimal ε-cover of the constraint set to develop the final bound.

(HC1) – Union bound and concentration inequality

In this first step we split the generalization error into three separate terms. To achieve this, we use the property that the constraint set $\mathfrak{D}$ that parameterizes the hypothesis class given by $\mathrm{Ob}(p, d)$ is compact and has a finite ε-covering. By Definition 4.3 this cover is defined by the set $\{\mathbf{D}_i \in \mathfrak{D} : i \in \{1, \ldots, N^{\mathrm{cov}}(\mathfrak{D}, \varepsilon)\}\}$ and for every $\mathbf{D} \in \mathfrak{D}$ there exists a $j \in \{1, \ldots, N^{\mathrm{cov}}(\mathfrak{D}, \varepsilon)\}$ such that $\mathrm{d}(\mathbf{D}_j, \mathbf{D}) \leq \varepsilon$ for the distance measure d defined on $\mathfrak{D}$. This enables us to rewrite the generalization error as

$$\Phi(\mathbf{X}) \coloneqq \sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \tag{4.14}$$

$$= \sup_{\mathbf{D}\in\mathfrak{D}} \Big|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \tag{4.15}$$
$$+ \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j)$$
$$+ \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)]\Big|$$

$$\leq \underbrace{\sup_{j\in\{1,\ldots,N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)]\right|}_{\text{(I)}} \tag{4.16}$$
$$+ \underbrace{\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j)\right|}_{\text{(II)}}$$
$$+ \underbrace{\sup_{\mathbf{D}\in\mathfrak{D}} \left|\mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right|}_{\text{(III)}}.$$

The next step is to individually bound the separate terms (I), (II), and (III) on the right-hand side. In summary, expression (I) is bounded using a union bound argument and Hoeffding's inequality whereas bounds for (II) and (III) are provided using the Lipschitz property of $f_{\mathbf{x}}$. Each step is discussed in detail in the following.

Since by assumption the function $f_{\mathbf{x}}(\mathbf{D})$ assumes values between 0 and $b > 0$, we can bound the term (I) using Hoeffding's inequality. With these parameters Theorem 4.2 simplifies to

$$\Pr\left[\left|\tfrac{1}{n}\sum\nolimits_{i=1}^{n}\left(\phi(X_i) - \mathbb{E}[\phi(X_i)]\right)\right| \geq t\right] \leq 2\exp\left(-\frac{2nt^2}{b^2}\right). \tag{4.17}$$

To provide an upper bound for the term (I) in (4.14) we examine the probability that it

exceeds some bound t. Expressed as formulas this yields

$$\Pr\left[\sup_{j\in\{1,\dots,N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| \geq t\right] \tag{4.18}$$

$$\leq \sum_{j=1}^{N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| \geq t\right] \tag{4.19}$$

$$\leq 2N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\exp\left(-\frac{2nt^2}{b^2}\right), \tag{4.20}$$

where inequality (4.19) follows from the union bound argument, which was defined in Theorem 4.1, and the last inequality (4.20) follows from Hoeffding's inequality applied to each summand. By setting the right-hand side equal to δ and solving for t, we can say that with probability at least $1-\delta$ the inequality

$$\sup_{j\in\{1,\dots,N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| \leq b\sqrt{\frac{\log(N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)) + \log(2/\delta)}{2n}} \tag{4.21}$$

holds. The union bound argument is source for a large part of the slackness of the final bounds since we have to ensure that the probability holds for every event individually. In the same fashion as during the derivation of the VC-based bounds in Chapter 2 it incorporates the covering number of the class of hypotheses, which is ultimately responsible for the $\sqrt{\log n}$ term in the final upper bound.

(HC2) – Lipschitz properties

To find a bound for terms (II) and (III) in Equation (4.14), we observe that $\mathrm{d}(\mathbf{D}_j, \mathbf{D}) \leq \varepsilon$ holds for all $j \in \{1, \dots, N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\}$. Thus, if the cost function $f_{\mathbf{x}}(\mathbf{D})$ is Lipschitz w.r.t. $\mathbf{D}$, we can bound (II) and (III) using the product of the Lipschitz constant and the distance of the covering. The main challenge here is to find a Lipschitz bound L that is independent of the training samples. We will show this in detail in Section 4.4 for the cost function of dictionary learning, among others. For now, we assume the Lipschitz continuity condition is fulfilled and we have $|f_{\mathbf{x}}(\mathbf{D}) - f_{\mathbf{x}}(\mathbf{D}')| \leq L\|\mathbf{D} - \mathbf{D}'\|_F$. Under this assumption, we can

bound (II) via

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j)\right| \leq \tfrac{1}{n}\sum_{i=1}^{n} L\varepsilon = L\varepsilon. \tag{4.22}$$

Likewise, when invoking the law of large numbers we see that (III) is bounded by

$$\begin{aligned} &\sup_{\mathbf{D}\in\mathfrak{D}} \left|\mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \\ &= \sup_{\mathbf{D}\in\mathfrak{D}} \left| \lim_{k\to\infty} \tfrac{1}{k}\sum_{i=1}^{k} f_{\mathbf{x}_i}(\mathbf{D}_j) - \lim_{k\to\infty} \tfrac{1}{k}\sum_{i=1}^{k} f_{\mathbf{x}_i}(\mathbf{D})\right| \\ &\leq L\varepsilon. \end{aligned} \tag{4.23}$$

So in summary, all three terms that occur on the right-hand side of Equation (4.14) are bounded and we obtain the preliminary result

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq b\sqrt{\frac{\log(N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)) + \log(2/\delta)}{2n}} + 2L\varepsilon \tag{4.24}$$

with probability at least $1-\delta$.

(HC3) – Final Covering & Lipschitz bound

The final step is to investigate the behavior of the minimal ε-covering of the constraint set that describes the class of hypotheses and use this result to provide a final bound on the generalization error. We have previously seen that the covering number for a p-dimensional unit ball is upper bounded by $N^{\mathrm{cov}}(B_1,\varepsilon) \leq (1+\frac{2}{\varepsilon})^p$. In general, the covering number of many metric spaces can be expressed in a form similar to this. In particular, all metric spaces that we consider in this work have an upper bound of the shape $N^{\mathrm{cov}}(\mathfrak{D},\varepsilon) = (C/\varepsilon)^h$, where C is a positive constant, h is called the covering dimension of the hypothesis class.

In particular, for the class of $\mathbb{R}^{p\times d}$ matrices with unit norm columns the covering number is upper bounded by $N^{\mathrm{cov}}(\mathfrak{D},\varepsilon) \leq (3/\varepsilon)^{dp}$. This yields the bound

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq b\sqrt{\frac{dp\log(3/\varepsilon) + \log(2/\delta)}{2n}} + 2L\varepsilon. \tag{4.25}$$

Since the above holds for all $0 < \varepsilon < 1$ we can further concretize the bounds by picking

specific values for ε. For example, choosing $\varepsilon = 1/\sqrt{n}$, a commonly used choice for ε, cf. [58, 90], yields the bound

$$\begin{aligned}\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \\ \leq \frac{b}{2}\sqrt{\frac{dp\log n}{n}} + \frac{1}{\sqrt{n}}\left(b\sqrt{\frac{dp\log 3 + \log(2/\delta)}{2}} + 2L\right)\end{aligned} \tag{4.26}$$

that holds with probability of at least $1-\delta$. A more sophisticated choice for ε proposed in [33] is

$$\varepsilon \coloneqq \frac{b}{2L}\sqrt{\frac{\beta\log n}{n}} \tag{4.27}$$

$$\text{with } \beta \coloneqq dp\max\left(\log\left(\tfrac{6L}{b}\right), 1\right). \tag{4.28}$$

Under the assumptions that $dp \geq 1$ and $\log(n) \geq 1$, where the first one is fulfilled by definition and the second one is true when n is large enough, we can further bound this constraint by substituting the definitions and using the triangle inequality on the square root. We get

$$b\sqrt{\frac{dp\log(3/\varepsilon) + \log(2/\delta)}{2n}} + 2L\varepsilon \tag{4.29}$$

$$\leq \frac{b}{\sqrt{2n}}\sqrt{dp\log\left(\frac{6L}{b\sqrt{\beta}}\right) + \tfrac{dp}{2}\log\frac{n}{\log n} + \log(2/\delta)} + b\sqrt{\frac{\beta\log n}{n}} \tag{4.30}$$

$$\leq \frac{b}{\sqrt{2n}}\sqrt{dp\log\left(\frac{6L}{b\sqrt{dp\max(\log\frac{6L}{b}, 1)}}\right) + \tfrac{\beta}{2}\log n + \log(2/\delta)} + b\sqrt{\frac{\beta\log n}{n}} \tag{4.31}$$

$$\leq \frac{b}{\sqrt{2n}}\sqrt{dp\log\left(\frac{6L}{b}\right) + \tfrac{\beta}{2}\log n + \log(2/\delta)} + b\sqrt{\frac{\beta\log n}{n}} \tag{4.32}$$

$$\leq b\sqrt{\frac{\beta + \log(2/\delta)}{2n}} + \left(1+\tfrac{1}{2}\right)b\sqrt{\frac{\beta\log n}{n}}. \tag{4.33}$$

Inequality (4.30) is simply a result of substituting in ε and β. The next line (4.31) follows from the definition of β, where $pd \leq \beta$. Line (4.32) follows from the fact that the term under

the square root in the denominator is greater than 1 by definition, and therefore removing it from the denominator increases the overall value. The final inequality (4.33) is due to the definition of β and the subadditivity of the square root.

In summary, the final generalization bound using the HC approach is

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq b\sqrt{\frac{\beta + \log(2/\delta)}{2n}} + 2b\sqrt{\frac{\beta \log n}{n}} \tag{4.34}$$

with probability at least $1 - \delta$.

4.3.2 Bounding framework using McDiarmid & Rademacher

The workflow when deriving the generalization bounds with this method follows the subsequently outlined steps.

- **(MR1) – McDiarmid's Lemma:** Use McDiarmid's Lemma to find an upper bound to $\sup_{\mathbf{D}\in\mathfrak{D}} |\frac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]|$ for the ordered set of samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$. The samples $\mathbf{x}_i$ are drawn according to the underlying distribution $\mathbb{P}$ which for this bounding scheme are distributions within the unit sphere.
- **(MR2) – Rademacher Complexity:** Via a symmetrization argument this bound is upper bounded again by a term dependent on the empirical Rademacher complexity. Then Gaussian complexity is used as upper bound for Rademacher complexity. This is achieved by utilizing Lemma 4.9.
- **(MR3) – Slepian's Lemma:** The empirical Gaussian complexity that results from the previous step is upper bounded via Slepian's Lemma. To achieve this, define a random process that fulfills the conditions required for Slepian's Lemma and has the additional property of being easily upper bounded.

In the following, we explore this framework in more detail.

(MR1) – McDiarmid's Lemma

The first step is to apply McDiarmid's inequality. To be able to do this, we first have to ascertain that all the necessary conditions are fulfilled. We can assume that the samples are

picked independently and identically distributed. In order to be able to apply McDiarmid's theorem, we need to confirm that changing a single variable only alters the function value of the generalization error

$$\Phi(\mathbf{X}) \coloneqq \sup_{\mathbf{D}\in\mathfrak{D}} \left| \tfrac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \right| \tag{4.35}$$

by a bounded amount. Let $\mathbf{X}'$ be a set of samples that differs from $\mathbf{X}$ only in the j-th component, i.e., $\mathbf{X}' \coloneqq [\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}'_j, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_n]$. Then for the absolute value of the difference $|\Phi(\mathbf{X}) - \Phi(\mathbf{X}')|$ we get

$$\begin{aligned}
&|\Phi(\mathbf{X}) - \Phi(\mathbf{X}')| \\
&\quad = \left| \sup_{\mathbf{D}\in\mathfrak{D}} \left| \tfrac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \right| \right. \\
&\qquad \left. - \sup_{\mathbf{D}\in\mathfrak{D}} \left| \tfrac{1}{n} \sum_{i\neq j} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] + \tfrac{1}{n} f_{\mathbf{x}'_j}(\mathbf{D}) \right| \right| \\
&\quad \leq \left| \sup_{\mathbf{D}\in\mathfrak{D}} \tfrac{1}{n} \left| f_{\mathbf{x}'_j}(\mathbf{D}) - f_{\mathbf{x}_j}(\mathbf{D}) \right| \right| \\
&\quad \leq \frac{b}{n}.
\end{aligned} \tag{4.36}$$

Here, the first inequality holds since the difference of suprema does not exceed the supremum of the difference and the second holds due to the boundedness of $f_{\mathbf{x}}$. This property allows us to apply McDiarmid's inequality as defined in Theorem 4.5 to $\Phi(\mathbf{X})$ which yields

$$\begin{aligned}
&\sup_{\mathbf{D}\in\mathfrak{D}} \tfrac{1}{n} \Big| \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \Big| \\
&\quad \leq \mathbb{E}_{\mathbf{X}} \left[\sup_{\mathbf{D}\in\mathfrak{D}} \left| \tfrac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \right| \right] + b\sqrt{\frac{\log(1/\delta)}{2n}}
\end{aligned} \tag{4.37}$$

with probability at least $1 - \delta$.

(MR2) – Symmetrization and Rademacher Complexity

The next step is to bound the expectation $\mathbb{E}_{\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\left|\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right|\right]$. In order to keep the discussion comprehensive, we only consider the situation when $\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) \geq \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]$ in the following. The discussion of the other case is identical. As a first step for obtaining a bound on the expectation we employ a symmetrization argument. By introducing a set of ghost samples $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n]$, which are drawn according to the same distribution as $\mathbf{X}$, we obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\left(\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right)\right] \\
&\quad= \mathbb{E}_{\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\mathbb{E}_{\tilde{\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \frac{1}{n}\sum_{i=1}^{n} f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right]\right] \\
&\quad= \mathbb{E}_{\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\mathbb{E}_{\tilde{\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right)\right]\right].
\end{aligned} \tag{4.38}$$

The supremum function is convex and we can apply Jensen's inequality, see [43], which yields

$$\mathbb{E}_{\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\mathbb{E}_{\tilde{\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right)\right]\right] \leq \mathbb{E}_{\mathbf{X},\tilde{\mathbf{X}}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\frac{1}{n}\sum_{i=1}^{n}\left(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right)\right]. \tag{4.39}$$

For the next step, we again use the ghost sampling technique. For each sample pair $\mathbf{x}_i, \tilde{\mathbf{x}}_i \in \mathbf{X}, \tilde{\mathbf{X}}$, swap the two elements with probability 1/2. The resulting two sets of examples are denoted by $\mathbf{T}, \tilde{\mathbf{T}}$. Since $\mathbf{X}, \tilde{\mathbf{X}}$ contain i.i.d. samples from $\mathbb{P}$, the distribution of these sets does not change, i.e., $\mathbf{X}, \tilde{\mathbf{X}} \sim \mathbf{T}, \tilde{\mathbf{T}}$. This implies

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D})) \\
&\quad= \frac{1}{n}\sum_{i=1}^{n}\begin{cases}(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D})) & \text{with probability } 1/2\\ (f_{\tilde{\mathbf{x}}_i}(\mathbf{D}) - f_{\mathbf{x}_i}(\mathbf{D})) & \text{with probability } 1/2\end{cases} \\
&\quad= \frac{1}{n}\sum_{i=1}^{n}\sigma_i(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D}))
\end{aligned} \tag{4.40}$$

with the Rademacher variables σ_i. Thus, both terms $\frac{1}{n}\sum_{i=1}^{n}(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D}))$ and $\frac{1}{n}\sum_{i=1}^{n}\sigma_i(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D}))$ are identically distributed and the same holds true for their supremum. Since both sets of samples are drawn according to the same distribution, taking the expectation over $\boldsymbol{\sigma}$ does not alter the result. Additionally, using Jensen's inequality on the supremum and the fact that σ_i and $-\sigma_i$ are identically distributed we get

$$\begin{aligned}
\mathbb{E}_{\mathbf{X},\tilde{\mathbf{X}}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D}))\right] &= \mathbb{E}_{\sigma,\mathbf{X},\tilde{\mathbf{X}}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}\sigma_i(f_{\mathbf{x}_i}(\mathbf{D}) - f_{\tilde{\mathbf{x}}_i}(\mathbf{D}))\right] \\
&\leq \mathbb{E}_{\sigma,\mathbf{X},\tilde{\mathbf{X}}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}\sigma_i f_{\mathbf{x}_i}(\mathbf{D}) + \sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}-\sigma_i f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right] \\
&= \mathbb{E}_{\sigma,\mathbf{X}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}\sigma_i f_{\mathbf{x}_i}(\mathbf{D})\right] + \mathbb{E}_{\sigma,\tilde{\mathbf{X}}}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\tfrac{1}{n}\sum_{i=1}^{n}\sigma_i f_{\tilde{\mathbf{x}}_i}(\mathbf{D})\right] \\
&= 2\,\mathfrak{R}_n(\mathfrak{F}_{\mathfrak{D}}).
\end{aligned} \tag{4.41}$$

In summary, we can say that the Rademacher complexity of the function class $\mathfrak{F}_{\mathfrak{D}}$ can be used to bound the expectation of the supremum $\mathbb{E}_{\mathbf{X}}[\Phi(\mathbf{X})] \leq 2\,\mathfrak{R}_n(\mathfrak{F}_{\mathfrak{D}})$ where $\mathfrak{F}_{\mathfrak{D}}$ is the class of hypotheses parameterized via the constraint set $\mathfrak{D}$. Since the value of $\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}})$ also varies by at most b/n when changing a single sample, we can apply McDiarmid's inequality again and get $\mathfrak{R}_n(\mathfrak{F}_{\mathfrak{D}}) \leq \widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}}) + b\sqrt{\log(1/\delta)/(2n)}$ with probability at least $1-\delta$. By using confidence bounds of $\delta/2$ for both estimates we can bound $\Phi(\mathbf{X})$ with probability at least $1-\delta$. The final part of this step is introducing the Gaussian complexity since it is easier to manage Gaussian random variables. By Lemma 4.9 we can bound the empirical Rademacher by the empirical Gaussian complexity and get

$$\sup_{\mathbf{D}\in\mathfrak{D}}\left|\tfrac{1}{n}\sum_{i=1}^{n}f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq \sqrt{2\pi}\,\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}}) + 3b\sqrt{\frac{\log(2/\delta)}{2n}} \tag{4.42}$$

which holds with probability at least $1-\delta$.

(MR3) – Slepian's Lemma

The final step consists of bounding the empirical Gaussian complexity. As Definition 4.8 states, it is defined as $\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}}) = \mathbb{E}_{\gamma}\left[\sup_{\mathbf{D}\in\mathfrak{D}}\frac{1}{n}\sum_{i=1}^{n}\gamma_i f_{\mathbf{x}_i}(\mathbf{D})\right]$. Determining a concrete

bound for this expression directly is infeasible. In order to provide a final bound for the generalization error, we have to find an expression that serves as an upper bound for the expectation of the supremum over the Gaussian process $G_{\mathbf{D}} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i f_{\mathbf{x}_i}(\mathbf{D})$. This can be achieved by applying Slepian's Lemma 4.11. Given the Gaussian process $G_{\mathbf{D}}$ we have to find another Gaussian process $H_{\mathbf{D}}$ that fulfills the conditions required by Slepian's Lemma, i.e., $\mathbb{E}[(G_{\mathbf{D}} - G_{\mathbf{D}'})^2] \leq \mathbb{E}[(H_{\mathbf{D}} - H_{\mathbf{D}'})^2]$ with the additional requirement that $\mathbb{E}[\sup_{\mathbf{D}\in\mathfrak{D}} H_{\mathbf{D}}]$ can be readily computed. This is easily accomplished under the assumption that $f_{\mathbf{x}}(\mathbf{D})$ is Lipschitz w.r.t. $\mathbf{D}$ with constant L. As in the previous section, we assume this to be true for now and show it later in detail.

When we define a Gaussian process $H_{\mathbf{D}} \coloneqq \langle \boldsymbol{\Xi}, \mathbf{D}\rangle_F = \sum_{i,j} \xi_{ij} d_{ij}$ with $\xi_{ij} \sim \mathcal{N}(0,1)$ for all $i \in \{1,\ldots,p\}$, $j \in \{1,\ldots,d\}$, then we get

$$\mathbb{E}_\gamma\big[(G_{\mathbf{D}} - G_{\mathbf{D}'})^2\big] \leq \frac{L^2}{n}\|\mathbf{D} - \mathbf{D}'\|_F^2 = \mathbb{E}_\xi\big[(H_{\mathbf{D}} - H_{\mathbf{D}'})^2\big]. \tag{4.43}$$

Thus, the conditions for applying Slepian's Lemma are fulfilled and we can upper bound the expectation of the supremum by

$$\mathbb{E}_\gamma\left[\sup_{\mathbf{D}\in\mathfrak{D}} G_{\mathbf{D}}\right] \leq \mathbb{E}_\xi\left[\sup_{\mathbf{D}\in\mathfrak{D}} H_{\mathbf{D}}\right]. \tag{4.44}$$

Furthermore, since $\mathbf{D} \in \mathrm{Ob}(p,d)$ and therefore has unit norm columns, we get

$$\begin{aligned}\mathbb{E}_\xi\left[\sup_{\mathbf{D}\in\mathfrak{D}} H_{\mathbf{D}}\right] &= \tfrac{L}{\sqrt{n}}\mathbb{E}_\xi\left[\sup_{\mathbf{D}\in\mathfrak{D}} \langle \boldsymbol{\Xi}, \mathbf{D}\rangle_F\right] \\ &= \tfrac{L}{\sqrt{n}}\mathbb{E}_\xi\left[\sum_{j=1}^{d} \|\boldsymbol{\xi}_j\|_2\right] \leq Ld\sqrt{\frac{p}{n}}.\end{aligned} \tag{4.45}$$

The second equality follows from the fact that the unit norm vector $\mathbf{x} \in S_{(p-1)}$ that maximizes $\langle \boldsymbol{\xi}, \mathbf{x}\rangle_2$ is the $\mathbf{x} \in \mathbb{R}^p$ that has the same direction as $\boldsymbol{\xi}$, and is also normalized, i.e., it is given as $\boldsymbol{\xi}/\|\boldsymbol{\xi}\|_2$. This implies that $\sup_{\mathbf{x}\in S_{(p-1)}} \langle \boldsymbol{\xi}, \mathbf{x}\rangle_2 = \|\boldsymbol{\xi}\|_2^2/\|\boldsymbol{\xi}\|_2 = \|\boldsymbol{\xi}\|_2$. The last inequality follows from Jensen's inequality and the fact that $\xi_{ij} \sim \mathcal{N}(0,1)$.

Combining Equations (4.37), (4.42), and (4.45) then yields the generalization error bound obtained via the scheme using McDiarmid's inequality and Rademacher averages. It is given

by

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq Ld\sqrt{\frac{2\pi p}{n}} + 3b\sqrt{\frac{\log(2/\delta)}{2n}} \tag{4.46}$$

with probability at least $1-\delta$.

4.4 Dictionary Learning Bounds in Depth

In the previous section, we saw that the results achieved with the HC bounding framework are consistently worse by the mild factor $\sqrt{\log n}$ than the ones obtained via MR. This inferior performance is a result of the generality of the first approach. However, due to this generality, it is possible to extend the achieved results for the HC technique when using more refined bounding arguments adapted to the concrete representation learning model. An example of how the first framework can be specialized for application to sparse dictionary learning is proposed in *"On the Sample Complexity of Dictionary Learning"* by **Seibert** *et al.*, [33, 77]. The following discussion is based on the above publications and proofs of the assertions can be found therein. The upcoming results extend the bounds obtained in the previous section to more general penalty functions and data distributions.

4.4.1 Penalty functions & probability distributions

Norm-like penalty functions

First of all, we slightly extend the requirements from the previous section to more specific conditions on the penalty function.

- **A1:** The function g is non-negative.
- **A2:** The function g is lower semi-continuous: $\liminf_{\boldsymbol{\alpha}\to\boldsymbol{\alpha}_0} g(\boldsymbol{\alpha}) \geq g(\boldsymbol{\alpha}_0)$.
- **A3:** The function g is coercive: $g(\boldsymbol{\alpha}) \to \infty$ as $\|\boldsymbol{\alpha}\| \to \infty$.
- **A4:** The loss function passes through zero: $g(\mathbf{0}) = 0$.

It should be noted that assumption **A4** is included mainly for convenience and should not be taken too literally. Any penalty function that fulfills **A1**–**A3** and has a global minimum

at $\mathbf{0}$ can be manipulated to fulfill **A4**. Later in the analysis of conditions **A1**–**A4** we will need a way to measure the magnitude of the ℓ_1-norm of the sparse coefficient w.r.t. the chosen penalty function. This is achieved by the auxiliary function

$$\bar{g}(t) \coloneqq \sup_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^d,\\ g(\boldsymbol{\alpha})\leq t}} \|\boldsymbol{\alpha}\|_1,\ t \geq 0. \tag{4.47}$$

The ℓ_1-norm that appears here stems from the $\ell_{1\to 2}$-norm that is used to measure the covering numbers later on. This matrix-norm is defined as $\ell_{1\to 2}(\mathbf{D}) \coloneqq \max_i \|\mathbf{d}_i\|_2$, i.e., the column with the largest ℓ_2-norm.

These type of conditions apply to a large variety of penalty functions. But naturally, there are other commonly used penalties that do not fall into this category. In the next section, we investigate a set of functions that do not fulfill conditions **A1**–**A4**, but with some alternative assumptions can still be processed with the provided framework in the following section.

Joint assumptions on g and the constraint set

While assumptions **A1**–**A4** cover a wide variety of penalty functions, they do not apply popular penalty functions related to the ℓ_0-quasi-norm. The ℓ_0-quasi-norm is defined as the number of non-zero entries of a vector. A typical example is the indicator function of a set (also called characteristic function). The indicator function of a set $\mathcal{K}$ is zero for all elements in $\mathcal{K}$ and infinite anywhere else. In order to provide a unified treatment of these types of penalty function we need a more refined set of assumptions that are jointly applied to the function g as well as the constraint set.

- **B1:** $g_{\mathcal{K}}$ is an indicator function of a set $\mathcal{K}$.

$$g_{\mathcal{K}}(\boldsymbol{\alpha}) = \begin{cases} 0, & \text{if } \boldsymbol{\alpha} \in \mathcal{K}; \\ \infty, & \text{otherwise.} \end{cases} \tag{4.48}$$

- **B2:** There exists a constant $\kappa > 0$ such that for any $\boldsymbol{\alpha} \in \mathcal{K}$ and $\mathbf{D} \in \mathfrak{D}$ the inequality $\kappa\|\boldsymbol{\alpha}\|_1^2 \leq \|\mathbf{D}\boldsymbol{\alpha}\|_2^2$ holds.

- **B3:** The set $\mathcal{K}$ contains the origin: $\mathbf{0} \in \mathcal{K}$.

- **B4:** $\mathfrak{D}$ is convex.

In analogy to the auxiliary function defined in Equation (4.47) for norm-like penalty functions we define

$$\bar{g}(t) \coloneqq 2\sqrt{2t/\kappa} \tag{4.49}$$

for indicator penalty functions.

Assumption on the probability distribution of the data

Finally, the generalization bound results rely on some assumptions on the probability distribution underlying the data samples. First, we need to control the behavior of the Lipschitz constant $L_{\mathbf{X}}(\bar{g})$ for large sample sizes. Second, we need to control the concentration of the empirical average around its expectation.

The first property is quantified by the probability

$$\Gamma_n(\gamma) \coloneqq \sup_{\mathbf{D}\in\mathfrak{D}} \Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| > \gamma\right], \tag{4.50}$$

while the second is measured via

$$\Lambda_n(L) \coloneqq \Pr\left[\tfrac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2 \cdot \bar{g}\left(\tfrac{1}{2}\|\mathbf{x}_i\|_2^2\right) > L\right]. \tag{4.51}$$

The assumptions we make on the distribution of the data samples are given by

- **C1:** The moment of $\bar{g}$ is bounded, i.e.,

$$L_{\mathbb{P}}(\bar{g}) \coloneqq \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}\left[\|\mathbf{x}\|_2 \cdot \bar{g}\left(\frac{\|\mathbf{x}\|_2^2}{2}\right)\right] < \infty. \tag{4.52}$$

- **C2:** The concentration of the empirical average of $f_{\mathbf{x}}$ is bounded close to its expectation. Concretely, there exist $c > 0$ and $T \in (0, \infty)$ such that

$$\Gamma_n(c\tau) \leq 2\exp(-n\tau^2) \quad \forall\, \tau \in [0, T],\ \forall\, n. \tag{4.53}$$

By observing assumption **C1**, we see that if $L > L_{\mathbb{P}}(\bar{g})$, then $\lim_{n\to\infty} \Lambda_n(L) = 0$ which will

be useful later. These two assumptions cover many data distributions of which we examine two classes in particular at the end of this section.

4.4.2 Lipschitz continuity

After establishing the necessary conditions for the penalty function and the probability distribution of the data we are ready to establish the Lipschitz constant, which is primarily driven by the penalty function g. The following Lemma introduces a one-sided Lipschitz property with an additional quadratic distance term. In order to state this first major goal, we need to relax the optimality conditions on the sparse code slightly. For this, we introduce the following notation.

Definition 4.12. Define the set of ε-admissible solutions for the samples $\mathbf{x}_i$ $i \in \{1, \dots, n\}$ and $\mathbf{D} \in \mathfrak{D}$ as

$$\mathfrak{A}_\varepsilon(\mathbf{X}, \mathbf{D}) \coloneqq \big\{\mathbf{A} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n] : \\ \boldsymbol{\alpha}_i \in \mathbb{R}^d,\ \tfrac{1}{n}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + g(\boldsymbol{\alpha}_i) \leq f_{\mathbf{x}_i}(\mathbf{D}) + \varepsilon,\ i \in \{1, \dots, n\}\big\}.$$

The subsequent Lemma gives us some insight into the existence and properties of this set of solutions.

Lemma 4.13. *For any $\varepsilon > 0$ the set of ε-admissible solutions $\mathfrak{A}_\varepsilon$ is not empty. Furthermore, if g fulfills conditions* **A1**–**A4***, then $\mathfrak{A}_0$ is not empty and bounded.*

With this Lemma, we are able to formulate the first Lipschitz-property for the empirical average $\frac{1}{n}\sum_{i=1}^n f_{\mathbf{x}_i}(\cdot)$.

Lemma 4.14. *Let $\|\cdot\|$ be some norm for (p, d)-matrices and $\|\cdot\|_\star$ its dual norm. For any $\mathbf{D}$, $\mathbf{D}' \in \mathfrak{D}$ we have*

$$\tfrac{1}{n}\sum_{i=1}^n f_{\mathbf{x}_i}(\mathbf{D}') \leq \tfrac{1}{n}\sum_{i=1}^n f_{\mathbf{x}_i}(\mathbf{D}) + L_{\mathbf{X}}(\mathbf{D})\|\mathbf{D}' - \mathbf{D}\| + C_{\mathbf{X}}(\mathbf{D})\|\mathbf{D}' - \mathbf{D}\|^2,$$

where

$$L_{\mathbf{X}}(\mathbf{D}) \coloneqq \inf_{\varepsilon>0} \sup_{\mathbf{A}\in\mathfrak{A}_\varepsilon} \tfrac{1}{n}\|(\mathbf{X}-\mathbf{D}\mathbf{A})\mathbf{A}^\top\|_\star, \tag{4.54}$$

$$C_{\mathbf{X}}(\mathbf{D}) \coloneqq \inf_{\varepsilon>0} \sup_{\mathbf{A}\in\mathfrak{A}_\varepsilon} \tfrac{C}{2n}\sum_{i=1}^{n}\|\boldsymbol{\alpha}_i\|_1^2. \tag{4.55}$$

The constant C that is used in the definition of $C_{\mathbf{X}}(\mathbf{D})$ only depends on the choice of norm $\|\cdot\|$ and the dimensions of the signal and the coefficient.

The dual norm that occurs here is defined as $\|\mathbf{U}\|_\star \coloneqq \sup_{\mathbf{D},\|\mathbf{D}\|\leq 1}\langle \mathbf{U},\mathbf{D}\rangle_F$. Lemma 4.14 implies the following statement.

Corollary 4.15. *Let $\|\cdot\|$ be some norm of (p,d)-matrices and $\|\cdot\|_\star$ its dual norm, and let $\mathfrak{D}$ be a class of dictionaries. If the Lipschitz variables $L_{\mathbf{X}}(\mathbf{D})$, $C_{\mathbf{X}}(\mathbf{D})$ are bounded, i.e.,*

$$\sup_{\mathbf{D}\in\mathfrak{D}} L_{\mathbf{X}}(\mathbf{D}) \leq L_{\mathbf{X}}(\mathfrak{D}),$$
$$\sup_{\mathbf{D}\in\mathfrak{D}} C_{\mathbf{X}}(\mathbf{D}) \leq C_{\mathbf{X}}(\mathfrak{D}),$$

then for any $\mathbf{D},\mathbf{D}'\in\mathfrak{D}$, $\mathbf{D}\neq\mathbf{D}'$ we have

$$\frac{\left|\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}') - \frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D})\right|}{\|\mathbf{D}'-\mathbf{D}\|} \leq L_{\mathbf{X}}(\mathfrak{D})\big(1+\tfrac{C_{\mathbf{X}}(\mathfrak{D})}{L_{\mathbf{X}}(\mathfrak{D})}\|\mathbf{D}'-\mathbf{D}\|\big).$$

As a result of Corollary 4.15 the empirical average $\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\cdot)$ is uniformly locally Lipschitz over the class $\mathfrak{D}$ for any constant $L \geq L_{\mathbf{X}}(\mathfrak{D})$. These bounds are concretized in the subsequent Lemma for the norm $\|\cdot\|_{1\to 2}$. This choice is inspired by the fact that in dictionary learning the dictionary columns are typically constrained to have unit ℓ_2-norm.

For the next lemma, we need to ensure that every signal in the training set can be represented via sparse code, i.e., the set $\mathfrak{A}_0$ for penalty functions that satisfy conditions **A1**–**A4** is not empty and bounded. Since g is non-negative, this also holds for $\frac{1}{n}\sum_i \|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}_i)$. Furthermore, we have $\lim_{k\to\infty}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}_k\|_2^2 + g(\boldsymbol{\alpha}_k) = \infty$ when $\lim_{k\to\infty}\|\boldsymbol{\alpha}_k\| = \infty$. This implies that the function $\mathbf{A}\mapsto \frac{1}{n}\sum_i \|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + g(\boldsymbol{\alpha}_i)$ has bounded sublevel sets. And since g is lower semi-continuous by assumption **A2**, the same applies to the function

$\frac{1}{n}\sum_i \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}_i)$. Therefore, there exists a set of coefficients $\mathbf{A} \coloneqq [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ such that $\sum_i \frac{1}{n}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + g(\boldsymbol{\alpha}_i) = \frac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{D})$.

Lemma 4.16. *Under assumptions* **A1**–**A4** *on the penalty function g for any training set $\mathbf{X}$ and any dictionary $\mathbf{D} \in \mathfrak{D}$, $L_{\mathbf{X}}(\mathbf{D})$ and $C_{\mathbf{X}}(\mathbf{D})$ with the norm $\|\cdot\| = \|\cdot\|_{1\to 2}$ are bounded by*

$$L_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{1}{n}\sum_{i=1}^{n} \sqrt{2 f_{\mathbf{x}_i}(\mathbf{D})} \cdot \bar{g}(f_{\mathbf{x}_i}(\mathbf{D})) \quad \text{and}$$
$$C_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{1}{2n}\sum_{i=1}^{n} \big(\bar{g}(f_{\mathbf{x}_i}(\mathbf{D}))\big)^2,$$

with $\bar{g}$ as defined in (4.47).

By leveraging the assumptions **A1**–**A4** on the penalty g, we see that

$$0 \leq g(\boldsymbol{\alpha}_i) \leq f_{\mathbf{x}_i}(\mathbf{D}) \leq \tfrac{1}{2}\|\mathbf{x}_i\|_2^2 \quad \text{and} \quad \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2 \leq \sqrt{2 f_{x_i}(\mathbf{D})} \leq \|\mathbf{x}_i\|_2^2. \tag{4.56}$$

Thus, Lemma 4.16 yields

Corollary 4.17. *Under the assumption that g fulfills conditions* **A1**–**A4**, *for any training set $\mathbf{X}$ and any dictionary $\mathbf{D} \in \mathfrak{D}$ the upper bounds in Lemma 4.16 can be concretized to*

$$L_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2 \cdot \bar{g}\big(\tfrac{1}{2}\|\mathbf{x}_i\|_2^2\big) \eqqcolon L_{\mathbf{X}}(\bar{g}) \quad \text{and} \tag{4.57}$$
$$C_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{1}{2n}\sum_{i=1}^{n} \big(\bar{g}\big(\tfrac{1}{2}\|\mathbf{x}_i\|_2^2\big)\big)^2 \eqqcolon C_{\mathbf{X}}(\bar{g}). \tag{4.58}$$

Up until now we only considered "norm-like" penalty functions that fulfill assumptions **A1**–**A4**. In the next Lemma, we give a Lipschitz result for indicator-type penalty functions.

Lemma 4.18. *Under assumptions* **B1**–**B4** *for any set of training samples $\mathbf{X}$ and any dictionary $\mathbf{D} \in \mathfrak{D}$, the conditions defined in* (4.54) *and* (4.55) *for the choice $\|\cdot\| = \|\cdot\|_{1\to 2}$ fulfill the bounds*

$$L_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{2}{n\sqrt{\kappa}}\|\mathbf{X}\|_F^2 \eqqcolon L_{\mathbf{X}}(\bar{g}), \tag{4.59}$$
$$C_{\mathbf{X}}(\mathbf{D}) \leq \tfrac{2}{n\kappa}\|\mathbf{X}\|_F^2 \eqqcolon C_{\mathbf{X}}(\bar{g}). \tag{4.60}$$

Finally, we can use all the previous results to obtain the following lemma that holds for both types of considered penalties.

Lemma 4.19. *Let g fulfill the conditions* **A1**–**A4** *or* **B1**–**B4**. *Then for any* $\mathbf{X}$ *and any* $\mathbf{D}, \mathbf{D}' \in \mathfrak{D}$ *we have*

$$\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}')\right| \leq L_{\mathbf{X}}(\bar{g})\|\mathbf{D}-\mathbf{D}'\|_{1\to 2}, \tag{4.61}$$

with the Lipschitz variable for **A1**–**A4** *being defined in* (4.57), *and for* **B1**–**B4** *as defined in* (4.59). *For the expected cost function we have*

$$\left|\mathbb{E}[f_{\mathbf{x}}(\mathbf{D})] - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}')]\right| \leq L\|\mathbf{D}-\mathbf{D}'\|_{1\to 2} \tag{4.62}$$

with probability at least $1-\Lambda_n(L)$. *If* $L > L_{\mathbb{P}}(\bar{g})$, *then* (4.62) *holds with probability* 1.

The proof of this lemma is the only time that **B4**, i.e., the convexity of $\mathfrak{D}$, is required. It is in fact possible to show a similar result that does not require $\mathfrak{D}$ to be convex, which we will not discuss here. The interested reader is referred to the detailed discussion of this case in [33]. With Lemma 4.19 we have shown the global Lipschitz property of $\frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D})$ and $\mathbb{E}[f_{\mathbf{x}}(\mathbf{D})]$ and are able to bound the generalization error.

4.4.3 Final bounds

We are now ready to bound the generalization error. In the following, the set $\left\{\mathbf{D}_i : i \in \{1,\dots,N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\}\right\} \subset \mathfrak{D}$ is a minimal ε-covering of $\mathfrak{D}$. For the assumptions **A1**–**A4** or **B1**–**B4** and **C1**–**C2** by using the triangle inequality the same way as in Section 4.3.1, we

get

$$\begin{aligned}\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| &\leq \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j)\right| \\ &\quad + \sup_{1\leq j\leq N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| \\ &\quad + \left|\mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j)] - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D})]\right| \\ &\leq \sup_{1\leq j\leq N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| + 2L\varepsilon.\end{aligned} \tag{4.63}$$

In order to estimate the upper bound of the supremum in (4.63), we investigate the probability that it is greater than some $\gamma > 0$.

$$\begin{aligned}&\Pr\left[\sup_{1\leq j\leq N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| > \gamma\right] \\ &\qquad \leq \Pr\left[\bigcup_{j=1}^{N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \left(\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| > \gamma\right)\right] \\ &\qquad \leq \sum_{j=1}^{N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)} \Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_j) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_j)]\right| > \gamma\right] \\ &\qquad = N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\cdot \Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}_1) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D}_1)]\right| > \gamma\right] \\ &\qquad \leq N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\cdot \sup_{\mathbf{D}\in\mathfrak{D}} \Pr\left[\left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}[f_{\mathbf{x}}(\mathbf{D})]\right| > \gamma\right] \\ &\qquad = N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\cdot \Gamma_n(\gamma),\end{aligned} \tag{4.64}$$

using Boole's inequality as defined in Theorem 4.1 in the second and the definition of $\Gamma_n(\gamma)$ (4.50) in the last line. With this, we get the temporary result that

$$\sup_{\mathbf{D}\in\mathfrak{D}} \left|\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| \leq 2L\varepsilon + \gamma \tag{4.65}$$

holds with probability at least $1 - \big(\Lambda_n(L) + N^{\mathrm{cov}}(\mathfrak{D},\varepsilon)\cdot\Gamma_n(\gamma)\big)$.

Define the parameters

$$\varepsilon \coloneqq \frac{c}{2L}\sqrt{\frac{\beta \log n}{n}}, \tag{4.66}$$

$$\tau \coloneqq \sqrt{\frac{h \log(C/\varepsilon) + t}{n}} \tag{4.67}$$

where $N^{\mathrm{cov}}(\mathfrak{D}, \varepsilon) \leq (C/\varepsilon)^h$ and $\beta \coloneqq h \cdot \max(\log \frac{2LC}{c}, 1)$. Under the assumption that the sample size is large enough we can write for any $0 \leq t \leq nT^2 - \beta \log n$ that

$$\begin{aligned} 2L\varepsilon + \gamma &= c\sqrt{\frac{\beta \log n}{n}} + c\sqrt{h \log\left(\frac{2LC}{c\sqrt{\beta}}\right) + \log\left(\frac{n}{\log n}\right) + t}\Big/\sqrt{n} \\ &\leq 2c\sqrt{\frac{\beta \log n}{n}} + c\sqrt{\frac{\beta + t}{n}}. \end{aligned} \tag{4.68}$$

Further, we can simplify the term governing the probability by plugging in the parameters

$$\begin{aligned} N^{\mathrm{cov}}(\mathfrak{D}, \varepsilon) \cdot \Gamma_n(c\tau) &\leq (C/\varepsilon)^h \cdot 2\exp(-n\tau^2) \\ &= (C/\varepsilon)^h \cdot 2\exp\left(-n \cdot \frac{h \log(C/\varepsilon) + t}{n}\right) \\ &= 2\exp(-t). \end{aligned} \tag{4.69}$$

Therefore, the final bound for the generalization error is

$$\sup_{\mathbf{D} \in \mathfrak{D}} \left| \tfrac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})] \right| \leq 2c\sqrt{\frac{\beta \log n}{n}} + c\sqrt{\frac{\beta + t}{n}} \tag{4.70}$$

which holds with probability at least $1 - \big(\Lambda_n(L) + 2e^{-t}\big)$.

As a final note, the assumptions on the probability distributions **C1**, **C2** are fulfilled under some standard assumptions. One example are data distributions that generate data in a ball of radius R, i.e., $\Pr[\|\mathbf{x}\|_2 \leq R] = 1$. Then for any $\mathbf{D} \in \mathfrak{D}$ we have

$$0 \leq f_{\mathbf{x}_i}(\mathbf{D}) \leq \tfrac{1}{2}\|\mathbf{x}\|_2^2 \leq R^2/2. \tag{4.71}$$

We can therefore apply Hoeffding's inequality

$$\begin{aligned} \Pr\left[\left|\tfrac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D})]\right| > \gamma\right] &\leq 2\exp\left(-\frac{2n^2\gamma^2}{n(R^2/2)^2}\right) \\ &= 2\exp\left(-\frac{8n\gamma^2}{R^4}\right), \end{aligned} \tag{4.72}$$

and get that

$$\Gamma_n(c\tau) \leq 2\exp(-n\tau^2) \tag{4.73}$$

with $c \coloneqq R^2/\sqrt{8}$ for any $\tau > 0$ which shows that **C2** is fulfilled. Furthermore, **C1** holds with $L_{\mathbb{P}}(\bar{g}) \leq R\bar{g}(R^2/2)$ since the ℓ_2-norm of $\mathbf{x}_i$ is smaller than or equal to R for all $i \in \{1, \ldots, n\}$ and $\Lambda_n(R\bar{g}(R^2/2)) = 0$ for all n.

In addition to distributions within the unit ball, which is the standard assumption made in literature regarding sample complexity and generalization error bounds, assumptions **C1** and **C2** also cover sub-Gaussian data distributions, i.e., distributions that for some $A > 0$ fulfill $\Pr[\|\mathbf{x}\|_2^2 \geq At] \leq \exp(-t)$ for all $t > 1$, see [17]. This class of distributions contains all bounded zero-mean random variables. In this scenario **C2** holds for $c = 12A$ and $T = 1$ and **C1** is fulfilled as soon as $\bar{g}$ has at most polynomial growth which follows from results in [33, 42] which are primarily based on Bernstein's inequality.

Chapter 5

Case Studies

In this chapter we apply the generalization error bounding strategies that were devised in the previous chapter to various representation learning models. As a first case study, we review the generalization error bound for principal component analysis, an example for classic unsupervised feature learning. Further, we investigate co-sparse analysis operator learning, which is a close relative to sparse dictionary learning, first in its standard specification and then in a separable variant that enforces additional structure on the hypothesis class. Finally, the generalization error bound for the supervised task-driven dictionary learning algorithm is analyzed.

5.1 Principal Component Analysis

The goal of principal component analysis is to find a representation for a given set of data samples $\mathbf{X} \in \mathbb{R}^{p \times n}$ in a lower dimensional space such that the new coordinates are uncorrelated [44, 69]. This new set of coordinates is referred to as principal components. Finding a feature space that allows for such an uncorrelated representation is typically achieved by using either the eigenvalue decomposition of the covariance matrix of the data or the singular value decomposition of the centered raw data. In the following discussion, we assume that the data $\mathbf{X}$ is centered, i.e., we subtract the sample mean and divide by the sample standard deviation. This can be done without loss of generality. Note that finding the principal components as outlined above can also be stated as the minimization problem

$$\min_{\mathbf{D} \in \mathrm{St}(p \times d)} \tfrac{1}{2} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2, \tag{5.1}$$

with $d < p$. The optimization is conducted over the Stiefel manifold $\mathrm{St}(p, d)$, which is characterized as the (p, d)-matrices with pairwise orthogonal columns, i.e.,

$$\mathrm{St}(p, d) \coloneqq \{\mathbf{D} \in \mathbb{R}^{p \times d} \,:\, \mathbf{D}^\top \mathbf{D} = \mathbf{I}_d\} \tag{5.2}$$

with the identity matrix $\mathbf{I}_d \in \mathbb{R}^{d \times d}$.

The minimization problem (5.1) almost has the shape of optimization problem we consider in the presented bounding frameworks. All that is missing is the penalty function. This can be easily resolved by stating the optimization problem as

$$\min_{\mathbf{D} \in \mathrm{St}(p,d)} \tfrac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \tfrac{1}{n} \sum_i g(\boldsymbol{\alpha}_i) \tag{5.3}$$

with the indicator penalty function for d-sparse vectors $g(\boldsymbol{\alpha}) = \chi_{d\text{-sparse}}(\boldsymbol{\alpha})$. Since for the coefficient vector we have $\boldsymbol{\alpha}_i \in \mathbb{R}^d$, the penalty function is equivalent to zero for all i.

Recall that the quality of how well a single sample $\mathbf{x}$ is encoded by a dictionary $\mathbf{D}$ is measured by $f_{\mathbf{x}}(\mathbf{D}) = \inf_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha})$. By construction of the loss function there exists a coefficient $\boldsymbol{\alpha}$ such that the infimum is assumed. We denote the coefficient that achieves the optimal representation for a data sample $\mathbf{x}$ by

$$\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}} \coloneqq \arg\min_{\boldsymbol{\alpha}} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}), \tag{5.4}$$

which we require for the discussion of the MR bounding approach.

Extension of results from Section 4.4

As a consequence of the indicator penalty function a direct application of the bounding scheme (HC) is not feasible. Alternatively, we resort to the framework devised in Section 4.4. The achieved results are comparable to [82] where a behavior $\eta \propto \sqrt{\log(n)/n}$ is derived, but they are more pessimistic than the $\eta \propto \sqrt{1/n}$ bounds presented in [9] for distributions within the unit ball that are driven by $\beta \propto d$. Due to the generality of the assumptions we made in Section 4.4, the results shown below are applicable to more general data distributions as they encompass sub-Gaussian distributions.

Proposition 5.1. *For PCA the technique proposed in Section 4.4 yields the generalization*

error bound

$$\Phi(\mathbf{X}) \leq 2c\sqrt{\frac{\beta \log n}{n}} + c\sqrt{\frac{\beta + t}{n}}$$

for training set size n with probability greater than $1 - e^{-t}$ for $t > 0$. The parameter β is governed by the underlying data distribution. For data distributed in the unit ball β is defined in (5.6)*. For sub-Gaussian distributed data it is defined in* (5.7)*.*

The bounding procedure is initially identical to the analysis of sparse dictionary learning presented in Section 4.4. The analysis of PCA differs in the final step, where the dimensionality of the constraint set is considered. For training data distributions within the unit sphere, that is $\Pr[\|\mathbf{x}_i\|_2 \leq 1] = 1$ for all $i \in \{1, \ldots, n\}$, we obtain the sample complexity bound as derived in Equation (4.70) with the parameters $c = 1/\sqrt{8}$, and the covering number parameters $h = pd - d(d+1)/2$, $C = 3\pi e^{\pi}$ since $\mathrm{St}(p, d)$ is a subset of $\{\mathbf{D} \in \mathbb{R}^{p \times d} : \|\boldsymbol{\alpha}\|_2^2 \leq \|\mathbf{D}\boldsymbol{\alpha}\|_2^2 \text{ for all } \boldsymbol{\alpha} \in \mathbb{R}^d\}$. Thus, the sample complexity is given by

$$\Phi(\mathbf{X}) \leq 2c\sqrt{\frac{\beta \log n}{n}} + c\sqrt{\frac{\beta + t}{n}} \tag{5.5}$$

with probability greater than or equal to $1 - e^{-t}$ and the driving parameter

$$\beta_{\mathrm{PCA}} = \left(pd - \frac{d(d+1)}{2}\right) \cdot \log(12\sqrt{2}\pi e^{\pi} L). \tag{5.6}$$

For sub-Gaussian distributions, i.e., distributions that fulfill $\Pr[\|\mathbf{x}\|_2^2 \geq At] < \exp(-t)$ for some fixed $A > 0$ and all $t > 1$, the growth variable of the sample complexity is given by

$$\beta_{\text{sub-Gaussian-PCA}} = \left(pd - \frac{d(d+1)}{2}\right) \cdot \max\left(\log\left(\frac{\pi e^{\pi}\sqrt{d}}{A}\right),\ 1\right). \tag{5.7}$$

McDiarmid & Rademacher

Compared to the result derived with the previous scheme the following bound loses the dependency on the signal dimension p and the $\sqrt{\log n}$ term, and recovers the results from [9, 82] apart from having a slightly stronger dependence on the coefficient dimension d and the stability parameters λ, L. This is achieved under the more rigorous assumption that

the data is distributed within the unit ball. The additional parameters are a result of the generality of the proposed bounding technique.

Proposition 5.2 (**MR**). *The MR approach yields the following generalization error bounds for PCA that maps p-dimensional data distributed within the unit ball to a d-dimensional reduced feature space. For a training set* $\mathbf{X}$ *consisting of* n *samples the inequality*

$$\Phi(\mathbf{X}) \leq \sqrt{\frac{2\pi}{n}}\lambda d + 3L\sqrt{\frac{\log(2/\delta)}{2n}}$$

holds with probability at least $1-\delta$*. The parameter* λ *is a stability parameter of the reduced coefficient and* L *is the Lipschitz constant of the cost function.*

In addition to the assumption that the data points $\mathbf{x}$ are distributed within a unit ball, we assume that the sparse code is stable, i.e., it is Lipschitz w.r.t. the matrix $\mathbf{D}$ with constant λ. In particular, this means that the optimal coefficient vector fulfills the property

$$\|\boldsymbol{\alpha}^{\star}_{\mathbf{x},\mathbf{D}} - \boldsymbol{\alpha}^{\star}_{\mathbf{x},\mathbf{D}'}\|_2^2 \leq \lambda^2 \|(\mathbf{D}-\mathbf{D}')^{\top}\mathbf{x}\|_2^2, \tag{5.8}$$

a property that is necessary for the step (**MR3**). This assumption is reasonable and becomes quite intuitive when we recall the standard definition of PCA. The optimal coefficient of a centered vector $\mathbf{x}$ w.r.t. an orthonormal base $\mathbf{D}$ with $\mathbf{D}^{\top}\mathbf{D} = \mathbf{I}_d$ is determined by $\boldsymbol{\alpha}^{\star}_{\mathbf{x},\mathbf{D}} = \mathbf{D}^{\top}\mathbf{x}$. The above equation therefore immediately follows.

The first step (**MR1**) is achieved by realizing that varying a single sample the McDiarmid's inequality yields a change in the generalization error $\Phi(\mathbf{X})$ of at most $2L/n$, where L is the Lipschitz constant of f. We can apply McDiarmid's inequality and obtain

$$\Phi(\mathbf{X}) \leq \mathbb{E}_{\mathbf{X}}[\Phi(\mathbf{X})] + L\sqrt{\frac{2\log(1/\delta)}{n}} \tag{5.9}$$

with probability at least $1-\delta$.

For (**MR2**) we first use the standard symmetrization and ghost sampling technique to replace the expectation over $\Phi(\mathbf{X})$ by the Rademacher complexity and then use the trivial connection between Rademacher and Gaussian complexity to get an expression that is based

on Gaussian random variables. This yields

$$\Phi(\mathbf{X}) \leq \sqrt{2\pi}\, \widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathrm{St}(p,d)}) + 3L\sqrt{\frac{\log(2/\delta)}{2n}}$$

which holds with probability at least $1-\delta$ with the empirical Gaussian complexity

$$\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathrm{St}(p,d)}) = \mathbb{E}_\gamma\left[\sup_{\mathbf{D}\in\mathrm{St}(p,d)} \tfrac{1}{n} G_{\mathbf{D}}\right]$$
$$\text{with } G_{\mathbf{D}} \coloneqq \sum_i \gamma_i f_{\mathbf{x}_i}(\mathbf{D}).$$

For the final step (**MR3**), we need to construct a Gaussian process $H_{\mathbf{D}}$ that fulfills the conditions of Slepian's Lemma, i.e., $\mathbb{E}[(G_{\mathbf{D}} - G_{\mathbf{D}'})^2] \leq \mathbb{E}[(H_{\mathbf{D}} - H_{\mathbf{D}'})^2]$. Using the Lipschitz continuity of $f_{\mathbf{x}}$ and the stable sparse code property (5.8) we can make the following estimations.

$$\begin{aligned}
\mathbb{E}_\gamma[(G_{\mathbf{D}} - G_{\mathbf{D}'})^2] &= \sum_{i=1}^n \left(f_{\mathbf{D}}(\mathbf{x}_i) - f_{\mathbf{D}'}(\mathbf{x}_i)\right)^2 \\
&\leq \sum_{i=1}^n L^2 \|\boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}} - \boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}'}\|_2^2 \\
&\leq \sum_{i=1}^n L^2\lambda^2 \|(\mathbf{D}-\mathbf{D}')^\top \mathbf{x}_i\|_2^2 \\
&= \mathbb{E}_\gamma[(H_{\mathbf{D}} - H_{\mathbf{D}'})^2],
\end{aligned} \tag{5.10}$$

where $H_{\mathbf{D}}$ is defined with the Gaussian vector $\boldsymbol{\Gamma}_i \in \mathbb{R}^d$ for all $i \in \{1,\dots,n\}$ by

$$H_{\mathbf{D}} \coloneqq \sum_i \lambda \cdot \langle \boldsymbol{\Gamma}_i, \mathbf{D}^\top \mathbf{x}_i \rangle. \tag{5.11}$$

The construction of $H_{\mathbf{D}}$ enables us to bound the expectation over its supremum by

$$\begin{aligned}\mathbb{E}_\gamma\left[\sup_{\mathbf{D}\in\mathfrak{D}} H_{\mathbf{D}}\right] &= \mathbb{E}_\gamma\left[\sup_{\mathbf{D}}\sum_i \lambda\cdot\langle\mathbf{\Gamma}_i, \mathbf{D}^\top\mathbf{x}_i\rangle\right] \\ &\leq \lambda\,\mathbb{E}_\gamma\left[\sup_{\mathbf{D}}\sum_j \|\sum_i \gamma_{ij}\mathbf{x}_i\|_2\|\mathbf{d}_j\|_2\right] \\ &\leq \lambda d\,\mathbb{E}_\gamma\left[\|\sum_{i=1}^n \gamma_{i1}\mathbf{x}_i\|_2\right] \\ &\leq \lambda d\sqrt{n},\end{aligned} \tag{5.12}$$

where the first inequality is achieved by an application of the Cauchy-Schwarz inequality, the second is due to the i.i.d. nature of γ_{ij} and the unit norm of the columns of elements in $\mathrm{St}(p,d)$, and the final inequality follows from Jensen's inequality and the normal distributed γ_{ij}. Thus, we obtain

$$\frac{\sqrt{2\pi}}{n}\mathbb{E}_\gamma\left[\sup_{\mathbf{D}\in\mathrm{St}(p,d)} H_{\mathbf{D}}\right] \leq \sqrt{\frac{2\pi}{n}}\lambda d. \tag{5.13}$$

We are now able to state the final bound on the generalization error obtained via the McDiarmid & Rademacher approach as

$$\Phi(\mathbf{X}) \leq \sqrt{\frac{2\pi}{n}}\lambda d + 3L\sqrt{\frac{\log(2/\delta)}{2n}} \tag{5.14}$$

which holds with probability at least $1-\delta$.

5.2 Co-sparse Analysis Operator

While dictionary learning is based on the sparse synthesis model, the closely related co-sparse analysis operator learning model relies, as the name implies, on analyzing signals. This means that instead of synthesizing the data from atoms of a dictionary, a data point $\mathbf{x}$ is analyzed by multiplication with an operator $\mathbf{\Omega}$. The co-sparse analysis model expressed in formulas reads as

$$\mathbf{\Omega}\mathbf{x} \approx \boldsymbol{\alpha}, \qquad \boldsymbol{\alpha} \text{ is sparse.}$$

$\mathbf{\Omega} \in \mathbb{R}^{d \times p}$ is referred to as the *co-sparse analysis operator*. Its rows can be interpreted as filters that when applied to signals of its corresponding domain yield sparse responses. It was shown in [65] that the filters which yield zero response determine the subspace to which the signal belongs. Let $\boldsymbol{\omega}_i^\top$ be the i-th row of $\mathbf{\Omega}$, i.e., $\mathbf{\Omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_d]^\top$. Let $\Lambda \subset \{1, \ldots, d\}$ be the index subset such that for all $i \in \Lambda$ it holds that $\boldsymbol{\omega}_i^\top \mathbf{x} = 0$. The signal $\mathbf{x}$ lies in the intersection of all hyperplanes to which $\boldsymbol{\omega}_i$ with $i \in \Lambda$ are normal vectors and the information contained within the signal is encoded in the set Λ. This is in contrast to the dictionary learning model where the signal is encoded by the non-zero elements of the coefficient vector.

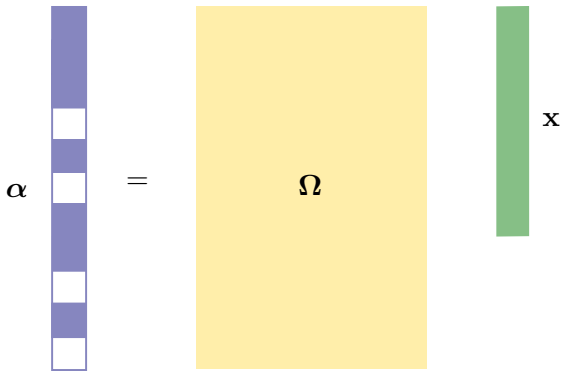


Figure 5.1: In the co-sparse analysis model a signal $\mathbf{x} \in \mathbb{R}^p$ is transformed by an operator $\mathbf{\Omega} \in \mathbb{R}^{d \times p}$, $d \geq p$ such that the resulting coefficient vector $\boldsymbol{\alpha} = \mathbf{\Omega}\mathbf{x}$ is co-sparse.

Co-sparse analysis operators trained on the domain of real images have proven to perform well in various tasks such as image denoising, segmentation, and classification. For more in depth information about this topic we refer the reader to the following articles [30, 37, 38, 75, 97] where co-sparse representations and their performance are thoroughly investigated.

Since we investigate an implementation of the co-sparse analysis operator framework in Chapter 6, we provide an in depth discussion of the problem setting and the involved penalty functions in the following. In order to determine how well a co-sparse analysis operator $\mathbf{\Omega}$ represents a given signal $\mathbf{x}$, a typical penalty function is provided by

$$\begin{aligned} g(\boldsymbol{\alpha}) &:= \sum_{j=1}^{d} \log(1 + \nu \alpha_j^2) \\ &\text{with } \boldsymbol{\alpha} = \mathbf{\Omega}\mathbf{x}, \end{aligned} \tag{5.15}$$

as proposed by Hawe *et al.* in [38]. The function g is a sparsity promoting function that acts as a smooth approximation of the ℓ_0-quasi-norm. In addition to this sparse representation quality measure we also have to enforce certain properties on the co-sparse operator $\mathbf{\Omega}$ during training to avoid trivial solutions. Typical properties are mutual incoherence, i.e., a certain degree of independence between the rows of $\mathbf{\Omega}$, as well as full rank of the operator. For this purpose we follow [38] and define the functions h, which penalizes rank deficient operators, and r, which measures the mutual incoherence, as

$$h(\mathbf{\Omega}) \coloneqq -\tfrac{1}{p\log(p)} \log\det\left(\tfrac{1}{d}\mathbf{\Omega}^\top\mathbf{\Omega}\right), \tag{5.16}$$

$$r(\mathbf{\Omega}) \coloneqq -\sum_{k<l} \log\left(1-(\boldsymbol{\omega}_k^\top\boldsymbol{\omega}_l)^2\right), \tag{5.17}$$

where $\boldsymbol{\omega}_k^\top, \boldsymbol{\omega}_l^\top$ are the k-th and l-th row of $\mathbf{\Omega}$. The resulting cost function for co-sparse analysis operator learning for a single sample $\mathbf{x}$ has the shape $f_\mathbf{x}(\mathbf{\Omega}) = g(\mathbf{\Omega}\mathbf{x}) + p(\mathbf{\Omega})$ with the structural penalty function $p(\mathbf{\Omega}) = \mu_1 h(\mathbf{\Omega}) + \mu_2 r(\mathbf{\Omega})$. The parameters $\mu_1, \mu_2 > 0$ control the impact of the respective penalty.

In addition to these regularizers an additional restriction on the co-sparse analysis operator is required in order to avoid scaling issues. Analogously to the dictionary learning approach, where each operator column was normalized to unit norm, a commonly applied constraint is to require that the rows of $\mathbf{\Omega}$ are normalized. As previously stated, this constraint set is referred to as the oblique manifold. By slight abuse of notation we denote this constraint set as

$$\mathrm{Ob}(p,d) \coloneqq \{\mathbf{\Omega}^\top \in \mathbb{R}^{p\times d} \,:\, (\mathbf{\Omega}\mathbf{\Omega}^\top)_{ii} = 1,\ i = 1,\dots,d\}, \tag{5.18}$$

where instead of having normalized columns, cf. Equation (4.11), this set features normalized rows. This enables us to state the optimization problem for finding a co-sparse analysis operator that produces the most co-sparse signal representation of a training set $\mathbf{X} \in \mathbb{R}^{p\times n}$ as

$$\operatorname*{arg\,min}_{\mathbf{\Omega}\in\mathrm{Ob}(p,d)} \tfrac{1}{n}\sum_{i=1}^{n} g(\mathbf{\Omega}\mathbf{x}_i) + p(\mathbf{\Omega}). \tag{5.19}$$

After having established the optimization problem we are now able to discuss the sample complexity of this sparse representation learning algorithm.

The subsequently presented results are novel bounds for the generalization error of

co-sparse analysis operator learning. In the case of the Hoeffding & Covering bounding scheme the results behave with $\eta \propto \sqrt{\log(n)/n}$ that is driven by the factor $\sqrt{pd}$ and the Lipschitz constant of the cost function. The (HC) results are slightly more pessimistic than the results derived with (MR). The McDiarmid & Rademacher bounding scheme loses the mild $\sqrt{\log n}$ term and yields the rate $\eta \propto 1/\sqrt{n}$. The driving parameters here are again $\sqrt{p}, \sqrt{d}, L$, but instead of encountering the product of the signal and coefficient dimension they occur separately, resulting in a tighter bound.

Hoeffding & Covering

Proposition 5.3 (HC). *The generalization error of co-sparse analysis operator learning for n training samples, data distribution within the unit ball, L-Lipschitz cost function, and analysis operators $\mathbf{\Omega} \in \mathrm{Ob}(p, d)$ is given by*

$$\Phi(\mathbf{X}) \leq b\sqrt{\frac{dp\log(n)}{n}} + \frac{2b\sqrt{dp\log(3) + \log(2/\delta)} + \sqrt{8}L}{\sqrt{2n}}$$

with probability at least $1 - \delta$.

In order to fit the optimization problem into the framework of the Lipschitz continuity and covering number technique, we need to reformulate it slightly. Evaluating the sparsity of the co-sparse coefficient as in Equation (5.15) is equivalent to

$$f_{\mathbf{x}}(\mathbf{\Omega}) = \inf_{\boldsymbol{\alpha}} \tfrac{1}{2}\|\boldsymbol{\alpha} - \mathbf{\Omega}\mathbf{x}\|_2^2 + g(\boldsymbol{\alpha})$$

and the cost function that can be used to determine the optimal operator for a set of samples $\mathbf{x}_i$ is then given by

$$\tfrac{1}{n}\sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{\Omega}) + p(\mathbf{\Omega}).$$

The additional term $p(\mathbf{\Omega})$ that appears in this expression does not affect the outcome of the discussion of the generalization error, as it gets eliminated by the difference of the empirical and expected risk.

(HC1)

As per usual, we first introduce a minimal ε-cover $\{\mathbf{\Omega}_1, \dots, \mathbf{\Omega}_{N^{\mathrm{cov}}(\mathrm{Ob},\varepsilon)}\}$ of $\mathrm{Ob}(p,d)$ and then split the generalization error into three components

$$\begin{aligned}\Phi(\mathbf{X}) &= \sup_{\mathbf{\Omega}} \left|\tfrac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{\Omega}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega})]\right| \\ &\leq \sup_{j\in\{1,\dots,N^{\mathrm{cov}}(\mathrm{Ob},\varepsilon)\}} \left|\tfrac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{\Omega}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega}_j)]\right| \\ &\quad + \sup_{\mathbf{\Omega}} \left|\tfrac{1}{n}\sum_i (f_{\mathbf{x}_i}(\mathbf{\Omega}) - f_{\mathbf{x}_i}(\mathbf{\Omega}_j))\right| \\ &\quad + \sup_{\mathbf{\Omega}} \left|\mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega})]\right|.\end{aligned} \tag{5.20}$$

To bound the first term, we apply Hoeffding's inequality, which requires the boundedness of $f_{\mathbf{x}}$. The sparsity measure g as defined in Equation (5.15) fulfills this condition since by construction the rows of $\mathbf{\Omega}$ have unit norm and the signals are within the unit ball. Thus, the sparse coefficients are bounded by $0 \leq \alpha_i \leq 1$ for all i, which in turn implies that $\log(1+\nu\alpha_i^2) \leq \log(1+\nu) =: b$ for all i. The union bound argument and Hoeffding's inequality then yield

$$\sup_{j\in\{1,\dots,N^{\mathrm{cov}}(\mathrm{Ob},\varepsilon)\}} \left|\tfrac{1}{n}\sum_i f_{\mathbf{x}_i}(\mathbf{\Omega}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega}_j)]\right| \leq b\sqrt{\frac{\log(N^{\mathrm{cov}}(\mathrm{Ob},\varepsilon)) + \log(2/\delta)}{2n}} \tag{5.21}$$

with probability at least $1-\delta$.

(HC2)

By the stability assumption we have

$$|f_{\mathbf{x}}(\mathbf{\Omega}) - f_{\mathbf{x}}(\mathbf{\Omega}')| \leq L\|\mathbf{\Omega} - \mathbf{\Omega}'\|_F, \tag{5.22}$$

which allows us to bound the second term in (5.20) by $L\|\mathbf{\Omega} - \mathbf{\Omega}_j\|_F$. Observing that $\mathbf{\Omega}_j$ is an element of a minimal ε-cover of $\mathrm{Ob}(p,d)$, there exists a j such that for any $\mathbf{\Omega} \in \mathrm{Ob}(p,d)$ we have $\|\mathbf{\Omega} - \mathbf{\Omega}_j\|_F \leq \varepsilon$. The same holds for the third term, as the expectation can be

written as the limit of $\frac{1}{n}\sum_i f_{x_i}(\mathbf{D})$, and we get

$$\sup_{\mathbf{\Omega}} \left|\tfrac{1}{n}\sum_i (f_{\mathbf{x}_i}(\mathbf{\Omega}) - f_{\mathbf{x}_i}(\mathbf{\Omega}_j))\right| + \sup_{\mathbf{\Omega}} \left|\mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{\Omega})]\right| \leq 2L\varepsilon. \tag{5.23}$$

(HC3)

The final covering is achieved by again realizing that the oblique manifold is a product of spheres, and therefore the covering number can be expressed by $N^{\mathrm{cov}}(\mathrm{Ob}(p,d),\varepsilon) = (3/\varepsilon)^{dp}$. In summary, we have

$$\Phi(\mathbf{X}) \leq b\sqrt{\frac{dp\log(3/\varepsilon) + \log(2/\delta)}{2n}} + 2L\varepsilon \tag{5.24}$$

with probability at least $1-\delta$. For $\varepsilon = 1/\sqrt{n}$ this yields the proposition.

McDiarmid & Rademacher

Proposition 5.4 (**MR**). *The generalization error of co-sparse analysis operator learning for n training samples, data distributions within the unit ball, L-Lipschitz cost function, and analysis operators $\mathbf{\Omega} \in \mathrm{Ob}(p,d)$ can be bounded by*

$$\Phi(\mathbf{X}) \leq Ld\sqrt{\frac{2\pi p}{n}} + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}$$

which holds with probability greater than or equal to $1-\delta$.

(MR1)

We begin by introducing another set of samples $\mathbf{X}'$ which differs from $\mathbf{X}$ only in its j-th component denoted by $\mathbf{x}'_j$. In order to apply McDiarmid's inequality, we need to make sure that the absolute difference between $\Phi(\mathbf{X})$ and $\Phi(\mathbf{X}')$ is bounded. By using the fact that the difference of the supremum is smaller than or equal to the supremum of the differences, the absolute value of a supremum is smaller than or equal to the supremum of the absolute value, and by applying the triangle inequality we get

$$|\Phi(\mathbf{X}) - \Phi(\mathbf{X}')| \leq \sup_{\mathbf{\Omega}} \left|\tfrac{1}{n}(g(\mathbf{\Omega}\mathbf{x}_j) - g(\mathbf{\Omega}\mathbf{x}'_j))\right|. \tag{5.25}$$

Since $\mathbf{\Omega}$ is an element of the constraint set $\mathrm{Ob}(p, d)$, its largest singular value is bounded by $\sqrt{d}$. Leveraging the properties that g is L-Lipschitz, the bounded singular value of $\mathbf{\Omega}$, and the unit norm property of $\mathbf{x}_i$ yields the result that altering a single sample changes the value of $\Phi(\mathbf{X})$ by at most $2L\sqrt{d}/n$.

Therefore, all assumptions of McDiarmid's inequality are fulfilled, and we obtain the bound

$$\Phi(\mathbf{X}) \leq \mathbb{E}[\Phi(\mathbf{X})] + L\sqrt{\frac{2d\log(1/\delta)}{n}} \tag{5.26}$$

which holds with probability at least $1 - \delta$.

(MR2)

The typical symmetrization and second application of McDiarmid's inequality then yield with probability at least $1 - \delta$

$$\Phi(\mathbf{X}) \leq 2\,\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}}) + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}. \tag{5.27}$$

Lemma 4.9 allows us to replace the Rademacher with the Gaussian complexity $\widehat{\mathfrak{R}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}}) \leq \sqrt{\pi/2}\,\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathfrak{D}})$ and all that remains is to bound the Gaussian process $\frac{1}{n}\sum_{i=1}^{n}\gamma_i f_{\mathbf{x}_i}(\mathbf{\Omega})$.

(MR3)

The last bounding step is based on Slepian's Lemma. With the first Gaussian random process $G_{\mathbf{\Omega}} = \frac{1}{n}\sum_{i=1}^{n}\gamma_i f_{\mathbf{x}_i}(\mathbf{\Omega})$, the conditions of Slepian's Lemma are fulfilled by the Gaussian process $H_{\mathbf{\Omega}} = \frac{L}{\sqrt{n}}\langle \mathbf{\Gamma}, \mathbf{\Omega}\rangle_F$. The application of Slepian's Lemma then yields

$$\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} G_{\mathbf{\Omega}}\right] \leq \mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} H_{\mathbf{\Omega}}\right], \tag{5.28}$$

and with the definition of $H_{\mathbf{\Omega}}$ we can estimate this bound by

$$\begin{aligned}\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} H_{\mathbf{\Omega}}\right] &= \tfrac{L}{\sqrt{n}}\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}}\langle \mathbf{\Gamma}, \mathbf{\Omega}\rangle_F\right] \\ &= \tfrac{L}{\sqrt{n}}\mathbb{E}_\gamma\left[\sum_{j=1}^{d}\|\boldsymbol{\gamma}_j\|_2\right] \leq Ld\sqrt{\tfrac{p}{n}}.\end{aligned} \tag{5.29}$$

In combination with (5.26) we get the final generalization bound

$$\Phi(\mathbf{X}) \leq Ld\sqrt{\frac{2\pi p}{n}} + 3L\sqrt{\frac{2d\log(2/\delta)}{n}} \tag{5.30}$$

which holds with probability at least $1-\delta$.

5.3 Separable Co-sparse Analysis Operator

The co-sparse analysis operator model introduced in the previous section is designed to work with vectorized signals. That means that regardless of the signal domain the input data has to be reshaped to vector form, potentially losing relevant local information. This approach has the additional drawback that when working with inherently multidimensional data, such as real images, the accumulation of numerical cost restricts the algorithm to relatively small patches extracted from the data. In order to cope with this problem we borrow a technique known from image processing. As we briefly mentioned in the previous section, the rows of the analysis operator can be interpreted as filters. In image processing a filter is called separable if it can be written as the product of two or more (simple) filters [84]. Separability has the advantage that it reduces the amount of computations and retains local signal information. It was successfully applied to the dictionary learning model in [39, 73]. Separable co-sparse analysis learning for signals of arbitrary dimensionality was proposed in *"Separable cosparse analysis operator learning"* by **Seibert** *et al.*, [79], which merges the co-sparse analysis learning framework proposed in [38] and the separable dictionary learning scheme proposed in [39]. Separability of co-sparse analysis operators was also considered by Qi *et al.* in [71] in a setting limited to two-dimensional data.

In order to extend the non-separable co-sparse analysis operator model to a separable setting, we require some additional tools from the field of multilinear algebra. In particular, we adopt the notation introduced in [24].

Definition 5.5 (k-mode product)**.** Given a q-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_q}$ and the matrix $\mathbf{\Omega} \in \mathbb{R}^{J_k \times I_k}$ with $1 \leq k \leq q$. Their k-mode product is denoted by

$$\mathcal{X} \times_k \mathbf{\Omega}.$$

The resulting tensor has the size $I_1 \times I_2 \times \ldots \times I_{k-1} \times J_k \times I_{k+1} \times \ldots \times I_q$ with entries

$$(\mathcal{X} \times_k \mathbf{\Omega})_{i_1 i_2 \ldots i_{k-1} j_k i_{k+1} \ldots i_q} = \sum_{i_k=1}^{I_k} x_{i_1 i_2 \ldots i_q} \cdot \omega_{j_k i_q} \quad \text{for } j_k = 1, \ldots, J_k.$$

The k-mode product allows the manipulation of a single slice of a tensor, i.e., one dimension of the tensor, by a matrix. A visualization of this process is provided in Figure 5.2. This illustration gives an indication of the benefit of training a co-sparse analysis operator. Instead of working with vectorized patches which destroy the structure of the data (which for gray scale images would be two-dimensional), we work with the original data (or patches from that data, to be more precise). Maintaining the inherent data structure allows for a more efficient computation of the matrix multiplications involved in the learning process. Additionally, storing the separable operator requires less memory since instead of storing a large matrix, only the smaller component matrices need to be kept in memory. In summary, this approach conserves the inherent structure of the data and allows for processing larger image patches when working with real image data.

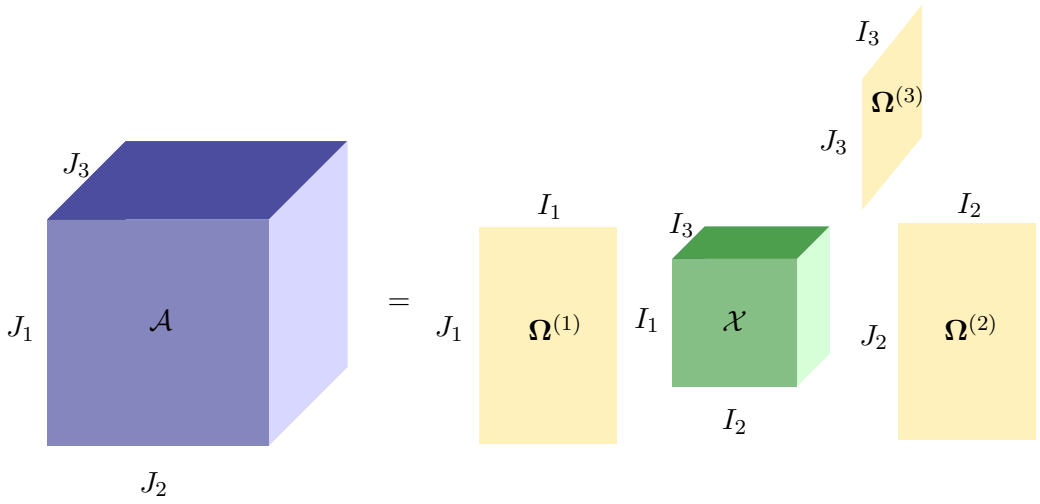


Figure 5.2: An illustration of the k-mode product for a three-dimensional tensor $\mathcal{X}$. The k-mode product allows the modification of slices of the tensor via a matrix.

For the later discussion it should be noted that the k-mode product from Definition 5.5 can also be expressed as a matrix-vector multiplication by using the Kronecker product

$\otimes$ and the vec operator which rearranges a tensor into a column vector. For the matrices $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{p \times r}$ the Kronecker product is defined as

$$\mathbf{A} \otimes \mathbf{B} \coloneqq \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix},$$

while the vec operator is defined as

$$\operatorname{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \in \mathbb{R}^{mn}.$$

With these tools we can rewrite the k-mode product as

$$\begin{aligned} & \mathcal{A} = \mathcal{S} \times_1 \mathbf{\Omega}^{(1)} \ldots \times_q \mathbf{\Omega}^{(q)} \\ \Leftrightarrow \quad & \operatorname{vec}(\mathcal{A}) = \left(\mathbf{\Omega}^{(1)} \otimes \mathbf{\Omega}^{(2)} \otimes \ldots \otimes \mathbf{\Omega}^{(q)} \right) \cdot \operatorname{vec}(\mathcal{S}). \end{aligned} \tag{5.31}$$

In the later discussion we require a succinct notation for computing the Kronecker product of the separable operator, effectively embedding it in a higher-dimensional space. For this purpose we define the embedding

$$\begin{aligned} \iota \colon \mathbb{R}^{J_1 \times I_1} \times \ldots \times \mathbb{R}^{J_q \times I_q} \quad & \to \quad \mathbb{R}^{\prod_k J_k \times \prod_k i_k}, \\ \left(\mathbf{\Omega}^{(1)}, \ldots, \mathbf{\Omega}^{(q)} \right) \quad & \mapsto \quad \mathbf{\Omega}^{(1)} \otimes \ldots \otimes \mathbf{\Omega}^{(q)}. \end{aligned} \tag{5.32}$$

Note, that the Kronecker product of two matrices with unit row norm is again a matrix that has unit row norm since for two vectors $\boldsymbol{\psi}, \boldsymbol{\omega}$ with $\|\boldsymbol{\psi}\|_2 = \|\boldsymbol{\omega}\|_2 = 1$ the equality $\|\boldsymbol{\psi} \otimes \boldsymbol{\omega}\|_2^2 = \sum_i \psi_i^2 \|\boldsymbol{\omega}_2\|_2^2 = \|\boldsymbol{\psi}\|_2^2 = 1$ holds. Thus, the embedding ι maps from the direct product of oblique manifolds which we denote by $\mathrm{Ob}_\times \coloneqq \mathrm{Ob}(p_1, d_1) \times \ldots \times \mathrm{Ob}(p_q, d_q)$ to the oblique manifold $\mathrm{Ob}(\prod_i p_i, \prod_i d_i)$.

All other components such as the sparsity promoting function g and the penalty function p remain the same as in the non-separable learning scenario and are defined in (5.15) and (5.16) & (5.17), respectively. The only difference is that the structural penalty p is applied to each component operator separately and that each $\mathbf{\Omega}^{(i)}$ is element of an appropriately sized

smaller oblique manifold $\mathrm{Ob}(p_i, d_i)$ for $i \in \{1, \dots, q\}$. For a signal $\mathcal{X} \in \mathfrak{X}$ with $\mathfrak{X} \subset \mathbb{R}^{p_1 \times \dots \times p_q}$ and operator $(\mathbf{\Omega}^{(1)}, \dots, \mathbf{\Omega}^{(q)})$ the sparse code $\mathcal{A}$ and penalty function g are defined as

$$\begin{aligned} \mathcal{A} &= \mathcal{X} \times_1 \mathbf{\Omega}^{(1)} \times_2 \mathbf{\Omega}^{(2)} \dots \times_q \mathbf{\Omega}^{(q)} \\ g(\mathcal{A}) &= \sum_i \log(1 + \nu \alpha_i^2), \end{aligned} \tag{5.33}$$

where the sum is taken over all entries of $\mathcal{A}$.

In order to discuss the sample complexity of the separable case, we need to make a slight modification to the cost function f. Instead of directly taking the separable $\mathbf{\Omega}$ as input, we first embed the operator via the function ι as defined in (5.32). Formally, we define the auxiliary function

$$\begin{aligned} \hat{f} \colon \mathrm{Ob}_\times \times \mathbb{R}^p &\to \mathbb{R} \\ (\mathbf{\Omega}, \mathbf{x}) &\mapsto g(\iota(\mathbf{\Omega})\mathbf{x}). \end{aligned} \tag{5.34}$$

This notation enables us to derive the sample complexity bounds in large parts analogously to the non-separable case.

The results stated in the following constitute the first discussion of generalization error bounds for separable co-sparse analysis operator learning and highlight the effect of the additional structure of the constraint set. The separability constraint directly affects the generalization error and results in a reduced driving constant, whereas the general behavior with respect to the number of samples remains the same as in the previously discussed case.

Hoeffding & Covering

Proposition 5.6 (**HC**). *For data distributions within the p-dimensional unit ball, L-Lipschitz cost function, and operators $\mathbf{\Omega}_i \in \mathrm{Ob}(p_i, d_i)$ for all $i \in \{1, \dots, q\}$ the following generalization error bounds for the separable co-sparse analysis operator learning algorithm trained on n samples.*

$$\Phi(\mathbf{X}) \leq \frac{b}{2} \sqrt{\frac{(\sum_{j=1}^q d_j p_j) \log(n)}{n}} + b \, \frac{\sqrt{(\sum_{j=1}^q d_j p_j) \log(3) + \log(2/\delta)} + \sqrt{8} L}{\sqrt{2n}}$$

which holds with at least $1 - \delta$ probability.

The first two steps (**HC1**) and (**HC2**) for the separable case remain exactly the same as for non-separable co-sparse analysis operator learning. In (**HC1**) no benefits can be gained from the application of the union bound argument and Hoeffding's inequality to the separable architecture, and for (**HC2**) the Lipschitz argument is identical to the non-separable discussion. We get

$$\Phi(\mathbf{X}) \leq b\sqrt{\frac{\log(N^{\mathrm{cov}}(\mathrm{Ob}_{\times}, \varepsilon)) + \log(2/\delta)}{2n}} + 2L\varepsilon \tag{5.35}$$

as a preliminary bound for the sample complexity which holds with probability at least $1-\delta$.

(HC3)

The third step is determining a covering number bound for the product of oblique manifolds. There are two ways to approach this problem. The first naive approach is to look at the embedding oblique manifold $\mathrm{Ob}(\prod_j p_j, \prod_j d_j)$, which yields the covering number

$$N^{\mathrm{cov}}\big(\mathrm{Ob}(\textstyle\prod_j p_j, \prod_j d_j), \varepsilon\big) \leq \left(\frac{3}{\varepsilon}\right)^{\prod_j d_j p_j} \tag{5.36}$$

since each row of $\mathrm{Ob}(\prod_j p_j, \prod_j d_j)$ has unit norm. Using this covering number provides no benefit over the bounds derived in the discussion of non-separable co-sparse analysis operators and is therefore not discussed any further.

The second approach is designed to conserve the inherent structure of direct products during the analysis, establish the covering number bound of each component oblique individually, and then combine the results. The covering number of a direct product of metric spaces is the product of the covering numbers and we get

$$N^{\mathrm{cov}}(\mathrm{Ob}_{\times}, \varepsilon) = \prod_j N^{\mathrm{cov}}(\mathrm{Ob}_j, \varepsilon) \leq \prod_j \left(\frac{3}{\varepsilon}\right)^{d_j p_j} = \left(\frac{3}{\varepsilon}\right)^{\sum_j d_j p_j}. \tag{5.37}$$

Equation (5.37) improves the covering number from (5.36) by replacing the product in the exponent with a sum, which yields a significant decrease in dimensionality. The resulting generalization error bound that holds with probability at least $1-\delta$ is

$$\Phi(\mathbf{X}) \leq b\sqrt{\frac{(\sum_{j=1}^{q} d_j p_j)\log(3/\varepsilon) + \log(2/\delta)}{2n}} + 2L\varepsilon. \tag{5.38}$$

With the choice of $\varepsilon = 1/\sqrt{n}$ we obtain the proposition.

Compared to the result for the non-separable case from Equation (5.30) the gains in terms of sample complexity can be entirely attributed to the lower covering number of the constraint set.

McDiarmid & Rademacher

Proposition 5.7 (**MR**). *For separable co-sparse analysis operator learning on n training samples distributed within the unit ball, operators $\mathbf{\Omega}_i \in \mathrm{Ob}(p_i, d_i)$ for $i \in \{1, \ldots, q\}$, and L-Lipschitz cost functions the MR scheme yields the generalization error bound*

$$\Phi(\mathbf{X}) \leq L \sum_{i=1}^{q} d_i \sqrt{\frac{2\pi p_i}{n}} + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}$$

that holds with probability at least $1 - \delta$.

We showed in the previous chapter on sparse dictionary learning that f is L-Lipschitz w.r.t. the dictionary $\mathbf{D}$. Therefore, the same holds true for the currently considered cost function w.r.t. the operator $\mathbf{\Omega}$. We are now working with the Kronecker product of matrices that each have unit row norm. By definition of the Kronecker product the rows of elements of $\mathrm{Ob}(p_1, d_1) \times \ldots \times \mathrm{Ob}(p_q, d_q)$ also have unit norm and are therefore element of an oblique manifold $\mathrm{Ob}(\prod_k p_k, \prod_k d_k)$. Thus, the Lipschitz property also holds for $\hat{f}$.

Steps (**MR1**) and (**MR2**) are equivalent to the discussion of the non-separable co-sparse analysis model and yield the preliminary bound $\Phi(\mathbf{X}) \leq \sqrt{2\pi}\,\widehat{\mathfrak{G}}_{\mathbf{X}}(\mathfrak{F}_{\mathrm{Ob}_\times}) + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}$ with probability at least $1 - \delta$. The generalization bound analysis for separable operators diverges in the final step which we investigate in the following.

(MR3)

We define the two Gaussian processes $G_{\mathbf{\Omega}} = \frac{1}{n}\sum_{i=1}^{n}\gamma_i \hat{f}(\mathbf{\Omega}, \mathbf{x}_i)$ with $\mathbf{\Omega} \in \mathrm{Ob}_{\times}$, and $H_{\mathbf{\Omega}} = \frac{L}{\sqrt{n}}\sum_{i=1}^{q}\langle \mathbf{\Gamma}^i, \mathbf{\Omega}^{(i)}\rangle_F$. Just as in the non-separable case, the inequality

$$\mathbb{E}_\gamma[(G_{\mathbf{\Omega}} - G_{\mathbf{\Omega}'})^2] \leq \tfrac{L^2}{n}\|\mathbf{\Omega} - \mathbf{\Omega}'\|_F^2 = \mathbb{E}_\gamma[(H_{\mathbf{\Omega}} - H_{\mathbf{\Omega}'})^2] \tag{5.39}$$

holds and we can apply the corollary of Slepian's Lemma to obtain the inequality $\mathbb{E}_\gamma[\sup_{\mathbf{\Omega}} G_{\mathbf{\Omega}}] \leq \mathbb{E}_\gamma[\sup_{\mathbf{\Omega}} H_{\mathbf{\Omega}}]$. The set $\mathrm{Ob}_{\times}$ is the direct product of oblique manifolds. Therefore, we can upper bound the expectation of the supremum of $H_{\mathbf{\Omega}}$ as follows.

$$\begin{aligned}
\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} H_{\mathbf{\Omega}}\right] &= \mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} \tfrac{L}{\sqrt{n}}\sum_{i=1}^{q}\operatorname{tr}\left((\mathbf{\Gamma}^{(i)})^\top \mathbf{\Omega}^{(i)}\right)\right] \\
&\leq \tfrac{L}{\sqrt{n}}\sum_{i=1}^{q}\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}^{(i)}}\operatorname{tr}\left((\mathbf{\Gamma}^{(i)})^\top \mathbf{\Omega}^{(i)}\right)\right] \\
&= \tfrac{L}{\sqrt{n}}\sum_{i=1}^{q}\mathbb{E}_\gamma\left[\sum_{j=1}^{d_i}\|\boldsymbol{\gamma}_j^{(i)}\|_2\right] \\
&\leq \tfrac{L}{\sqrt{n}}\sum_{i=1}^{q} d_i\sqrt{p_i}.
\end{aligned} \tag{5.40}$$

The vector $\boldsymbol{\gamma}_j^{(i)}$ denotes the j-th row of $\mathbf{\Gamma}^{(i)}$. The last inequality follows from applying Jensen's inequality and since the entries of $\mathbf{\Gamma}^{(i)}$ are all $\mathcal{N}(0,1)$ random variables. The final sample complexity bound that holds with probability at least $1-\delta$ then is given as

$$\Phi(\mathbf{X}) \leq L\sum_{i=1}^{q} d_i\sqrt{\frac{2\pi p_i}{n}} + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}. \tag{5.41}$$

In comparison, using the generalization error for a non-separable operator that has the same size as the Kronecker product of the sub-operators would be upper bounded by $L\prod_i d_i\sqrt{\frac{2\pi p_i}{n}} + 3L\sqrt{\frac{2d\log(2/\delta)}{n}}$.

5.4 Supervised Dictionary Learning

All algorithms presented up until now were unsupervised methods in the sense that only unlabeled training data was being processed. However, the proposed bounding frameworks are also capable of tackling supervised learning methods that attempt to recover a label corresponding to a data point. We will be focusing on a supervised learning method closely related to sparse dictionary learning presented in Chapter 4 in this section. Supervised dictionary models, which are also referred to as task-driven dictionary learning and predictive sparse coding, consist of two stages. In the first stage an encoder generates a sparse data representation, which is then used by a decoder that maps the sparse representation to the label space. Typical examples of task-driven dictionary learning are presented in [53–55] which all exhibit the structure described above.

In contrast to unsupervised dictionary learning methods, supervised methods seek to minimize a (supervised) loss by training a dictionary and a linear decoder in such a way that the decoder takes the sparse representations of the data as an input. In the supervised setting each input data point is actually a tuple $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is the actual data sample from $\mathbb{R}^p$, and the corresponding label $\mathbf{y}$ is an element of the label space $\mathbb{R}^q$. Given a data point $\mathbf{x}$ and the dictionary $\mathbf{D}$ the optimal sparse code is given as

$$\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}} \coloneqq \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + g(\boldsymbol{\alpha}), \tag{5.42}$$

where the sparse dictionary $\mathbf{D}$ is an element of the oblique manifold with unit norm columns. The authors of [53] propose using an elastic-net penalty function $g(\boldsymbol{\alpha}) \coloneqq r_1\|\boldsymbol{\alpha}\|_1 + r_2\|\boldsymbol{\alpha}\|_2^2$ with the regularization parameters $r_1, r_2 \in \mathbb{R}^+$. The decoder is then trained using a convex loss function that measures how accurately the label $\mathbf{y}$ can be approximated by observing the sparse code $\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}}$. There are several choices of loss function proposed by the authors. We discuss the least squares approach. The entire learning scheme for the encoder, i.e., the dictionary, and the decoder is described by the optimization problem

$$\min_{\mathbf{D},\boldsymbol{\Omega}\in\mathfrak{D}_e,\mathfrak{D}_d} \tfrac{1}{2n}\sum_{i=1}^{n} f_{(\mathbf{x}_i,\mathbf{y}_i)}(\mathbf{D},\boldsymbol{\Omega}) \coloneqq \min_{\mathbf{D},\boldsymbol{\Omega}\in\mathfrak{D}_e,\mathfrak{D}_d} \tfrac{1}{2n}\sum_{i=1}^{n} \|\mathbf{y}_i - \boldsymbol{\Omega}\boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}}\|_2^2. \tag{5.43}$$

There are several assumptions we have to make in order to be able to derive generalization

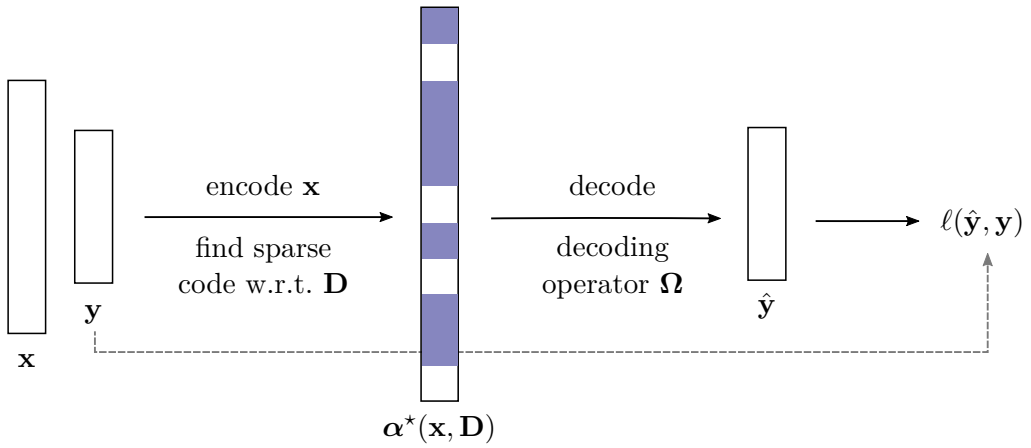


Figure 5.3: A schematic overview of supervised dictionary learning. Given a set of training data consisting of tuples $(\mathbf{x}, \mathbf{y})$ a dictionary $\mathbf{D}$ is trained that can be used to generate a sparse coefficient vector $\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}}$ for an element of the signal space. Additionally, a linear decoder $\boldsymbol{\Omega}$ is learned that maps the sparse code to the label space via $\hat{\mathbf{y}} = \boldsymbol{\Omega}\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}}$. The quality of the prediction is measured by the cost function ℓ.

bounds. The first assumption is that the encoder is Lipschitz w.r.t. the dictionary $\mathbf{D}$ with constant L_{enc} and bounded. The McDiarmid technique requires an additional condition on the sparse code. For two dictionaries $\mathbf{D}$, $\mathbf{D}'$ the optimal sparse code should not vary by more than a bound dependent on the dictionaries. Mairal *et al.* in [53] employ the elastic-net $\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + r_2\|\boldsymbol{\alpha}\|_2^2 + r_1\|\boldsymbol{\alpha}\|_1$ as a way to promote sparsity. As shown in [101] the solution to the elastic-net problem is equivalent to $\boldsymbol{\alpha}^\star$ that minimizes $\boldsymbol{\alpha}^\top(\mathbf{D}^\top\mathbf{D} + r_2\mathbf{I})\boldsymbol{\alpha}/(1 + r_2) - 2\mathbf{x}^\top\mathbf{D}\boldsymbol{\alpha} + r_1\|\boldsymbol{\alpha}\|_1$. Hence, the sparse codes are inherently influenced by the terms $\mathbf{D}^\top\mathbf{D}$ and $\mathbf{D}^\top\mathbf{x}$ and a suitable bound for the Lipschitz property is given by $\|\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}} - \boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}'}\|_2^2 \leq \lambda_1^2\|\mathbf{D}^\top\mathbf{D} - (\mathbf{D}')^\top\mathbf{D}'\|_F^2 + \lambda_2^2\|(\mathbf{D} - \mathbf{D}')^\top\mathbf{x}\|_2^2$ for some $\lambda_1, \lambda_2 > 0$. Further, the ℓ_2-norm of the encoder has to be bounded as well. We denote the maximal value by $\|\boldsymbol{\alpha}^\star_{\mathbf{x},\mathbf{D}}\|_2 \leq \alpha_{\max}$ which is a natural consequence to unit norm constrained signals and dictionaries with normalized columns. Next, we require the loss to be Lipschitz w.r.t. $(\mathbf{D}, \boldsymbol{\Omega})$ with Lipschitz constant L_{loss}, and the function value to be bounded by some constant b. This property holds similar to previous discussions for sample distributions within the unit ball. Finally, we assume that the hypothesis class $(\mathfrak{D}_e, \mathfrak{D}_d)$ can be covered using an ε-covering. This implies that the Frobenius norm of each weight matrix is bounded. In particular, for

the decoder we have $\|\mathbf{\Omega}\|_F \leq \omega_{\max}$. This condition is fulfilled by the oblique manifold which is the typical constraint set for the encoder and decoder, respectively.

The sample complexity of these supervised sparse coding methods has been investigated in [59] using covering numbers over the function space that is parameterized by $(\mathbf{D}, \mathbf{\Omega}) \in (\mathfrak{D}_e, \mathfrak{D}_d)$. For signals of dimension p and q-dimensional labels the resulting bounds decay as $\eta \propto \sqrt{\log(n)/n}$ with a driving factor of $\sqrt{pq}$. These bounds are constrained to the LASSO encoder [88] and univariate labels while the following results also hold for the more general elastic-net penalty and q-dimensional label space. The proposed HC bounding scheme yields results with the typical behavior $\eta \propto \sqrt{\log(n)/n}$ and expresses the dependence on the structure of the encoder and decoder via the factor $\sqrt{pqd^2}$. Since the following results incorporate labels of dimension q, a dependency on this variable is to be expected. The additional $\sqrt{d}$ stems from the separate consideration of encoding and decoding. The MR bound is primarily influenced by the factors $\sqrt{qd}$ and d^2, while being independent of the signal dimension. These results highlight the importance of the dimension of the sparse encoding in the supervised learning setting.

Hoeffding & Covering

Proposition 5.8 (HC). *For task-driven dictionary learning HC yields the following bound for data distributions within the p-dimensional unit ball, q-dimensional class labels, and assumptions as stated above. For training set size of n the inequality*

$$\Phi(\mathbf{X}) \leq \frac{b}{2}\sqrt{\frac{\log n}{n}} + \frac{1}{\sqrt{n}}\left(\sqrt{\frac{pqd^2\log(3) + \log(2/\delta)}{2}} + 2L\right)$$

holds with probability at least $1 - \delta$.

(HC1)

We begin as usual by introducing a minimal ε-covering of $(\mathfrak{D}_e, \mathfrak{D}_d)$ with elements $(\mathbf{D}_j, \mathbf{\Omega}_j)$, $j \in \{1, \dots, N^{\text{cov}}((\mathfrak{D}_e, \mathfrak{D}_d), \varepsilon)\}$ and then separating the generalization error to

$$\begin{aligned}
\Phi(\mathbf{X}) &= \sup_{\mathbf{D},\mathbf{\Omega}} \left| \tfrac{1}{n} \sum\nolimits_i f_{\mathbf{x}_i}(\mathbf{D}, \mathbf{\Omega}) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega})] \right| \\
&\leq \sup_{j \in \{1,\dots,N^{\text{cov}}\}} \left| \tfrac{1}{n} \sum\nolimits_i f_{\mathbf{x}_i}(\mathbf{D}_j, \mathbf{\Omega}_j) - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j, \mathbf{\Omega}_j)] \right| \\
&\quad + \sup_{\mathbf{D},\mathbf{\Omega}} \left| \tfrac{1}{n} \sum\nolimits_i f_{\mathbf{x}_i}(\mathbf{D}, \mathbf{\Omega}) - \tfrac{1}{n} \sum\nolimits_i f_{\mathbf{x}_i}(\mathbf{D}_j, \mathbf{\Omega}_j) \right| \\
&\quad + \sup_{\mathbf{D},\mathbf{\Omega}} \left| \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}_j, \mathbf{\Omega}_j)] - \mathbb{E}_{\mathbf{x}}[f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega})] \right|.
\end{aligned} \tag{5.44}$$

Under the assumption of boundedness on the loss function we can apply Hoeffding's inequality the first term is bounded by $b\sqrt{(\log(N^{\text{cov}}((\mathfrak{D}_e, \mathfrak{D}_d), \varepsilon)) + \log(2/\delta))/2n}$ with probability at least $1 - \delta$.

(HC2)

From the Lipschitz property of the loss function and bounded encoder property we know that

$$\|f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega}) - f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega}')\|_2 \leq \alpha_{\max} L_{\text{loss}} \cdot \|\mathbf{\Omega} - \mathbf{\Omega}'\|_2. \tag{5.45}$$

Furthermore, from the Lipschitz property of the encoder and the fact that there exists an ε-cover for $\mathfrak{D}_d$, we get

$$\|f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega}) - f_{\mathbf{x}}(\mathbf{D}', \mathbf{\Omega})\|_2 \leq \omega_{\max} L_{\text{loss}} L_{\text{enc}} \cdot \|\mathbf{D} - \mathbf{D}'\|_2. \tag{5.46}$$

By combining these two observations we see that the cost function $f_{\mathbf{x}}(\mathbf{D}, \mathbf{\Omega})$ is Lipschitz w.r.t. $(\mathbf{D}, \mathbf{\Omega}) \in (\mathfrak{D}_e, \mathfrak{D}_d)$ with constant $L \coloneqq L_{\text{loss}}(\alpha_{\max} + \omega_{\max} \cdot L_{\text{enc}})$.

With this result, terms two and three in Equation (5.44) can each be bounded by $L\varepsilon$ since there exists a $j \in \{1, \dots, N^{\text{cov}}((\mathfrak{D}_e, \mathfrak{D}_d), \varepsilon)\}$ such that $\mathrm{d}((\mathbf{D}, \mathbf{\Omega}), (\mathbf{D}_j, \mathbf{\Omega}_j)) \leq \varepsilon$ by definition of the ε-cover.

(HC3)

Combining the previous results yields

$$\Phi(\mathbf{X}) \leq b\sqrt{\frac{\log(N^{\mathrm{cov}}((\mathfrak{D}_e, \mathfrak{D}_d), \varepsilon)) + \log(2/\delta)}{2n}} + 2L\varepsilon \tag{5.47}$$

with probability at least $1 - \delta$.

For the final bound, we have to take a closer look at the constraint sets $\mathfrak{D}_e \subset \mathbb{R}^{p\times d}$ and $\mathfrak{D}_d \subset \mathbb{R}^{q\times d}$. To be concrete, we adopt the paradigm proposed for the dictionary learning setting and set $\mathfrak{D}_e$ to be the oblique manifold $\mathrm{Ob}(p, d)$ with normalized columns, cf. Equation (4.11), and $\mathfrak{D}_d$ to be the oblique $\mathrm{Ob}(q, p)$ with normalized rows, cf. Equation (5.18). Under these circumstances, the covering number of the direct product of these two constraint sets can be bounded by $N^{\mathrm{cov}}((\mathfrak{D}_e, \mathfrak{D}_d), \varepsilon) \leq (3/\varepsilon)^{pqd^2}$. With this result and by setting $\varepsilon = 1/\sqrt{n}$ we obtain the bound in the proposition.

McDiarmid & Rademacher

Proposition 5.9 (MR). *The generalization error bound achieved by the MR framework for task-driven dictionary learning and the same conditions as in Proposition 5.8 is given by*

$$\Phi(\mathbf{X}) \leq \frac{1}{\sqrt{n}}\left(\sqrt{2\pi}L_{loss}\omega_{\max}\big(\alpha_{\max}\sqrt{qd} + \lambda_1 d^2 + \lambda_2 d\big) + 3b\sqrt{\log(2/\delta)/2}\right)$$

which holds with probability at least $1 - \delta$.

(MR1)

Under the bounded loss assumption we can apply McDiarmid's inequality to upper bound the generalization error by

$$\Phi(\mathbf{X}) \leq \mathbb{E}_{\mathbf{X}}\big[\Phi(\mathbf{X})\big] + b\sqrt{\frac{\log(1/\delta)}{2n}} \tag{5.48}$$

with probability at least $1 - \delta$.

(MR2)

Using ghost sampling and the boundedness of the set $(\mathfrak{D}_e, \mathfrak{D}_d)$ we can again apply McDiarmid's inequality to bound the expectation in the previous term by the Rademacher complexity via $\mathbb{E}_{\mathbf{X}}[\Phi(\mathbf{X})] \leq 2\,\mathfrak{R}_n(\mathfrak{F}_{(\mathfrak{D}_e,\mathfrak{D}_d)})$. Due to the relation of Rademacher and Gaussian complexity we get with probability at least $1-\delta$

$$\Phi(\mathbf{X}) \leq \sqrt{2\pi}\,\widehat{\mathfrak{G}}_{\mathbf{X}}\big(\mathfrak{F}_{(\mathfrak{D}_e,\mathfrak{D}_d)}\big) + 3b\sqrt{\frac{\log(2/\delta)}{2n}} \tag{5.49}$$

with the empirical Gaussian complexity

$$\widehat{\mathfrak{G}}_{\mathbf{X}}\big(\mathfrak{F}_{(\mathfrak{D}_e,\mathfrak{D}_d)}\big) = \mathbb{E}_{\gamma}\left[\sup_{(\mathbf{D},\mathbf{\Omega})\in(\mathfrak{D}_e,\mathfrak{D}_d)} \tfrac{1}{n} G_{(\mathbf{D},\mathbf{\Omega})}\right] \tag{5.50}$$

and the Gaussian process

$$G_{(\mathbf{D},\mathbf{\Omega})} \coloneqq \sum_{i=1}^{n} \gamma_i f_{\mathbf{x}_i}(\mathbf{D},\mathbf{\Omega}). \tag{5.51}$$

(MR3)

Proposition 5.10. *The Gaussian process*

$$\begin{aligned} H_{(\mathbf{D},\mathbf{\Omega})} \coloneqq \sum\nolimits_{i=1}^{n} L_{loss}\Big(&\alpha_{\max}\cdot\langle\mathbf{\Gamma}_i^{(1)},\mathbf{\Omega}\rangle_F \\ &+\lambda_1\cdot\omega_{\max}\cdot\langle\mathbf{\Gamma}_i^{(2)},\mathbf{D}^\top\mathbf{D}\rangle_F \\ &+\lambda_2\cdot\omega_{\max}\cdot\langle\mathbf{\Gamma}_i^{(3)},\mathbf{D}^\top\mathbf{x}_i\rangle_F\Big) \end{aligned} \tag{5.52}$$

with $\mathbf{\Gamma}_i^{(1)} \in \mathbb{R}^{q\times d}$, $\mathbf{\Gamma}_i^{(2)} \in \mathbb{R}^{d\times d}$, *and* $\mathbf{\Gamma}_i^{(3)} \in \mathbb{R}^{d}$ *for all* $i \in \{1,\ldots,n\}$ *fulfills the condition* $\mathbb{E}_\gamma[(G_{(\mathbf{D},\mathbf{\Omega})} - G_{(\mathbf{D}',\mathbf{\Omega}')})^2] \leq \mathbb{E}_\gamma[(H_{(\mathbf{D},\mathbf{\Omega})} - H_{(\mathbf{D}',\mathbf{\Omega}')})^2]$ *with respect to the Gaussian process* (5.51).

Proof. Due to the i.i.d. nature of the standard normal distributed γ_i we get

$$\mathbb{E}_\gamma\big[(G_{\mathbf{D},\mathbf{\Omega}} - G_{\mathbf{D}',\mathbf{\Omega}'})^2\big] = \sum_{i=1}^{n}\big(f_{\mathbf{x}_i}(\mathbf{D},\mathbf{\Omega}) - f_{\mathbf{x}_i}(\mathbf{D}',\mathbf{\Omega}')\big)^2.$$

Then for any $i \in \{1, \dots, n\}$ by employing the Lipschitz continuity of the function f, the triangle inequality, the compactness of the set $\mathfrak{D}_d$, and the Lipschitz-continuity and boundedness of the encoder, we get the following series of inequalities

$$\begin{aligned}
&\big(f_{\mathbf{x}_i}(\mathbf{D}, \mathbf{\Omega}) - f_{\mathbf{x}_i}(\mathbf{D}', \mathbf{\Omega}')\big)^2 \\
&\quad \leq L_{\text{loss}}^2 \|\mathbf{\Omega}\boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}} - \mathbf{\Omega}'\boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}'}\|_2^2 \\
&\quad \leq L_{\text{loss}}^2 \left(\alpha_{\max}^2 \|\mathbf{\Omega} - \mathbf{\Omega}'\|_F^2 + \omega_{\max}^2 \|\boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}} - \boldsymbol{\alpha}^\star_{\mathbf{x}_i,\mathbf{D}'}\|_2^2 \right) \\
&\quad \leq L_{\text{loss}}^2 \left(\alpha_{\max}^2 \|\mathbf{\Omega} - \mathbf{\Omega}'\|_F^2 + \omega_{\max}^2 \big(\lambda_1^2 \|\mathbf{D}^\top\mathbf{D} - (\mathbf{D}')^\top\mathbf{D}'\|_F^2 + \lambda_2^2 \|(\mathbf{D} - \mathbf{D}')^\top \mathbf{x}_i\|_2^2\big) \right).
\end{aligned}$$

In the next equation we drop the parameters L_{loss}, $\alpha_{\max}$, $\omega_{\max}$, λ_1, and λ_2 to simplify the notation and improve legibility. By taking the sum of the last line of the previous equation over i, we see that

$$\begin{aligned}
&\sum_{i=1}^{n} \left(\|\mathbf{\Omega} - \mathbf{\Omega}'\|_F^2 + \|\mathbf{D}^\top\mathbf{D} - (\mathbf{D}')^\top\mathbf{D}'\|_F^2 + \|(\mathbf{D} - \mathbf{D}')^\top \mathbf{x}_i\|_2^2 \right) \\
&= \mathbb{E}_\gamma \left[\sum_i \big(\langle \mathbf{\Gamma}_i^{(1)}, \mathbf{\Omega} - \mathbf{\Omega}' \rangle_F^2 + \langle \mathbf{\Gamma}_i^{(2)}, \mathbf{D}^\top\mathbf{D} - (\mathbf{D}')^\top\mathbf{D}' \rangle_F^2 + \langle \mathbf{\Gamma}_i^{(3)}, (\mathbf{D} - \mathbf{D}')^\top \mathbf{x}_i \rangle_F^2 \big) \right] \\
&= \mathbb{E}_\gamma \Bigg[\Bigg(\big(\textstyle\sum_i \langle \mathbf{\Gamma}_i^{(1)}, \mathbf{\Omega} \rangle_F + \langle \mathbf{\Gamma}_i^{(2)}, \mathbf{D}^\top\mathbf{D} \rangle_F + \langle \mathbf{\Gamma}_i^{(3)}, \mathbf{D}^\top \mathbf{x}_i \rangle_F \big) \\
&\qquad - \big(\textstyle\sum_i \langle \mathbf{\Gamma}_i^{(1)}, \mathbf{\Omega}' \rangle_F + \langle \mathbf{\Gamma}_i^{(2)}, (\mathbf{D}')^\top\mathbf{D}' \rangle_F + \langle \mathbf{\Gamma}_i^{(3)}, (\mathbf{D}')^\top \mathbf{x}_i \rangle_F \big) \Bigg)^2 \Bigg]
\end{aligned}$$

with appropriately sized matrices $\mathbf{\Gamma}_i^{(1)}, \mathbf{\Gamma}_i^{(2)}, \mathbf{\Gamma}_i^{(3)}$, $i = 1, \dots, n$ with i.i.d. normal Gaussian entries. This yields the proposition.

□

With Proposition 5.10 we can now apply Slepian's Lemma which yields $\mathbb{E}_\gamma[\sup_{\mathbf{D},\mathbf{\Omega}} G_{(\mathbf{D},\mathbf{\Omega})}] \leq \mathbb{E}_\gamma[\sup_{\mathbf{D},\mathbf{\Omega}} H_{(\mathbf{D},\mathbf{\Omega})}]$. In order to upper bound the expectation of the right-hand side, we investigate its three summands, namely $\alpha_{\max} L_{\text{loss}} \cdot \langle \mathbf{\Gamma}_i^{(1)}, \mathbf{\Omega} \rangle_F$, $\lambda_1 \omega_{\max} L_{\text{loss}} \cdot \langle \mathbf{\Gamma}_i^{(2)}, \mathbf{D}^\top\mathbf{D} \rangle_F$, and $\lambda_2 \omega_{\max} L_{\text{loss}} \cdot \langle \sum_{i=1}^n \mathbf{\Gamma}_i^{(3)}, \mathbf{D}^\top \mathbf{x}_i \rangle_F$, separately. The following estimates use the properties of the involved matrices, the i.i.d. Gaussian distribution of the

entries of $\mathbf{\Gamma}_i^{(k)}$ for $k = 1, 2, 3$ and $i = 1, \ldots, n$, the Cauchy-Schwarz inequality, and Jensen's inequality. For the first term, we get

$$\begin{aligned}
&\mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} \alpha_{\max} L_{\text{loss}} \cdot \langle \textstyle\sum_i \mathbf{\Gamma}_i^{(1)}, \mathbf{\Omega}\rangle_F\right] \\
&\quad\leq \mathbb{E}_\gamma\left[\sup_{\mathbf{\Omega}} \alpha_{\max} L_{\text{loss}} \cdot \|\textstyle\sum_i \mathbf{\Gamma}_i^{(1)}\|_F \|\mathbf{\Omega}\|_F\right] \\
&\quad\leq \alpha_{\max}\omega_{\max} L_{\text{loss}} \cdot \sqrt{\mathbb{E}_\gamma\left[\sum_{j,k=1}^{q,d} \big(\sum_{i=1}^{n} \gamma_{ijk}^{(1)}\big)^2\right]} \\
&\quad= \alpha_{\max}\omega_{\max} L_{\text{loss}} \cdot \sqrt{nqd}.
\end{aligned} \tag{5.53}$$

The second term is bounded via

$$\begin{aligned}
&\mathbb{E}_\gamma\left[\sup_{\mathbf{D}} \lambda_1\omega_{\max} L_{\text{loss}} \cdot \langle \textstyle\sum_i \mathbf{\Gamma}_i^{(2)}, \mathbf{D}^\top\mathbf{D}\rangle_F\right] \\
&\quad\leq \mathbb{E}_\gamma\left[\sup_{\mathbf{D}} \lambda_1\omega_{\max} \cdot L_{\text{loss}} \|\textstyle\sum_i \mathbf{\Gamma}_i^{(2)}\|_F \|\mathbf{D}^\top\mathbf{D}\|_F\right] \\
&\quad\leq \lambda_1\omega_{\max} L_{\text{loss}} d \cdot \sqrt{\mathbb{E}_\gamma\left[\sum_{j,k=1}^{d,d} \big(\sum_{i=1}^{n} \gamma_{ijk}^{(2)}\big)^2\right]} \\
&\quad= \lambda_1 L_{\text{loss}}\omega_{\max} d^2 \cdot \sqrt{n}.
\end{aligned} \tag{5.54}$$

And finally for the third summand we obtain

$$\begin{aligned}
&\mathbb{E}_\gamma\left[\sup_{\mathbf{D}} \lambda_2\omega_{\max} L_{\text{loss}} \cdot \langle \textstyle\sum_{i=1}^n \mathbf{\Gamma}_i^{(3)}, \mathbf{D}^\top\mathbf{x}_i\rangle_F\right] \\
&\quad= \mathbb{E}_\gamma\left[\lambda_2\omega_{\max} L_{\text{loss}} \cdot \sup_{\mathbf{D}} \textstyle\sum_{j=1}^d \langle \sum_{i=1}^n \gamma_{ij}^{(3)}\mathbf{x}_i, \mathbf{d}_j\rangle_2\right] \\
&\quad\leq \mathbb{E}_\gamma\left[\lambda_2\omega_{\max} L_{\text{loss}} \cdot \sup_{\mathbf{D}} \textstyle\sum_{j=1}^d \|\sum_{i=1}^n \gamma_{ij}^{(3)}\mathbf{x}_i\|_2 \|\mathbf{d}_j\|_2\right] \\
&\quad\leq \mathbb{E}_\gamma\left[\lambda_2\omega_{\max} L_{\text{loss}} d \cdot \|\textstyle\sum_{i=1}^n \gamma_{i1}^{(3)}\mathbf{x}_i\|_2\right] \\
&\quad\leq \lambda_2\omega_{\max} L_{\text{loss}} d \cdot \sqrt{\mathbb{E}_\gamma\big[\textstyle\sum_{j=1}^p (\sum_{i=1}^n \gamma_{i1}^{(3)}(\mathbf{x}_i)_j)^2\big]} \\
&\quad= \lambda_2\omega_{\max} L_{\text{loss}} d \cdot \sqrt{\textstyle\sum_{i,j=1}^{n,p} (\mathbf{x}_i)_j^2} \\
&\quad\leq \lambda_2\omega_{\max} L_{\text{loss}} d \cdot \sqrt{n}.
\end{aligned} \tag{5.55}$$

Combining all three estimates then yields the bound

$$\frac{\sqrt{2\pi}}{n}\mathbb{E}_\gamma\left[\sup_{\mathbf{D},\mathbf{\Omega}} H_{(\mathbf{D},\mathbf{\Omega})}\right] \leq \sqrt{\frac{2\pi}{n}}\omega_{\max}L_{\text{loss}}\big(\alpha_{\max}\sqrt{qd}+\lambda_1 d^2+\lambda_2 d\big). \tag{5.56}$$

By combining (5.48) and (5.56) we obtain the proposed generalization error bound of the MR scheme.

Chapter 6

Experimental Evaluation

In this chapter we provide an implementation of the co-sparse analysis learning framework for both separable and non-separable co-sparse analysis operators as defined in the sections 5.2 and 5.3. Both learning schemes are used to recover a ground truth operator from synthetic training data. The performance of both training scenarios is measured via the distance of the operator generated by the algorithm to the ground truth throughout the optimization procedure. The training is conducted via a geometric stochastic gradient descent algorithm that utilizes mini-batches, i.e., instead of relying on a single sample for a SGD update, each update step is based on a mini-batch of a fixed size k, where k of the n available training samples are picked at random in each iteration. We denote the mini-batch in the i-th iteration by $\mathbf{x}_{\{i\}}$ with the index set $\{i\} \subset \{1, \dots, n\}$ and $|\{i\}| = k$.

As discussed in Chapter 2 classic (stochastic) gradient descent methods defined in Euclidean space determine the next iterate by following the negative gradient starting from the current iterate for a certain distance. The concrete choice of this distance will be discussed later. This update strategy works in the Euclidean domain. However, when the updated parameter is constrained to a specific subset there is no guarantee that the next iterate fulfills the required conditions. Furthermore, the accurate definition of the gradient is slightly more complex when dealing with constrained subspaces. Constraint sets that occur in machine learning can often be interpreted as smooth sub-manifolds of Euclidean space. As discussed in detail in [1] the gradient of a function on a smooth sub-manifold is actually an element of the so-called tangent bundle which is the union of all tangent spaces of the manifold. For example, consider the p-dimensional unit sphere $S_{(p-1)}$. The gradient of a smooth function f defined on the sphere at a point $\mathbf{x}_i \in S_{(p-1)}$ is a vector

$\mathbf{g} \in \mathbb{R}^p$ that fulfills the condition $\mathbf{g}^\top \mathbf{x}_i = 0$. Assume that $\mathbf{g}$ is unequal to $\mathbf{0}_p$, then naively using a gradient descent upgrade with some step length $a > 0$ yields $\mathbf{x}_{i+1} = \mathbf{x}_i - a\mathbf{g}$. But $\|\mathbf{x}_{i+1}\|_2^2 = \langle \mathbf{x}_i - a\mathbf{g}, \mathbf{x}_i - a\mathbf{g} \rangle = \|\mathbf{x}_i\|_2^2 + \|\mathbf{g}\|_2^2 - 2a\langle \mathbf{x}_i, \mathbf{g} \rangle = \|\mathbf{x}_i\|_2^2 + \|\mathbf{g}\|_2^2 > 1$, and therefore $\mathbf{x}_{i+1}$ is no longer an element of $S_{(p-1)}$.

In order to provide a viable learning scheme, the update strategy has to ensure that all iterates are contained within the constraint set. A simple remedy that guarantees that the next iterate fulfills this condition is to take a step in the ambient space, i.e., compute the sum of the current iteration point and its scaled gradient, and then orthogonally project the result back onto the manifold. This technique is referred to as a retraction and was first introduced in [2] as a way to approximate more complex exponential maps and was used in [100] to train co-sparse analysis operators.

While this update strategy works well when only taking very small steps, i.e., staying within close proximity of the suspension point, the quality of the approximation degrades quickly as the step size increases. A more accurate way to perform the update is to search for the next iterate along the geodesic in the direction of the steepest descent. Hawe *et al.* show in [38] that using geodesics results in a much better performing co-sparse operator learning algorithm compared to geometric algorithms based on retractions. Geodesics are the equivalent to straight lines on manifolds and are in general difficult to determine as their computation requires solving a computationally expensive ordinary differential equation. As we have seen in the problem statement of co-sparse analysis operator learning in Section 5.2 the optimization is performed on the oblique manifold which admits a closed form solution for geodesics. Updating along geodesics is therefore a viable alternative to retractions in the considered learning scenario. In the following, we present a geometric optimization algorithm with adaptive step size selection.

6.1 Optimization on Matrix Manifolds

The research field that copes with optimization on manifolds is often referred to as geometric optimization. The work of Edelman *et al.* [28] provides a comprehensive introduction to first and second order optimization methods for matrices with orthogonality constraints. Absil *et al.* offer an exhaustive discussion of this topic in [1]. We are interested in geometric variants of stochastic gradient descent methods as proposed in [10], where a geometric SGD

variant with fixed step size was proposed. In *"Learning co-sparse analysis operators with separable structures"* by **Seibert** *et al.*, [80] we proposed a geometric stochastic gradient method with variable step size selection. In the following, we give a brief introduction to first order methods on Riemannian sub-manifolds and then propose a geometric learning algorithm for the co-sparse analysis model.

The constraint set relevant for co-sparse analysis operator learning as discussed in Section 5.2 is the set of matrices with unit norm rows, also called the oblique manifold

$$\mathrm{Ob}(p,d) \coloneqq \{\boldsymbol{\Omega}^\top \in \mathbb{R}^{p\times d} \,:\, (\boldsymbol{\Omega}\boldsymbol{\Omega}^\top)_{ii} = 1,\ i = 1,\dots,d\}, \tag{6.1}$$

which is a sub-manifold of the embedding Euclidean space $\mathbb{R}^{p\times d}$. In order to simplify the notation, we omit the dimensions and simply write Ob in the following.

First, in order to compute the gradient of a smooth function f defined on the manifold Ob, we consider an extension of f to $\mathbb{R}^{p\times d}$, denoted as $\hat{f}$, and compute the Euclidean gradient $\nabla\hat{f}(\boldsymbol{\Omega})$ in $\mathbb{R}^{d\times p}$. This gradient is computed with respect to the standard Frobenius inner product, i.e., $\langle \mathbf{A}, \mathbf{B}\rangle_F = \mathrm{tr}(\mathbf{A}\mathbf{B}^\top)$. The Riemannian gradient is then equivalent to the projection of $\nabla\hat{f}(\boldsymbol{\Omega})$ onto the tangent space of Ob at $\boldsymbol{\Omega}$. For an embedded manifold the tangent space at some point can be described as the vector space spanned by the velocities of all curves going through the respective point. For the oblique manifold it is defined as

$$T_{\boldsymbol{\Omega}}\,\mathrm{Ob} \coloneqq \{\mathbf{A} \in \mathbb{R}^{d\times p} \,:\, (\boldsymbol{\Omega}\mathbf{A}^\top)_{ii} = 0,\ i = 1,\dots,d\}, \tag{6.2}$$

i.e., the set of all matrices with rows orthogonal to the rows of $\boldsymbol{\Omega}$. The projection of a matrix $\mathbf{A} \in \mathbb{R}^{d\times p}$ onto this tangent space is achieved by the function

$$\Pi_{T_{\boldsymbol{\Omega}}\,\mathrm{Ob}}(\mathbf{A}) \coloneqq \mathbf{A} - \mathrm{diag}(\boldsymbol{\Omega}\mathbf{A}^\top)\cdot\boldsymbol{\Omega}, \tag{6.3}$$

where $\mathrm{diag}(\boldsymbol{\Omega}\mathbf{A}^\top)$ returns a diagonal matrix with the same diagonal entries as $\boldsymbol{\Omega}\mathbf{A}^\top$. In summary the Riemannian gradient of the smooth function f at some point $\boldsymbol{\Omega}$ is defined as the (d,p)-matrix

$$\mathbf{G}(\boldsymbol{\Omega}) \coloneqq \Pi_{T_{\boldsymbol{\Omega}}\,\mathrm{Ob}}(\nabla\hat{f}(\boldsymbol{\Omega})). \tag{6.4}$$

In order to define a geometric update step, we have to follow the geodesic emanating

from the current iterate in direction of the negative gradient. While in general computing a geodesic requires solving a differential equation, for the oblique manifold geodesics can be computed in closed form. This is achieved by taking advantage of the fact that the oblique manifold is simply a product of spheres, see [46].

Definition 6.1 (Geodesic on the sphere)**.** Given $\mathbf{x} \in S_{(p-1)}$ and $\mathbf{g} \in T_{S_{(p-1)}}\mathbf{x}$, i.e., $\langle \mathbf{g}, \mathbf{x} \rangle = 0$. The geodesic emanating from $\mathbf{x}$ in direction $\mathbf{g}$ is defined as

$$\mu_{\mathbf{x},\mathbf{g}}(t) = \begin{cases} \mathbf{x}, & \text{for } \|\mathbf{g}\| = 0; \\ \mathbf{x}\cos(t\|\mathbf{g}\|) + \mathbf{g}\frac{\sin(t\|\mathbf{g}\|)}{\|\mathbf{g}\|}, & \text{otherwise} \end{cases} \tag{6.5}$$

for $t > 0$.

Geodesics on Ob then are given as follows.

Definition 6.2 (Geodesic on the oblique manifold)**.** Given $\mathbf{\Omega} \in \mathrm{Ob}$ and $\mathbf{G} \in T_{\mathbf{\Omega}}\,\mathrm{Ob}$. The geodesic emanating from $\mathbf{\Omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_d]^\top$ in direction $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_d]^\top$ is defined as

$$\gamma_{\mathbf{\Omega},\mathbf{G}}(t) = [\mu_{\boldsymbol{\omega}_1,\mathbf{g}_1}(t), \ldots, \mu_{\boldsymbol{\omega}_d,\mathbf{g}_d}(t)]^\top \tag{6.6}$$

for $t > 0$.

Figure 6.1 provides an illustration of the update procedure on manifolds.

6.2 Averaged Armijo Step Size Selection

Like in the Euclidean setting there are various strategies to determine the length of the step taken along the geodesic for each update. The simplest way of doing so is to use a fixed step size which, when chosen appropriately, can lead to an algorithm that exhibits good convergence behavior. However, finding an appropriate value requires arduous fine tuning of the step size parameter. Instead, the algorithm we propose utilizes an adaptive step size method which starts with an initial step size a_0 and shrinks it until the resulting step size a suffices a condition that is based on the well-known Armijo condition. The original definition of the Armijo condition is

$$f\big(\gamma_{\mathbf{\Omega}_i,-\mathbf{G}_i}(a), \mathbf{x}_{\{i\}}\big) \leq f(\mathbf{\Omega}_i, \mathbf{x}_{\{i-1\}}) + a \cdot c \cdot \|\mathbf{G}_i\|_2^2, \tag{6.7}$$

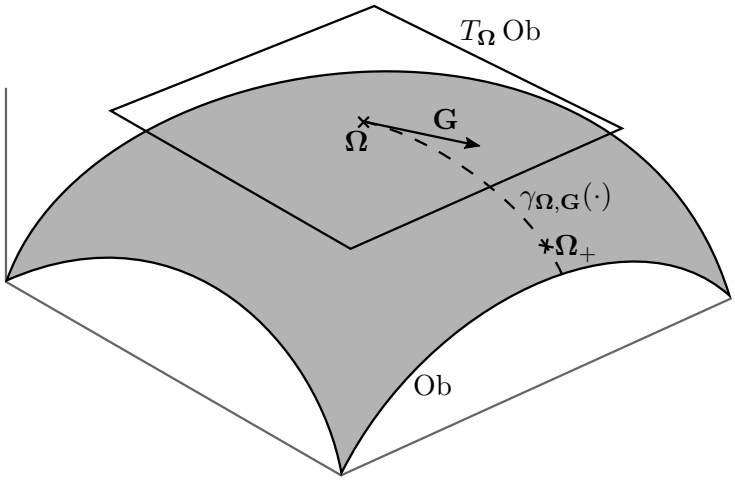


Figure 6.1: Geometric gradient descent on the oblique manifold Ob. Starting in $\mathbf{\Omega}$ we follow the direction $\mathbf{G}$ along the geodesic $\gamma_{\mathbf{\Omega},\mathbf{G}}(\cdot)$ until we reach the next iteration point $\mathbf{\Omega}_+$. $\mathbf{G}$ is an element of $T_{\mathbf{\Omega}}\,\mathrm{Ob}$, the tangent space of Ob at $\mathbf{\Omega}$.

where $\mathbf{G}_i \coloneqq \Pi_{T_{\mathbf{\Omega}}\,\mathrm{Ob}}(\nabla_{\mathbf{\Omega}_i}\hat{f}(\mathbf{\Omega}_i, \mathbf{x}_{\{i\}}))$ is the Riemannian gradient and $c \in (0,1)$ is some predefined parameter. Fulfillment of this condition leads to a more controlled convergence behavior of the algorithm. In fact, the Armijo condition is required to formally prove convergence of first order optimization algorithms such as gradient descent, see [98].

Using step size selection in conjunction with stochastic methods causes a new challenge. Since mini-batch SGD methods use only a subset of the available data, the updates of the cost function do not minimize the overall objective. In other words, the iterates generated by SGD approach the minimum of the optimization problem only on average. Thus, comparing the function value of the prospective next iterate to only the last function value is not a meaningful measure of the quality for the step size. Stochastic line search methods are further hindered by the fact that the new direction is not guaranteed to be a descent direction which is required for the Armijo step size selection by definition. As a result, situations can arise where the Armijo condition can never be fulfilled although the algorithm is not converged.

These issues are mitigated in part by the variation of the Armijo condition which we propose in the following. Instead of comparing the candidate for the next iterate to only the last function value, we average over a sliding window of the last w function values in order to capture the information gathered in the preceding mini-batches.

Definition 6.3 (Averaged Armijo condition)**.** Given the smooth cost function f on the oblique manifold, the geodesic γ as defined (6.6), the operators $\mathbf{\Omega}_i \in \mathrm{Ob}$, and let $\mathbf{x}_{\{i\}}$ be subsets of training samples of size k for $i \in \mathbb{N}$. The average over the last w function values is defined as

$$\bar{f}(\mathbf{\Omega}_i, \mathbf{X}) \coloneqq \frac{1}{w} \sum_{j=0}^{w-1} f(\mathbf{\Omega}_{i-j}, \mathbf{x}_{\{i-j-1\}}). \tag{6.8}$$

A step size $a > 0$ fulfills the *averaged Armijo condition* if

$$f\left(\gamma_{\mathbf{\Omega}_i, -\mathbf{G}_i}(a), \mathbf{x}_{\{i\}}\right) \leq \bar{f}(\mathbf{\Omega}_i, \mathbf{X}) + a \cdot c \cdot \|\mathbf{G}_i\|_2^2 \tag{6.9}$$

holds, with the Riemannian gradient $\mathbf{G}_i = \Pi_{T_{\mathbf{\Omega}_i}\,\mathrm{Ob}}(\nabla_{\mathbf{\Omega}} \hat{f}(\mathbf{\Omega}_i, \mathbf{x}_{\{i\}}))$ and the control parameter $c \in (0, 1)$.

We use the value $w = 2000$ and $c = 10^{-4}$ in the following experiments. The pseudo code for the averaged Armijo condition is presented in Algorithm 1.

Remark. In situations where (6.9) cannot be fulfilled, i.e., when $f(\gamma_{\mathbf{\Omega}_i, -\mathbf{G}_i}(a), \mathbf{x}_{\{i\}})$ is greater than $\bar{f}(\mathbf{\Omega}_i, \mathbf{X})$ for all $a > 0$, the averaging Armijo step size condition can never be fulfilled. In order to deter the algorithm from becoming stuck and to avoid unnecessary computations, we restrict the number of iterations for the line search algorithm to a maximum of $l_{\max}$, which we set to 40 in our experiments. If no step size that fulfills the averaged Armijo condition can be determined, we do not update $\mathbf{\Omega}_i$, and move on to the next mini-batch of training samples.

With these components we are almost able to state the optimization algorithm. The last missing aspect is a convergence criterion for the algorithm. It is well known that the typical convergence property of norm zero of the gradient does not hold for stochastic gradient methods as we briefly discussed in Section 2.2. Therefore, it is not feasible to use a threshold on the norm of the gradient to abort the iteration of the algorithm. Instead, we use the change of the function value as a measure. Concretely, we take an average over the v most

Algorithm 1 Averaged Armijo step size selection

Input: Scalars: a_0, $\beta \in (0,1)$, $c \in (0,1)$
Current operator: $\boldsymbol{\Omega}_i$
Training data: $\mathbf{X}$, $\mathbf{x}_{\{i\}}$
Cost function: $f(\cdot)$
Geodesic: $\gamma(\cdot)$
Integer: $l_{\max}$
$a \leftarrow a_0$
$l \leftarrow 0$
$\mathbf{G} \leftarrow \Pi_{T_{\boldsymbol{\Omega}_i}\,\mathrm{Ob}}\big(\nabla_{\boldsymbol{\Omega}} \hat{f}(\boldsymbol{\Omega}_i, \mathbf{x}_{\{i\}})\big)$
$\boldsymbol{\Omega}_{\text{temp}} \leftarrow \gamma_{\boldsymbol{\Omega}_i, -\mathbf{G}}(a)$
while $f(\boldsymbol{\Omega}_{\text{temp}}, \mathbf{x}_{\{i\}}) > \bar{f}(\boldsymbol{\Omega}_i, \mathbf{X}) + a \cdot c \cdot \|\mathbf{G}\|_2^2$
and $l < l_{\max}$ **do**
    $a \leftarrow \beta \cdot a$
    $l \leftarrow l + 1$
    $\boldsymbol{\Omega}_{\text{temp}} \leftarrow \gamma_{\boldsymbol{\Omega}_i, -\mathbf{G}}(a)$
end while
if $l < l_{\max}$ **then**
    $a_i \leftarrow a$
else
    $a_i \leftarrow 0$
end if
Output: a_i

recent iterations and compare them to the average of all function values up to this point. When this quantity falls below a certain threshold, the algorithm terminates. The function value for the current mini-batch is given as $f(\boldsymbol{\Omega}_i, \mathbf{x}_{\{i\}}) \coloneqq \frac{1}{k} \sum_{j \in \{i\}} f(\boldsymbol{\Omega}_i, \mathbf{x}_j)$. We denote the average of all previous iterations, and the average over the last v iterations respectively by

$$\phi_i \coloneqq \tfrac{1}{i} \sum_{j=1}^{i} f(\boldsymbol{\Omega}_j, \mathbf{x}_{\{j\}}), \tag{6.10}$$

$$\bar{\phi}_i \coloneqq \tfrac{1}{v} \sum_{j=i-v+1}^{i} \phi_j. \tag{6.11}$$

The learning algorithm is said to have converged when the value

$$\nu = (|\phi_i - \bar{\phi}_i|)/\bar{\phi}_i \tag{6.12}$$

falls below a certain threshold ε. We use the values $v = 200$ and $\varepsilon = 5 \cdot 10^{-5}$ in the following experiments.

We are now able to present the entire code for geometric stochastic gradient descent method with averaged Armijo condition. It is summarized in Algorithm 2.

Algorithm 2 Geometric mini-batch stochastic gradient descent

Input: Initial operator: $\mathbf{\Omega}_0$
  Training data: $\mathbf{X}$
  Cost function: $f(\cdot)$
  Geodesic: $\gamma(\cdot)$
  Parameters: $\varepsilon > 0$, $i_{\max} \in \mathbb{N}$
$i \leftarrow 0$
$\mathbf{G} \leftarrow \Pi_{T_{\mathbf{\Omega}_i}\mathrm{Ob}}\big(\nabla_{\mathbf{\Omega}} \hat{f}(\mathbf{\Omega}_i, \mathbf{x}_{\{i\}})\big)$
$\nu \leftarrow 1$
while $\nu > \varepsilon$ and $i < i_{\max}$ **do**
  Set step size a_i according to Algorithm 1.
  $\mathbf{\Omega}_{i+1} \leftarrow \gamma_{\mathbf{\Omega}_i, -\mathbf{G}}(a_i)$
  $\mathbf{G} \leftarrow \Pi_{T_{\mathbf{\Omega}_i}\mathrm{Ob}}\big(\nabla_{\mathbf{\Omega}} \hat{f}(\mathbf{\Omega}_i, \mathbf{x}_{\{i\}})\big)$
  Compute ϕ_i and $\bar{\phi}_i$ as defined in (6.10) and (6.11).
  $\nu \leftarrow |\phi_i - \bar{\phi}_i|/\bar{\phi}_i$
  $i \leftarrow i + 1$
end while
Output: $\mathbf{\Omega}_i$

6.3 Operator Recovery

For the experiment we first train a separable ground truth operator $\mathbf{\Omega}_{\mathrm{gt}}$ on 50 000 patches of real image data. The patches have size 7×7 and are randomly extracted from a set of training images. This operator is separable and due to the two-dimensional nature of the training data consists of two small analysis operators, each with 8 rows and 7 columns. The full operator therefore is an element of $\mathbb{R}^{64 \times 49}$.

In order to simulate operator recovery, we use the ground truth operator to generate signals that have a predefined co-sparsity of at least 15 by choosing a random support set $\Lambda \subset \{1, \dots, 64\}$, $|\Lambda| = 15$ for each sample. By $\mathbf{\Omega}_\Lambda$ we denote the restriction of $\mathbf{\Omega}$ to the rows corresponding to the support. We then use Gram-Schmidt orthonormalization to generate a matrix $\mathbf{B}_{\mathbf{\Omega}_\Lambda}$ such that the rows of $\mathbf{B}_{\mathbf{\Omega}_\Lambda}$ have the same span as the rows of $\mathbf{\Omega}_\Lambda$. The span of the rows of $\mathbf{\Omega}_\Lambda$, denoted by $\operatorname{rowspan}(\mathbf{\Omega}_\Lambda)$, are all vectors that can be expressed as linear combination of the rows of $\mathbf{\Omega}_\Lambda$. That is, $\mathbf{B}_{\mathbf{\Omega}_\Lambda}$ is chosen such that $\mathbf{B}_{\mathbf{\Omega}_\Lambda}\mathbf{B}_{\mathbf{\Omega}_\Lambda}^\top = \mathbf{I}_{15}$ and $\operatorname{rowspan}(\mathbf{B}_{\mathbf{\Omega}_\Lambda}) = \operatorname{rowspan}(\mathbf{\Omega}_\Lambda)$. Next, for each sample we generate a vector $\mathbf{x} \in \mathbb{R}^p$ with normal Gaussian entries and then project it to the space that is orthogonal to the row span of $\mathbf{\Omega}_\Lambda$ via $\Pi_{\operatorname{rowspan}(\mathbf{\Omega}_\Lambda)}(\mathbf{x}) = (\mathbf{I}_{49} - \mathbf{B}_{\mathbf{\Omega}_\Lambda}^\top \mathbf{B}_{\mathbf{\Omega}_\Lambda})\mathbf{x}$. The generated training samples are then scaled to unit norm and finally disturbed by some small Gaussian noise of standard deviation 0.05 to simulate small disturbances that occur in real world training data. For the following experiment we created 10 different synthetic data sets with 500 000 samples each. The initial estimation of the co-sparse analysis operator that is used to instantiate the learning algorithm is obtained by generating a matrix (or several matrices in the separable case) of appropriate size with Gaussian normal entries and then normalizing the rows to unit norm. With the training data and initial operator available we start the learning procedure with the same hyperparameters as set previously.

In order to compare how well the trained operator represents the ground truth, it is not sufficient to simply measure the Frobenius norm of the difference between the ground truth and the learned operator. There is no guarantee that the row position of the learned operator corresponds to the row position in the ground truth operator. To account for this misalignment, we use a metric specifically designed to cope with this variability. It is based on the so-called assignment problem [63] which consists of finding the minimum weight matching in a weighted bipartite graph. To adopt it to our setting, we compute the confusion matrix of the ground truth matrix $\mathbf{\Omega}_{\text{gt}}$ and the trained $\mathbf{\Omega}_{\text{train}}$ and subtract it from a matrix of corresponding size of all ones, i.e., $\mathbf{C} = \mathbf{1}_d\mathbf{1}_d^\top - \mathbf{\Omega}_{\text{gt}}\mathbf{\Omega}_{\text{train}}^\top \in \mathbb{R}^{d\times d}$. The entries of this matrix are between 0 and 1, and the closer to 0 an entry is, the more similar the corresponding rows of the operators are. Then we search for the permutation of d elements $\pi \in \operatorname{Sym}_d$ such that the sum over the entries of $\mathbf{C}$ corresponding to the indices $(i, \pi(i))$,

$i \in \{1, \ldots, d\}$ is minimized.

$$H(\mathbf{C}) := \min_{\pi \in \mathrm{Sym}_d} \sum_{i=1}^{d} c_{i,\pi(i)}. \tag{6.13}$$

An efficient algorithm to achieve this is the Hungarian method proposed in [63].

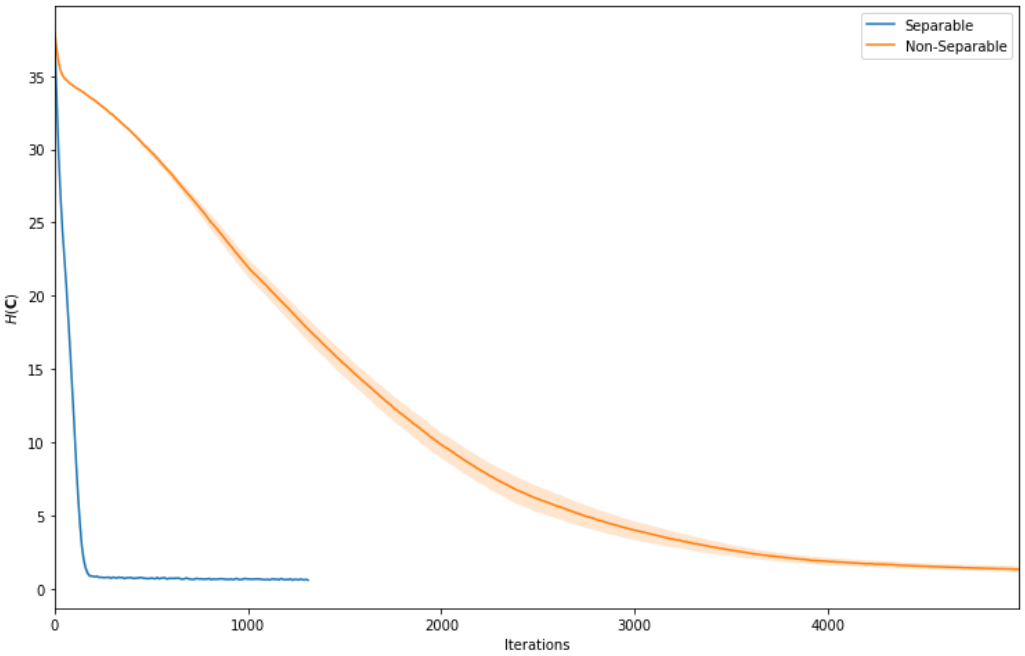


Figure 6.2: Development of the training error for recovering the ground truth operator for separable and non-separable learning with mini-batch size 500. The ground truth operator is a separable operator trained on real image data. The plotted lines are the mean of the reconstruction error measured via $H(\cdot)$ over 10 different data sets for the two considered methods. The lighter colored regions show the bootstrapped region of the 95% confidence intervals.

In order to evaluate the progression of the training algorithm, we observe the distance of the trained operator to the ground truth throughout the entire optimization process. The pictured plots in Figure 6.2 show the distance $H(\mathbf{C})$ of the learned to the ground truth operator for both separable and the non-separable learning algorithm for a mini-batch size of 500. The average distance over all training sets is represented by the solid line, and the bootstrapped 95% confidence interval is displayed as the shaded region around the graphs. Both algorithms start out at a distance of 40 to the ground truth operator. For the separable learning algorithm the distance rapidly decreases. At 100 iterations it is already below 10

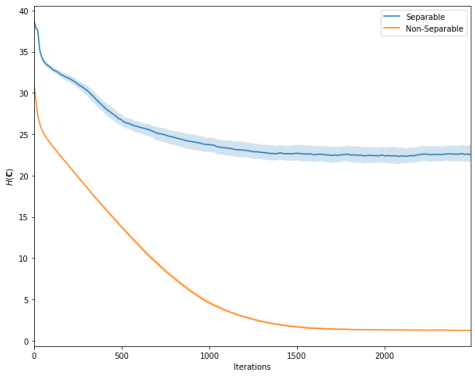
and at iteration 170 the average distance reaches values smaller than 1. After 1300 iterations the distance to the ground truth is at 0.2 and the algorithm terminates shortly after that. In contrast, the distance to the ground truth for the non-separable learning algorithm decreases much slower in this test setting. The average distance over the 10 synthetic training sets to the ground truth takes values below 10 after 2000 iterations and it requires 5000 iterations of the learning algorithm until the distance saturates at values slightly greater than 1.2.

In addition to the previous experiment where we measure the distance to a separable ground truth operator, we conduct the experiment with identical settings to recover a non-separable operator. That is, we train a non-separable operator on real image data and then use it to generate 10 sets of training data where each individual signal has a co-sparsity of approximately 15. For each of these 10 training sets we use both the separable as well as the non-separable algorithm to recover the ground truth operator.

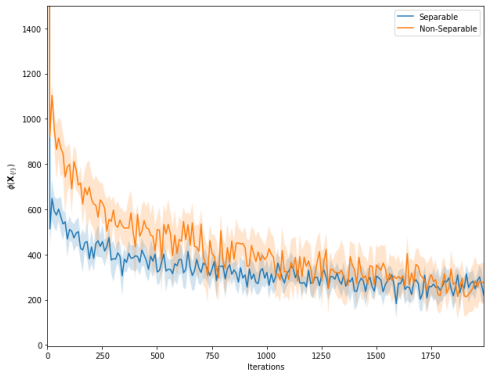
When plotting the distance to the ground truth in Figure 6.3a using $H(\mathbf{C})$ as the distance measure, we see that the separable co-sparse analysis operator learning procedure initially reduces the distance to the ground truth operator. However, after the initial progress the distance does not decrease significantly any more after iteration 1000. This is not surprising, as the ground truth operator to which we measure the distance to is itself non-separable but the separable learning algorithm is only able to produce separable operators and therefore faces an inherent performance disadvantage. The non-separable learning scheme on the other hand is able to recover an operator that by the measure $H(\mathbf{C})$ is much closer to the ground truth.

In order to provide another measure of relative performance, we compare a proxy to the generalization error of the two algorithms. Figure 6.3b shows the distance of the function value $f_{\mathbf{X}_{\{i\}}}(\mathbf{\Omega}_i)$ of the current operator $\mathbf{\Omega}_i$ w.r.t. the currently considered mini-batch to the expected empirical risk of the ground truth operator $\mathcal{E}_n(f_{\mathbf{\Omega}_{\mathrm{gt}}})$. We only plot the first 2500 iterations since after that the behavior of the algorithms does not change significantly. It is evident that both algorithms reach a similar distance to their respective empirical risk. It is noteworthy, however, that the separable method is initially faster in decreasing the generalization error proxy, and it takes the non-separable method more than 1000 iterations to reach a similar level.

The recovery performance of the separable operator in the case of separable ground truth data is remarkable when considering that the non-separable model has considerably more



(a) Distance of the learned operator to a non-separable ground truth operator measured via $H(\cdot)$.



(b) Plot of a proxy to the generalization error. The distance between the function value of the current operator on the respective mini-batch and the empirical risk of the function defined by the ground truth operator $\mathbf{\Omega}_{\text{gt}}$.

Figure 6.3: Experiments conducted with non-separable ground truth $\mathbf{\Omega}_{\text{gt}}$.

degrees of freedom during the training process. The separable model in the considered scenario has 112 trainable parameters while the non-separable one has 3136, i.e., 28 times more parameters. Still the non-separable model does not achieve a comparable accuracy. In a similar vein, the computation time of the separable algorithm is significantly lower than the non-separable one. While performing 1000 iterations for a mini-batch size of 500 takes the non-separable method on average 32 seconds when recovering the separable ground truth, the separable learning scheme only requires 16 seconds, a speed up of factor 2.

6.4 Variable Batch Size

In this experiment we examine the performance of the learning algorithm for separable and non-separable operators with different batch sizes. We again generate 10 training sets, each consisting of 500 000 training samples with a set co-sparsity of 15 and train both separable and the non-separable operators on 10 different mini-batch sizes. We abort the learning procedure after a maximum of 10 000 iterations. We then measure the distance of the learned operator to the ground truth via the distance measure $H(\mathbf{C})$ as defined in (6.13). The results are presented as a box plot in Figure 6.4. Each box represents the distance of the trained operator at convergence (or at the point when the maximum number of iterations is met)

to the ground truth operator gauged by the error measure defined in Equation (6.13). The horizontal black line within each box represents the median over the 10 training sets, the colored boxes indicate the quartiles, and the whiskers mark the minimum and maximum values for the respective batch size and method.

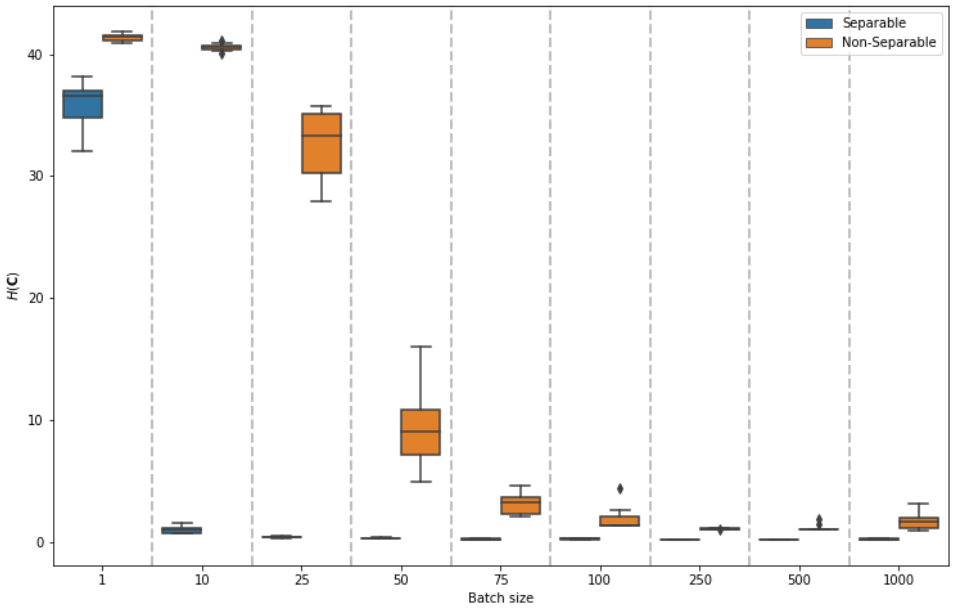


Figure 6.4: Distance of the learned operator to the ground truth operator measured via $H(\mathbf{C})$ for varying batch sizes.

By limiting the number of total iterations, this chart provides a way to relate the algorithmic performance of the learning procedure to the sample complexity results since for smaller mini-batch sizes the algorithm is limited in the number of samples it incorporates into the training procedure. It is only for batch sizes larger than 50 that all available training samples are included in the training at least once.

When using a single sample for each stochastic gradient descent step neither the separable, nor the non-separable operator show good results. This is mainly due to the fact that we abort after 10 000 iterations, and thus both algorithms only see as many samples. The

median distance to the ground truth for both algorithms is at 36 for the separable and 41 for the non-separable learning scheme. When the sample size increases, the error of the separable operator learning procedure decreases much more rapidly. Concretely, for a mini-batch size of 50, the average distance over all 10 training sets to the ground truth operator is already at $H(\mathbf{C}) = 0.7$. After that, larger batch sizes do not dramatically alter the result.

The non-separable algorithm still has an average distance of $H(\mathbf{C}) = 5.6$ for mini-batches of 50 samples. The batch size has to be increased to 500 samples for the non-separable algorithm to achieve results of similar quality as the separable approach. It achieves a median distance to the ground truth of slightly greater than 1.

The performance of non-separable and separable co-sparse analysis operator learning algorithms behave as proposed by the sample complexity results derived in the previous section. While this experiment cannot be used to reproduce the sample complexity bounds exactly, it does corroborate the theoretical findings from Chapter 5 that the additional structure that is enforced during the learning procedure for the separable approach decreases the amount of required training data significantly when the data can be adequately represented in a separable fashion.

Chapter 7

Conclusion

This thesis provided an analysis of the generalization error bounds and sample complexity of representation learning algorithms with a focus on sparse models. We devised two frameworks that offer different routes to derive generalization error bounds. Both bounding schemes were constructed in a readily applicable step-by-step manner. The representation models covered by both methods relied on the assumption that the information contained within a signal can be expressed via a dictionary which is applicable to a wide variety of representation models. The first technique labeled (HC) used the central argument of Hoeffding's concentration inequality in conjunction with the Lipschitz property of the empirical mean of the cost function. The final bound was obtained by a covering number argument that is applied to the constraint set that describes the considered class of hypotheses. The second technique (MR) employed McDiarmid's bounded difference concentration inequality and Rademacher/Gaussian complexity to measure the expressiveness of the hypothesis class. The final bounds were achieved by using a corollary of Slepian's Lemma which captures the structure of the parameterizing constraint set. The resulting bounds were inherently different. For n training samples (HC) yielded precision bounds $\eta \propto \sqrt{\log(n)/n}$. The second scheme (MR) resulted in bounds of order $\eta \propto \sqrt{1/n}$, which is a typical rate observed for error bounds obtained via techniques based on empirical processes. The behavior of both bounds was dependent on algorithm specific properties and the concrete bounds for each model highlighted the influences of the dimension of the signal, the structure of the dictionary employed in the respective model, and properties of the penalty function.

The two frameworks were applied to a variety of representation learning algorithms that cover the spectrum of sparsity-based representations and new bounds were derived for these

models. The first case study that also served as a tutorial to the bounding procedure was an analysis of the sparse dictionary learning model. The bounds achieved by the frameworks recovered previously existing results from literature for data distributed within the unit ball. A specialized investigation revealed that the scope of the (HC) results can be extended to sub-Gaussian data distributions due to the basic nature of the involved bounding steps. Furthermore, generalization error bounds for principal component analysis were provided. While the results did not achieve the same performance as state-of-the-art bounds from publications specialized on PCA, which are independent of the signal dimension, the achieved results exhibited the same behavior in regards of the number of available samples. That the proposed frameworks are also applicable to supervised representation models was shown by analyzing supervised dictionary learning with elastic-net penalty function. Novel bounds were devised for co-sparse analysis operator learning, a sparse analysis model closely related to dictionary learning. In addition to the standard, unstructured model, a variant that enforces a separable structure on the trained co-sparse operator was developed. The achieved results further supported the observation that enforcing additional structure on the dictionaries allows for more nuanced generalization error bounds. This thesis also illustrated the relation of the generalization error to the optimization error encountered in stochastic optimization methods, and recalled the bound of $E_{\text{est}} \leq 2\eta$ for the estimation error.

In the final chapter, an implementation was proposed for the co-sparse analysis learning problem. This chapter leveraged the inherent geometric properties of the learning problem to propose a geometric stochastic optimization algorithm that features a novel variable step size selection. The algorithm was used in experiments to recover a ground truth operator based on synthetic training data for both separable and non-separable co-sparse analysis operators. The performance of the recovery was tracked throughout the training procedure in order to provide insights into the actual speed of convergence of both model variants. The conducted experiments supported the theoretical findings of the previous chapter. As the sample complexity results suggested, adding a separable structure to the dictionary resulted in a faster recovery of the optimal operator. While the separable as well as the non-separable learning algorithm achieved a similar quality of recovery of the ground truth, the separable model approached the optimal solution much faster.

Bibliography

1. P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
2. R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub, "Newton's method on Riemannian manifolds and a geometric model for the human spine", *IMA Journal of Numerical Analysis*, vol. 22, no. 3, pp. 359–390, 2002.
3. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
4. P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds", *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.
5. P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results", *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.
6. P. L. Bartlett and S. Mendelson, "Empirical minimization", *Probability Theory and Related Fields*, vol. 135, no. 3, pp. 311–334, 2006.
7. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
8. D. P. Bertsekas, *Nonlinear programming*. Athena Scientific Belmont, 1999.
9. G. Blanchard, O. Bousquet, and L. Zwald, "Statistical properties of kernel principal component analysis", *Machine Learning*, vol. 66, no. 2-3, pp. 259–294, 2007.
10. S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds", *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

11. L. Bottou, "On-line learning and stochastic approximations", *On-line Learning in Neural Networks*, 9–42, Cambridge University Press, 1998.

12. L. Bottou, "Stochastic learning", *Advanced Lectures on Machine Learning*, pp. 146–168, 2004.

13. L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning", *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

14. O. Bousquet, "Concentration inequalities and empirical processes theory applied to the analysis of learning algorithms", PhD thesis, Ecole Polytechnique, 2002.

15. O. Bousquet and L. Bottou, "The tradeoffs of large scale learning", *Advances in Neural Information Processing Systems*, pp. 161–168, 2008.

16. A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images", *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

17. V. V. Buldygin and Y. V. Kozachenko, *Metric characterization of random variables and random processes.* American Mathematical Society, 2000.

18. E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?", *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

19. S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

20. D. Cohn and G. Tesauro, "How tight are the Vapnik-Chervonenkis bounds?", *Neural Computation*, vol. 4, no. 2, pp. 249–269, 1992.

21. L. Comtet, *Advanced Combinatorics: The Art of Finite and Infinite Expansions.* D. Reidel Publishing Company, 1974.

22. C. Cortes and V. Vapnik, "Support vector machine", *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

23. F. Cucker and S. Smale, "On the mathematical foundations of learning", *Bulletin of the American Mathematical Society*, vol. 39, pp. 1–49, 2002.

24. L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition", *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

25. L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition.* Springer, 1996.

26. R. M. Dudley, "Universal Donsker classes and metric entropy", *The Annals of Probability*, vol. 15, no. 4, pp. 1306–1326, 1987.

27. J. J. Duistermaat and J. A. Kolk, *Multidimensional Real Analysis I: Differentiation*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2004.

28. A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints", *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

29. A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant, "A general lower bound on the number of examples needed for learning", *Information and Computation*, vol. 82, no. 3, pp. 247–261, 1989.

30. M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors", *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

31. R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points – online stochastic gradient for tensor decomposition", *Proceedings of the 28th Conference on Learning Theory*, pp. 797–842, 2015.

32. R. Gribonval, R. Jenatton, and F. Bach, "Sparse and spurious: Dictionary learning with noise and outliers", *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6298–6319, 2015.

33. R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert, "Sample complexity of dictionary learning and other matrix factorizations", *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3469–3486, 2015.

34. M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent", *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1225–1234, 2016.

35. D. Haussler, “Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension”, *Journal of Combinatorial Theory, Series A*, vol. 69, no. 2, pp. 217–232, 1995.

36. D. Haussler, M. Kearns, H. S. Seung, and N. Tishby, “Rigorous learning curve bounds from statistical mechanics”, *Machine Learning*, vol. 25, no. 2–3, pp. 195–236, 1996.

37. S. Hawe, “Learning sparse data models via geometric optimization with applications to image processing”, PhD thesis, Technische Universität München, 2013.

38. S. Hawe, M. Kleinsteuber, and K. Diepold, “Analysis operator learning and its application to image reconstruction”, *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.

39. S. Hawe, M. Seibert, and M. Kleinsteuber, “Separable dictionary learning”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438–445, 2013.

40. W. Hoeffding, “Probability inequalities for sums of bounded random variables”, *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

41. M. Horvath and G. Lugosi, “A data-dependent skeleton estimate and a scale-sensitive dimension for classification”, 1997.

42. R. Jenatton, R. Gribonval, and F. Bach, “Local stability and robustness of sparse dictionary learning in the presence of noise”, Research Report, 2012.

43. J. L. W. V. Jensen, “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”, *Acta Mathematica*, vol. 30, no. 1, pp. 175–193, 1906.

44. I. T. Jolliffe, *Principal component analysis*. Springer, 1986.

45. J. Kiefer and J. Wolfowitz, “Stochastic estimation of the maximum of a regression function”, *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

46. M. Kleinsteuber and H. Shen, “Intrinsic Newton’s method on oblique manifolds for overdetermined blind source separation”, *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*, 2010.

47. V. Koltchinskii, “Rademacher penalties and structural risk minimization”, *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1902–1914, 2001.

48. V. Koltchinskii and D. Panchenko, “Rademacher processes and bounding the risk of function learning”, in *High Dimensional Probability II*, Springer, 2000, pp. 443–459.

49. M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and processes*. Springer Science & Business Media, 1991.

50. J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent converges to minimizers", *29th Annual Conference on Learning Theory*, Proceedings of Machine Learning Research, pp. 1246–1257, 2016.

51. G. Lugosi, "Improved upper bounds for probabilities of uniform deviations", *Statistics & probability letters*, vol. 25, no. 1, pp. 71–77, 1995.

52. G. Lugosi and M. Pintér, "A data-dependent skeleton estimate for learning", *Proceedings of the ninth Annual Conference on Computational Learning Theory*, pp. 51–56, 1996.

53. J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

54. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis", *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

55. J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning", pp. 1033–1040, 2009.

56. S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

57. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., 1982.

58. A. Maurer and M. Pontil, "K-dimensional coding schemes in Hilbert spaces", *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5839–5846, 2010.

59. N. Mehta and A. G. Gray, "Sparsity-based generalization bounds for predictive sparse coding", *Proceedings of the 30th International Conference on Machine Learning*, pp. 36–44, 2013.

60. R. Meir and T. Zhang, "Generalization error bounds for bayesian mixture algorithms", *Journal of Machine Learning Research*, vol. 4, pp. 839–860, 2003.

61. S. Mendelson, "Rademacher averages and phase transitions in Glivenko-Cantelli classes", *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 251–263, 2002.

62. M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT Press, 2012.

63. J. Munkres, "Algorithms for the assignment and transportation problems", *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

64. N. Murata, "A statistical study of on-line learning", *On-line Learning in Neural Networks*, pp. 63–92, 1998.

65. S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms", *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

66. J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

67. B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?", *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

68. J. M. R. Parrondo and C. Van den Broeck, "Vapnik-Chervonenkis bounds for generalization", *Journal of Physics A: Mathematical and General*, vol. 26, no. 9, pp. 2211–2223, 1993.

69. K. Pearson, "On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 6, pp. 559–572, 1901.

70. D. Pollard, *Convergence of stochastic processes*. Springer Science & Business Media, 1984.

71. N. Qi, Y. Shi, X. Sun, J. Wang, and W. Ding, "Two dimensional analysis sparse model", *IEEE International Conference on Image Processing*, pp. 310–314, 2013.

72. R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data", *Proceedings of the 24th International Conference on Machine Learning*, pp. 759–766, 2007.

73. R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua, "Learning separable filters", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2754–2761, 2013.

74. H. Robbins and S. Monro, “A stochastic approximation method”, *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

75. R. Rubinstein, T. Peleg, and M. Elad, “Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model”, *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.

76. N. Sauer, “On the density of families of sets”, *Journal of Combinatorial Theory, Series A*, vol. 13, no. 1, pp. 145–147, 1972.

77. M. Seibert, M. Kleinsteuber, R. Gribonval, R. Jenatton, and F. Bach, “On the sample complexity of dictionary learning”, *IEEE Statistical Signal Processing Workshop*, pp. 244–247, 2014.

78. M. Seibert, M. Kleinsteuber, and K. Hüper, “Properties of the BFGS method on Riemannian manifolds”, *Mathematical System Theory: Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday*, pp. 395–412, 2013.

79. M. Seibert, J. Wörmann, R. Gribonval, and M. Kleinsteuber, “Separable cosparse analysis operator learning”, *Proceedings of the 22nd European Signal Processing Conference*, pp. 770–774, 2014.

80. M. Seibert, J. Wörmann, R. Gribonval, and M. Kleinsteuber, “Learning co-sparse analysis operators with separable structures”, *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 120–130, 2016.

81. S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

82. J. Shawe-Taylor, C. K. I. Williams, N. Cristianini, and J. Kandola, “On the eigenspectrum of the Gram matrix and the generalization error of kernel PCA”, *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2510–2522, 2005.

83. D. Slepian, “The one-sided barrier problem for Gaussian noise”, *Bell Labs Technical Journal*, vol. 41, no. 2, pp. 463–501, 1962.

84. S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*. California Technical Publishing, 1997.

85. J.-L. Starck, E. J. Candès, and D. L. Donoho, “The curvelet transform for image denoising”, *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

86. M. Talagrand, "Sharper bounds for Gaussian and empirical processes", *The Annals of Probability*, pp. 28–76, 1994.

87. T. Tao, *Topics in random matrix theory*, ser. Graduate Studies in Mathematics, vol. 132. American Mathematical Society, 2012.

88. R. Tibshirani, "Regression shrinkage and selection via the LASSO", *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

89. I. Tošić and P. Frossard, "Dictionary learning", *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

90. D. Vainsencher, S. Mannor, and A. M. Bruckstein, "The sample complexity of dictionary learning", *Journal of Machine Learning Research*, vol. 12, pp. 3259–3281, 2011.

91. A. W. Van Der Vaart and J. A. Wellner, "Weak convergence", in *Weak Convergence and Empirical Processes*, Springer, 1996.

92. V. Vapnik, *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

93. V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities", *Theory of Probability and its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

94. V. N. Vapnik and A. Y. Chervonenkis, "Theory of pattern recognition: Statistical problems of learning", 1974.

95. N. Vayatis and R. Azencott, "Distribution-dependent Vapnik-Chervonenkis bounds", *European Conference on Computational Learning Theory*, pp. 230–240, 1999.

96. R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices", in *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012, pp. 210–268.

97. Z. Wen, B. Hou, and L. Jiao, "Discriminative nonlinear analysis operator learning: When cosparse model meets image classification", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3449–3462, 2017.

98. P. Wolfe, "Convergence conditions for ascent methods", *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.

99. J. Wörmann, S. Hawe, and M. Kleinsteuber, "Analysis based blind compressive sensing", *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 491–494, 2013.

100. M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations", *Proceedings of the 19th European Signal Processing Conference*, pp. 1470–1474, 2011.

101. H. Zou and T. Hastie, "Regularization and variable selection via the elastic net", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.