Iterative Soft-Thresholding from a Statistical Learning Perspective

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Ekkehard Schnoor, M.Sc.

aus

Stuttgart, Deutschland

Berichter: Prof. Dr. Holger Rauhut
Prof. Dr. Hartmut Führ

Tag der mündlichen Prüfung: 20. Juni 2024

Eidesstattliche Erklärung

Ich, Ekkehard Schnoor,

erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
3. Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
4. Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
5. Alle wesentlichen Quellen von Unterstützung wurden benannt;
6. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
7. Ein Teil oder Teile dieser Arbeit wurden zuvor veröffentlicht und zwar in:
   - M. Tiomoko, E. Schnoor, M.E.A. Seddik, I. Colin and A. Virmaux. "Deciphering Lasso-based Classification Through a Large Dimensional Analysis of the Iterative Soft-Thresholding Algorithm". In *Proceedings of the 39th International Conference on Machine Learning* (PMLR 162:21449-21477), 2022.
   - A. Behboodi, H. Rauhut and E. Schnoor. "Compressive Sensing and Neural Networks from a Statistical Learning Perspective". In *Compressed Sensing in Information Processing* (pp. 247-277). Birkhäuser, Cham, 2022.
   - E. Schnoor, A. Behboodi and H. Rauhut. "Generalization Error Bounds for Iterative Recovery Algorithms Unfolded as Neural Networks". Accepted for publication in *Information and Inference: A Journal of the IMA 12.3* (pp. 2267-2299), 2023.

Dresden, 4. Juni 2023

Abstract

This dissertation explores connections between the areas of compressive sensing and machine learning. It is centered around the so-called iterative soft-thresholding algorithm (ISTA), an iterative algorithm to solve the ℓ_1-regularized least squares problem also known as LASSO (least absolute shrinkage and selection operator) that has various applications in statistics and signal processing.

We will investigate two statistical learning problems that can be regarded as two different interpretations of the same underlying optimization problem and its solution through ISTA. While both are different, in common they have a generalization perspective, *i.e.,* we aim for finding performance guarantees at inference, that is when applying the trained model to new data samples that have not been used for training, but can be regarded as samples from the same underlying (but in practice, typically unknown) distribution. Thus, the contribution of this thesis lies in providing novel investigations of the iterative soft-thresholding algorithm from the viewpoint of statistical learning theory. We heavily rely on tools from high-dimensional probability theory to prove our results.

The first of the problems we consider deals with an interpretation of ISTA as a neural network, a topic which attracted attention with the rise of deep learning in the past decade. As a first step to introduce trainable parameters, we address a rather simple model, where a dictionary is learned implicitly. Then, we extend our results to a greatly generalized setup including a variety of ISTA-inspired neural networks, ranging from recurrent ones to architectures more similar to feedforward neural networks. Based on estimates of the Rademacher complexity of the corresponding hypothesis classes, we derive the first generalization error bounds for such specific neural network architectures and compare our theoretical findings to numerical experiments. While previous works strongly focused on generalization of deep learning in the context of classification tasks, we provide theoretical results in the context of inverse problems, which is much less explored in the literature.

The second problem considers the application of LASSO in a classification context, where the solution found through ISTA plays the role of a sparse linear classifier. Under realistic assumptions on the training data, we show that this induces a concentration on the distribution over the corresponding hypothesis class. This enables us to derive an algorithm to predict the classification accuracy based solely on statistical properties of the training data, which we confirm with the help of numerical experiments.

Zusammenfassung in deutscher Sprache

Die vorliegende Dissertation beschäftigt sich mit Themen an der Schnittstelle des Compressive Sensing und des Maschinellen Lernens. Schwerpunkt sind dabei Anwendungen des sogenannten iterativen Soft-Thresholding-Algorithmus (ISTA), eines iterativen Algorithmus zur Lösung des auch als LASSO (engl. *least absolute shrinkage and selection operator*) bekannten ℓ_1-regularisierten kleinste-Quadrate-Problems, mit zahlreichen Anwendungen in der Statistik und Signalverarbeitung.

Wir untersuchen zwei Probleme der statistischen Lerntheorie, die als zwei unterschiedliche Interpretationen des selben zugrundeliegenden Optimierungsproblems mitsamt der Lösung durch ISTA angesehen werden können. Obwohl sie verschieden sind, ist gemeinsam eine Untersuchung in Hinblick auf ihre Generalisierung, d.h. wir untersuchen die Genauigkeit der Vorhersage der Modelle auf Daten der zugrundeliegenden (und üblicherweise nicht bekannten) Verteilung, die jedoch nicht für die Optimierung des Modells verwendet wurden. Der Beitrag dieser Dissertation liegt somit in neuen Untersuchungen des iterativen Soft-Thresholding-Algorithmus aus Sicht der statistischen Lerntheorie. Für die Herleitung unserer Ergebnisse stützen wir uns stark auf Resultate aus der hochdimensionalen Wahrscheinlichkeitstheorie.

Das erste Problem betrifft die Interpretation von ISTA als neuronales Netz, ein Thema, das mit den Durchbrüchen im tiefen Lernen im letzten Jahrzehnt an Aufmerksamkeit gewonnen hat. In einem ersten Schritt zur Einführung trainierbarer Parameter befassen wir uns mit einem recht einfachen Modell, bei dem ein sogenanntes *dictionary* implizit gelernt wird. Anschließend dehnen wir unsere Ergebnisse auf ein deutlich allgemeineres Modell aus, das eine Vielzahl von ISTA-inspirierten neuronalen Netzen umfasst, von rekurrenten Netzen bis hin zu Architekturen, die klassischen sogenannten *feedforward*-Netzen ähnlicher sind. Wir leiten die ersten Schranken für den Generalisierungsfehler für diese Netzwerkarchitekturen her, die auf Abschätzungen der Rademacher-Komplexität der entsprechenden Hypothesenklassen beruhen, und vergleichen unsere theoretischen Ergebnisse mit numerischen Experimenten. Während frühere Arbeiten sich stark auf die Generalisierung des tiefen Lernens im Kontext von Klassifikationsaufgaben konzentrierten, liefern wir theoretische Ergebnisse mit den in diesem Zusammenhang weniger untersuchten inversen Problemen.

Das zweite Problem betrifft die Anwendung von LASSO im Kontext einer Klassifizierungsaufgabe, in dem die durch ISTA gefundene Lösung die Rolle eines dünnbesetzten linearen Klassifizierers spielt. Unter realistischen Annahmen über die Trainingsdaten zeigen wir, dass dies eine Konzentration auf die Verteilung über die entsprechende Hypothesenklasse induziert. Das ermöglicht es uns, einen Algorithmus zur Vorhersage der Klassifizierungsgenauigkeit zu entwickeln, der ausschließlich auf den statistischen Eigenschaften der Trainingsdaten beruht, und dies bestätigen wir mit Hilfe numerischer Experimente.

Notation

Let us introduce a set of notations and conventions that will be used throughout the entire thesis, besides the standard notation used in the mathematical literature.

Conventions. Throughout the thesis, we follow the conventions from mathematical statistics to denote the dimension by p and the sample size by n. As common in the compressive sensing literature, m is the number of measurements and s stands for the sparsity level of some sparse signal. Further, in case a flexible formulation is convenient, we usually employ the letter d (as in $\mathbb{R}^d$) for a general dimension with no fixed meaning.

Numbers, Vectors and Matrices. Based on the standard notation $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ for numbers, we will use (self-explanatory) expressions like $\mathbb{R}_{>0} := \{x \in \mathbb{R} : x > 0\}$ to denote the set of (strictly) positive real numbers. To make it easier to distinguish scalars from higher-dimensional objects, the latter are printed in bold type. Vectors $\boldsymbol{v} \in \mathbb{R}^p$ and matrices $\boldsymbol{A} \in \mathbb{R}^{p \times n}$ are denoted with bold (minuscule or capital, respectively) letters. $\mathbb{1}_p \in \mathbb{R}^p$ is the vector of all ones and $\boldsymbol{I}_p$ denotes the $p \times p$ identity matrix; if the size is clear from the context, we simply write $\boldsymbol{I}$. General norms are denoted by $\| \,.\, \|$. Concrete norms will always be specified, in particular we will make use of the vector ℓ_p-norm $\| \,.\, \|_p$ for $1 \leq p \leq \infty$ and the spectral norm $\|\boldsymbol{A}\|_{2 \to 2}$ as well as the Frobenius norm $\|\boldsymbol{A}\|_F$ for matrices. The transpose of a matrix $\boldsymbol{A}$ is $\boldsymbol{A}^\top$, and the trace of a quadratic matrix $\boldsymbol{A}$ is denoted by $\operatorname{tr}(\boldsymbol{A})$. The (decreasingly ordered) singular values of a rank-k matrix $\boldsymbol{A}$ are denoted by $\sigma_1(\boldsymbol{A}), \dots, \sigma_k(\boldsymbol{A})$, or shortly just $\sigma_1, \dots, \sigma_k$ if $\boldsymbol{A}$ is clear from the context. The Hadamard (or entrywise) product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of same size is written as $\boldsymbol{A} \odot \boldsymbol{B}$. The expression $\ker(\boldsymbol{A})$ denotes the kernel of the the linear mapping associated to $\boldsymbol{A}$. The notation $\mathcal{D}(\boldsymbol{A}) \in \mathbb{R}^p$ for a square matrix $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ is a vector containing the diagonal elements of $\boldsymbol{A}$. Furthermore, for a vector $\boldsymbol{v} \in \mathbb{R}^p$, we write $\operatorname{diag}(\boldsymbol{v})$ for the $(p \times p)$ diagonal matrix with $\boldsymbol{v}$ on its main diagonal. Sometimes functions $f : \mathbb{R} \to \mathbb{R}$ are applied to vectors $\boldsymbol{x} \in \mathbb{R}^p$, when $f(\boldsymbol{x}) \in \mathbb{R}^p$ stands for the element-wise application of f (analogously for matrices). Similarly, for mappings $f : \mathbb{R}^p \to \mathbb{R}^d$ we may apply them to matrices $\boldsymbol{X} \in \mathbb{R}^{p \times n}$, when $f(\boldsymbol{X}) \in \mathbb{R}^{d \times n}$ has to be understood as a column-wise application of f. Sometimes, we collect vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$ as columns in a matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$, expressed as $[\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$.

Sets, metric and normed spaces. Sets are usually denoted by italic letters like $\mathcal{S}$. Metric spaces are denoted by $(\mathcal{S}, d)$, and normed spaces by $(\mathcal{V}, \| \,.\, \|)$. Covering numbers of metric or normed spaces (or subsets thereof) are denoted by $\mathcal{N}(\mathcal{S}, d, \varepsilon)$ or $\mathcal{N}(\mathcal{V}, \| \,.\, \|, \varepsilon)$. The unit ball of an n-dimensional normed space $(\mathcal{V}, \| \,.\, \|)$ is denoted by $\mathcal{B}^n_{\|\cdot\|} := \{\boldsymbol{x} \in \mathcal{V} : \|\boldsymbol{x}\| \leq 1\}$, or simply by $\mathcal{B}^n_{\mathcal{V}}$ or $\mathcal{B}^n$, and similarly the sphere $\mathcal{S}^{n-1}_{\|\cdot\|} := \{\boldsymbol{x} \in \mathcal{V} : \|\boldsymbol{x}\| = 1\}$ or $\mathcal{S}^{n-1}_{\mathcal{V}}$, or shortly $\mathcal{S}^{n-1}$. As usual, $\mathcal{A}$ being a subset of $\mathcal{B}$ is written as $\mathcal{A} \subset \mathcal{B}$ (in the non-strict sense). The letter $\mathcal{H}$ is reserved for hypothesis spaces in a statistical learning setting. By $O(p)$ we denote the orthogonal group and $SO(p)$ is the special orthogonal group. For any $k \in \mathbb{N}$, we write $[k] := \{1, \dots, k\}$.

Probability theory and statistics. This thesis relies heavily on stochastical tools. Probability spaces $(\Omega, \mathcal{E}, \mathbb{P})$ (*i.e.,* a probability measure $\mathbb{P}$ defined on a σ-algebra $\mathcal{E}$ of an underlying set Ω) are typically omitted, as often we are only interested in the distribution of random variables X defined on some underlying probability space rather than the probability space itself. For instance, as usually done in the literature with a slight abuse of notation, we shortly write $\mathbb{P}(X > \lambda)$ instead of $\mathbb{P}(\{\omega \in \Omega : X(\omega) > \lambda\})$ for a real-valued random variable X defined on some underlying set Ω, and any scalar $\lambda \in \mathbb{R}$. Similarly, following the usual conventions in the literature, statements like "Let $\boldsymbol{x} \in \mathbb{R}^p$ be a random vector ..." have to be understood in the sense that $\boldsymbol{x} : \Omega \to \mathbb{R}^p$ is a random variable taking values in $\mathbb{R}^p$. One of the most important distributions is the Gaussian distribution. To denote a univariate Gaussian random variable of mean μ and variance σ^2, we write $\mathcal{N}(\mu, \sigma^2)$, and in particular $\mathcal{N}(0, 1)$ for the standard normal distribution. In the multivariate case $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean (vector) $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, e.g. $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for the most simple case of an isotropic Gaussian random vector with mean zero. Expectations of a random variable X (including vector- or matrix valued ones) are denoted by $\mathbb{E}[X]$, or shortly just by $\mathbb{E}X$ or $\bar{X}$, and the variance by $\mathrm{Var}(X)$. For a random vector $\boldsymbol{v} \in \mathbb{R}^p$, the matrix $\mathrm{Cov}(\boldsymbol{v}) = \boldsymbol{\Sigma}_v \in \mathbb{R}^{p \times p}$ denotes its covariance matrix (if it exists).

$\mathcal{O}$-Notation. The following is relevant to Chapter 3.

- Consider $\boldsymbol{x} = (1, \ldots, 1) \in \mathbb{R}^p$, so that $\|\boldsymbol{x}\|_2 = \sqrt{p}$. Interpreted as a sequence $(\boldsymbol{x}_p)_{p \in \mathbb{N}}$, but as usual omitting the index p, we have that $\|\boldsymbol{x}\|_2 = \mathcal{O}(\sqrt{p})$. Note that, for any absolute constant $\alpha \in \mathbb{R} \setminus \{0\}$ (the case of $\alpha = 0$ is not of interest anyways), there is also $\|\alpha \cdot \boldsymbol{x}\|_2 = \mathcal{O}(\sqrt{p})$.

  Example: In a binary classification problem, for feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ and labels $y_1, \ldots, y_n$ collected in $\boldsymbol{y} \in \{-1, 1\}^n$, there is $\|\boldsymbol{y}\|_2 = \mathcal{O}(\sqrt{n})$.

- This motivates to normalize $\boldsymbol{x} = (1/\sqrt{p}, \ldots, 1/\sqrt{p}) \in \mathbb{R}^p$, so that $\|\boldsymbol{x}\|_2 = 1$. Interpreted as a sequence, this is a sequence of unit vectors and we write $\|\boldsymbol{x}\|_2 = \mathcal{O}(1)$. We use this expression whenever there is no dimension-dependency and the limit depends only on absolute constants. For example, for any absolute $\alpha \in \mathbb{R}$ and $\alpha \boldsymbol{x} = (\alpha/\sqrt{p}, \ldots, \alpha/\sqrt{p}) \in \mathbb{R}^p$, there is $\|\alpha \boldsymbol{x}\|_2 = \alpha$ and we still write $\|\alpha \cdot \boldsymbol{x}\|_2 = \mathcal{O}(1)$. As another example, for $\boldsymbol{x} = (1/p, \ldots, 1/p) \in \mathbb{R}^p$ there is $\|\boldsymbol{x}\|_2 = \mathcal{O}(p^{-1/2})$.

- Moving to matrices and matrix norms, for the identity matrix $\boldsymbol{I}_p$ (or any multiple thereof) we have (independently of p) that $\|\boldsymbol{I}_p\|_{2\to 2} = 1$, so of course $\|\boldsymbol{I}_p\|_{2\to 2} = \mathcal{O}(1)$. However, there is a dimension-dependency with respect to the Frobenius norm, as $\|\boldsymbol{I}_p\|_F = \mathcal{O}(\sqrt{p})$. Considering a matrix $\mathbb{1}_{p \times p}$ whose entries are all 1 (or $\alpha \neq 0$), of course $\|\alpha \cdot \mathbb{1}_{p \times p}\|_F = \mathcal{O}(p)$.

- Note that when $p/n \to c \in (0, \infty)$, when the number of rows and columns of a random matrix are of the same order of magnitude, they can even be used interchangeable with respect to their asymptotic behavior. Thus, the assumption of a commensurable convergence rate is also helpful to enable simplifications.

- To ensure convergence, usually a normalization is required, and typically the data is normalized by $1/\sqrt{n}$ (or $1/\sqrt{p}$), which is suitable for most appropriate data models [LC18b].

  Examples: One application of this normalization (in the single asymptotic $n \to \infty$) is the introductory example of ridge regression (1.21). We may also need to balance the asymptotic growth rate of the data with that of hyperparameters of the algorithm; an example of this will be discussed in Assumption 3.9 in Chapter 3.

Contents

List of Figures

Summary

This dissertation explores connections between compressive sensing and statistical learning theory. It is centered around the classical ℓ_1-regularized least squares problem

$$\operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{0.1}$$

also known as the LASSO (least absolute shrinkage and selection operator) [Tib96]. Here, a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times p}$ and a vector $\boldsymbol{y} \in \mathbb{R}^m$ are given; the ℓ_1-regularizer is well-known to promote sparse (that is, containing only few non-zero entries) solutions $\boldsymbol{x} \in \mathbb{R}^p$. This convex but non-smooth optimization problem has no explicit solution, but iterative algorithms with convergence guarantees exist, of which here we focus on ISTA [DDDM04]. It can be derived using tools from convex optimization theory (proximal mappings) and be regarded as a *projected gradient descent*, consisting of a gradient descent step (of the non-regularized problem) followed by an application of the sparsity-promoting soft-thresholding function. Formally, after initializing $\boldsymbol{x}^0 = \boldsymbol{0}$, we compute recursively

$$\boldsymbol{x}^{k+1} = S_{\tau\lambda}\left(\boldsymbol{x}^k + \tau \boldsymbol{A}^\top (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^k)\right) \tag{0.2}$$

for $k \geq 0$, where the threshold $\lambda > 0$ and the step size $\tau > 0$ are parameters of the algorithm, and S_λ (applied entrywise) is the *soft-thresholding* or *shrinkage operator* defined as $S_\lambda(x) = \operatorname{sign}(x) \cdot \max(0, |x| - \lambda)$ for any $x \in \mathbb{R}$. It is well-known [DDDM04] that $(\boldsymbol{x}^k)_{k \in \mathbb{N}}$ converges to a minimizer of the optimization problem, if $\tau \|\boldsymbol{A}\|_{2 \to 2}^2 < 2$. While being well-understood from an optimization viewpoint, this thesis investigates two different interpretations of (0.1) as problems of statistical learning theory, and analyses them from a generalization viewpoint. Thus, we consider different machine learning tasks (such as regression or classification) using hypothesis classes that are build upon ISTA and are combined with appropriate loss functions that measure the performance of the models on given samples.

In Chapter 2, we take the viewpoint of compressive sensing, where $\boldsymbol{A} \in \mathbb{R}^{m \times p}$ is a *measurement matrix*, taking m linear measurements, where we aim to reconstruct the original signal $\boldsymbol{x} \in \mathbb{R}^p$ from the measurement vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m$. To make this task feasible, compressive sensing considers a sparsity prior on the signals $\boldsymbol{x} \in \mathbb{R}^p$, *i.e.*, it assumes that $\boldsymbol{x}$ contains only few non-zero entries. As a variant of this, we assume that $\boldsymbol{x} \in \mathbb{R}^p$ is sparse with respect to some (orthogonal) dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{p \times p}$, if $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{z}$ for $\boldsymbol{z} \in \mathbb{R}^p$ being the sparse representation of $\boldsymbol{x}$ with respect to $\boldsymbol{\Phi}$. By inserting this into (0.2) and by rearranging the terms, one ISTA iteration takes the form

$$\boldsymbol{z}^{k+1} = S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau \boldsymbol{\Phi}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{\Phi}\right)\boldsymbol{z} + \tau(\boldsymbol{A}\boldsymbol{\Phi})^\top \boldsymbol{y}\right],$$

which can be interpreted as a layer of a neural network with the weight matrix $\boldsymbol{I} - \tau\boldsymbol{\Phi}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{\Phi}$, bias $\tau(\boldsymbol{A}\boldsymbol{\Phi})^\top \boldsymbol{y}$ and activation function $S_{\tau\lambda}$, where the trainable parameters are the entries of $\boldsymbol{\Phi}$. (For now, we assume the step size τ and the threshold λ to be hyperparameters.) Generally, such an interpretation of ISTA as a neural network has

been first proposed in [GL10] in 2010, inspiring intense research activity combining compressive sensing and deep learning in the recent years. Most of this work has been experimental, while we approach this problem as a statistical learning problem and aim to prove bounds on the generalization error. Roughly speaking, this is defined as the difference between the training and test errors and measures how well the model generalizes to unseen data. To that end, we formulate a statistical learning problem and derive a generalization error bound based on a well-established general result employing the Rademacher complexity [BM02], a complexity measure for hypothesis classes in statistical learning theory. The main difficulty lies in the derivation of sharpest-possible estimates of the Rademacher complexity. A key ingredient of the proof is a generalized contraction principle [Mau16] for vector-valued function classes; furthermore, we apply Dudley's inequality [Dud67] to upper bound the Rademacher complexity and therefore derive estimates of the covering numbers in Dudley's integral.

Our main result is rigorously stated in Theorem 2.1 in Chapter 2, and establishes a uniform bound on the generalization error for this recovery problem, measuring the reconstruction error simply with respect to the ℓ_2-norm. Informally, denoting its true and empirical risk by $\mathcal{L}(h)$ and $\hat{\mathcal{L}}(h)$, the generalization error bound can be bounded as

$$|\mathcal{L}(h) - \hat{\mathcal{L}}(h)| \lesssim \sqrt{\frac{pm\log(L) + p^2\log(L)}{n}},$$

uniformly for all $h \in \mathcal{H}$, where n denotes the sample size and L stands for the number of iterations of ISTA, or layers of the ISTA-inspired neural network. Remarkably, we achieve an only logarithmic dependence of the generalization error on the number of layers, while it is generally well-known to be challenging to derive generalization error bounds of deep neural networks with a mild dependence on the depth that are able to explain the surprisingly good generalization often observed in practice.

Then, we extend our findings to a much more general scenario, including the dictionary learning problem above, and many other examples of practical interest. For instance, it includes the case that step sizes τ and thresholds λ are trainable in each layer, and a very general setup for weight matrices, so that we can cover a wide range of architectures from recurrent neural networks to networks more similar to standard feedforward neural networks. Furthermore, we allow flexible dimensions (*widths*, in the deep learning terminology), such that the model is suitable for other tasks than just reconstruction. Again, it is remarkable that we obtain a fairly mild linear dependence of the generalization error on the number of layers in the most general scenario covered in Theorem 2.11, which reduces to a logarithmic dependence in important special cases (Remark 2.12). The described results are layed out in detail in Chapter 2, which is based on the book chapter [BRS22] as well as the journal paper [SBR21], both of which are co-authored by the author of this thesis, together with Arash Behboodi and Holger Rauhut.

Chapter 3 is based on the same underlying optimization problem and its solutions through ISTA, but takes a different perspective by considering a binary classification problem. Here, with a different notation (0.1) reads as

$$\boldsymbol{\omega}^\star = \operatorname*{arg\,min}_{\boldsymbol{\omega}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{\omega}\|_2^2 + \lambda\|\boldsymbol{\omega}\|_1, \tag{0.3}$$

with the data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p\times n}$ collecting the samples from two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ and $\boldsymbol{y} \in \{-1, 1\}^n$ containing the corresponding labels. As the solution $\boldsymbol{\omega}^\star$ from (0.3), we obtain a linear classifier and thus a hypothesis class consisting of functions of the type $h_\omega(\boldsymbol{x}) = \text{sign}(\boldsymbol{\omega}^\top \boldsymbol{x})$, a basic linear model in machine learning. It is a building block of some of the most fundamental machine learning models including logistic regression with the logistic loss function $\ell(x, y, h_\omega) = \log(1 + \exp(-y h_\omega(x)))$, and support vector machines with a hinge loss $\ell(\boldsymbol{x}, y, \boldsymbol{\omega}) = \max(0, 1 - y \cdot \boldsymbol{x}^\top \boldsymbol{\omega})$. In contrast, the least-squares loss used in (0.3) is more established for regression than classification problems. The ℓ_1-regularizer promotes sparse solutions that discard most of the features; thus, it is useful in applications when only few of the features discrimate between the two classes.

Again, we are interested in the generalization of this model with respect to its accuracy (thus, using different loss functions for training and inference). Upper bounds for the generalization error for linear models are straightforward to obtain through Rademacher complexity and VC dimension bounds that can easily be computed for such simple models. Instead, we derive an algorithm that predicts the precise classification accuracy based on statistical properties of the training data, and depending on the regularization parameter λ, enabling applications to hyperparameter optimization. Here, it will be useful to consider the following random fixed point equation

$$\boldsymbol{\omega}^\star = \Phi(\boldsymbol{X})(\boldsymbol{\omega}^\star) = S_{\tau\lambda}\left(\boldsymbol{\omega}^\star + \tau \boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}^\star)\right), \tag{0.4}$$

where the randomness in the data matrix $\boldsymbol{X}$ induces randomness in the function $\Phi(\boldsymbol{X}) : \mathbb{R}^p \to \mathbb{R}^p$, and therefore a probability distribution over the solution $\boldsymbol{\omega}^\star$ of the fixed point equation. Thus, if well defined, $\boldsymbol{\omega}^\star$ as implicitly given in (0.4), is a random vector in $\mathbb{R}^p$ whose properties depend on the distribution of $\boldsymbol{X}$. As an assumption that is both capable of modelling realistic data, and restrictive enough from a mathematical viewpoint, we assume that the data points $\boldsymbol{x} \in \mathbb{R}^p$ satisfy the concentration inequality

$$\mathbb{P}\left(|\varphi(\boldsymbol{x}) - \mathbb{E}[\varphi(\boldsymbol{x})]| \geq t\right) \leq Ce^{-(t/c)^2} \qquad \forall t > 0 \tag{0.5}$$

for any 1-Lipschitz (with respect to the ℓ_2-norm) real valued function $\varphi : \mathbb{R}^p \to \mathbb{R}$. Here, $c, C > 0$ are absolute constants that do not depend on n and p and thus enable the application of tools from random matrix theory to problems from high-dimensional statistics [CL22]. Variants of (0.5) are ubiquitous in high-dimensional probability theory and are generally based on the so-called *concentration of measure phenomenon* [Led01] in high dimensions. A key finding formulated in Theorem 3.13 in Chapter 3 is that, under the above assumption on the underlying data distribution, the random vector $\boldsymbol{\omega}^\star$ as implicitly defined by (0.4) is tightly concentrated. This result is obtained through an application of a probabilistic variant and extension [LC20, Theorem 5] of the classical fixed point theorem of Banach. Thus, in contrast to generalization error bounds based on uniform convergence, we obtain a distribution on the hypothesis class, and we propose an algorithm to approximately compute its mean and its covariance, or, more precisely, so-called *deterministic equivalents* thereof. Namely, by being able to express the performance of the classifier in terms of few *scalar observations* like the Lipschitz functions φ above, this is sufficient to predict the performance of the classifier. A great technical challenge arises from the intricate dependencies induced by the iterative procedure of ISTA. A key tool for dealing with those dependencies is the so-called *leave-one-out approach* [EK09], where we may omit the ith sample $\boldsymbol{x}_i$ from the data matrix $\boldsymbol{X}$, e.g. replacing it by the zero vector, to obtain the modified data matrix $\boldsymbol{X}_{-i}$ that is deprived of the ith entry and independent of $\boldsymbol{x}_i$. This work will be presented in more detail in Chapter 3. It is based on the conference paper [TSSCV22] which is co-authored by the author of this thesis, together with Malik Tiomoko, Mohamed El Amine Seddik, Igor Colin and Aladin Virmaux.

I would like to express my gratitude to my PhD advisor Prof. Holger Rauhut for giving me the opportunity to do a PhD under his supervision in first place - at the intersection of two modern and fascinating topics of applied mathematics, Compressive Sensing and Machine Learning - and for providing both a lot of freedom as well as his expert guidance when needed, and support throughout the past years. Also, I want to thank Prof. Hartmut Führ for agreeing to be the second examiner.

Furthermore, I would like to acknowledge the exchange with and contributions from my colleagues and coauthors: Bella Naumova and Sebastian Lubjuhn (Aachen) and Dr. Igor Colin, Dr. Mohamed El Amine Seddik and Dr. Aladin Virmaux (Paris). Especially I had the immense pleasure to work closely together with Dr. Arash Behboodi (Aachen/Amsterdam) during the earlier phases of this work, and with Dr. Malik Tiomoko (Grenoble/Paris) in the final project.

I am grateful for interesting discussions with Vicky Kouni (Athens) and Dr. Cosme Louart (Beijing/Grenoble), and for proofreading and feedback on earlier versions of this thesis provided by Wiebke Bartolomaeus, Dr. Leonardo Galli and Laura Paul (Aachen) and Dr. Fateme Ghayem with Dr. Mostafa Sadeghi (Grenoble/Paris) that helped improving the readability.

My interest in some of the topics underlying this dissertation dates back to the time before joining RWTH Aachen: I was able to build up on my experience gained from my Master thesis under supervision of Prof. Jan Vybíral (Berlin/Prague) and my research internship in the group of Prof. Romain Couillet (Grenoble).

Finally, I would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG) by the project *Compressive Sensing via Neural Network Learning* (SCoSNeL, MA 1184/36-1) within the SPP 1798 *Compressed Sensing in Information Processing* (CoSIP) that made this research possible.

Dresden, May 1, 2023 Ekkehard Schnoor

1 Introduction

This first chapter serves as a light introduction for the main topics in this thesis. We will lay out the core ideas and recall some key concepts and results that are the foundation for the later chapters. Note that this introduction does not contain any novel ideas and results; rather, most can be considered common sense and folklore in the respective fields. The material presented here can be found in various monographs and textbooks; in particular and as main references, let us point the textbooks [MRT18; SSBD14a] on machine learning, and [FR13] for compressive sensing. A classical and elaborate reference for neural networks from the perspective of statistical learning theory is [AB99], which, however, does not take into account many of the more recent advances from the last few years. Other, related references include [Ver18] on high-dimensional probability with applications in data science, and [Wai19] on high-dimensional statistics from an non-asymptotic viewpoint. We will take both non-asymptotic *and* asymptotic viewpoints; the later being linked to asymptotic random matrix theory [CD11; Tao12]; of particular interest for us are its applications to machine learning [CL22]. In contrast to the spirit of this chapter, results that are of a more technical nature and that will be required for the proofs in Chapters 2 and 3, the main content of this thesis, but are not necessary for a general understanding of the main ideas, are provided (accompanied by proofs or literature references) in the appendices, Appendix A, Appendix B and Appendix C.

1.1 Motivation

To motivate the main content to follow later on in Chapter 2 and Chapter 3, let us begin by informally introducing the two main scenarios studied throughout this thesis, namely the problem of *sparse recovery*, and *sparse linear classifiers*. While both rely on the same underlying optimization problem and a specific algorithm to solve it, they still differ in the task to be solved and the tools used to derive mathematical performance guarantees.

1.1.1 Sparse Recovery

The area of *inverse problems* studies (in the finite-dimensional case) the recovery of signals $\boldsymbol{x} \in \mathbb{R}^p$ from measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. In the most simple noiseless case, we aim to solve problems of the type

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \tag{1.1}$$

i.e., solving linear systems of equations in a finite-dimensional setting. In this case, $\boldsymbol{A} \in \mathbb{R}^{m \times p}$ is the measurement matrix (whose m rows are interpreted as *measurements*, taken by inner products of the rows of $\boldsymbol{A}$ with $\boldsymbol{x}$) applied to the original signal $\boldsymbol{x}$ to obtain the measurement vector $\boldsymbol{y} \in \mathbb{R}^m$. Aiming for reconstruction from as few measurements as possible (*i.e.,* $\boldsymbol{A}$ has fewer rows than columns, $m \ll p$) leads to an under-determined linear system with an infinite number of solutions, if any. However, incorporating prior knowledge on $\boldsymbol{x}$ (e.g. based on modelling the class of vectors of interest, typically from some low-complexity distribution) changes the situation: Reconstruction may become possible, from surprisingly few measurements. Of particular interest are *sparse* signals, *i.e.*, vectors $\boldsymbol{x}$ containing only a few (say $s < p$) non-zero entries, when reconstruction is possible from as few as $O(s \log(p))$ measurements, much less than predicted by Shannons classical sampling theory [Sha48]. This turned out to be a useful model as many natural signals and data allow an (approximate) sparse representation with respect to a suitable basis or frame; in the past almost two decades, a whole branch of applied mathematics called *compressive sensing* (or *compressed sensing*, or *compressive sampling*, or *sparse recovery*) has been devoted to this study of inverse problems with sparsity constraints. It was initiated by Candès and Tao [CT05] and Donoho [Don06] in 2006, even though it has earlier origins in [DS89] and is related to the least absolute shrinkage and selection operator (LASSO) [Tib96]. While the first breakthroughs were achieved on the theoretical side, compressive sensing quickly entered applications such as medical imaging [LSLDP05], radar [BS07], and wireless communications [BHSN10; PAW07]. Let us very briefly recall some main ideas. Formally, $\boldsymbol{x} \in \mathbb{R}^p$ is called *s-sparse* (with $s \leq p$, and typically $s \ll p$), if

$$\|\boldsymbol{x}\|_0 := |\operatorname{supp}(\boldsymbol{x})| := \big|\{j : x_j \neq 0\}\big| \leq s. \tag{1.2}$$

Further, we denote the set of s-sparse vectors by $\Sigma_s = \{\boldsymbol{x} \in \mathbb{R}^p : \|\boldsymbol{x}\|_0 \leq s\}$. (Note that $\|\,.\,\|_0$ is not a norm; however, the notation may be justified by the observation of $\lim_{p \to 0} \|\boldsymbol{x}\|_p^p = \|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathbb{R}^p$.) Searching for a solution of (1.1) that is as sparse as possible, it is tempting to consider the optimization problem

$$\operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{x}\|_0 \qquad \text{subject to} \qquad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{P_0}$$

Unfortunately, it can be shown that the *ℓ_0-minimization problem* is NP-hard [Nat95]. (Note that a s-sparse vector $\boldsymbol{x} \in \mathbb{R}^p$ can have $\binom{p}{s}$ different support patterns, which becomes intractably large for sufficiently large values of p and s of practical interest.) However, it turned out that (P_0) is often equivalent to the following *ℓ_1-minimization problem*, its convex relaxation, the so-called *basis pursuit*, which was introduced in [CDS01], for which efficient reconstruction algorithms exist:

$$\operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{x}\|_1 \qquad \text{subject to} \qquad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{P_1}$$

passing from rom $\|\,.\,\|_0$ (see (1.2) above) to the ℓ_1-norm. Next we turn to sufficient conditions on the matrix $\boldsymbol{A}$, so that the problems (P_0) and (P_1) are equivalent. One such condition is the so-called *null space property* (NSP) that, informally stated, requires the elements of $\ker(\boldsymbol{A})$ to be "well-spread", in the sense that they are not supported on an index set of a small size [CDD09]. While it again suffers from the typical combinatorial problems due to the number of different support patterns (and thus is difficult to check on a given matrix), a sufficient condition for the NSP to hold is the so-called *restricted isometry property* (RIP), a fundamental concept in compressive sensing introduced in [CT05]. We say that $\boldsymbol{A} \in \mathbb{R}^{m \times p}$ has the RIP of order s (with $s \in [n]$), when there exists a so-called *RIP constant* $0 \leq \delta_s = \delta_s(\boldsymbol{A}) < 1$, such that

$$(1 - \delta)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1 + \delta)\|\boldsymbol{x}\|_2^2 \qquad \forall \boldsymbol{x} \in \Sigma_s. \tag{1.3}$$

Intuively, (1.3) means that the restriction of $\boldsymbol{A}$ to the set of s-sparse vectors acts as an approximate isometry, explaining the nomenclature. Again, it turns out that (deterministic) RIP matrices are difficult to construct (or, similarly, checking a given matrix whether it possesses the RIP). However, it can be shown that certain random matrices have the RIP (for sufficiently large s and small δ) with high probability. The prototype of a random measurement matrix is (an appropriately normalized) *Gaussian measurement matrix*,

$$\boldsymbol{A} \in \mathbb{R}^{m \times p} \qquad \text{with} \qquad A_{kl} \sim \mathcal{N}(0, 1/m^2) \qquad \text{i.i.d.}$$

By the concentration of measure phenomenon, $\|\boldsymbol{Ax}\|_2^2$ concentrates strongly around its mean $\mathbb{E}\|\boldsymbol{Ax}\|_2^2 = \|\boldsymbol{x}\|_2^2$ for any $\boldsymbol{x} \in \mathbb{R}^p$; thus, one may establish the restricted isometry property. There is a close connection to the classical Johnson-Lindenstrauss Lemma [Joh84] often applied for random projections of finite point clouds for linear dimensionality reduction (approximately distant-preserving embeddings similar as in the RIP): Random matrices that can be used in the Johnson-Lindenstrauss Lemma also have the RIP with high probability [BDDW08]; on the other hand, a given (deterministic) RIP matrix with randomized column signs (i.i.d. ±1 with probability $1/2$) is also a Johnson-Lindenstrauss embedding [KW11]. Note that besides the standard case of Gaussian measurement matrices, various other cases like *Bernoulli random matrices* or different *structured random matrices* have been studied in the literature (see [Rau] and the references therein).

Thus, with the use of appropriate random measurement matrices, we may consider the more accessible convex relaxation (P_1) of the intractable problem (P_0). Besides this *encoding* procedure (which refers to taking measurements to obtain a compression $\boldsymbol{y}$ of $\boldsymbol{x}$) just described, we are interested in efficient and robust *decoders*, *i.e.,* algorithms to solve (P_1). Such algorithms are typically based on techniques from convex optimization [Ber09] and also referred to as *reconstruction algorithms* as they aim to recover the original signal (or at least, a good approximation thereof) $\boldsymbol{x}$. We may also investigate their *robustness*, which refers to situations of noisy measurements, or when the signal of interest is not exactly sparse, but may be well-approximated by a sparse vector.

Let us also remark that it has become apparent that the encoding scheme of Gaussian measurements described above is *universal* in the sense that it works well for other data types beyond the classical assumption of sparsity. For instance, [BCDH10] considers model-based compressive sensing and [BW09] random projections of smooth manifolds, aiming for distance-preserving embeddings with respect to the geodesic distance. With the rise of deep learning, techniques from compressive sensing have also been experimentally applied to highly realistic data models such as generative models [BJPD17] and investigated from a theoretical viewpoint [HHHV21; HV18; HV19]. We will return to the problem of data models in Section 1.3.

From the various reconstruction algorithms, the iterative soft-thresholding algorithm [DDDM04] (short: ISTA) will be our focus within this thesis. We will introduce it in Section 1.2 and observe that it can be interpreted as a neural network. By introducing trainable parameters, it is possible to solve inverse problems in a data-driven (rather than model-based) fashion, which has become popular in the past decade in connection with the spectacular empirical success of machine learning with artificial neural networks. Later on, this approach will be investigated in Chapter 2. Before moving to the main content of this thesis, we recall some aspects of statistical learning theory in 1.2.

Finally, let us note that for simplicity we restrict ourselves to real signals and measurement matrices throughout this thesis, even though it is often straightforward, and for certain applications such as *synthetic aperture radar* (SAR), even desirable to extend the theory of compressive sensing to the complex case. On the other hand, neural networks with complex weights or processing complex-valued input data are, for the time being, much less established. Still, they could be desirable to be able to process certain data types that are naturally suited to be represented and processed using complex numbers. An adaption of the backpropagation algorithm to this case is given in [Nit97], recently, complex neural networks have been investigated from an approximation-theoretic viewpoint [CLMPV22]. For a comparison between equivalent architectures of complex-valued and real-valued networks see [BRMVO22] and for possible merits of complex-valued neural networks in applications for PolSAR image segmentation we refer to [BRMVO]. For recent surveys on the topic, see also [BQL21; LHG22] and the references therein. Extending the work from Chapter 2 to complex-valued signals and network parameters could be an interesting extension for future work, as well as practical applications of ISTA-inspired neural networks for complex data.

Extensions of sparse recovery are low-rank matrix recovery (we refer to the survey [DR16] and the references therein) and low-rank tensor recovery: [GQ14; HMGW14; RSS17; ZWZM19].

1.1.2 Sparse Classifiers

As a second scenario (and later the topic of Chapter 3), we revisit the following basic binary classification problem. Consider a simple linear classifier $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega} = \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle$, which, based on the sign of $g(\boldsymbol{x})$, assigns $\boldsymbol{x} \in \mathbb{R}^p$ to either class $\mathcal{C}_1$ or $\mathcal{C}_2$, through a separating hyperplane $\{\boldsymbol{y} \in \mathbb{R}^p : \boldsymbol{y}^\top \boldsymbol{\omega} = 0\}$. We assume that the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are associated to the labels ±1, which can be directly predicted by $h_{\boldsymbol{\omega}}(\boldsymbol{x}) = \text{sign}(g(\boldsymbol{x})) = \text{sign}(\boldsymbol{x}^\top \boldsymbol{\omega})$ for a given weight vector $\boldsymbol{\omega}$, where the sign function is given by[1]

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 1, & x \geq 0. \end{cases} \tag{1.4}$$

This is a fundamental example for a machine learning problem, which will be discussed more in general in the following Section 1.3. Let us here briefly and informally describe the situation we are interested in in the context of Chapter 3. Denoting the label of a given feature vector $\boldsymbol{x}$ by $y \in \{-1, 1\}$, we may train the model to satisfy

$$g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega} \approx y.$$

Given a whole training sequence $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ (collected as columns in the data matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$) and corresponding labels $y_1, \ldots, y_n \in \mathbb{R}$, collected in the label vector $\boldsymbol{y} \in \mathbb{R}^n$, we may pass to a compact matrix notation to obtain the desired condition

$$\boldsymbol{X}^\top \boldsymbol{\omega} \approx \boldsymbol{y}.$$

The discrepancy between the desired output (*i.e.*, the true label), and the output provided by a specific *hypothesis* $h_{\boldsymbol{\omega}}$ (characterized/parameterized by $\boldsymbol{\omega}$) is measured through a *loss function*. Of the many possibilities, let us first mention the simple least-square problem $(\boldsymbol{x}^\top \boldsymbol{\omega} - y)^2$, that is, across the whole training set, we may aim to minimize

$$\|\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y}\|_2^2.$$

(For convenience when calculating gradients, a factor 1/2 may be added.) In the case only a few features are responsible to distinguish between the two classes, and the majority of the features are not informative to the classifier, it makes sense to use a *sparse classifier*, *i.e.*, a weight vector $\boldsymbol{\omega}$ containing many zero entries. This has turned out to

[1] One may modify the rule of the classifier at the decision boundary $\{\boldsymbol{x} \in \mathbb{R}^p : g(\boldsymbol{x}) = 0\}$, but this is a technical detail and putting $\text{sign}(0) = 1$ is convenient for our purposes.

be useful in high-dimensional applications with comparably small sample sizes such as being encountered in bioinformatics [Con+17]. If chosen appropriately, the non-zero entries of $\boldsymbol{\omega}$ correspond to the features relevant to the classifiers decision, with their individual weights. Making use of the sparsity-promoting effect of the ℓ_1-norm, we add an ℓ_1-regularizer to the least squares problem to obtain the very well-known LASSO [Tib96]

$$\boldsymbol{\omega}^\star = \operatorname*{arg\,min}_{\boldsymbol{\omega}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{\omega}\|_2^2 + \lambda\|\boldsymbol{\omega}\|_1. \tag{1.5}$$

The regularized minimization problem is known to be equivalent to a corresponding constraint minimimization problem, *i.e.,* for an appropriate choice of $\lambda' > 0$ the solution of (1.5) coincides with the solution of

$$\boldsymbol{\omega}^\star = \operatorname*{arg\,min}_{\boldsymbol{\omega}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{\omega}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\omega}\|_1 \leq \lambda'.$$

Again, we will employ ISTA to (approximately) solve the above optimization problem and obtain a classifier $h_{\boldsymbol{\omega}^\star}$ given through $\boldsymbol{\omega}^\star$. Often we combine two loss functions in machine learning, as we may use one loss function for training and another one for the performance on the test dataset. For instance, after obtaining $h_{\boldsymbol{\omega}^\star}$ using ISTA, we may be interested in the *accuracy* (*i.e.,* probability of correct classification) of the classifier. Ideally, we would like to find guarantees to predict its performance.

This *generalization* perspective is common to both situations outlined in this section, firstly sparse recovery using ISTA-inspired neural networks, and secondly sparse linear classifiers obtained through ISTA. They are both based on the same well-known underlying mathematical problem, yet describe different situations, and will be tackled using different tools. Before recalling important concepts from statistical learning theory more rigourosly in Sections 1.3 and 1.4, let us finally introduce ISTA.

1.2 The Iterative Soft-Thresholding Algorithm

Note that in this section we mainly use the notation from the *sparse recovery problem* (measurement matrix $\boldsymbol{A}$ applied to a signal $\boldsymbol{x}$ to obtain a measurement vector $\boldsymbol{y}$) introduced in Section 1.1. The case of *sparse classifiers* with a data matrix $\boldsymbol{X}$, weight vector $\boldsymbol{\omega}$ and label vector $\boldsymbol{y}$ is analog. Later one, by using different notations we avoid confusion between the two models under consideration; the only "overlap" of $\boldsymbol{y}$ (denoting both the measurement vector and the label vector) will be not problematic as it will always be obvious which is being considered.

Regularized Least Squares Problems. One of the most fundamental and classical methods in statistics and numerical analysis, attributed to both Legendre and Gauß [Gau87] (who famously used it for his astronomical calculations) is the following least-square problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \tag{1.6}$$

assuming an approximate linear relation between $\boldsymbol{y}$ and $\boldsymbol{x}$ through $\boldsymbol{A}$. The functional appearing in (1.6) is differentiable and its gradient (with respect to $\boldsymbol{x}$) is given by

$$\nabla\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 = \boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}).$$

One is often interested in regularized variants of (1.6), *i.e.,* minimization problems of the type

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\mathcal{R}(\boldsymbol{x}),$$

with a regularizer $\mathcal{R}(\boldsymbol{x})$ and regularization parameter $\lambda > 0$ influencing the strength of the regularizer as part of the whole expression. Regularizers can be seen as a way to incorporate prior knowledge on $\boldsymbol{x}$; from a statistical learning perspective, they can also be interpreted as a form of (implicit) dataset augmentation to prevent overfitting (more details in Section 1.3). In the case $\mathcal{R}(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$, we obtain the so-called *ridge regression* problem

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^p} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2. \tag{1.7}$$

Here, an explicit solution of (1.7) exists and it is straightforward to derive the minimizer

$$\boldsymbol{x}^\star = (\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^\top\boldsymbol{y}. \tag{1.8}$$

We will return to the problem (1.7) later for illustrational purposes. However, with regard to the main content of this thesis we are mainly interested in the more challenging case of a sparsity-promoting ℓ_1-regularizer, that is

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^p} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{LASSO}$$

as already seen above in (1.5). In either case, for $\lambda = 0$ this boils down again to the ordinary regression problem (1.6), which is therefore also called *ridgless regression*. The problem given in (LASSO) is a convex, but non-smooth optimization problem. Thus, there exists no explicit solution, and standard gradient-descent methods are not directly applicable. However, using proximal gradient methods from convex optimization, one can derive the so-called iterative soft-thresholding algorithm (ISTA) as a practical method to solve (LASSO), which lies at the heart of this thesis. Even though we stick to the original setup explained above, let us point out that also variants of (LASSO) exist that employ a differentiable approximations of the ℓ_0-norm and the sign function [Sad+19].

Iterative Soft-Thresholding Algorithm. An actual algorithm for computing such minimizer is ISTA [DDDM04], where we initialize $\boldsymbol{x}^0 = \boldsymbol{0}$, and then recursively compute

$$\boldsymbol{x}^{k+1} = S_{\tau\lambda}\left(\boldsymbol{x}^k + \tau\boldsymbol{A}^\top(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}^k)\right), \tag{1.9}$$

where λ and τ are parameters of the algorithm, and S_λ (applied entrywise) is the *soft-thresholding* or *shrinkage operator* defined as

$$S_\lambda : \mathbb{R} \to \mathbb{R}, \qquad x \ \mapsto \begin{cases} 0, & \text{if } |x| \ \le \lambda, \\ x - \lambda\,\mathrm{sign}(x), & \text{if } |x| \ > \lambda, \end{cases}$$

or, equivalently, in closed form $S_\lambda(x) = \mathrm{sign}(x)\cdot\max(0,|x|-\lambda)$ for any $x \in \mathbb{R}$. It is well-known, see e.g. [DDDM04], that $\boldsymbol{x}^k$ converges to a minimizer of (LASSO) under the condition

$$\tau\|\boldsymbol{A}\|_{2\to 2}^2 < 2. \tag{1.11}$$

Let us add a few remarks that will be useful later. Firstly, simply by rearranging terms in (1.9) we can rewrite one ISTA iteration as follows [GL10]

$$\boldsymbol{x}^{k+1} = S_{\tau\lambda}\left((\boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{x}^k + \tau \boldsymbol{A}^\top \boldsymbol{y}\right).$$

Thus, we observe that one ISTA iteration takes the form of a layer of a neural network with weight matrix $\boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A}$, bias $\tau \boldsymbol{A}^\top \boldsymbol{y}$ and activation function $S_{\tau\lambda}$ applied entrywise. Note that for now it only takes the form of a neural network; in Chapter 2 we will formulate it more rigorously as a machine learning problem, introducing a hypothesis class through trainable parameters. Also a fixed point equation viewpoint will be useful. As this will show up in Chapter 3, let us use the notation from there and note that, if ISTA converges, its limit $\boldsymbol{\omega}^\star$ satisfies the fixed point equation

$$\boldsymbol{\omega}^\star = S_{\lambda\tau}\left((\boldsymbol{I} - \tau \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}^\star + \tau \boldsymbol{X}\boldsymbol{y}\right).$$

Due to the randomness in the data matrix $\boldsymbol{X}$, this leads to a random fixed point equation of the type $\boldsymbol{\omega} = \Phi(\boldsymbol{X})(\boldsymbol{\omega})$, with $\Phi(\boldsymbol{X})$ being a random function taking $\boldsymbol{\omega}$ as an input. Such situations have been studied in detail in [LC20].

Let us briefly review other reconstruction algorithms and alternatives for ISTA, even though they will play no role for the remainder of this thesis. A closely related algorithm is the so-called hard-thresholding algorithm [BD09], which however has the disadvantage of being non-continuous. Another variant is to select the, say s largest entries after the gradient descent step and set all remaining entries to zero: this is problematic (for instance, in the context of neural networks) as a comparison of all entries is required, instead of a straightforard entrywise application of the activation function. Another algorithm for sparse reconstruction is orthogonal matching pursuit [Zha11]. ISTA is of particular interest to us due to the possible formulations as neural network and fixed point equations seen above. Finally, let us remark that we collect a few properties of the soft-thresholding function and ISTA in Appendix C.

1.3 Statistical Learning Theory

In this section, we first recall some basic concepts such as hypothesis classes, data models and distributions, and loss function in machine learning, and then we discuss the topic of *generalization* in statistical learning theory, that plays a central role in this thesis. For textbooks providing thorough introductions to such topics, we refer to [CL22; Jun22; MRT18; SSBD14b].

Firstly, let us note that here we focus exclusively on *supervised learning*, which aims at learning a function $f : \mathcal{X} \to \mathcal{Y}$ based on *labeled* training data. That is, we have access to *data points*, or *feature vectors*, or *samples* $\boldsymbol{x}_i \in \mathcal{X}$ and labels y_i (if scalar-valued; or $\boldsymbol{y}_i$ in case of vector-valued) from some set $\mathcal{Y}$, being collected in a training set $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$. Throughout this thesis, we will only encounter the case of Euclidean data, *i.e.,* $\mathcal{X} \subset \mathbb{R}^p$; for a binary classification problem we have labels $y_i \in \{-1, 1\}$, and $\mathcal{Y} \subset \mathbb{R}^m$ in case of a regression problem. Typically, the $(\boldsymbol{x}_i, y_i)$ are assumed to be i.i.d. samples from some joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Thus, $f : \mathcal{X} \to \mathcal{Y}$ is a function from features to labels, that shall satisfy $f(\boldsymbol{x}_i) \approx y_i$. We consider such functions that are elements of some *hypothesis class* $\mathcal{H}$, and for any element $h \in \mathcal{H}$ of this hypothesis class we use a *loss function* to measure the discrepancy between $\hat{y} = h(\boldsymbol{x})$ and y

$$\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to [0, \infty), \qquad (h, \boldsymbol{x}, \boldsymbol{y}) \mapsto \ell(h, \boldsymbol{x}, \boldsymbol{y}). \tag{1.12}$$

Ideally, we would like to find a hypothesis $h \in \mathcal{H}$ that minimizes the *true risk*, *i.e.,* the expectation of the loss with respect to the data distribution $\mathcal{D}$,

$$\mathcal{L}(h) = \mathbb{E}_{x,y \sim \mathcal{D}}(\ell(h, \boldsymbol{x}, \boldsymbol{y})). \tag{1.13}$$

For real applications, the distribution $\mathcal{D}$ is typically not known or cannot be fully described. Furthermore, minimizing the expression in (1.13) may be challenging or even intractable. Therefore, a hypothesis $h_{\mathcal{S}} \in \mathcal{H}$ is learned in practice from sampled training data $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, typically through minimizing

$$\hat{\mathcal{L}}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, \boldsymbol{x}_i, \boldsymbol{y}_i), \tag{1.14}$$

i.e., applying the well-known principle of *empirical risk minimization* (short: ERM) for the *training* of our model, which is given through the hypothesis class $\mathcal{H}$. If no explicit solution is available, usually (a variant of) the gradient descent method is employed to solve the optimization problem.

However, ultimately, we are not interested in a low training error only, but we would like the hypothesis to *generalize well*, *i.e.,* to perform well (or, more precisely: to perform *similarly* well) on the true distribution. This way, we want to prevent the common problem of *overfitting*, which describes the situation of the learned hypothesis performing well on the training data (*i.e.,* $f(\boldsymbol{x}_i) \approx y_i$), but not being able to make good predictions on new samples from the same distribution, but not belonging to the training set. While again the true distribution is typically not known, we may instead use a *test* or *validation set* for comparison, which was not used for training. This motivates to introduce the so-called generalization error (GE(h_A)) of $h_A \in \mathcal{H}$ (learned through some algorithm A, e.g. empirical risk minimization), defined as the gap between the true and empirical error,

$$\mathrm{GE}(h_A) = |\hat{\mathcal{L}}(h_A) - \mathcal{L}(h_A)|. \tag{1.15}$$

Ideally, we would like to find mathematical guarantees to ensure a certain performance of our machine learning method of interest, and conditions that enable us to estimate or upper bound the generalization error (1.15). It turns out such conditions usually strongly depend on the ambient dimension p and the sample size n, and are connected with the notion of *sample complexity* that denotes the minimal number of training samples required to successfully learn a target function.

After this informal and rather brief overview, we will discuss more details of the above, and review some examples of interest for the remainder of this work, in the rest of this section. We conclude with a brief summary, that also serves as an overview for the more detailed coverage in Chapters 2 and 3.

1.3.1 Hypothesis Classes

From the large amount of hypothesis classes considered in machine learning, we will mainly consider models from the opposites extremes in terms of complexity. On the one hand, we will deal both with simple linear hypothesis classes and on the other hand, we will investigate particular instances of neural networks, very rich hypothesis classes containing highly non-linear functions.

Linear Hypothesis Classes. Let us consider the basic linear model $\boldsymbol{\omega}^\top \boldsymbol{x} = \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle$, *i.e.,* the inner product of a *weight vector* $\boldsymbol{\omega} \in \mathbb{R}^p$ with some *feature vector* $\boldsymbol{x} \in \mathbb{R}^p$, which can be used for either regression or classification purposes. Depending on the combination with different loss functions, this may result in different machine learning models such as logistic regression or support vector machines (SVMs).

Geometrically, this operation divides the p-dimensional space, through the *linearly separating hyperplane* $\{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{x} = 0\}$, into two distinct areas where either $\boldsymbol{\omega}^\top \boldsymbol{x} < 0$ or $\boldsymbol{\omega}^\top \boldsymbol{x} > 0$. Given linear separability in a binary classification problem, our predictor takes the form

$$h_\omega(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}),$$

with the so-called *decision boundary* corresponding to the separating hyperplane, *i.e.,* the region $\{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{x} = 0\}$. Formally, without any further assumptions on $\boldsymbol{\omega}$, our hypothesis class in this linear model is the set of functions given by

$$\mathcal{H} := \left\{ h : \mathbb{R}^p \to \{-1, 1\} \,|\, h_\omega(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}),\, \boldsymbol{\omega} \in \mathbb{R}^p \right\}.$$

However, typically we assume that the models parameter are contained in a bounded set. This is usually needed to derive theoretical guarantees; also this restriction is natural anyways regarding the practical applicability (e.g. hardware constraints etc.). The easiest choice would be to incorporate an ℓ_2-norm constraint on the parameter vector $\boldsymbol{\omega}$, that is

$$\mathcal{H}_2 := \left\{ h : \mathbb{R}^p \to \{-1, 1\} \,|\, h_\omega(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}),\, \boldsymbol{\omega} \in \mathbb{R}^p,\, \|\boldsymbol{\omega}\|_2 \le B_2 \right\}.$$

For sparse linear classifiers, we can again make use of the sparsity-promoting effect of an ℓ_1-norm constraint on $\boldsymbol{\omega}$ (also note Lemma 1.5 below). Therefore, let us define

$$\mathcal{H}_1 := \left\{ h : \mathbb{R}^p \to \{-1, 1\} \,|\, h_\omega(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}),\, \boldsymbol{\omega} \in \mathbb{R}^p,\, \|\boldsymbol{\omega}\|_1 \le B_1 \right\}.$$

Later on in this introduction, we will review different combinations of such simple models with loss functions, as well as discuss their generalization behavior. A particular instance of this, namely the LASSO-based classification using (1.5) will be the topic of Chapter 3.

Note that linear models are fairly simple and not expressive enough for many applications. Therefore, they are often used in combination with other methods as a part of a larger machine learning pipeline. For instance, one may first apply a non-linear feature map to increase the separability of classes, followed by a simple linear classifier. An important example of this are neural networks, which often contain a simple linear classifier in their last layer that may be trained together in an *end-to-end* fashion with the non-linear feature map (all previous layers). Let us briefly discuss neural networks in the next section.

Artificial Neural Networks. Artificial neural networks form a class of functions that are loosely inspired by biological neurons and can be traced back to [Ros58]. However, their practical breakthrough, first in computer vision applications, was possible only due to the increased availability of sufficiently large datasets as well as advances in hardware (graphics processing units). In their most simple form, one *layer* of a (fully-connected feedforward) neural network,

$$f(\boldsymbol{x}) = \rho(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}), \tag{1.16}$$

is an affine linear function $x \mapsto \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ with a *weight matrix* $\boldsymbol{W}$ and a *bias* $\boldsymbol{b}$ of appropriate sizes, followed by a non-linear activation function $\rho : \mathbb{R} \to \mathbb{R}$ applied entrywise. A (deep) neural network is a concatenation of several (many) layers of the form (1.16), resulting in function classes of very strong expressive power. In its basic form, the collection of all weight matrices and biases form the set of trainable parameters. Note that this describes classical feed-forward neural networks; many particular or related models (including convolutional neural networks, recurrent neural networks, or residual neural networks etc.) exist. They can be applied to classical machine learning tasks such as classification and regression, but also for various other applications like dimensionality reduction or representation learning (e.g. through so-called *autoencoders* that map the original data onto some latent space, as introduced in [HS06] and further developements such as *variational autoencoders* [KW13] that aim to estimate the parameters of the latent space distributions, and *sparse autoencoders* [MF13; Ng+11]). Moreover, data generation [Goo+20] (which will also be discussed in the next subsection) or playing games (with a first series of breakthroughs in Chess, Go and Shogi [Sil+16; Sil+17a; Sil+17b; Sil+18] inspiring many subsequent efforts) where often they represent the state of the art.

Despite the empirical success, many questions are still open from a theoretical point of view. Of particular interest for us is the generalization of (often deep and overparameterized) neural networks, which behaves completely differently than traditional techniques. Later, we will study this problem at the hand of a particular instance of neural networks, namely the ones that are inspired by the iterative soft-thresholding algorithm (see Section 1.2). This will be discussed in detail in Chapter 2, after introducing the necessary background on the topic of generalization in the next section.

There is a vast and fast-growing amount of literature on deep learning, often focusing on numerical experiments. For a recent monograph describing the current mathematical understanding of this topic, we refer to [GK22] and the references therein.

1.3.2 Data Models and Distributions

In this section, we briefly discuss different data models and distributions that are encountered in machine learning, and that are relevant to the remainder of the thesis. For instance, we focus exclusively on feature vectors or signals that can be interpreted as vectors in $\mathbb{R}^p$.

Modeling a data distribution, or estimating certain parameters of the distribution underlying some given samples, is a key problem in machine learning and statistics. Often, distributions of datasets of practical interest are not accessible. Nevertheless, many useful models have been developed and we recall a few of them below that are of interest for the upcoming Chapters 2 and 3. From a generalization perspective (more details in the following section), let us note that generalization error bounds based on the Rademacher complexity (see Definition 1.6 below) or the VC dimension (see Definition 1.2 below) are *distribution-free* (or at least, can be applied when having access to only finitely many samples rather than the full distribution). This situation is encountered in Chapter 2, where the hypothesis class of interest is adapted to be suitable for sparse reconstruction tasks, but the derived generalization bounds are *data-independent* and hold for any arbitrary distribution (and thus, may be somewhat pessimistic). In Chapter 3 we will take a different approach, where our goal will be to predict the generalization in terms of few scalar quantities that are derived from the training data. For now, let us recall the following data models.

Sparsity has attracted considerable attention in the past almost two decades with the rise of compressive sensing and related topics. Examples include dictionary learning (see [Ela10] and the references therein) and methods from applied harmonic analysis (such as wavelets [Mal99], and an abundance of further, often multiscale representation systems that have been developed). To illustrate, we say some (possibly dense, *i.e.,* having nonzero entries) signal $\boldsymbol{x} \in \mathbb{R}^p$ is sparse with respect to some dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{p \times p}$, if $\boldsymbol{x} = \boldsymbol{\Phi z}$ for $\boldsymbol{z} \in \mathbb{R}^p$ being sparse, *i.e.,* $\|\boldsymbol{z}\|_0$ small. Note that sparsity turned out useful not only in the context of machine learning, but in general for many applications in signal and image processing, in particular for compression purposes (such as the JPEG compression standard for compression of digital image files). Sparsity plays an important role in this thesis as well: We will encounter it both as a data model in the sparse reconstruction problem in Chapter 2 and as a model of the parameters in the case of a sparse linear classifier in Chapter 3.

Gaussian Mixture Model. The *Gaussian mixture model* (GMM) is a classical example for a data model, which may be used in either supervised or unsupervised settings. We focus on a two-class Gaussian mixture model; the generalization to the multi-class case is straightforward. Let $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{C}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{C}_2)$ be two multivariate normal distributions with characteristic means $\boldsymbol{\mu}_l, \boldsymbol{C}_l, l = 1, 2$ corresponding to the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$. In the GMM, we draw with probability $p_1 \in (0, 1)$ from class $\mathcal{C}_1$, *i.e.,* $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{C}_1)$, and with probability $p_2 = 1 - p_2$, we sample from class $\mathcal{C}_2$, *i.e.,* $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{C}_2)$. If both classes are *balanced*, $p_1 = p_2 = 1/2$. While not being a realistic model for complex real-life datasets, it is still a useful toy model as we will also see in the next section.

Generative Models. Many real-life datasets (such as natural images) have been known to be very difficult to model and are often thought of as a distribution over a submanifold in a high-dimensional space that eludes a precise mathematical description. *Generative adversarial networks (GANs)* [Goo+20] have changed this situation, by allowing to generate synthetic data that strongly resembles real datasets, like certain image classes. Based on concepts from two-player-games originating in game theory, the idea is to train two connected neural networks that try to outperform each other: Firstly, a so-called *generator* that generates samples from a random input, aiming for highly realistic outputs that, secondly, the *discriminator* tries to distinguish from real datasets (thus, classifies between *real* and *fake* images). In the last few years, generative adversarial networks have received tremendous attention and can nowadays create strikingly realistic artificial samples. This research activity has led to an unmanageable amount of literature available; as a recent survey article, let us mention [GSWTY21] and the references therein. Generative models as a signal prior, rather than the traditional assumption of sparsity, have then also been employed in compressive sensing, firstly showing highly favorable behavior in experiments [BJPD17] followed by efforts from a theoretical perspective [HHHV21; HV18; HV19]. Mathematically speaking, such *generative models* are Lipschitz continuous transformations of Gaussian random vectors, but remarkably, in some sense, behave *as if they were Gaussian* [SLTC20]. This motivates us to consider the class of random vectors obeying a certain concentration property, by being both a realistic model and at the same time mathematically more accessible.

Concentrated Random Vectors. Let us recall the *Gaussian concentration inequality* [Led01; LT11], stating that for a Gaussian random vector $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ in $\mathbb{R}^p$ and any L-Lipschitz function $f : \mathbb{R}^p \to \mathbb{R}$ (with Lipschitz continuity with respect to the ℓ_2-norm), there is

$$\mathbb{P}\left(|f(\boldsymbol{x}) - \mathbb{E}f(\boldsymbol{x})| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2L\|\boldsymbol{\Sigma}\|_{2\to 2}}\right). \tag{1.17}$$

This is an instance of the so-called *concentration of measure phenomenon*, where *Lipschitz observations* $f(\boldsymbol{x})$ of $\boldsymbol{x}$ concentrate strongly around their mean $\mathbb{E}f(\boldsymbol{x})$. Let us also point out that the bound on the right hand side of the inequality is independent from p (up to a possible implicit dependence of p due to $\|\boldsymbol{\Sigma}\|_{2\to 2}$; however, for instance when $\boldsymbol{\Sigma} = \boldsymbol{I}_p$ the expression would be entirely independent from p). We are more generally interested in distributions that behave similar to (1.17).

Definition 1.1 (q-exponential concentration; observable diameter) Let $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ be a normed vector space and $q > 0$. A random vector $\boldsymbol{x} \in \mathcal{X}$ is said to be q-exponentially concentrated if for any 1-Lipschitz continuous (with respect to $\|\cdot\|_{\mathcal{X}}$) real-valued function $\varphi : \mathcal{X} \to \mathbb{R}$ there exists $C \geq 0$ independent of $\dim(\mathcal{X})$ and $\sigma > 0$ such that, for all $t \geq 0$,

$$\mathbb{P}\left(|\varphi(\boldsymbol{x}) - \mathbb{E}\varphi(\boldsymbol{x})| \geq t\right) \leq Ce^{-(t/\sigma)^q}. \tag{1.18}$$

This is denoted as $\boldsymbol{x} \propto \mathcal{E}_q(\sigma \,|\, \mathcal{X}, \|\cdot\|_{\mathcal{X}})$, where σ is called the *observable diameter*. If σ does not depend on $\dim(\mathcal{X})$, we write $\boldsymbol{x} \propto \mathcal{E}_q(1 \,|\, \mathcal{X}, \|\cdot\|_{\mathcal{X}})$.

Let us point out again that the observable diameter may depend on $\dim(\mathcal{X})$, unlike the absolute constant $C \geq 0$. Even more so, it can turn out to be very interesting to study this dimensionality-dependence of the observable diameter σ: on the one hand, when σ decreases with increasing $\dim(\mathcal{X})$, there is a *concentration of measure* effect in the sense that the random vector behaves close to deterministic in large dimensions (consider $\mathcal{X} = \mathbb{R}^p$ and φ to be the coordinate projections). On the other hand, when (1.18) holds only for a relatively large or increasing (with the dimension) observable diameter σ, the concentration of $\varphi(\boldsymbol{x})$ becomes *weaker*, *i.e.*, it may fluctuate more around its mean $\mathbb{E}\varphi(\boldsymbol{x})$. In this sense, the observable diameter can be interpreted as a *degree of concentration of* $\boldsymbol{x}$ *under scalar Lipschitz observations* with respect to the dimension.

Furthermore, let us remark that the mean in (1.18) may be replaced by the median, possibly with slight adjustments. The concentration property in Definition 1.1 is slightly more general than the *convex concentration property* considered in [Ada05; Ada15; KR17; Led01; MS12; VW15] where convex (rather than Lipschitz continuous) functions are considered in the special case where $q = 2$. Let us discuss a few examples and properties of concentrated random vectors, that are either fundamental or of interest in the context of this thesis, in particular in view of Chapter 3. For more examples, see [KR17, p. 10].

1. Returning to the Gaussian concentration inequality (1.17) above, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ is 2-exponentially concentrated, *i.e.*, $\boldsymbol{x} \propto \mathcal{E}_2(\sigma \,|\, \mathbb{R}^p, \|\cdot\|_2)$ with $\sigma = \sqrt{2L\|\boldsymbol{\Sigma}\|_{2\to 2}}$. For $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ (*i.e.*, the covariance being the identity), σ is ensured to be dimension-independent so that, using the convention introduced in Definition 1.1, we then write $\boldsymbol{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^p, \|\cdot\|_2)$.

2. We denote the spherical distribution, or uniform distribution over the sphere in $\mathbb{R}^p$ with radius $\sqrt{p}$ by $\boldsymbol{x} \sim \operatorname{Unif}\left(\sqrt{p}S^{p-1}\right)$. (Rigorously, $\boldsymbol{x}$ is uniformly distributed on the sphere $\sqrt{p}S^{p-1}$ if for every Borel subset $\mathcal{E} \subset \sqrt{p}S^{p-1}$ the ratio of the $(p-1)$-dimensional volumes of $\mathcal{E}$ and $\sqrt{p}S^{p-1}$ equals the probability of the event $\boldsymbol{x} \in \mathcal{E}$.) By the concentration of Lipschitz functions on the sphere (see, for instance, [Ver18, Theorem 5.1.4]), we have $\boldsymbol{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^p, \|\cdot\|_2)$ as for Gaussian random vectors.

3. An important property of the concentration property described in (1.18) is its *stability under Lipschitz transforms*. More precisely, if $f : \mathcal{Z} \to \mathcal{X}$ (for $(\mathcal{Z}, \|\,.\,\|_{\mathcal{Z}})$ being another normed space) is a K-Lipschitz function and $\boldsymbol{x} \in \mathcal{X}$ with $\boldsymbol{x} \propto \mathcal{E}_q(\sigma \,|\, \mathcal{X}, \|\cdot\|_{\mathcal{X}})$ then $f(\boldsymbol{x}) \propto \mathcal{E}_q(K \,|\, \mathcal{Z}, \|\cdot\|_{\mathcal{Z}})$ where K might depend on $\dim(\mathcal{X})$. As neural networks represent Lipschitz continuous mappings, generative adversarial networks are Lipschitz continuous transforms of (isotropic) Gaussian random vectors and can therefore be modeled using concentration properties such as (1.18) [SLTC20].

1.3.3 Loss functions

Let us return to the concept of loss functions already introduced in (1.12). Here, we discuss some examples of loss functions that are relevant in the context of this work.

Mean Squared Error. A typical choice of the loss function, mainly for regression tasks (including the sparse reconstruction task), is

$$\ell_{\text{MSE}} = \|h(\boldsymbol{x}) - \boldsymbol{y}\|_2^2, \tag{1.19}$$

giving rise to the so-called *mean squared error* (MSE). However, deviations from this are possible as well. In Chapter 2, we simply measure the reconstruction error with respect to the (non-squared) ℓ_2-norm, that is $\ell = \|h(\boldsymbol{x}) - \boldsymbol{y}\|_2$, which has the advantage of being Lipschitz continuous (even on unbounded domains). Also, even though possibly somewhat unusual, we will use the MSE (or variants of it, such as with added regularization) also for classification tasks, using labels $y \in \{-1, 1\}$ instead of vectors $\boldsymbol{y} \in \mathbb{R}^d$. We will discuss other possible choices for the loss function in the following paragraphs.

Accuracy, or 0/1-loss Function. For the linear binary classifier $h_\omega(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ with (true) label $y \in \{-1, 1\}$ and prediction $\hat{y} = h_\omega(\boldsymbol{x})$, we may consider the *0/1-loss function* defined as follows by

$$\ell_{0/1}\left((\boldsymbol{x}, y), h_\omega\right) = \begin{cases} 0, & \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}) = y, \\ 1, & \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}) \neq y. \end{cases}$$

Here, the loss equals zero, if the prediction was correct, and one, if the prediction was not correct. When taking the empirical error on a sufficiently large training set, this corresponds to the percentage of misclassified points in the training set; in expectation, this can be interpreted as the *probability of misclassification*.

Logistic Regression. A classical example for a linear machine learning model is *logistic regression* and the corresponding *logistic loss*, which assumes the following probabilistic model on the data. Given some data point $\boldsymbol{x} \in \mathbb{R}^p$ and model parameters $\boldsymbol{\omega} \in \mathbb{R}^p$, the (conditional) probabilities of the corresponding label $y \in \{-1, 1\}$ are assumed to be given by (depending on $\boldsymbol{\omega}^\top \boldsymbol{x}$)

$$\mathbb{P}_\omega(y = 1 | x) = \frac{1}{1 + \exp(-\boldsymbol{\omega}^\top \boldsymbol{x})},$$

for the label $y = 1$, and consequently for $y = -1$, using the relation $\mathbb{P}_\omega(y = -1 | x) = 1 - \mathbb{P}_\omega(y = 1 | x)$,

$$\mathbb{P}_\omega(y = -1 | \boldsymbol{x}) = \frac{1}{1 + \exp(\boldsymbol{\omega}^\top \boldsymbol{x})}.$$

Given i.i.d. realizations $\mathcal{S} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ of a joint distribution $\mathcal{D}$ of features $\boldsymbol{x}_i$ and corresponding labels y_i (assumed to obey the probabilistic model above), the *maximum-likelihood estimator* $\boldsymbol{\omega}^\star$ for the parameter $\boldsymbol{\omega}$ is given by (through independence)

$$\begin{aligned}\boldsymbol{\omega}^\star :&= \operatorname*{arg\,max}_{\boldsymbol{\omega} \in \mathbb{R}^p} \mathbb{P}(y_1|\boldsymbol{x}_1, \ldots, y_n|\boldsymbol{x}_n) = \operatorname*{arg\,max}_{\boldsymbol{\omega} \in \mathbb{R}^p} \prod_{i=1}^n \mathbb{P}(y_i|\boldsymbol{x}_i) \\ &= \operatorname*{arg\,max}_{\boldsymbol{\omega} \in \mathbb{R}^p} \prod_{i=1}^n \frac{1}{1 + \exp(-y_i \boldsymbol{\omega}^\top \boldsymbol{x})}.\end{aligned}$$

Maximizing this positive function is equivalent to maximizing its logarithm, *i.e.,*

$$\begin{aligned}\boldsymbol{\omega}^\star &= \operatorname*{arg\,max}_{\boldsymbol{\omega} \in \mathbb{R}^p} \sum_{i=1}^n - \log\left(1 + \exp(-y_i \boldsymbol{\omega}^\top \boldsymbol{x})\right) \\ &= \operatorname*{arg\,min}_{\boldsymbol{\omega} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \log\left(1 + \exp(-y_i \boldsymbol{\omega}^\top \boldsymbol{x})\right).\end{aligned}$$

We have rewritten this maximum-likelihood estimator as the *empirical risk minimizer* of the so-called *logistic loss function*, which is given by

$$\ell_{\text{logloss}}(\boldsymbol{x}, y, h_\omega) := \log(1 + \exp(-y h_\omega(\boldsymbol{x}))).$$

Note that ℓ_{logloss} is non-negative, convex and differentiable with respect to the parameters contained in the weight vector $\boldsymbol{\omega}$ and can thus be solved by using gradient descent. One may also consider an ℓ_1-regularized version of the logistic loss, *i.e.,*

$$\log(1 + \exp(-y h_\omega(\boldsymbol{x}))) + \lambda \|\boldsymbol{\omega}\|_1,$$

which however is not differentiable anymore.

Support Vector Machines (SVMs) are another class of linear classifiers based on the model $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}$ (often used after applying a non-linear feature map first to increase the linear separability of classes). Building upon the *hinge loss* given by

$$\ell(\boldsymbol{x}, y, \boldsymbol{\omega}) = \max(0, 1 - y \cdot \boldsymbol{x}^\top \boldsymbol{\omega}),$$

they classically add a ℓ_2-regularization term to obtain

$$\ell(\boldsymbol{x}, y, \boldsymbol{\omega}) = \max(0, 1 - y \cdot \boldsymbol{x}^\top \boldsymbol{\omega}) + \lambda \|\boldsymbol{\omega}\|_2.$$

Again, one may replace the ℓ_2-regularizer with a ℓ_1-regularizer, promoting a sparse classifier that performs a *feature selection*, when only few features are expected to be relevant to discriminate the classes. This has been proposed in [BM98; ZRTH03] and was latter analysed in [KV17]. For an exhaustive treatment of support vector machines, we refer to the monography [SC08] and the references therein.

1.4 Generalization: Asymptotic and Non-Asymptotic Approaches

In this section, we want to take a closer look at the topic of *generalization*, which plays a key role in statistical learning theory, and also in this thesis. In (1.15) earlier in this section, we have already introduced the generalization error. Here, we consider different approaches on how to derive mathematical performance guarantees for machine learning models, which typically depend on some notion of *complexity* or *richness* of the underlying hypothesis space (to be defined more precisely), and crucial parameters like the *sample size* (available training data) and the *feature dimension*. This will be made more precise in this section, where we then recall generalization error bounds (based on the Rademacher complexity and VC dimension in the non-asymptotic case), as well as asymptotic approaches that aim to derive the asymptotically precise performance. We will discuss the following cases.

1. **Finite p, finite n:** Finite sample complexity bounds for a finite-dimensional setting. This is well-established in the machine learning literature (e.g. uniform convergence through Rademacher complexity/VC dimension bounds).

2. **Finite p, $n \to \infty$:** Known through many classical results (e.g. central limit theorem, law of large numbers etc.); however, low-dimensional intuition behind machine learning methods (e.g. kernels) may collapse in a high-dimensional setting [CBG16].

3. **Large p, large n:** Models a scenario of large-dimensional statistics where both the dimension and the sample size are large, through the limit $n, p \to \infty$ with $p/n \to c \in (0, \infty)$, with tools based on (asymptotic) random matrix theory (RMT).

We discuss advantages and disadvantages of the different approaches. Thus, we are well prepared for the upcoming chapters, where Chapter 2 corresponds to the first cases, and Chapter 3 corresponds to the third of the three cases.

1.4.1 Finite n, finite p: Uniform Convergence through non-asymptotic Bounds

VC Dimension. The *Vapnik-Chervonenkis dimension* (or short: *VC dimension*) is possibly the most classical notion to quantitatively measure what is meant by the idea of the *complexity* or *richness* of a hypothesis class [VC15]. It is defined as follows.

Definition 1.2 (Vapnik-Chervonenkis dimension) The VC dimension $\mathrm{VCdim}(\mathcal{H})$ of $\mathcal{H}$ is the size n of the largest set of samples that can be *shattered* by $\mathcal{H}$, *i.e.,* for any of the 2^n possible assignments of binary labels, there exists an hypothesis $h \in \mathcal{H}$ that realizes this assignment of labels.

As the definition already suggests, the VC dimension is mainly suitable for classification tasks. Recalling the simple example of linear binary classifiers allocating $\boldsymbol{x}$ to either class $\boldsymbol{x} \to \mathcal{C}_1$ or $\boldsymbol{x} \to \mathcal{C}_2$ based on the test (for some $\eta \in \mathbb{R}$)

$$g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x} \underset{\mathcal{C}_1}{\overset{\mathcal{C}_2}{\gtrless}} \eta,$$

geometrically dividing the space $\mathbb{R}^p$ in two half spaces via the separating hyperplane $\{\boldsymbol{y} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{y} = \eta\}$. As can be shown easily, the VC dimension of this hypothesis class of hyperplane separators in $\mathbb{R}^p$ is $p + 1$; as a reference, see Example 3.12 and Theorem

3.13 in [MRT18]. (In this case, the VC dimension is close to the vector space dimension of $\mathbb{R}^p$; note, however, that the VC dimension may be in stark contrast to the vector space dimension, if the hypothesis class has a vector space structure at all.) Generally, given an estimate of the VC dimension $\mathrm{VCdim}(\mathcal{H})$ of the hypothesis class $\mathcal{H}$ of interest, we obtain upper bounds for the generalization error as follows.

Theorem 1.3 (Generalization via the VC Dimension - Binary Classification) *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$ with VC dimension $\mathrm{VCdim}(\mathcal{H}) = d$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in \mathcal{H}$ (with sample size n):*

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{1/\delta}{2n}}.$$

For a reference and proof, see [MRT18, Corollary 3.19]. Note that from the inequality given in the theorem it is straightforward to obtain bounds on the generalization error as defined in (1.15). Results of this type are called *uniform*, explaining also the notion of *uniform convergence*, as the bounds are valid for all $h \in \mathcal{H}$ (in particular, for the empirical risk minimizer). For the linear classifier, ignoring constants and logarithmic terms we roughly have

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + \mathcal{O}\left(\sqrt{\frac{p}{n}}\right),$$

which crucially depends on the ratio of the dimension p and sample size n. Theorem 1.3, and similar theorems below relying on the Rademacher complexity instead of the VC dimension, can be applied as follows in a practical setting.

1. Train a model on a training set of size n to obtain some hypothesis $h \in \mathcal{H}$!
2. Calculate the empirical error $\hat{\mathcal{L}}(h)$ (track during training).
3. Given VC dimension of $\mathcal{H}$ (and desired probability δ), calculate an upper bound for the true (or test) error $\mathcal{L}(h)$.
4. "Rule of thumb" to obtain meaningful bounds: $n \sim 100 \cdot p$.

Let us also remark that VC dimension bounds for neural networks have been studied in [AB99]; however, it is difficult to obtain sharp VC dimension bounds that are able to explain the generalization behaviour observed in practice [NK19; ZBHRV17].

Rademacher Complexity. As another complexity measure for hypothesis classes, which is suitable both for classification and regression tasks, we discuss the so-called *Rademacher complexity* [BM02]. It will play an important role in Chapter 2. Let us begin by considering the Rademacher complexity of general subsets $\mathcal{A}$ of $\mathbb{R}^n$ before moving to Rademacher complexities of hypothesis classes in the context of statistical learning theory.

Definition 1.4 (Rademacher complexity of sets $\mathcal{A} \subset \mathbb{R}^n$) For a set $\mathcal{A} \subset \mathbb{R}^n$ with elements $\boldsymbol{a} = (a_1, \dots, a_n) \in \mathcal{A}$, its Rademacher complexity $\mathcal{R}(\mathcal{A})$ is defined as follows by

$$\mathcal{R}(\mathcal{A}) = \mathbb{E}_{\varepsilon} \sup_{\boldsymbol{a} \in \mathcal{A}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i a_i,$$

where $\boldsymbol{\varepsilon}$ is a Rademacher vector, *i.e.*, a vector of independent Rademacher variables ε_i, $i = 1, \dots, n$, taking the values ± 1 with equal probability.

Lemma 1.5 *The Rademacher complexity of the convex hull of $\mathcal{A}$ equals the Rademacher complexity of the set $\mathcal{A}$ itself,* i.e., $\mathcal{R}(\mathcal{A}) = \mathcal{R}(\mathrm{conv}(\mathcal{A}))$.

Proof. See Lemma 26.7 in [SSBD14b]. ∎

We can easily extend Definition 1.4 to the case of hypothesis classes, whose elements are evaluated at finitely many points (data points in the context of machine learning).

Definition 1.6 (Rademacher complexity of function classes) For a class $\mathcal{G}$ of real-valued functions g, and samples drawn from $\boldsymbol{x}_i \sim \mathcal{D}$ (where $\mathcal{D}$ is a distribution on the joint domain of the functions $g \in \mathcal{G}$), the empirical Rademacher complexity is defined as

$$\mathcal{R}_{\mathcal{S}}(\mathcal{G}) = \mathbb{E}_{\varepsilon} \sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i g(\boldsymbol{x}_i), \tag{1.20}$$

where again the expectation is taken with respect to a Rademacher vector $(\varepsilon_1, \dots, \varepsilon_n)$. The (true) Rademacher complexity is then given by $\mathcal{R}_n(\mathcal{G}) = \mathbb{E}_{\mathcal{S} \sim \mathcal{D}^n} \mathcal{R}_{\mathcal{S}}(\mathcal{G})$.

Intuitively, the Rademacher complexity measures the capability of the function class $\mathcal{G}$ to "match" random labels $\varepsilon_i \in \{-1, 1\}$. While this reasoning is based on a classification problem, the Rademacher complexity is also applicable to regression problems. Let us also already refer to Appendix A containing more details on suprema of stochastic processes (including technical details with regard to measurability), of which (1.20) is a special instance, and introducing *Dudley's integral*, an important tool to find sharp upper bounds for Rademacher complexities.

Let us illustrate the definition of a Rademacher complexity by employing a rather simple example of linear hypothesis classes. We will revisit and discuss a different approach to this later on in Chapter 3. Instead, in Chapter 2 we will estimate the Rademacher complexities of much more complicated hypothesis classes.

Lemma 1.7 *Consider the hypothesis class of linear functionals from $\mathbb{R}^p$ to $\mathbb{R}$ with ℓ_2-norm constraint,* i.e., $\mathcal{H} := \{g : \mathbb{R}^p \to \mathbb{R} \,|\, g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x},\ \boldsymbol{\omega} \in \mathbb{R}^p,\ \|\boldsymbol{\omega}\|_2 \leq B_2\}$ *and samples $\mathcal{S} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ with $\boldsymbol{x}_i \in \mathbb{R}^p$. We obtain the following upper bound for the Rademacher complexity $\mathcal{R}_{\mathcal{S}}(\mathcal{H})$*

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}) \leq B_2 \sqrt{\frac{\max_i \|\boldsymbol{x}_i\|_2^2}{n}}.$$

Despite being fairly simple, let us give the easy proof for illustrational purposes. It can be found in various textbooks on statistical learning theory, such as [MRT18; SSBD14b].

Proof. Firstly, by the Cauchy-Schwartz inequality and the boundedness of $\boldsymbol{\omega}$, we obtain

$$\begin{aligned} n\mathcal{R}_{\mathcal{S}}(\mathcal{H}) =& \mathbb{E} \sup_{g \in \mathcal{H}_2} \sum_{i=1}^{n} \varepsilon_i g(\boldsymbol{x}_i) = \mathbb{E} \sup_{\|\boldsymbol{\omega}\|_2 \leq B_2} \left\langle \boldsymbol{\omega}, \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\rangle \leq \mathbb{E} \sup_{\|\boldsymbol{\omega}\|_2 \leq B_2} \|\boldsymbol{\omega}\|_2 \left\| \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\|_2 \\ =& B_2 \cdot \mathbb{E} \left\| \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\|_2 . \end{aligned}$$

Next, by Jensens inequality for the concave square root function, we get

$$\mathbb{E} \left\| \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\|_2 = \mathbb{E} \left(\left\| \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\|_2^2 \right)^{1/2} \leq \left(\mathbb{E} \left\| \sum_{i=1}^{n} \varepsilon_i \boldsymbol{x}_i \right\|_2^2 \right)^{1/2} .$$

By a basic rearrangement of terms and the independence of the Rademacher variables ε_i,

$$\begin{aligned}\mathbb{E}\left\|\sum_{i=1}^{n}\varepsilon_i \boldsymbol{x}_i\right\|_2^2 =&\mathbb{E}\left\langle \sum_{i=1}^{n}\varepsilon_i \boldsymbol{x}_i, \sum_{j=1}^{n}\varepsilon_j \boldsymbol{x}_j\right\rangle = \sum_{i\neq j}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \mathbb{E}[\varepsilon_i\varepsilon_j] + \sum_{i=1}^{n}\langle \boldsymbol{x}_i, \boldsymbol{x}_i\rangle \mathbb{E}[\varepsilon_i^2] \\ =& \sum_{i=1}^{n}\|\boldsymbol{x}_i\|_2^2 \leq n \cdot \max_{i=1,\dots,n}\|\boldsymbol{x}_i\|_2^2.\end{aligned}$$

Finally, combining our findings yields the bound on the Rademacher complexity $\mathcal{R}_{\mathcal{S}}(\mathcal{H})$,

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}) \leq \frac{B_2}{n}\cdot\sqrt{n}\cdot\sqrt{\max_{i=1,\dots,n}\|\boldsymbol{x}_i\|_2^2} = B_2\sqrt{\frac{\max_i \|\boldsymbol{x}_i\|_2^2}{n}},$$

finishing the proof of the lemma. ■

An analog to Theorem 1.3, relying on the Rademacher complexity instead of the VC dimension, is the following result [MRT18, Theorem 3.5] that provides generalization error bounds in a binary classification setting.

Theorem 1.8 (Generalization via the Rademacher Complexity - Binary Classification) *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, 1\}$ and let $\mathcal{D}$ be the distribution over the input space $\mathcal{X}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample $\mathcal{S}$ of size n drawn according to $\mathcal{D}$, each of the following holds for any $h \in \mathcal{H}$:*

$$\begin{aligned}\mathcal{L}(h) &\leq \hat{\mathcal{L}}(h) + \mathcal{R}_{\mathcal{S}}(\mathcal{H}) + \frac{\log(1/\delta)}{2n}, \\ \mathcal{L}(h) &\leq \hat{\mathcal{L}}(h) + \mathcal{R}(\mathcal{H}) + 3\frac{\log(2/\delta)}{2n}.\end{aligned}$$

For regression problems such as the reconstruction problem, which will be the topic of Chapter 2, we rely on the following result. As for the previous result, a key ingredient for its proof is McDiarmid's inequality [McD].

Theorem 1.9 (Generalization via the Rademacher Complexity - Regression) *Let $\mathcal{H}$ be a family of functions, $\mathcal{S}$ the training set drawn from $\mathcal{D}^n$, and ℓ a real-valued bounded loss function satisfying $|\ell(h, z)| \leq c$ for all $h \in \mathcal{H}, z \in Z$. Then, for $\delta \in (0, 1)$, with probability at least $1-\delta$ we have, for all $h \in \mathcal{S}$,*

$$\begin{aligned}\mathcal{L}(h) &\leq \hat{\mathcal{L}}(h) + 2\mathcal{R}(\ell \circ \mathcal{H}) + c\sqrt{\frac{2\log(2/\delta)}{n}}, \\ \mathcal{L}(h) &\leq \hat{\mathcal{L}}(h) + 2\mathcal{R}_{\mathcal{S}}(\ell \circ \mathcal{H}) + 4c\sqrt{\frac{2\log(4/\delta)}{n}}.\end{aligned}$$

In Chapter 2, we will derive generalization error bounds based on Theorem 1.9. Note that the Rademacher complexity terms in Theorem 1.9 are Rademacher complexities of the function class that results from the concatenation of the hypothesis class $\mathcal{H}$ and the loss function ℓ. In such cases (and assuming that $\mathcal{H}$ consists of real-valued functions), when the loss function is K-Lipschitz continuous we may apply Talagrands contraction principle (see Theorem B.3) to obtain

$$\mathcal{R}_{\mathcal{S}}(\ell \circ \mathcal{H}) \leq K \cdot \mathcal{R}_{\mathcal{S}}(\mathcal{H}),$$

so that in the end the problem boils down to finding Rademacher complexities of the hypothesis class itself anyways. In the case of vector-valued function classes (such as high-dimensional regression problems, a situation also encountered in Chapter 2 in this thesis), the Rademacher complexity as in Definition 1.6 is not well-defined. To still be able to make use of Theorem 1.9 nevertheless, we will employ a generalized contraction principle due to Maurer which will be discussed in the Appendix B in Theorem B.4.

1.4.2 Finite $p, n \to \infty$: Classical Statistics

This situation of a finite, fixed dimension p and a sample size n tending to infinity is ubiquitous in many classical results from probability theory and statistics. As a prominent example, let us recall the classical *central limit theorem* (CLT), the deeper reason why the normal distribution is ubiquitous in many phenomena.

Theorem 1.10 (Central Limit Theorem) *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of i.i.d. real-valued random variables with mean $\mu = \mathbb{E}[X_i] < \infty$ and variance $\sigma^2 = \mathrm{Var}(X_i) < \infty$ with $\sigma^2 \neq 0$. Then, the sequence $(Z_n)_{n\in\mathbb{N}}$ of random variables given by*

$$Z_n = \frac{X_1 + \cdots + X_n - n\mu}{\sqrt{n}\sigma}$$

converges in distribution towards the standard normal distribution $\mathcal{N}(0,1)$. In particular (see also (B.4) *in the appendix), for any $a, b \in \mathbb{R}$ with $a < b$, as $n \to \infty$,*

$$\mathbb{P}\left(Z_n \in [a,b]\right) \to \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2}\,\mathrm{d}x.$$

We will make use of the Central Limit Theorem later on in Chapter 3. Let us here also introduce the following variant of the central limit theorem, showing that inner products with concentrated random vectors behave approximately like Gaussians. It will be important in Chapter 3 to justify that the classification scores $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ of linear classifiers behave like Gaussians in high dimensions. It is originally due to [FGP07; Kla07]; here, we follow the version and notation provided [SLCT21, Theorem 3.2]. (Recall Definition 1.1 and the comment thereafter for the notion of concentrated random vectors, and the notation $\boldsymbol{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^p, \|\cdot\|_2)$ is explained.)

Theorem 1.11 (CLT for Concentrated Random Vectors) *Let $\boldsymbol{x} \in \mathbb{R}^p$ be a random vector with $\mathbb{E}[\boldsymbol{x}] = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{I}_p$. Further, let η be the uniform measure on the sphere $\mathcal{S}^{p-1} \subset \mathbb{R}^p$ of radius* 1. *Then, if $\boldsymbol{x}$ follows the concentration $\boldsymbol{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^p, \|\cdot\|_2)$, there exist two constants $C, c > 0$ and a set $\Omega \subset \mathcal{S}^{p-1}$ such that $\eta(\Omega) \geq 1 - C\sqrt{p}e^{-c\sqrt{p}}$ and*

$$\sup_{t\in\mathbb{R}} \left| \mathbb{P}(\boldsymbol{\omega}^\top \boldsymbol{x} \geq t) - F_{0,1}(t) \right| \leq p^{-1/4} \qquad \forall \boldsymbol{\omega} \in \Omega,$$

where $F_{0,1}$ is the cumulative distribution function of the standard normal distribution $\mathcal{N}(0,1)$; see equation (B.1) *in the appendix.*

If $\boldsymbol{x}$ is a general (possibly non-centered and non-isotropic) Gaussian random vector, an analog result can be obtained through appropriate shifting and scaling (similar for $\boldsymbol{\omega}$ not being normalized).

Let us again consider a linear classifier, but in its ℓ_2-regularized version (1.7) with the corresponding solution (1.8). Note that in general, in the limit $n \to \infty$, the solution provided in (1.8) does not need to converge. Thus, it may be required to normalize appropriately to ensure convergence when passing to the asymptotics $n \to \infty$. Usually, the data is normalized by the factor of $1/\sqrt{n}$, which is appropriate for typical assumptions[1] on $\boldsymbol{X}$; see also [CL20]. Thus, we may consider

$$\underset{\boldsymbol{\omega}\in\mathbb{R}^p}{\arg\min} \left\| \frac{1}{\sqrt{n}} \boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y} \right\|_2^2 + \lambda \|\boldsymbol{\omega}\|_2^2 \tag{1.21}$$

and from (1.7) we immediately obtain the minimizer of (1.21) to be

$$\boldsymbol{\omega}_n^\star = \left(\frac{1}{n} \boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_p \right)^{-1} \frac{1}{\sqrt{n}} \boldsymbol{X}\boldsymbol{y}, \tag{1.22}$$

which, depending on the distribution over $\boldsymbol{X}$, may converge with $\lim_{n\to\infty} \boldsymbol{\omega}_n^\star = \boldsymbol{\omega}^\star \in \mathbb{R}^p$. Then, the (asymptotic) classification score $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}^\star$ is given by

$$\frac{1}{\sqrt{n}} \boldsymbol{x}^\top \boldsymbol{\omega}^\star = \frac{1}{n} \boldsymbol{x}^\top \left(\frac{1}{n} \boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_p \right)^{-1} \boldsymbol{X}\boldsymbol{y}. \tag{1.23}$$

The expression $(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_p)^{-1}$ appearing in (1.23) is called the *resolvent* of the *sample covariance matrix* (assuming zero mean) $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top$. The resolvent is very difficult to analyse in case of large n alone, even under a Gaussian mixture model (with few attempts such as [TB20] available in the literature). However, it is a classical object being studied in (asymptotic) random matrix theory, when also $p \to \infty$, which is naturally linked to the setting when both n and p are large. We discuss this in the next section.

1.4.3 Large *n*, large *p*: Large-dimensional Statistics

As we have seen in the previous section, it may be difficult to derive results when $n \to \infty$ alone. Furthermore, this approach may also be insufficient to model correctly situations where the dimension p is similarly large (or even larger, in extreme cases) as the sample size n. While many machine learning algorithms are designed based on low-dimensional intuitions, their behavior in a large-dimensional setting may be very different. As an illustration, let us refer to the large n, large p investigation [CBG16] of kernel spectral clustering [NJW01; SM00], revealing insights into its inner workings that are very different to the original reasoning [VL07].

This approach of the double asymptotics is thus naturally linked to non-asymptotic random matrix theory, which studies spectral properties of random matrices when $n, p \to \infty$, typically at commensurable rate $p/n \to c \in (0, \infty)$. Here, in contrast to a non-asymptotic setting, there are some differences: For instance, pointwise convergence does *not* imply convergence with respect to some matrix norms; furthermore, equivalence of norms no longer holds.

Note this is different from the use of random measurement matrices in compressive sensing, where one is usually interested in non-asymptotic bounds. Nevertheless, some asymptotic approaches in compressive sensing exist as well, such as an asymptotic analysis of the RIP [CEG15].

After early works due to Wishart [Wis28] and Wigners *semi-circle law* [Wig55] (inspired

[1] For an isotropic Gaussian random vector $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ there is $\mathbb{E}\|\boldsymbol{x}\|_2 = \sqrt{p}$, with a strong concentration around the mean. (Thus, the large dimensional Gaussian behaves very differently than our low-dimensional intuition would suggest, and in high dimension it is similar to the spherical distribution.)

by applications in physics), the field greatly advanced with the work of Marčenko and Pastur [MP67] (see Theorem 1.12 below). While originally mainly intended for applications in statistics (especially covariance estimation) and physics, later on it became an important tool for applications in wireless communications (see [CD11], and the references therein) and machine learning [CL22].

Now let us turn to the classical work [MP67], providing the limiting ($n, p \to \infty$) spectral behavior (*i.e.,* the convergence of the *discrete* distribution of eigenvalues converges to a *continuous spectral density*) of the sample covariance matrix

$$\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top \in \mathbb{R}^{p\times p}$$

of some data matrix $\boldsymbol{X} \in \mathbb{R}^{p\times n}$. (Note that more precisely, this is a sequence of random matrices indexed by n and p, with $p/n \to c \in (0,\infty)$. Typically in the literature, the index is omitted for convenience.)

Theorem 1.12 (Marčenko–Pastur) *Let $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ be a random matrix with i.i.d. mean-zero, unit variance entries. Then, the spectral density of $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top$ in the limit of $n, p \to \infty$ with $p/n \to c \in (0,\infty)$ is given by*

$$\mu(dx) = (1-c^{-1})^+\delta_0(x) + \frac{1}{2\pi c x}\sqrt{(x-E_-)^+(E_+-x)^+}\mathrm{d}x,$$

where $E_\pm = (1\pm\sqrt{c})^2$ and $(x)^+ = \max\{0,x\}$ and δ_0 is a Dirac measure.

Remark 1.13 Note that the Dirac measure δ_0 in zero takes (isolated) zero eigenvalues into account for the rank-deficient case (when $p > n$, so that $c > 1$). If $\boldsymbol{X}$ is *not* rank-deficient (*i.e.,* when $n > p$, *i.e.,* $c \in (0,1)$), then $\mu(dx)$ has compact support $[E_-, E_+]$ and the probability of eigenvalues lying in $[a,b] \subset [E_-, E_+]$ is given by

$$\int_a^b \frac{1}{2\pi c x}\sqrt{(x-E_-)^+(E_+-x)^+}\,\mathrm{d}x,$$

with the density given by (and integration with respect to) the Lebesgue measure. ◊

Remark 1.14 Let us remark that different versions of the above theorem exist. Originally (and as stated above), i.i.d. mean-zero and unit variance have been considered. However, this makes the original results unsuitable for applications in statistical learning theory, where the typical assumption of having i.i.d. samples $\boldsymbol{x}_i$ is already a strong assumption, but certainly an i.i.d. assumption on the entries of each feature vector vector realistic data types (such as natural signals and images) would be inappropriate. However, many results such as the version of the Marčenko–Pastur law as stated above in Theorem 1.12 have been generalized to the case of having i.i.d. columns (or rows, respectively), thus enabling applications in statistical learning theory. Of particular importance is the case of concentrated random vectors introduced above, which is both a more realistic data model [SLTC20] and is compatible with random matrix theory [EK09]. ◊

Let us return to the problem of ridge regression (1.22), considered previously in the large n alone setting. Note that when in (1.22) passing to the limit of both $n, p \to \infty$, it is unclear how to describe convergence of (1.22) in $\mathbb{R}^p$ when $p \to \infty$, even provided knowledge of Theorem 1.12 and thus its resolvent $(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_p)^{-1}$. Furthermore, in more difficult situations, the limiting eigenvalue distribution of the involved expressions may not be accessible at all. However, it turns out that for applications in statistical learning theory, a full description of asymptotic eigenvalue distributions is not strictly required, as the generalization behavior of some algorithms can be described through knowing a (few) scalar quantities, whose asymptotic behavior is more likely to be tractable. Namely, we will use the important idea that often quantities such as regression errors, or misclassification rates, may be described as functionals of random matrices, whose (scalar-valued) asymptotic behavior may be tracked through so-called *deterministic equivalents* [HLN07]. In this example, let us note that (1.23) is a quadratic form of the resolvent $(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_p)^{-1}$ of $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top$, and deterministic equivalents have been derived in [LC20]. In general, we may say that (the sequence; omitting the index n) $\bar{\boldsymbol{Q}} \in \mathbb{R}^{n\times n}$ is a deterministic equivalent for the symmetric random matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ (again, more precisely: in the limit $n \to \infty$) if, for (sequences of) deterministic vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ of unit (Euclidean) norm

$$\boldsymbol{a}^\top(\boldsymbol{Q} - \bar{\boldsymbol{Q}})\boldsymbol{b} \to 0,$$

as $n \to \infty$, with convergence in probability or almost surely. Thus, the idea is to find deterministic objects that asymptotically behave, under operations like quadratic forms, similar to the random object of interest. Note that we may also consider different situations than quadratic forms: In Chapter 3, we will also encounter functionals involving the trace and then consider deterministic equivalents with respect to this operation. More details on deterministic equivalents can be found in B.

While classically being used for topics such as covariance estimation for decades, only in recent years this approach has been adapted to the analysis of more elaborate machine learning algorithms. This section could only provide a glimpse into this area of research; for a systematic treatment of this topic, we refer to the recent monograph [CL22]. Note that the main challenges of this approach lie in dealing with dependencies and the non-linear nature of many advanced machine learning methods, such as neural networks. Besides, generally speaking, such techniques prefer *dense over sparse expressions* (see also the discussion at the end of [CL22, Section 2.6.2]). Furthermore, this approach works better with a simple training procedure (for instance, for a convex loss surface with available convergence guarantees, or more generally when the minimizer of the loss function can be expressed implicitly as the solution of a fixed-point equation). It may become intractable in case of highly complicated models, such as the highly non-linear functions represented by deep neural networks, with an inaccessible training procedure (non-convex loss surface, stochastic gradient descent). Nevertheless, some works on (shallow, typically having only one hidden layer) neural networks exist, and use *full-batch* training rather than stochastic gradient descent [ASS20; LC18a; LLC18; SMG13].

However, when it is possible to deal with the technical challenges posed, often a good agreement between theoretical (asymptotic) predictions and empirical (non-asymptotical) observations can be found. This is also due to the speed of convergence at the rate of $\mathcal{O}(\sqrt{pn})$ in the double limit compared to $\mathcal{O}(\sqrt{n})$ in the central limit theorem (single asymptotics when $n \to \infty$ only).

Overview

We conclude this introduction, and prepare the upcoming two chapters, with the following overview.

Chapter 2 - Uniform Convergence

ISTA (introduce trainable parameters):

$$\boldsymbol{x}^{k+1} = S_{\tau\lambda}\left((\boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{x}^k + \tau \boldsymbol{A}^\top \boldsymbol{y}\right),$$

- Measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times p}$,
- (sparse) signal $\boldsymbol{x} \in \mathbb{R}^p$,
- measurement vector $\boldsymbol{y} \in \mathbb{R}^m$.

Generalization of trained decoder (error measured by the ℓ_2-norm):

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + \mathcal{O}\left(\frac{\mathcal{R}(\mathcal{H})}{\sqrt{n}}\right) + \ldots$$

- *Finite* dimension p and sample size n.
- Estimate the Rademacher complexity $\mathcal{R}(\mathcal{H})$ (or other complexity measure).
- Algorithm-independent: holds for all $h \in \mathcal{H}$ (in particular ERM).
- Distribution-free (no explicit use any assumptions on the data distribution, such as sparsity).
- **Advantage:** concrete bound for any given values of p and n.
- **Disdvantage:** Only upper bound for generalization error!

Chapter 3 - Asymptotic Approach

ISTA (fixed point formulation):

$$\boldsymbol{\omega}^\star = S_{\lambda\tau}\left(\boldsymbol{\omega}^\star + \tau \boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}^\star)\right).$$

- Data matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$,
- label vector $\boldsymbol{y} \in \{-1, 1\}^n$,
- (sparse) linear classifier $\boldsymbol{\omega}^\star \in \mathbb{R}^p$.

Generalization of the linear classifier obtained (using accuracy/ 0/1-loss):

$$\text{Compute} \quad \mathbb{E}\mathcal{L}(h^\star)$$

- $n, p \to \infty$ with $p/n \to c \in (0, 1)$.
- Data distribution induces distribution (and concentration!) over $h^\star$.
- Relying on (simple) algorithm to compute $h^\star \in \mathcal{H}$.
- Employing a concentration of measure framework; estimation of first and second order moments.
- **Advantage:** Precise performance guarantees (asymptotically).
- **Disdvantage:** Intractable for highly complicated models (NNs).

2 Unfolded Neural Networks for Sparse Reconstruction

In this chapter, we will study neural networks that arise from unfolding the iterative soft-thresholding algorithm. This chapter is based on [BRS22; SBR21] where our main contribution is to provide a novel analysis of the generalization error of a general class of neural networks inspired by ISTA. Thus, a main goal of this chapter is to connect the areas of inverse problems and statistical learning theory, whereas so far research on generalization of neural networks has strongly focused on classification problems. The chapter is structured as follows. After the introductury Section 2.1, Section 2.2 studies the problem of generalization of ISTA-inspired networks at the hand of a comparatively simple example, namely that of learning an orthogonal dictionary, being included in ISTA as the trainable parameters. By avoiding the presentation to become overly technical, this allows us to focus on the main methods of the proof. This will be greatly generalized in Section 2.3 to a much larger class of ISTA-inspired neural networks, that take a flexible choice of parameters into account, and even allow trainable thresholds and stepsizes, which may also differ from layer to layer. Furthermore, this general scenario contains both recurrent neural networks and ones more similar to feedforward neural networks, and in particular covers the dictionary learning problem studied previously as a special case. The proof relies on classical bounds of the generalization error via estimates of the Rademacher complexity of the hypothesis class; however, an important ingredient is a generalized contraction principle for vector-valued hypothesis classes. In Section 2.4 we present results of numerical experiments and compare with our theoretical findings from the previous sections. In Section 2.5 we discuss various related topics, extensions, and open questions.

Before delving into this chapter, let us remark that the notation used here differs from the one in the original papers, in order to be consistent with the rest of this thesis. Namely, we adapt the notation from statistics where p (not N) is the ambient dimension and n denotes the sample size (rather than m, which is here reserved for the number of measurements, as common in the compressive sensing literature anyways).

2.1 Introduction

In Chapter 1, we have introduced the iterative soft-thresholding algorithm. Let us recall that ISTA, for fixed stepsize $\tau > 0$ and fixed threshold $\lambda > 0$, consists in first computing

$$f_1(\boldsymbol{y}) = S_{\tau\lambda}(\tau \boldsymbol{A}^\top \boldsymbol{y}),$$

and then, iteratively for $l \leq 2$ up to a certain number $l = L$ of iterations,

$$\begin{aligned} f_l(\boldsymbol{z}) =& S_{\tau\lambda}\left[\boldsymbol{z} + \tau \boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A})\boldsymbol{z})\right] \\ =& S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A}\right)\boldsymbol{z} + \tau \boldsymbol{A}^\top \boldsymbol{y}\right]. \end{aligned}$$

(Note that for $l > 1$, all f_l coincide as functions on $\mathbb{R}^p$.) Let us point out an observation that is fundamental to this chapter, namely that one iteration of ISTA can be interpreted as a layer of a neural network with weight matrix $\boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A}$, bias $\tau \boldsymbol{A}^\top \boldsymbol{y}$ and non-linear activation function $S_{\tau\lambda}$ applied elementwise. In this context, the index l refers to the layer number of the neural network. Note that the neural networks studied here in some sense resemble autoencoders, classes of neural networks with the purpose to learn a lower-dimensional or structured representation of the data: This is typically achieved by training a neural network to reconstruct its input, but enforcing dimensionality reduction (and thus, avoiding trivial solutions like an identity map) through shrinking the dimensionality into a small central layer of latent variables. This concept was originally proposed in [HS06]; various variants of this approach have been developed, most notably maybe *variational autoencoders* [KW13]. Interesting in the context of this thesis are also *sparse autoencoders* [MF13; Ng+11], that aim to obtain a sparse representation of the data, possibly of *larger* dimension.

Here, taking the measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} =: \mathrm{enc}_{\boldsymbol{A}}(\boldsymbol{x})$ may be interpreted as *encoding* the signal $\boldsymbol{x}$ into $\boldsymbol{y}$, corresponding to a shallow, one-layer linear neural network (which is deterministic, when the measurement matrix $\boldsymbol{A}$ is considered to be fixed), the *decoder* is based on the unfolded version of the iterative soft thresholding algorithm (ISTA) with L iterations as follows. However, note that in the current form ISTA only takes the form of a neural network (in this context, also called *unfolded neural network*). Introducing parameters that are optimized with respect to some available training data, this leads to the notion of learned iterative soft-thresholding algorithms (LISTA). This has been observed for the first time in [GL10], and this combination of inverse problems and deep learning is interesting for various reasons:

- Firstly, a fundamental difference compared to traditional approaches is that it works in a data-driven manner. Thus, rather than using prior assumptions on the data of interest, the training may help the algorithm adapt to a specific data distribution. For instance, instead of plain sparsity additional structure (certain support patterns, correlations between entries etc.) may arise in applications, to which the flexible models like neural networks may easily adapt.

- Related to the previous point, one may hope for (approximate) reconstruction via trained decoders to be possible, at least in some cases, from (even) fewer measurements than predicted by classical compressive sensing (which itself already revolutionized sampling theory by massively improving classical bounds by Shannon [Sha48]).

- At *test time* (or *inference*), that is when applying the trained model to new data, the numerical evaluation of the neural network may be faster than traditional algorithms.

The recent years have witnessed considerable research activity at the intersection of deep learning and inverse problems. This chapter provides, to the best of our knowledge, a first detailed theoretical investigation of the generalization error of neural networks inspired by ISTA. By introducing trainable parameters $\mathcal{P}$ of the unfolded network, we would like to learn parameters that are suitable to perform a certain task, from a training sequence $\mathcal{S} = ((\boldsymbol{x}_i, \boldsymbol{y}_i))_{i=1,\dots,n}$ with i.i.d. samples drawn from an (unknown) distribution $\mathcal{D}$. Mainly, we consider the task of sparse reconstruction - Section 2.2 focusses exclusively on this, while the more flexible setup in Section 2.3 covers general regression tasks, including reconstruction. Formally, $\mathcal{D}$ is a distribution over the $\boldsymbol{x}_i$, and then the corresponding measurements $\boldsymbol{y}_i$ are given by $\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i$, with $\boldsymbol{A}$ being fixed.

While the algorithms studied are mostly suitable for sparse reconstruction tasks (but, apart from that, also general regression tasks), throughout our derivation we make *no explicit assumptions on the signals $\boldsymbol{x}$ of interest except that we presume that the signals $\boldsymbol{x}$ in the class are bounded by a certain value*, say B_{in}, in the ℓ_2-norm. Furthermore, for technical reasons that will become apparent later on, we will also introduce functions σ to be applied after the final layer to bound its output.

For a more rigorous formulation as a statistical learning problem, we will formally introduce a hypothesis class $\mathcal{H}$ (parameterized by the respective parameters $\mathcal{P}$) and a loss function. The hypothesis set essentially consists of all functions that can be expressed as L-step soft-thresholding with parameters from $\mathcal{P}$, and based on the training samples $\mathcal{S}$ and given the hypothesis space $\mathcal{H}$ (with a technical modification after the final layer which will be introduced below), a learning algorithm yields a function $h \in \mathcal{H}$ that aims at reconstructing $\boldsymbol{x}$ from the measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

Different choices for the loss function ℓ to measure the quality of the reconstruction $\hat{\boldsymbol{x}}_i = h(\boldsymbol{y}_i)$ compared to the original signal $\boldsymbol{x}_i$ are possible. A popular choice for regression problems is the mean squared error (1.19). Instead, throughout this chapter we use the the loss function

$$\ell(h, \boldsymbol{x}, \boldsymbol{y}) = \|h(\boldsymbol{y}) - \boldsymbol{x}\|_2, \tag{2.1}$$

which (in contrast to the squared norm, leading to the mean-squared-error) has the advantage of being 1-Lipschitz continuous, even on unbounded domains. For the notions of *empirical and true risk* and the definition of the *generalization error*, which are central objects of interests in this chapter, we refer to Section 1.3 in Chapter 1 (see equations (1.13), (1.14) and (1.15)).

Finally, let us also comment on the *measurement design*. Throughout the thesis, we will assume a fixed measurement matrix for theoretical investigations and an appropriately normalized Gaussian random matrix for the numerical experiments. Thus, the theoretical investigations focus on the reconstruction task. Let us note, however, that the approach described here can, in principle, also be applied to training the measurement matrix, either independently of the training the measurement matrix, or simultaneously in an end-to-end fashion. A rigorous theoretical investigation of the combined problem is highly challenging and remains an opportunity for future research. From an experimental viewpoint, such scenarios that include training the measurement matrix (to satisfy RIP-like conditions to be suitable for reconstruction tasks) have been considered previously in [WRL19; Wu+19].

2.2 LISTA for Dictionary Learning

In this section, we will derive generalization error bounds for a specific model learning an orthogonal dictionary suitable for reconstruction. By focusing on such a relatively simple example, we avoid an overly technical presentation and can focus on the developing the required proof methods. This section is based on the book chapter [BRS22], which is co-authored by the author of this thesis. Later on, we will adapt the proof to a much more general setup.

2.2.1 Dictionary Learning Model

To introduce trainable parameters, one may consider the following scenario. Namely, let us be given a class of signals $\boldsymbol{x} \in \mathbb{R}^p$ which are not necessarily sparse themselves, but sparsely representable with respect to a dictionary $\boldsymbol{\Phi}_0 \in \mathbb{R}^{p\times p}$. In other words, for each $\boldsymbol{x}$ there is a sparse vector $\boldsymbol{z} \in \mathbb{R}^p$ such that $\boldsymbol{x} = \boldsymbol{\Phi}_0\boldsymbol{z}$. The dictionary $\boldsymbol{\Phi}_0$ is assumed to be unknown. For a fixed stepsize $\tau > 0$, and a fixed $\lambda > 0$, the first layer is defined by $f_1(\boldsymbol{y}) = S_{\tau\lambda}(\tau(\boldsymbol{A\Phi})^\top\boldsymbol{y})$. For the iteration (or layer, respectively) $l > 1$, the output is given by

$$\begin{aligned} f_l(\boldsymbol{z}) =&S_{\tau\lambda}\left[\boldsymbol{z} + \tau(\boldsymbol{A\Phi})^\top(\boldsymbol{y} - (\boldsymbol{A\Phi})\boldsymbol{z})\right] \\ =&S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\boldsymbol{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A\Phi}\right)\boldsymbol{z} + \tau(\boldsymbol{A\Phi})^\top\boldsymbol{y}\right], \end{aligned} \tag{2.2}$$

which again can be interpreted as a layer of a neural network with weight matrix $\boldsymbol{I} - \tau\boldsymbol{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A\Phi}$, bias $\tau(\boldsymbol{A\Phi})^\top\boldsymbol{y}$ and activation function $S_{\tau\lambda}$, where the trainable parameters are the entries of $\boldsymbol{\Phi}$. Note that for $l > 1$, all f_l coincide as functions on $\mathbb{R}^p$. The index then refers to the iteration step or layer of the neural network, respectively. Then we denote the concatenation of l such layers as $f^l_{\boldsymbol{\Phi}}$, *i.e.*, for $\boldsymbol{\Phi}$ in every layer and given by

$$f^L_{\boldsymbol{\Phi}}(\boldsymbol{y}) = f_L \circ f_{L-1} \cdots \circ f_1(\boldsymbol{y}), \tag{2.3}$$

Note that, strictly speaking, the vector $\boldsymbol{y}$ will also be an input to the subsequent layers f_2, f_3 etc., but to simplify the notation, we do not write it explicitly after each layer. This point will not be of major importance for our derivations throughout this chapter.

For an actual reconstruction we need to apply the dictionary $\boldsymbol{\Phi}$ again after the final layer. This means, a decoder (for a fixed number of layers L) is a neural network with shared weights

$$\mathrm{dec}^L_{\boldsymbol{\Phi}}(\boldsymbol{y}) = \boldsymbol{\Phi} f_L \circ f_{L-1} \cdots \circ f_1(\boldsymbol{y}) = \boldsymbol{\Phi} f^L_{\boldsymbol{\Phi}}(\boldsymbol{y}).$$

For technical reasons which will become apparent later in the proofs in Section 3, we will add an additional function σ after the final layer. Different choices are possible here; we consider the choice

$$\sigma : \mathbb{R}^p \to \mathbb{R}^p, \qquad \boldsymbol{x} \mapsto \begin{cases} \boldsymbol{x} & \text{if } \|\boldsymbol{x}\|_2 \leq B_{\text{out}}, \\ B_{\text{out}}\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} & \text{if } \|\boldsymbol{x}\|_2 > B_{\text{out}}, \end{cases} \tag{2.4}$$

with some fixed constant B_{out}. Obviously, this ensures $\|\sigma(\boldsymbol{x})\|_2 \leq B_{\text{out}}$. Furthermore, note that σ is norm-contractive and 1-Lipschitz, *i.e.*,

$$\|\sigma(\boldsymbol{x})\|_2 \leq \|\boldsymbol{x}\|_2 \qquad \text{and} \qquad \|\sigma(\boldsymbol{x}_1) - \sigma(\boldsymbol{x}_2)\|_2 \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \tag{2.5}$$

for any $\boldsymbol{x}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^p$. The role of σ is to push the output of the network inside the ℓ_2-ball of radius B_{out}, which in many applications is approximately known. The prior knowledge about the range of the output (boundedness) can improve the reconstruction performance and generalization [WGLZ20]. The constant B_{out} may be simply chosen to be equal to B_{in}.

The hypothesis set consists of all functions that can be expressed as L-step soft thresholding, where the dictionary matrix $\boldsymbol{\Phi}$ parameterizes the hypothesis class, and with an additional σ after the final layer added. That is,

$$\mathcal{H}^L_1 = \{\sigma \circ f : \mathbb{R}^m \to \mathbb{R}^p : f(\boldsymbol{y}) = \boldsymbol{\Phi} f^L_{\boldsymbol{\Phi}}(\boldsymbol{y}), \boldsymbol{\Phi} \in O(p)\}. \tag{2.6}$$

The assumption that $\boldsymbol{\Phi}$ ranges over the orthogonal group $O(p)$ and is shared across the layers leads to a recurrent neural network with a moderate number of weights. Using weight sharing enables a straightforward interpretation of learning a dictionary for reconstruction. (Much more general scenarios are discussed later, including models without weight-sharing (or different degrees thereof), and models where also the threshold λ and the stepsize τ may be trainable, and even be altered from layer to layer.) Throughout, we will use the loss function (2.1) (which has the advantage of being 1-Lipschitz) and the notion of the generalization error as in (1.15).

2.2.2 Main result

Let us begin by stating the following result on the generalization error of the class of neural networks $\mathcal{H}_1^L$ introduced above in (2.6) with a learned orthogonal dictionary. We state our theorem here under the simplifying but reasonable assumption that $\tau\|\boldsymbol{A}\|_2^2 \leq 1$, satisfying the convergence condition (1.11). A more general version of the result will be presented in Section 2.2.7. Note that this, and similar results in this Chapter, are applications of Theorem 1.9.

Theorem 2.1 *Consider the hypothesis space $\mathcal{H}_1^L$, $L \geq 2$, defined in* (2.6) *and assume the samples $\boldsymbol{x}_i$, $i = 1, \ldots, n$, to be drawn i.i.d. at random according to some (unknown) distribution such that $\|\boldsymbol{x}_i\|_2 \leq B_{in}$ almost surely with $B_{in} = B_{out}$ in* (2.4)*. Let $\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i$ and assume that $\tau\|\boldsymbol{A}\|_{2\to 2}^2 \leq 1$. Then with probability at least $1-\delta$, for all $h \in \mathcal{H}_1^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) \quad + \quad & 8B_{out}\sqrt{\frac{pm}{n}}\sqrt{2\log(5L)} + 8B_{out}\frac{N\sqrt{\log(e+8eL)}}{\sqrt{n}} \\ & + B_{out}\sqrt{\frac{128\log(4/\delta)}{n}}.\end{aligned}$$

Proof. The proof of Theorem 2.1 is based on Theorem 2.8 and Corollary 2.9 further below in Section 2.2.7. ∎

Of course, the idea is to choose an h that minimizes the empirical loss $\hat{\mathcal{L}}(h)$, *i.e.,* the first term on the right hand side of (2.7), but in principle any h (computed by some algorithm) can be inserted into this bound. Since the samples are available, both $\hat{\mathcal{L}}(h)$ and the other terms can be computed (assuming B_{in} is known), so that the theorem allows to provide a concrete bound of the true risk $\mathcal{L}(h)$. Roughly speaking, *i.e.,* ignoring constants, the generalization error can be bounded as

$$|\mathcal{L}(h) - \hat{\mathcal{L}}(h)| \lesssim \sqrt{\frac{pm\log(L) + p^2\log(L)}{n}}. \tag{2.8}$$

In other words, once the number of training samples scales like $n \sim (pm + p^2)\log(L)$, the generalization error is guaranteed to be small with high probability.

Remarkably, the number L of layers only enters logarithmically, while some of the previously available bounds for deep neural networks (in the context of classification, however) scale even only exponentially with L (at least in many interesting settings).

The remainder of this section is devoted to the proof of the above statement. We will use the approach based on the Rademacher complexity as described in Chapter 1, in particular in Theorem 1.9. Hence, we need to estimate the Rademacher complexity

$$\mathcal{R}_n(\ell \circ \mathcal{H}_1^L) = \mathbb{E} \sup_{h \in \mathcal{H}_1^L} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i \|h(\boldsymbol{y}_i) - \boldsymbol{x}_i\|_2.$$

As explained in Chapter 1, the so-called contraction principle is often applied in such situations. However, since we are dealing with a hypothesis class of vector-valued functions, it is not applicable in its standard form. A crucial tool, upon which our proof relies, is a is a generalization to this situation of vector-valued functions due to [Mau16, Corollary 4], which is provided in the appendix in Lemma B.4. As both the ℓ_2-norm and (by assumption) the function σ from (2.4) are 1-Lipschitz, applying Lemma B.4 yields

$$\mathcal{R}_n(\ell \circ \mathcal{H}^L) \leq \sqrt{2}\mathbb{E} \sup_{h \in \mathcal{H}^L} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{p} \varepsilon_{ik} h_k(\boldsymbol{x}_i). \tag{2.9}$$

In order to derive a bound for the Rademacher complexity, we use chaining techniques. Roughly speaking, this refers to bounding the expectation of a stochastic process by geometric properties of its index set (covering numbers at different scales), equipped with an appropriate norm (or metric). We briefly provide the necessary results in the next section, for a more detailed introduction to the topic, we refer the reader to [LT11; Tal14].

2.2.3 Boundedness: Assumptions and Results

For technical reasons that will become apparent, we will introduce a separate dictionary for the linear transformation after the very final layer and consider the enlarged hypothesis class

$$\mathcal{H}_2^L = \{\sigma \circ h : \mathbb{R}^m \to \mathbb{R}^p : h(\boldsymbol{y}) = \boldsymbol{\Psi} f_{\boldsymbol{\Phi}}^L(\boldsymbol{y}), \boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(p)\}. \tag{2.10}$$

In order to apply Theorem 1.9, the loss function needs to be bounded. Therefore, and as commonly done in the machine learning literature, we assume (as already mentioned) that the input is bounded in the ℓ_2-norm by some constant B_{in}, *i.e.*,

$$\|\boldsymbol{x}\|_2 \leq B_{\text{in}}. \tag{2.11}$$

Furthermore, let us recall from (2.5) that the function σ is bounded by B_{out}. In particular, this means that every $h \in \mathcal{H}_2^L$ (analogously for $\mathcal{H}_1^L$) is also bounded by

$$\|h(\boldsymbol{y})\|_2 = \left\|\sigma\left(\boldsymbol{\Psi} f_{\boldsymbol{\Phi}}^L(\boldsymbol{y})\right)\right\|_2 \leq B_{\text{out}} \tag{2.12}$$

independently of $\boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(p)$. By passing to the matrix notation (*i.e.*, considering the matrix $\boldsymbol{Y}$ collecting all measurements, instead of a single measurement $\boldsymbol{y}$), we obtain the similar estimate

$$\|h(\boldsymbol{Y})\|_F \leq \sqrt{n} B_{\text{out}} \tag{2.13}$$

where the additional term of $\sqrt{n}$ takes the number of training points into account. By combining (2.11) and (2.12), we find that the loss function is bounded by $B_{\text{in}} + B_{\text{out}}$, as

$$\begin{aligned} \ell(h, \boldsymbol{y}, \boldsymbol{x}) &= \|h(\boldsymbol{y}) - \boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_2 + \|h(\boldsymbol{y})\|_2 \\ &\leq B_{\text{in}} + B_{\text{out}}, \end{aligned}$$

so that $B_{\text{in}} + B_{\text{out}}$ plays the role of c in Theorem 1.9. Besides these boundedness assumptions, we can also upper bound the output $f^l_{\mathbf{\Phi}}(\boldsymbol{Y})$ with respect to the Frobenius norm after any number of layers l (in particular for $l < L$, when the layer is not directly followed by an application of the σ function) as follows . This will be used later in the main technical result, Theorem 2.8.

Lemma 2.2 *For any $\mathbf{\Phi} \in O(p)$, $l \in \mathbb{N}$, and arbitrary $\tau, \lambda > 0$ in $S_{\tau\lambda}$ in the definition* (1.10) *of $f^l_{\mathbf{\Phi}}$, we have*

$$\left\|f^l_{\mathbf{\Phi}}(\boldsymbol{Y})\right\|_F \leq \left\|\tau(\boldsymbol{A}\mathbf{\Phi})^\top \boldsymbol{Y}\right\|_F \sum_{k=0}^{l-1} \left\|\boldsymbol{I} - \tau\mathbf{\Phi}^\top \boldsymbol{A}^\top \boldsymbol{A}\mathbf{\Phi}\right\|^k_{2\to 2} \tag{2.15}$$

$$\leq \tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F \sum_{k=0}^{l-1} \left\|\boldsymbol{I} - \tau\boldsymbol{A}^\top \boldsymbol{A}\right\|^k_{2\to 2}. \tag{2.16}$$

Before we prove this result, let us point out the following useful observation regarding the expression $\|\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\|_{2\to 2}$ that we will encounter more often in the sequel. By part (i) of Lemma C.2 below, it can be easily bounded under realistic assumptions. In particular, we can use it to simplify the above estimate to obtain for arbitrary $\mathbf{\Psi}, \mathbf{\Phi} \in O(p)$. Namely, under the condition of $\tau\|\boldsymbol{A}\|^2_{2\to 2} \leq 1$ and assuming $\boldsymbol{y}_i = \boldsymbol{A}(\boldsymbol{x}_i)$ we have

$$\begin{aligned}\left\|\mathbf{\Psi} f^L_{\mathbf{\Phi}}(\boldsymbol{Y})\right\|_2 = \left\|f^L_{\mathbf{\Phi}}(\boldsymbol{Y})\right\|_2 &\leq L\tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F = L\tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{A}\boldsymbol{X}\|_F \\ &\leq L\|\boldsymbol{X}\|_F \leq L\sqrt{n}B_{\text{in}},\end{aligned} \tag{2.17}$$

i.e., a linear growth with L. Note that this is a worst case bound, and might possibly be improved under additional assumptions. Now, let us return to Lemma 2.2 and prove this result.

Proof. Note that the second inequality (2.16) immediately follows from (2.15) due to the orthogonality of $\mathbf{\Phi}$. We will prove (2.15) via induction. Clearly, for $l = 1$, we have $\|f^1_{\mathbf{\Phi}}(\boldsymbol{Y})\|_F = \|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\|_F$. Assuming the statement is true for l, we obtain it for $l+1$ by the following chain of inequalities, using in particular the contractivity of $S_{\tau\lambda}$ with respect to the Frobenius norm,

$$\begin{aligned}
\left\|f^{l+1}_{\mathbf{\Phi}}(\boldsymbol{Y})\right\|_F &= \left\|S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\mathbf{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A}\mathbf{\Phi}\right) f^l_{\mathbf{\Phi}}(\boldsymbol{Y}) + \tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right]\right\|_F \\
&\leq \left\|\left(\boldsymbol{I} - \tau\mathbf{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A}\mathbf{\Phi}\right) f^l_{\mathbf{\Phi}}(\boldsymbol{Y})\|_F + \|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right\|_F \\
&\leq \left\|\boldsymbol{I} - \tau\mathbf{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A}\mathbf{\Phi}\right\|_{2\to 2}\left\|f^l_{\mathbf{\Phi}}(\boldsymbol{Y})\right\|_F + \left\|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right\|_F \\
&\leq \left\|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right\|_F\left(\sum_{k=0}^{l-1}\left\|\boldsymbol{I} - \tau\mathbf{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A}\mathbf{\Phi}\right\|^{k+1}_{2\to 2}\right) + \left\|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right\|_F \\
&= \left\|\tau(\boldsymbol{A}\mathbf{\Phi})^\top\boldsymbol{Y}\right\|_F\sum_{k=0}^{l}\left\|\boldsymbol{I} - \tau\mathbf{\Phi}^\top\boldsymbol{A}^\top\boldsymbol{A}\mathbf{\Phi}\right\|^k_{2\to 2},
\end{aligned}$$

where we have used the induction hypothesis to arrive at the fourth line. ∎

2.2.4 Bounding the Rademacher Complexity

Recalling our hypothesis spaces introduced above in equations (2.6) and (2.10), obviously $\mathcal{H}_1^L$ is embedded in $\mathcal{H}_2^L$, *i.e.,* we have the set inclusion

$$\mathcal{H}_1^L \subset \mathcal{H}_2^L.$$

For fixed number of layers $L \in \mathbb{N}$ and $i = 1, 2$ define the set $\mathcal{M}_i \subset \mathbb{R}^{p \times n}$ as follows by

$$\mathcal{M}_i = \left\{ [h(\boldsymbol{y}_1), \dots, h(\boldsymbol{y}_n)] \in \mathbb{R}^{p \times n} : h \in \mathcal{H}_i^L \right\}.$$

Concretely, in the case $i = 2$, the set $\mathcal{M}_2$ corresponding to the hypothesis space $\mathcal{H}_2^L$ reads as

$$\mathcal{M}_2 = \left\{ \sigma\left(\boldsymbol{\Psi} f_{\boldsymbol{\Phi}}^L(\boldsymbol{Y}) \right) \in \mathbb{R}^{p \times n} : \boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(p) \right\}. \tag{2.18}$$

Note that $\mathcal{M}_2$ is parameterized by $\boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(p)$ (as the hypothesis space $\mathcal{H}_2^L$ is), such that we can rewrite (2.9) as

$$\mathcal{R}_n(\ell \circ \mathcal{H}_2^L) \leq \sqrt{2}\mathbb{E} \sup_{\boldsymbol{M} \in \mathcal{M}_2} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{p} \varepsilon_{ik} M_{ki}. \tag{2.19}$$

We use Dudley's inequality, Theorem A.4, and a covering number argument to bound the Rademacher complexity term (2.19) . The appropriate (pseudo-)metric d from (A.2) appearing in Theorem A.4 turns out to be the Frobenius norm, since for any $\boldsymbol{M}, \hat{\boldsymbol{M}} \in \mathcal{M}_2$,

$$\begin{aligned} & d(\boldsymbol{M}, \hat{\boldsymbol{M}}) \\ &= \left(\mathbb{E} \left| \sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} M_{ki} - \sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} \hat{M}_{ki} \right|^2 \right)^{1/2} = \left(\mathbb{E} \left(\sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} (M_{ki} - \hat{M}_{ki}) \right)^2 \right)^{1/2} \\ &= \left(\sum_{i=1}^{n} \sum_{k=1}^{m_L} (M_{ki} - \hat{M}_{ki})^2 \right)^{1/2} = \|\boldsymbol{M} - \hat{\boldsymbol{M}}\|_F, \end{aligned} \tag{2.20}$$

where we have used that $\mathbb{E}[\varepsilon_{ik}^2] = 1$ and $\mathbb{E}[\varepsilon_{ik}\varepsilon_{jl}] = \mathbb{E}[\varepsilon_{ik}]\mathbb{E}[\varepsilon_{jl}] = 0$ whenever $i \neq j$ or $k \neq l$ due to independence of the Rademacher variables ε_{ik} and ε_{jl}. The Rademacher process defined in (2.19) is a sub-Gaussian process, *i.e.,* satisfying (A.3), as it is obviously centered and furthermore

$$\begin{aligned} \mathbb{E} \exp \left(\theta \left(\sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} M_{ki} - \sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} \hat{M}_{ki} \right) \right) &= \mathbb{E} \exp \left(\theta \left(\sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} (M_{ki} - \hat{M}_{ki}) \right) \right) \\ &\leq \exp(\theta^2 \|\boldsymbol{M} - \hat{\boldsymbol{M}}\|_F^2 / 2). \end{aligned}$$

Furthermore, for the set of matrices $\mathcal{M}_2$ defined above in (2.19), its radius $\Delta(\mathcal{M}_2)$ can be bounded by

$$\begin{aligned} \Delta(\mathcal{M}_2) = \sup_{h \in \mathcal{H}_2^L} \sqrt{ \mathbb{E} \left(\sum_{i=1}^{n} \sum_{k=1}^{p} \varepsilon_{ik} h_k(\boldsymbol{y}_i) \right)^2 } &\leq \sup_{h \in \mathcal{H}_2^L} \sqrt{ \mathbb{E} \sum_{i=1}^{n} \sum_{k=1}^{p} (h_k(\boldsymbol{y}_i))^2 } \\ \leq \sup_{h \in \mathcal{H}_2^L} \sqrt{ \sum_{i=1}^{n} \|h(\boldsymbol{y}_i)\|^2 } \leq \sqrt{n} B_{\text{out}}, \end{aligned}$$

with the last inequality already known from (2.13). Plugging our findings into Dudley's inequality (A.4), the Rademacher complexity term (2.19) can be upper bounded by

$$\mathcal{R}_n(\ell \circ \mathcal{H}_2^L) \leq \frac{4\sqrt{2}}{n} \int_0^{\sqrt{n}B_{\text{out}}/2} \sqrt{\log \mathcal{N}(\mathcal{M}_2, \|\cdot\|_F, \varepsilon)}\, \mathrm{d}\varepsilon. \tag{2.21}$$

We only need to find the covering numbers inside the integral. For that, we bound the covering number of the hypothesis class by the covering number of its parameter space. This is done using a perturbation analysis argument.

2.2.5 A Perturbation Result

The following theorem relates the effect of perturbation of the parameters on the function outputs. This result will be used to bound their covering numbers.

Theorem 2.3 *Consider the functions $f_{\mathbf{\Phi}}^L$ defined as in* (2.3) *with $L \geq 2$ and a dictionary $\mathbf{\Phi}$ in $O(p)$. Then, for any $\mathbf{\Phi}_1, \mathbf{\Phi}_2 \in O(p)$ we have*

$$\left\| f_{\mathbf{\Phi}_1}^L(\mathbf{Y}) - f_{\mathbf{\Phi}_2}^L(\mathbf{Y}) \right\|_F \leq K_L \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to2}, \tag{2.22}$$

where K_L is given by

$$\begin{aligned} K_L &= \tau\|\mathbf{Y}\|_F \|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\|_{2\to2}^{L-1} \\ &\quad + \tau\|\mathbf{Y}\|_F \sum_{l=2}^{L} \|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\|_{2\to2}^{L-l} \left(1 + 2\tau\|\mathbf{A}\|_{2\to2}^2 \sum_{k=0}^{l-2} \|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\|_{2\to2}^k\right). \end{aligned} \tag{2.23}$$

If $\tau\|\mathbf{A}\|_{2\to2}^2 \leq 1$, we have the simplified upper bound

$$K_L \leq \tau\|\mathbf{Y}\|_F L(L+3). \tag{2.24}$$

The bound (2.24) follows from the observation in part (i) of Lemma C.2.

Proof. We formally set $f_{\mathbf{\Phi}_1}^0(\mathbf{Y}) = f_{\mathbf{\Phi}_2}^0(\mathbf{Y}) = \mathbf{Y}$ for a unified treatment of all layers $l \geq 1$. Using the fact that $S_{\tau\lambda}$ is 1-Lipschitz we obtain

$$\begin{aligned} & \left\| f_{\mathbf{\Phi}_1}^l(\mathbf{Y}) - f_{\mathbf{\Phi}_2}^l(\mathbf{Y}) \right\|_F \\ \leq\ & \Big\| \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{A}\mathbf{\Phi}_1\right) f_{\mathbf{\Phi}_1}^{l-1}(\mathbf{Y}) + \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{Y} \\ & \quad - \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{A}\mathbf{\Phi}_2\right) f_{\mathbf{\Phi}_2}^{l-1}(\mathbf{Y}) - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{Y} \Big\|_F \\ \leq\ & \left\| \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{A}\mathbf{\Phi}_1\right) f_{\mathbf{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{A}\mathbf{\Phi}_2\right) f_{\mathbf{\Phi}_2}^{l-1}(\mathbf{Y}) \right\|_F \\ & \quad + \left\| \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{Y} - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{Y} \right\|_F \\ \leq\ & \left\| \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{A}\mathbf{\Phi}_1\right) f_{\mathbf{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{A}\mathbf{\Phi}_2\right) f_{\mathbf{\Phi}_2}^{l-1}(\mathbf{Y}) \right\|_F \\ & \quad + 2\tau \|\mathbf{Y}\|_F \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to2}. \end{aligned}$$

The term (2.25) is estimated further as follows.

$$\left\| \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_1)^\top\mathbf{A}\mathbf{\Phi}_1\right) f_{\mathbf{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\mathbf{\Phi}_2)^\top\mathbf{A}\mathbf{\Phi}_2\right) f_{\mathbf{\Phi}_2}^{l-1}(\mathbf{Y}) \right\|_F$$

$$
\begin{aligned}
&\leq \left\| \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_1\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right. \\
&\qquad + \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \Big\|_F \\
&\qquad + \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \Big\|_F \\
&\leq \left\| \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_1\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right. \\
&\qquad + \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \\
&\qquad \left. + \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) \left(f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y})\right) \right\|_F \\
&\leq \left\| \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_1 f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2 f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right. \\
&\qquad \left. + \tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{A\Phi}_2 f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2 f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F \\
&\qquad + \left\| \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \\
&\leq \left\| \tau(\boldsymbol{A\Phi}_1)^\top \right\|_{2\to 2} \left\| (\boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2) f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F \\
&\qquad + \tau \left\| (\boldsymbol{A\Phi}_1)^\top - (\boldsymbol{A\Phi}_2)^\top \right\|_{2\to 2} \left\| \boldsymbol{A\Phi}_2 f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F \\
&\qquad + \left\| \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \\
&\leq \tau \left\| \boldsymbol{A} \right\|_{2\to 2} \left\| \boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2 \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F + \tau \left\| \boldsymbol{A} \right\|_{2\to 2} \left\| \boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2 \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F \\
&\qquad + \left\| \left(\boldsymbol{I} - \tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{A\Phi}_2\right) \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \\
&= 2\tau \left\| \boldsymbol{A} \right\|_{2\to 2} \left\| \boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2 \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F + \left\| \boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A} \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F .
\end{aligned}
$$

Plugging this back into (2.25) gives us

$$
\begin{aligned}
& \left\| f^{l}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \\
\leq \;& \left\| \boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A} \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \\
& \quad + \tau \left(2 \left\| \boldsymbol{Y} \right\|_F + 2 \left\| \boldsymbol{A} \right\|_{2\to 2} \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) \right\|_F \right) \left\| \boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2 \right\|_{2\to 2} \\
\leq \;& A \left\| f^{l-1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{l-1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F + B_l \left\| \boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2 \right\|_{2\to 2},
\end{aligned}
$$

where A and B_l in the previous estimate (2.27) are given by

$$
\begin{aligned}
A &= \left\| \boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A} \right\|_{2\to 2}, \\
Z_0 &= 0, \qquad Z_l = \sum_{k=0}^{l-1} \left\| \boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A} \right\|_{2\to 2}^k, \quad l \geq 1, \\
B_l &= \tau \|\boldsymbol{Y}\|_F \left(2 + 2\tau \left\| \boldsymbol{A} \right\|_{2\to 2}^2 Z_{l-1} \right), \qquad l \geq 1.
\end{aligned}
$$

Using these abbreviations, the general formula for K_L in (2.23) has the compact form

$$
K_L = \sum_{l=1}^{L} A^{L-l} B_l, \qquad L \geq 1. \tag{2.28}
$$

Based on (2.27) we prove via induction that (2.22) holds for any number of layers $L \in \mathbb{N}$ with K_L given by (2.28). For $L = 1$, we can directly calculate the constant K_1 via

$$\begin{aligned} \left\| f^1_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^1_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F =& \left\| S_{\tau\lambda}(\tau(\boldsymbol{A\Phi}_1)^\top \boldsymbol{Y}) - S_{\tau\lambda}(\tau(\boldsymbol{A\Phi}_2)^\top \boldsymbol{Y}) \right\|_F \\ \leq& \tau \|\boldsymbol{Y}\|_F \, \|\boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2\|_{2\to 2}, \end{aligned}$$

so that $\tau\|\boldsymbol{Y}\|_F \leq 2\tau\|\boldsymbol{Y}\|_F = B_1 = K_1$, as claimed in (2.28). Now we proceed with the induction step, assuming formula (2.28) to hold for some $L \in \mathbb{N}$. Applying the estimate after (2.26) for the output after layer $L + 1$, we obtain

$$\begin{aligned} \left\| f^{L+1}_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^{L+1}_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F \leq& A \left\| f^L_{\boldsymbol{\Phi}_1}(\boldsymbol{Y}) - f^L_{\boldsymbol{\Phi}_2}(\boldsymbol{Y}) \right\|_F + B_{L+1} \, \|\boldsymbol{A\Phi}_2 - \boldsymbol{A\Phi}_1\|_{2\to 2} \\ \leq& AK_L \|\boldsymbol{A\Phi}_2 - \boldsymbol{A\Phi}_1\|_{2\to 2} + B_{L+1} \|\boldsymbol{A\Phi}_2 - \boldsymbol{A\Phi}_1\|_{2\to 2} \\ \leq& (AK_L + B_{L+1}) \|\boldsymbol{A\Phi}_2 - \boldsymbol{A\Phi}_1\|_{2\to 2}, \end{aligned}$$

and therefore,

$$K_{L+1} \quad = \quad AK_L + B_{L+1} = A \sum_{l=1}^{L} A^{L-l} B_l + B_{L+1} = \sum_{l=1}^{L+1} A^{(L+1)-l} B_l.$$

This is indeed the desired expression for K_{L+1}, finishing the proof of (2.22). It remains to prove the upper bound (2.24). In part (i) of Lemma C.2 we show that $\|\boldsymbol{I} - \tau\boldsymbol{A}^\top \boldsymbol{A}\|_{2\to 2} = 1$ when $\tau\|\boldsymbol{A}\|^2_{2\to 2} \leq 1$. Therefore we obtain

$$\begin{aligned} K_L \quad =& \quad \sum_{l=1}^{L} A^{L-l} B_l \leq \sum_{l=1}^{L} B_l = \tau\|\boldsymbol{Y}\|_F \sum_{l=1}^{L} \left(2 + 2\tau \, \|\boldsymbol{A}\|^2_{2\to 2} \, Z_{l-1} \right) \\ \leq& \quad 2L\tau\|\boldsymbol{Y}\|_F + 2\tau\|\boldsymbol{Y}\|_F \sum_{l=1}^{L} Z_{l-1} \leq 2L\tau\|\boldsymbol{Y}\|_F + 2\tau\|\boldsymbol{Y}\|_F \sum_{l=1}^{L} l \\ =& \quad \tau\|\boldsymbol{Y}\|_F L(L+3), \end{aligned}$$

finishing the proof of the theorem. ■

The following result is an adaptation of the previous theorem to take the special form of the final layer into account (a final linear transformation, followed by applying the function σ).

Corollary 2.4 *Consider the thresholding networks $\boldsymbol{\Psi} f^L_{\boldsymbol{\Phi}} \in \mathcal{H}^L_2$ as defined in Section 2.2.4, with $L \geq 2$ and $\boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(p)$. Then, for any $\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \in O(p)$ and $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in O(p)$ we have*

$$\begin{aligned} & \left\| \sigma(\boldsymbol{\Psi}_1 f^L_{\boldsymbol{\Phi}_1}(\boldsymbol{Y})) - \sigma(\boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_2}(\boldsymbol{Y})) \right\|_F \\ \leq & \quad M_L \|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to 2} + K_L \|\boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2\|_{2\to 2}, \end{aligned}$$

with K_L as in Theorem 2.3 and

$$M_L = \tau\|\boldsymbol{A}\|_{2\to 2} \|\boldsymbol{Y}\|_F \sum_{k=0}^{L-1} \left\| \boldsymbol{I} - \tau\boldsymbol{A}^\top \boldsymbol{A} \right\|^k_{2\to 2}. \tag{2.30}$$

Under the additional assumption that $\tau\|A\|_{2\to 2}^2 \leq 1$ *we have*

$$\begin{aligned}&\left\|\sigma(\Psi_1 f_{\Phi_1}^L(Y)) - \sigma(\Psi_2 f_{\Phi_2}^L(Y))\right\|_F \\ &\leq \tau\|Y\|_F \left(L\|A\|_{2\to 2}\|\Psi_1 - \Psi_2\|_{2\to 2} + L(L+3)\|A\Phi_1 - A\Phi_2\|_{2\to 2}\right).\end{aligned}$$

Proof. Let us begin with the following estimates, which now include the application of the measurement and the respective dictionary after the final layer. By the 1-Lipschitzness of σ, adding mixed terms and applying the triangle inequality, and finally using Theorem 2.3 for the second summand in the last step we obtain

$$\begin{aligned}&\left\|\sigma\left(\Psi_1 f_{\Phi_1}^L(Y)\right) - \sigma\left(\Psi_2 f_{\Phi_2}^L(Y)\right)\right\|_F \\ \leq\; &\left\|\Psi_1 f_{\Phi_1}^L(Y) - \Psi_2 f_{\Phi_1}^L(Y) + \Psi_2 f_{\Phi_1}^L(Y) - \Psi_2 f_{\Phi_2}^L(Y)\right\|_F \\ \leq\; &\left\|\Psi_1 f_{\Phi_1}^L(Y) - \Psi_2 f_{\Phi_1}^L(Y)\right\|_F + \left\|\Psi_2 f_{\Phi_1}^L(Y) - \Psi_2 f_{\Phi_2}^L(Y)\right\|_F \\ \leq\; &\left\|f_{\Phi_1}^L(Y)\right\|_F \|\Psi_1 - \Psi_2\|_{2\to 2} + \left\|f_{\Phi_1}^L(Y) - f_{\Phi_2}^L(Y)\right\|_F \\ \leq\; &\left\|f_{\Phi_1}^L(Y)\right\|_F \|\Psi_1 - \Psi_2\|_{2\to 2} + K_L\,\|A\Phi_1 - A\Phi_2\|_{2\to 2}.\end{aligned}$$

Now, (2.29) follows from Lemma 2.2. The additional simplified bounds then easily follow from the respective ones in Theorem 2.3 as well as in (2.17). ∎

Remark 2.5 One may try a similar computation like in the proof above for the hypothesis space $\mathcal{H}_1^L$ instead $\mathcal{H}_2^L$. However, after the analog estimate for $\Phi_1, \Phi_2 \in O(p)$,

$$\left\|\Phi_1 f_{\Phi_1}^L(Y) - \Phi_2 f_{\Phi_2}^L(Y)\right\|_F \leq \left\|f_{\Phi_1}^L(Y)\right\|_F \|\Phi_1 - \Phi_2\|_{2\to 2} + K_L\,\|A\Phi_1 - A\Phi_2\|_{2\to 2},$$

we need to consider both $\|A\Phi_1 - A\Phi_2\|_{2\to 2}$ and $\|\Phi_1 - \Phi_2\|_{2\to 2}$ for later covering number arguments. Using $\mathcal{H}_2^L$ helps to obtain more concise covering numbers for the class. Therefore, we decouple the single dictionary applied after the final layer from the previous layers (which all appear together with A). ♢

2.2.6 Covering number estimates

Our proof is built on Dudley's integral in (2.21). We need to compute covering numbers $\mathcal{N}(\mathcal{M}_2, \|\cdot\|_F, \varepsilon)$ at different scales $\varepsilon > 0$ to evaluate the integral for the space $\mathcal{M}_2$. The following lemma provides a covering number estimate of A applied to the orthogonal group. It is a straightforward application of the well-known Lemma A.2 in the appendix.

Lemma 2.6 *For a fixed matrix* $A \in \mathbb{R}^{m\times p}$ *consider the set* $\mathcal{W}$ *defined by*

$$\mathcal{W} = \{A\Phi : \Phi \in O(p)\} \subset \mathbb{R}^{m\times p}, \tag{2.31}$$

i.e., *A applied to the orthogonal group. The covering number estimate is given by*

$$\mathcal{N}(\mathcal{W}, \|\cdot\|_{2\to 2}, \varepsilon) \leq \left(1 + \frac{2\|A\|_{2\to 2}}{\varepsilon}\right)^{mp}.$$

Proof. First note that $\mathcal{W}$ can be rewritten as

$$\mathcal{W} = \left\{ \|\boldsymbol{A}\|_{2\to2} \frac{\boldsymbol{A\Phi}}{\|\boldsymbol{A}\|_{2\to2}} : \boldsymbol{\Phi} \in O(p) \right\}.$$

For the covering numbers of the orthogonal group $(O(p), \|\cdot\|_{2\to2})$ equipped with the spectral norm we have

$$\mathcal{N}\left(O(p), \|\cdot\|_{2\to2}, \varepsilon\right) \leq \left(1 + \frac{2}{\varepsilon}\right)^{p^2}.$$

This follows from the fact that the orthogonal group $O(p)$ is contained in $B^{p\times p}_{\|\cdot\|_{2\to2}}$, and therefore Lemma A.2 applies. This bound then gives

$$\begin{aligned}\mathcal{N}\left(\mathcal{W}, \|\cdot\|_{2\to2}, \varepsilon\right) =& \mathcal{N}\left(\left\{\boldsymbol{A\Phi}/\|\boldsymbol{A}\|_{2\to2} : \boldsymbol{\Phi} \in O(p)\right\}, \|\cdot\|_{2\to2}, \varepsilon/\|\boldsymbol{A}\|_{2\to2}\right) \\ \leq& \left(1 + \frac{2\|\boldsymbol{A}\|_{2\to2}}{\varepsilon}\right)^{mp}.\end{aligned}$$

■

Recall that for Dudleys inequality (Theorem A.4), we need to estimate the covering numbers $\mathcal{N}\left(\mathcal{M}_2, \|\cdot\|_{2\to2}, \varepsilon\right)$ of the set $\mathcal{M}_2$ defined in (2.18). In Corollary 2.4, we showed we can estimate distances in $\mathcal{M}_2$ via distances of the underlying parameters, $\|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to2}$ and $\|\boldsymbol{A\Phi}_1 - \boldsymbol{A\Phi}_2\|_{2\to2}$. We make use of this in the next corollary, which prepares the application of Dudleys inequality afterwards.

Corollary 2.7 *The (logarithms of the) covering numbers of the set $\mathcal{M}_2$ are bounded by*

$$\begin{aligned}&\log\left(\mathcal{N}\left(\mathcal{M}_2, \|\cdot\|_{2\to2}, \varepsilon\right)\right) \\ &\quad \leq p^2 \cdot \log\left(1 + \frac{4M_L}{\varepsilon}\right) + mp \cdot \log\left(1 + \frac{4\|\boldsymbol{A}\|_{2\to2}K_L}{\varepsilon}\right).\end{aligned}$$

Proof. Using the definition of the set (2.31), we have

$$\begin{aligned}\mathcal{N}\left(K_L\{\boldsymbol{A\Phi} : \boldsymbol{\Phi} \in O(p)\}, \|\cdot\|_{2\to2}, \varepsilon\right) &= \mathcal{N}\left(\{\boldsymbol{A\Phi} : \boldsymbol{\Phi} \in O(p)\}, \|\cdot\|_{2\to2}, \varepsilon/K_L\right) \\ &\leq \left(1 + \frac{2\|\boldsymbol{A}\|_{2\to2}K_L}{\varepsilon}\right)^{mp}.\end{aligned}$$

By the inclusion $O(p) \subset B^{p\times p}_{\|\cdot\|_{2\to2}}$, and by Lemma A.2 (with $\varepsilon/2$ instead of ε) we obtain

$$\begin{aligned}\mathcal{N}\left(M_L \cdot O(p), \|\cdot\|_{2\to2}, \varepsilon/2\right) &= \mathcal{N}\left(O(p), \|\cdot\|_{2\to2}, \varepsilon/(2M_L)\right) \\ &\leq \left(1 + \frac{4M_L}{\varepsilon}\right)^{p^2}.\end{aligned}$$

Applying Lemma A.3 for covering numbers estimates of product spaces (in the case $p = 2$) and the previous estimates, we can now bound the covering number of the set $\mathcal{M}_2$ by

$$\begin{aligned}\mathcal{N}\left(\mathcal{M}_2, \|\cdot\|_F, \varepsilon\right) &\leq \mathcal{N}\left(M_L \cdot O(p) \times K_L \cdot \mathcal{W}, \|\cdot\|_{2\to2}, \varepsilon\right) \\ &\leq \mathcal{N}\left(M_L \cdot O(p), \|\cdot\|_{2\to2}, \varepsilon/2\right) \mathcal{N}\left(K_L \cdot \mathcal{W}, \|\cdot\|_{2\to2}, \varepsilon/2\right) \\ &\leq \left(1 + \frac{4M_L}{\varepsilon}\right)^{p^2} \left(1 + \frac{4\|\boldsymbol{A}\|_{2\to2}K_L}{\varepsilon}\right)^{mp},\end{aligned}$$

which immediately gives us the desired statement after taking the logarithm. ■

2.2.7 Main result

Finally, we are able to state and prove the main result of this section. It is similar to Theorem 2.1; in fact, Theorem 2.1 will be derived from the following result, but considers the larger hypothesis class $\mathcal{H}_2^L$ instead of $\mathcal{H}_1^L$. Furthermore, note that the condition $\tau\|\boldsymbol{A}\|_{2\to 2}^2 \leq 1$ from Theorem 2.1 only appears below in Corollary 2.9, a special case of the following result. Furthermore, we do not yet assume $B_{\text{in}} = B_{\text{out}}$.

Theorem 2.8 *Consider the hypothesis space $\mathcal{H}_2^L$ defined in* (2.10) *and assume the samples $\boldsymbol{x}_i$, $i = 1, \dots, n$, to be drawn i.i.d. at random according to some (unknown) distribution such that $\|\boldsymbol{x}_i\|_2 \leq B_{in}$ almost surely with $B_{in} = B_{out}$ in* (2.4)*, with $\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i$. Then with probability at least $1-\delta$, for all $h \in \mathcal{H}_2^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \quad &\leq \quad \hat{\mathcal{L}}(h) + 8B_{out}\sqrt{\frac{pm}{n}}\sqrt{\log e\left(1+\frac{8K_L\|\boldsymbol{A}\|_{2\to 2}}{\sqrt{n}B_{out}}\right)} \\ &\quad +8B_{out}\frac{p}{\sqrt{n}}\sqrt{\log e\left(1+\frac{8M_L}{\sqrt{n}B_{out}}\right)} + 4(B_{in}+B_{out})\sqrt{\frac{2\log(4/\delta)}{n}},\end{aligned}$$

where K_L is the constant from (2.23) *in Theorem 2.3, and M_L is given in* (2.30).

Proof. For the proof it remains to bound the Rademacher complexity via Dudley's integral (2.21), for which in turn we use the covering number arguments from the previous subsection (Corollary 2.7) as follows,

$$\begin{aligned}\mathcal{R}_n(\ell\circ\mathcal{H}_2^L) \quad &= \quad \mathbb{E}\sup_{\boldsymbol{M}\in\mathcal{M}_2}\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\varepsilon_{ik}M_{ik} \\ &\leq \quad \frac{4\sqrt{2}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\mathcal{N}\left(\mathcal{M}_{2},, \|\cdot\|_F, \varepsilon\right)}\,\mathrm{d}\varepsilon \\ &\leq \quad \frac{4\sqrt{2}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{p^2\cdot\log\left(1+\frac{4M_L}{\varepsilon}\right)}\,\mathrm{d}\varepsilon \\ &\quad +\frac{4\sqrt{2}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{mp\cdot\log\left(1+\frac{4\|\boldsymbol{A}\|_{2\to 2}K_L}{\varepsilon}\right)}\,\mathrm{d}\varepsilon \\ &\leq \quad \frac{4\sqrt{2}p}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{4M_L}{\varepsilon}\right)}\,\mathrm{d}\varepsilon \\ &\quad +\frac{4\sqrt{2mp}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{4\|\boldsymbol{A}\|_{2\to 2}K_L}{\varepsilon}\right)}\,\mathrm{d}\varepsilon \\ &\leq \quad 2\sqrt{2}B_{\text{out}}\frac{p}{\sqrt{n}}\sqrt{\log\left(e\left(1+\frac{4M_L}{\sqrt{n}B_{\text{out}}/2}\right)\right)} \\ &\quad +2\sqrt{2}B_{\text{out}}\sqrt{\frac{pm}{n}}\sqrt{\log\left(e\left(1+\frac{4K_L\|\boldsymbol{A}\|_{2\to 2}}{\sqrt{n}B_{\text{out}}/2}\right)\right)}.\end{aligned}$$

where we have used the following inequality for the last step [FR13, Lemma C.9]

$$\int_0^\alpha \sqrt{\log\left(1+\frac{\beta}{t}\right)}\,\mathrm{d}t \leq \alpha\sqrt{\log\left(e(1+\beta/\alpha)\right)} \quad \text{for} \quad \alpha,\beta > 0.$$

The theorem is obtained using Theorem 1.9 with the upper bound $c = B_{\text{in}} + B_{\text{out}}$ for the functions output from (2.13), and bounding the Rademacher complexity term (2.9) with the generalized contraction principle Lemma B.4, which in turn is bounded using Dudleys integral as above. ■

Let us make the reasonable assumption that $\tau\|\boldsymbol{A}\|_{2\to 2} \leq 1$. Taking into account that $M_L \leq \tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F L$, see also (2.17), *i.e.*, that M_L scales at most linearly in L (which remains inside the logarithm), and since K_L depends quadratically on L, see (2.24), we have

$$\mathcal{L}(h) - \hat{\mathcal{L}}(h) \lesssim \frac{p}{\sqrt{n}}\sqrt{\log(L)} + \sqrt{\frac{pm}{n}}\sqrt{\log(L)} \sim \sqrt{\frac{\log(L)p(p+m)}{n}} \sim \sqrt{\frac{\log(L)p^2}{n}},$$

where the last relation holds under the reasonable assumption that $1 \leq m \leq p$. This estimate is stated more rigorously and with explicit constants in the following corollary.

Corollary 2.9 *In the situation of Theorem 2.8, let us assume additionally that $\tau\|\boldsymbol{A}\|_{2\to 2}^2 \leq 1$. With probability at least $1-\delta$, for all $h \in \mathcal{H}_2^L$, the generalization error is bounded as*

$$\begin{aligned} \mathcal{L}(h) \quad &\leq \quad \hat{\mathcal{L}}(h) + 8B_{out}\sqrt{\frac{pm}{n}}\sqrt{1+\log\left(2+\frac{8L(L+3)\tau\|\boldsymbol{Y}\|_F\|\boldsymbol{A}\|_{2\to 2}}{\sqrt{n}B_{out}}\right)} \\ &\quad + 8B_{out}\frac{p}{\sqrt{n}}\sqrt{\log e\left(1+\frac{8\tau L\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F}{\sqrt{n}B_{out}}\right)} + 4(B_{in}+B_{out})\sqrt{\frac{2\log(4/\delta)}{n}}. \end{aligned}$$

Proof. The statement of the corollary is obtained from Theorem 2.8 by inserting firstly $K_L \leq \tau\|\boldsymbol{Y}\|_F L(L+3)$ from (2.24) in Theorem 2.3, which holds under the assumption $\tau\|\boldsymbol{A}\|_{2\to 2}^2 \leq 1$, and by secondly inserting $M_L = \tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F L$; recall (2.30) for the definition of M_L and see also part (ii) of Lemma C.2. ■

Combining our findings so far, the main result Theorem 2.1 can be obtained easily from Theorem 2.8 and Corollary 2.9 as follows.

Proof of Theorem 2.1. Recall from (2.17) that $\tau\|\boldsymbol{A}\|_{2\to 2}\|\boldsymbol{Y}\|_F \leq \sqrt{n}B_{\text{in}}$. Further, by assumption of Theorem 2.1 we have $B_{\text{in}} = B_{\text{out}}$ and $L \geq 2$, such that $2 + 8L(L+3) \leq (5L)^2$. Therefore, the following term appearing in Corollary 2.9 has a much simpler upper bound,

$$\log\left(2+\frac{8L(L+3)\tau\|\boldsymbol{Y}\|_F\|\boldsymbol{A}\|_{2\to 2}}{\sqrt{n}B_{\text{out}}}\right) \leq \log(2+8L(L+3)) \leq 2\log(5L).$$

Plugging in this estimate and using $\mathcal{H}_1^L \subseteq \mathcal{H}_2^L$ gives the statement of Theorem 2.1. ■

2.3 LISTA: General Model

In this section we are going to derive generalization error bounds for a much more general model than the one studied in the previous section. This section is based on the paper [SBR21], which in turn builds up on the techniques previously developed by the same group of authors, including the author of this thesis, in [BRS22] .

2.3.1 A flexible ISTA model

We will now introduce a considerably more general setting, that goes far beyond the particular above example of learning an orthogonal dictionary suitable for reconstruction, but still contains it as a special case. We abandon the assumption of necessary weight-sharing between all layers. More precisely, the weight sharing can happen in any possible order, *i.e.,* between any arbitrary number of layers, appearing at any position in the neural network - in particular, weight sharing is not only possible among subsequent layers. (No weight sharing is also included.) Furthermore, we allow various additional (trainable) parameters, and include additional 1-Lipschitz operations after each soft thresholding step, such as pooling operations.

Formally, for L being the number of layers in the decoder, we introduce $J \leq L+1$ bounded parameter sets

$$\mathcal{W}^{(1)} \subset \mathbb{R}^{k_1}, \ldots, \mathcal{W}^{(J)} \subset \mathbb{R}^{k_J}, \qquad k_1, \ldots, k_J \in \mathbb{N},$$

where $\mathbb{R}^{k_j}$ is equipped with a norm $\| \cdot \|^{(j)}$. For each layer $l = 1, \ldots, L+1$ (including a final transform after the last layer), we introduce Lipschitz continuous mappings B_l (often linear) that provide the parameterization of a matrix $B_l(\boldsymbol{w}^{(j)}) \in \mathbb{R}^{m \times m_{l-1}}$ using a parameter $\boldsymbol{w}^{(j)} \in \mathcal{W}^{(j)}$, where $j = j(l)$ corresponds to the parameter set associated to the l-layer:

$$B_l : \mathcal{W}^{(j(l))} \to \mathbb{R}^{m \times m_{l-1}}, \qquad \boldsymbol{w}^{(j)} \mapsto B_l(\boldsymbol{w}^{(j)}). \tag{2.32}$$

Note that if $J = 1$, then all layers share the same weights; if $J = L+1$, there is no weight sharing and all layers, and the final transform after the last layer, have different underlying parameters. If l is either clear from the context, or not relevant, we may omit it in $j(l)$ and simply write j. If $j(l) = j(l')$ for any two different layers $l \neq l'$, the two layers share the same weights. Note that even in this situation still it may be that $B_l \neq B_{l'}$, since already the involved dimensions m_{l-1} and $m_{l'-1}$ may be different - this means that even if layers share the same *underlying parameters*, the *parameterizations* in the sense of the mappings B_l and $B_{l'}$ may still be different. (We typically denote the index refering to the parameter set as an *upper* index, and the index referring to the layer number as a *lower* index.) Let us also remark that $\mathcal{W}^{(i)} = \mathcal{W}^{(k)}$ is possible even when $i \neq k$.

To make the Lipschitz assumption precise, we require that for each $l \in [L+1]$, there exists a constant $D_l > 0$, such that

$$\|B_l(\boldsymbol{w}_1) - B_l(\boldsymbol{w}_2)\|_{2\to 2} \leq D_l \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|^{(j(l))} \qquad \forall\, \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}^{(j(l))}. \tag{2.33}$$

In order to introduce the network architecture, let $\boldsymbol{I}_k$ denote the $k \times k$ identity matrix, for some $k \in \mathbb{N}$, and S_λ the soft thresholding operator (1.10) acting componentwise. Further, we will use a 1-Lipschitz operation $P_l : \mathbb{R}^{m_{l-1}} \to \mathbb{R}^{m_L}$ such as pooling, which satisfies

$$\|P_l(\boldsymbol{z})\|_2 \leq \|\boldsymbol{z}\|_2 \qquad \forall\, \boldsymbol{z} \in \mathbb{R}^{m_{l-1}}.$$

Then $P_l \circ S_{\tau_l \lambda_l}$ is also 1-Lipschitz and norm contractive. (In many scenarios with $m_{l-1} = m_L$, P_l will simply be the identity; see also Remark 2.10 and the examples in Section 2.3.3.)

For $l = 1, \ldots, L$, and dimension (width) parameters $m_0, \ldots, m_L$, we then define the layer $f_l : \mathbb{R}^{m_{l-1}} \times \mathbb{R}^m \to \mathbb{R}^{m_L}$ as

$$f_l\,(\boldsymbol{z},\boldsymbol{y}) = P_l S_{\tau_l \lambda_l}\left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l B_l(\boldsymbol{w}^{(j(l))})^\top B_l(\boldsymbol{w}^{(j(l))})\right)\boldsymbol{z} + \tau_l B_l(\boldsymbol{w}^{(j(l))})^\top \boldsymbol{y}\right],$$

with parameter vector $\boldsymbol{w}^{(j(l))} \in \mathcal{W}^{(j(l))}$, stepsize τ_l, threshold λ_l. The input vector $\boldsymbol{y} \in \mathbb{R}^m$ may be $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m$ for some (a priori unknown) vector $\boldsymbol{x} \in \mathbb{R}^p$ in a compressive sensing scenario, but our setup allows more general regression tasks. The vector $\boldsymbol{z}$ will be initialized as $\boldsymbol{0}$ for the input of the first layer; afterwards it will be the output of the previous layer (see below). The stepsize $\tau_l > 0$ and the threshold $\lambda_l > 0$ in the soft thresholding activation function can be either trainable parameters, or fixed all the time. In the simplest case $\tau_l = \tau, \lambda_l = \lambda > 0$ are fixed and the same in each layers.

After the final layer f_L, we apply another linear transform $B_{L+1}(\boldsymbol{w}^{j(L+1)})$ followed by some function $\sigma : \mathbb{R}^{m_{L+1}} \to \mathbb{R}^{m_{L+1}}$ to be specified below, *i.e.*, ,

$$g_{L+1} : \mathbb{R}^{m_L} \to \mathbb{R}^{m_{L+1}}, \quad g_{L+1} = \sigma \circ B_{L+1}(\boldsymbol{w}^{j(L+1)}).$$

For reconstruction tasks, the function g_{L+1} projects the sparse representation onto the ambient space and controls the output norm. For technical reasons, we will require a function σ which will be applied after the final layer of the decoding networks. The function σ is assumed to be norm-contractive and norm-clipping as well as be 1-Lipschitz, *i.e.*,

$$\|\sigma(\boldsymbol{x})\|_2 \le \min\{\|\boldsymbol{x}\|_2, B_{\text{out}}\} \qquad \text{and} \qquad \|\sigma(\boldsymbol{x}_1) - \sigma(\boldsymbol{x}_2)\|_2 \le \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \tag{2.34}$$

for any $\boldsymbol{x}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{m_L}$ and some fixed constant $B_{\text{out}} > 0$. The technical reasons behind introducing σ will become apparent later in the proofs in Section 2.3.5. A typical choice for σ satisfying all requirements is

$$\sigma : \mathbb{R}^{m_{L+1}} \to \mathbb{R}^{m_{L+1}}, \qquad \boldsymbol{x} \mapsto \begin{cases} \boldsymbol{x} & \text{if } \|\boldsymbol{x}\|_2 \le B_{\text{out}}, \\ B_{\text{out}} \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} & \text{if } \|\boldsymbol{x}\|_2 > B_{\text{out}}, \end{cases} \tag{2.35}$$

The motivation for introducing σ, both regarding technical reasons and with respect to applications, is the same as in (2.4) in the previous section on the dictionary learning problem. Obviously we make essentially the same choice of σ here (and for simplicity continue to use the same notation), only taking a more general output dimension into account. (Typically, for reconstruction tasks we have $m_{L+1} = p$.) Note that for the first layer's input we have $m_0 = m$, *i.e.*, the number of measurements. A typical choice for the last layer's dimension is $m_{L+1} = p$, which corresponds to the setting of reconstruction problems; but note that our framework allows to consider different situations. Let us introduce the compact notation

$$\mathcal{W} := \mathcal{W}^{(1)} \times \cdots \times \mathcal{W}^{(J)} \subset \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_J} =: \mathcal{X}$$

for the set of K-dimensional weights $\boldsymbol{W} = (\boldsymbol{w}^{(1)}, \ldots \boldsymbol{w}^{(J)}) \in \mathcal{W}$, where K is the sum of the individual dimensions $k_j = \dim \mathcal{W}^{(J)}$, *i.e.*,

$$K := k_1 + \cdots + k_J. \tag{2.36}$$

In order to allow for learnable stepsizes and thresholds we introduce the set $\mathcal{T} \subset \mathbb{R}^L_{>0}$ of stepsize vectors $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_L)$ and the set $\Lambda \subset \mathbb{R}^L_{>0}$ of thresholding vectors $\boldsymbol{\lambda} =$

$(\lambda_1, \dots, \lambda_L)$. Then we define $f^L_{W,\tau,\lambda}$ to be the concatenation of all layers f_l,

$$f^L_{W,\tau,\lambda}(y) := f_L(\dots f_2(f_1(\mathbf{0}, y), y) \dots),$$

and the neural network – also called decoder – is obtained after an application of g_{L+1},

$$h(y) = h^L_{W,\tau,\lambda} := g_{L+1} \circ f^L_{W,\tau,\lambda}(y) = g_{L+1}(f_L(\dots f_2(f_1(\mathbf{0}, y), y) \dots)). \tag{2.37}$$

The fact that the input y is entered directly into each of the layers in addition to the input from the previous layers, may be interpreted as the network having so-called skip connections.

For the investigations in the following sections it will be convenient to view the parameter sets as subsets of normed spaces. The set $\mathcal{W}$ is contained in the K-dimensional product space $\mathcal{X} = \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_J}$, which we equip with the norm

$$\|W\|_{\mathcal{X}} := \max_{j=1,\dots,J} \|w^{(j)}\|^{(j)} \qquad \text{for} \qquad W = \left(w^{(1)}, \dots, w^{(J)}\right) \in \mathcal{X}, \tag{2.38}$$

where we recall that $\|\cdot\|^{(j)}$ is the norm on $\mathbb{R}^{k_j}$ used in (2.33). Denoting $B^L_{\|\cdot\|_\infty} = \{\tau \in \mathbb{R}^L : \|\tau\|_\infty \leq 1\}$ the unit ℓ_∞-ball, we assume that the set $\mathcal{T}$ of stepsizes and the Λ of thresholds are contained in shifted ℓ_∞-balls of radii r_1 and r_2, *i.e.*,

$$\mathcal{T} \subset \tau_0 + r_1 B^L_{\|\cdot\|_\infty} \qquad \Lambda \subset \lambda_0 + r_2 B^L_{\|\cdot\|_\infty}. \tag{2.39}$$

Setting $r_1 = r_2 = 0$ corresponds to the case of fixed stepsizes and thresholds while choosing $r_1, r_2 > 0$ corresponds to learned stepsizes and thresholds. The above conditions require that $\tau_j \in [\tau_{0,j} - r_1, \tau_{0,j} + r_1]$ for all $\tau \in \mathcal{T}$ and $\lambda_j \in [\lambda_{0,j} - r_1, \lambda_{0,j} + r_2]$ for all $\lambda \in \Lambda$. Recalling that $\mathcal{W}$ is assumed to be bounded, we can introduce the parameters

$$B_\infty := \sup_{\substack{W \in \mathcal{W} \\ l \in [L+1]}} \left\| B_l(w^{(j(l))}) \right\|_{2\to 2}, \qquad W_\infty := \sup_{W \in \mathcal{W}} \|W\|_{\mathcal{X}}, \tag{2.40}$$

$$\tau_\infty := \sup_{\tau < \in \mathcal{T}} \|\tau\|_\infty, \qquad \lambda_\infty := \sup_{\lambda \in \Lambda} \|\lambda\|_\infty. \tag{2.41}$$

Note that if the Lipschitzness assumption (2.33) on the mappings B_l (2.32), it holds that

$$B_\infty \leq W_\infty \max_{l \in [L+1]} D_l \leq W_\infty D_\infty, \qquad D_\infty := \max_{l \in [L+1]} D_l. \tag{2.42}$$

Moreover, there is $\tau_\infty \leq \|\tau_0\|_\infty + r_1$ and $\lambda_\infty \leq \|\lambda_0\|_\infty + r_2$ according to assumption (2.39).

Remark 2.10 Let us add some comments to motivate the setup above. Allowing weight-sharing between layers can be easily motivated, for instance, through the dictionary learning problem considered in Section 2.2. However, weight-sharing between the stepsizes and thresholds seems less realistic, which is why we consider different τ_l and λ_l for each layer, $l \in [L]$. It is possible to generalize this even further by training these parameters entrywise. However, to prevent the presentation from becoming even more technical, we focus on the problem at hand. $\Diamond$

Using the concepts and notation introduced above, given a number L of layers, we define our hypothesis space as the parameterized set of all h (see (2.37) for the definition of h), *i.e.*,

$$\mathcal{H}^L := \{h : \mathbb{R}^m \to \mathbb{R}^{m_{L+1}} \mid h = h^L_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}},\ \boldsymbol{W} \in \mathcal{W},\ \boldsymbol{\tau} \in \mathcal{T},\ \boldsymbol{\lambda} \in \Lambda\}, \tag{2.43}$$

Note that once again we will employ the loss function (2.1) and make use of its 1-Lipschitzness, and rely on the notion of the generalization error as in (1.15).

2.3.2 Main Result

Our main result uses the setup and the notation introduced in Section 2.3.1. In order to state it, we additionally introduce the following quantities, where we recall that B_∞, W_∞, τ_∞ and λ_∞ are defined in (2.40) and (2.41), the dimension K of the parameter set of weights in (2.36), and D_∞ in (2.42). We set

$$\alpha = \sup_{l\in[L]} \sup_{\boldsymbol{w}^{(j(l))}\in\mathcal{W}^{(j(l))}} \sup_{\boldsymbol{\tau}\in\mathcal{T}} \left\| \boldsymbol{I}_{m_{l-1}} - \tau_l B_l(\boldsymbol{w}^{(j(l))})^\top B_l(\boldsymbol{w}^{(j(l))}) \right\|_{2\to 2}, \tag{2.44}$$

and define $Z_0 = 0$,

$$Z_l = \tau_\infty B_\infty \sum_{k=0}^{l-1} \alpha^k = \begin{cases} \tau_\infty B_\infty \alpha \frac{1-\alpha^{(l-1)}}{1-\alpha} & \text{if } \alpha \neq 1 \\ \tau_\infty B_\infty (l-1) & \text{if } \alpha = 1 \end{cases} \quad l = 1, \dots, L. \tag{2.45}$$

Using this definition of Z_l, let us introduce the further abbreviations M_L, O_L, Q_L given by

$$M_L = \sum_{l=1}^{L} \left(\lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F (B_\infty Z_{l-1} + 1)\right) \alpha^{L-l}, \tag{2.46}$$

$$O_L = \sum_{l=1}^{L} \tau_\infty \sqrt{m_\infty n} \alpha^{L-l}, \tag{2.47}$$

$$Q_L = (B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L) D_\infty, \tag{2.48}$$

where K_L in the definition (2.48) of Q_L is given as follows by

$$K_L = \sum_{l=1}^{L} \tau_\infty \|\boldsymbol{Y}\|_F \left(1 + 2B_\infty Z_{l-1}\right) \alpha^{L-l}. \tag{2.49}$$

We assume that the data distribution $\mathcal{D}$ is such that for $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}$

$$\|\boldsymbol{y}\|_2 \leq B_{\text{in}} \quad \text{almost surely}$$

for some constant B_{in}. In particular, $\|\boldsymbol{y}_i\|_2 \leq B_{\text{in}}$ for all $i = 1, \dots, n$ (with probability 1). Furthermore, we require the function

$$\Psi(t) = \sqrt{\log(1+t) + t(\log(1+t) - \log(t))}, \quad t > 0, \quad \Psi(0) = 0. \tag{2.50}$$

Note that the function Ψ is continuous in $t = 0$, and satisfies the bound $\Psi(t) \leq \sqrt{\log(e(1+t))}$, see below in Lemma A.5. Our main theorem reads as follows.

Theorem 2.11 *Consider the hypothesis space $\mathcal{H}^L$ defined in* (2.43). *With probability at least*

$1-\delta$, the true risk for any $h \in \mathcal{H}^L$ is bounded as

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + 2\sqrt{2}\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) + 4(B_{in} + B_{out})\sqrt{\frac{2\log(4/\delta)}{n}}, \tag{2.51}$$

where the Rademacher complexity term is further bounded by

$$\begin{aligned}&\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) \\ \leq& 2\sqrt{2}B_{out}\left[\sqrt{\frac{K}{n}}\Psi\left(\frac{16W_\infty Q_L}{\sqrt{n}B_{out}}\right) + \sqrt{\frac{L}{n}}\Psi\left(\frac{16r_2 O_L}{\sqrt{n}B_{out}}\right) + \sqrt{\frac{L}{n}}\Psi\left(\frac{16r_1 M_L}{\sqrt{n}B_{out}}\right)\right].\end{aligned} \tag{2.52}$$

Remark 2.12 In the special case that the stepsizes $\boldsymbol{\tau}_0 \in \mathbb{R}^L$ and/or the thresholds $\boldsymbol{\lambda}_0 \in \mathbb{R}^L$ are fixed, so that $r_1 = 0$ and/or $r_2 = 0$, the above bound simplifies due to $\Psi(0) = 0$. For instance, if $r_1 = r_2 = 0$ then

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) \leq 2\sqrt{2}B_{\mathrm{out}}\sqrt{\frac{K}{n}}\Psi\left(\frac{16W_\infty Q_L}{\sqrt{n}B_{\mathrm{out}}}\right),$$

greatly simplifying the bound (2.52) in Theorem 2.11 above. ♢

While the constants M_L, O_L and Q_L look complicated in general and may actually scale exponentially in L, the expressions greatly simplify in the important special case that $\alpha \leq 1$. In fact, the motivating algorithm ISTA, corresponding to fixing $\boldsymbol{A\Phi}$ in (2.2) and letting $L \to \infty$, is known to converge under the condition that $\tau\|\boldsymbol{A\Phi}\|_{2\to 2} \leq 1$ (compare (1.11) in the introductury chapter) implying that $\|\boldsymbol{I} - \tau(\boldsymbol{A\Phi})^\top(\boldsymbol{A\Phi})\|_{2\to 2} \leq 1$; see Lemma C.2. These conditions correspond to $\tau_\infty B_\infty^2 \leq 1$ and $\alpha \leq 1$ in our general setup. This suggests to impose these condition on the hypothesis space (and therefore in the training of the network). The corresponding generalization result reads as follows.

Corollary 2.13 *Assume that $\tau_\infty B_\infty^2 \leq 1$, implying $\alpha \leq 1$. Set $m_\infty = \max_{l\in[L]} m_l$. Then the Rademacher complexity term in* (2.51) *is bounded by*

$$\begin{aligned}\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) \leq& 2\sqrt{2}B_{out}\left[\sqrt{\frac{K}{n}\log\left(e\left(1 + 16L(L+1)\tau_\infty B_\infty W_\infty D_\infty \frac{B_{in}}{B_{out}}\right)\right)}\right. \\ &+ \sqrt{\frac{L}{n}}\Psi\left(\frac{16r_2 L\tau_\infty\sqrt{m_\infty}}{B_{out}}\right) \\ &\left.+ \sqrt{\frac{L}{n}}\Psi\left(\frac{16r_1 L\left(\lambda_\infty^2 m_\infty\sqrt{n} + B_{in}(B_\infty\lambda_\infty\sqrt{m_\infty n} + (L-1)/2)\right)}{B_{out}}\right)\right].\end{aligned} \tag{2.53}$$

Proof. Note that $\|\boldsymbol{Y}\|_F \leq \sqrt{n}B_{\mathrm{in}}$. Under our assumptions, the constant K_L satisfies

$$\begin{aligned}K_L &= \sum_{l=1}^{L}\tau_\infty\|\boldsymbol{Y}\|_F\left(1 + 2B_\infty Z_{l-1}\right)\alpha^{L-l} \leq \sum_{l=1}^{L}\tau_\infty\|\boldsymbol{Y}\|_F\left(1 + 2(l-1)\tau_\infty B_\infty^2\right) \\ &\leq \tau_\infty\|\boldsymbol{Y}\|_F\left(L + 2\sum_{l=1}^{L}(l-1)\right) = \tau_\infty\|\boldsymbol{Y}\|_F\left(L + L(L-1)\right) = \tau_\infty\|\boldsymbol{Y}\|_F L^2 \leq \tau_\infty L^2\sqrt{n}B_{\mathrm{in}}.\end{aligned}$$

Hence,

$$Q_L = (B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L)D_\infty \leq \left[L^2\tau_\infty B_\infty\sqrt{n}B_{\mathrm{in}} + \sqrt{n}B_{\mathrm{in}}L\tau_\infty B_\infty\right]D_\infty$$

$$= L(L+1)\tau_\infty B_\infty D_\infty \sqrt{n} B_{\text{in}}.$$

For the constant O_L, we obtain $O_L = \sum_{l=1}^{L} \tau_\infty \sqrt{m_\infty n} \alpha^{L-l} \leq L\tau_\infty \sqrt{m_\infty n}$. The constant M_L satisfies

$$\begin{aligned} M_L &= \sum_{l=1}^{L} \left(\lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F (B_\infty Z_{l-1} + 1)\right) \alpha^{L-l} \\ &\leq L(\lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F) \lambda_\infty \sqrt{m_\infty n} + \sum_{l=1}^{L} \|\boldsymbol{Y}\|_F \tau_\infty B_\infty^2 (l-1) \\ &\leq L(\lambda_\infty \sqrt{m_\infty n} + B_\infty \sqrt{n} B_{\text{in}}) \lambda_\infty \sqrt{m_\infty n} + \sqrt{n} B_{\text{in}} \frac{L(L-1)}{2}. \end{aligned}$$

Plugging the above bounds into (2.52) and using that $\Psi(t) \leq \sqrt{\log(e(1+t))}$ completes the proof. ■

Note that in case the mappings B_l are linear it follows from $B_\infty \leq W_\infty$, see (2.42), that the assumption $\tau_\infty B_\infty^2 \leq 1$ is implied by $\tau_\infty B_\infty W_\infty D_\infty \leq 1$. Additionally assuming $B_{\text{in}} = B_{\text{out}}$, the first logarithmic term in (2.53) takes the simple form $\log(e(1 + 16L(L+1)))$.

In general, considering only the dependence in K, L and n and viewing all other terms as constants, the bound of Corollary (2.13) essentially reads as

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) \lesssim \sqrt{\frac{(K+L)\log(L)}{n}}.$$

Moreover, if the thresholds and stepsizes are fixed (not learned), so that $r_1 = r_2 = 0$, we obtain the bound

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) \lesssim \sqrt{\frac{K\log(L)}{n}}.$$

This is one of the main messages of our result: The dependence of the generalization error on the number of layers is only logarithmic in important cases in contrast to many previous results on the generalization error for deep learning, where the scaling in the number of layers is often exponential. It is furthermore interesting to compare with our findings from the previous section, as stated in the main result Theorem 2.1, and the discussion thereafter. Here, the price to pay for the more general setup is a possible linear dependence of the generalization error on the number of layers, while in important special we fall back to the logarithmic behavior observed above. Nevertheless, we can avoid loose bounds with an exponential dependence on the number of layers that naive approaches would produce.

2.3.3 Examples

Let us illustrate this general scenario considered in this section with a few different examples of practical interest. We apply our general main result to the specific situations.

Learning an orthogonal dictionary. Let us start by demonstrating how to recover the dictionary learning problem from Section 2.2. Here, we choose $J = 1$ (thus, $j(l) = 1$ for all l) and

$$\mathcal{W}^{(1)} = \{\boldsymbol{\Phi} : \boldsymbol{\Phi} \in O(p)\} \subset \mathbb{R}^{p \times p} \simeq \mathbb{R}^{k_1},$$

$$B_l(\mathbf{\Phi}) = \mathbf{A}\mathbf{\Phi}, \quad l = 1, \ldots, L, \quad B_{L+1}(\mathbf{\Phi}) = \mathbf{\Phi}, \tag{2.54}$$

where all dimensions $m_1 = m_2 = \cdots = m_L = p$ are equal, $k_1 = p^2$, and with $\tau_l = \tau$ and $\lambda_l = \lambda$ being fixed. We simply put $P_l = \mathbf{I}_p$ for all $l \in [L]$. Let us put $\| \cdot \|^{(1)} = \| \cdot \|_{2\to 2}$, so that for all $\mathbf{\Phi} \in O(p)$ and $l \in [L]$ we have

$$\|B_l(\mathbf{\Phi})\|_{2\to 2} = \|\mathbf{A}\mathbf{\Phi}\|_{2\to 2} \leq \|\mathbf{A}\|_{2\to 2}\|\mathbf{\Phi}\|^{(1)},$$

so that $D_l = \|\mathbf{A}\|_{2\to 2}$ for all $l \in [L]$ due to the linearity of B_l. Moreover, $\|B_l(\mathbf{\Phi})\|_{2\to 2} = \|\mathbf{\Phi}\|_{2\to 2} = \|\mathbf{\Phi}\|^{(1)}$ resulting in $D_{L+1} = 1$ and $B_\infty = W_\infty = D_\infty = \max\{1, \|\mathbf{A}\|_{2\to 2}\}$.

Assuming that $\tau \max\{1, \|\mathbf{A}\|_{2\to 2}\} \leq 1$ and considering that the thresholds and stepsizes are fixed, Corollary 2.13 states that the generalization error bound scales like (with high probability)

$$C\sqrt{\frac{p^2 \log(L)}{n}} \tag{2.55}$$

for a constant C depending on $\tau, \|\mathbf{A}\|_{2\to 2}, B_{\text{in}}, B_{\text{out}}$, as we have already shown (only for this specific case) in our previous work [BRS22] (noting that $p + m \asymp p$).

Overcomplete dictionaries. As another important class of dictionaries, let us consider overcomplete dictionaries. This case is similar to the previous one, but here we consider

$$\mathcal{W}^{(1)} = \{\mathbf{\Phi} : \mathbf{\Phi} \in \mathbb{R}^{p\times d}, \|\mathbf{\Phi}\|_{2\to 2} \leq \rho\} \subset \mathbb{R}^{m\times d} \simeq \mathbb{R}^{k_1}, \qquad d > p > m, \qquad \rho > 0,$$

with $k_1 = p \cdot d > m \cdot p$. The mappings B_l are defined as in (2.54). We have the input/output dimensions $m_0 = m_1 = m_2 = \cdots = m_{L-1} = d$, and $m_L = p$. We use again $\| \cdot \|^{(1)} = \| \cdot \|_{2\to 2}$, which, as above, leads to $D_\infty = B_\infty = W_\infty = \max\{\rho\|\mathbf{A}\|_\infty, 1\}$. Assuming constant stepsizes and thresholds and $\tau B_\infty^2 \leq 1$, Corollary 2.13 leads to a generalization bound scaling like

$$C\sqrt{\frac{pd \log(L)}{n}}.$$

This is slightly worse than for orthonormal dictionaries due to $p \geq d$.

Two (alternating) dictionaries. Similar to the first two examples, one may consider two "alternating" dictionaries. In case of orthogonal dictionaries, similar to the first example, we have $J = 2$ (thus, $j(l) = 1$ for all l) and

$$\mathcal{W}^{(1)} = \{\mathbf{\Phi}_1 : \mathbf{\Phi}_1 \in O(p)\} \subset \mathbb{R}^{p\times p} \simeq \mathbb{R}^{k_1}, \quad \mathcal{W}^{(2)} = \{\mathbf{\Phi}_2 : \mathbf{\Phi}_2 \in O(p)\} \subset \mathbb{R}^{p\times p} \simeq \mathbb{R}^{k_2}.$$

The mappings B_l are defined as in (2.54), but, for odd l, B_l operates on $\mathcal{W}^{(1)}$, while for even l it operates on $\mathcal{W}^{(2)}$. Here, $K = k_1 + k_2 = 2p^2$, which results in an additional factor of $\sqrt{2}$ appearing in (2.55). Analogously, one may obtain bounds for two alternating overcomplete dictionaries, or more than two alternating dictionaries, or other related scenarios. For example, we can consider the case without any weight-sharing between layers, *i.e.*, where $J = L$. Then $j(l) = l, k_j = p^2$ for all $j = 1, \ldots, N$ and $K = k_1 + \cdots + k_L = Lp^2$. This leads to an additional factor of $\sqrt{L}$. Of course, this may also be combined with trainable stepsizes and thresholds.

Convolutional LISTA. For input images, one natural choice of weight matrices is convolutional kernels. In this model, the layer l contains the following operation

$$B_l(\boldsymbol{w})(\boldsymbol{z}) = \boldsymbol{\Omega}(\boldsymbol{w}) * \boldsymbol{z},$$

where the length of the convolutional filter $\boldsymbol{w}_j$ is k_j, and the mapping $\boldsymbol{\Omega} : \mathbb{R}^{k_j} \to \mathbb{R}^p$ is its zero-padded version. This model is discussed in [SG18a]. Since K is merely dependent on the number of parameters k_j and not p, our result already shows that smaller convolutional filters lead to smaller overall K and therefore are expected to show better generalization.

2.3.4 Bounding the Rademacher Complexity via Dudley's Integral

Again, we employ the loss function ℓ from (2.1), and the same boundedness assumptions as in the previous section, such that

$$\ell(h, \boldsymbol{y}, \boldsymbol{x}) = \|h(\boldsymbol{y}) - \boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_2 + \|h(\boldsymbol{y})\|_2 \leq B_{\text{in}} + B_{\text{out}}.$$

Thus, with this choice of the loss function, again the main challenge and focus of this section is to bound the Rademacher complexity of $\ell \circ \mathcal{H}$,

$$\mathcal{R}_{\mathcal{S}}(\ell \circ \mathcal{H}) = \mathbb{E} \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i \left\| \boldsymbol{x}_i - h(\boldsymbol{y}_i) \right\|_2, \tag{2.56}$$

with $\mathcal{H} = \mathcal{H}^L$ as defined in (2.43) in the most general case studied here, or some other hypothesis spaces of interest. We proceed analogously to the previous section. For a fixed number $L \in \mathbb{N}$ of layers, and given a hypothesis space $\mathcal{H}$ of functions mapping from $\mathbb{R}^m$ to $\mathbb{R}^{m_{L+1}}$ (with $m_{L+1} = p$ for reconstruction tasks) let us define the set $\mathcal{M}_{\mathcal{H}} \subset \mathbb{R}^{m_{L+1} \times n}$ as

$$\mathcal{M}_{\mathcal{H}} := \left\{ [h(\boldsymbol{y}_1), \ldots, h(\boldsymbol{y}_n))] \in \mathbb{R}^{m_{L+1} \times n} : h \in \mathcal{H} \right\}.$$

From now on, we focus on the hypothesis space $\mathcal{H} = \mathcal{H}^L$, when the corresponding set is given by (using the compact matrix notation)

$$\mathcal{M}_{\mathcal{H}^L} = \left\{ h^L_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y}) \in \mathbb{R}^{m_{L+1} \times n} : \boldsymbol{W} \in \mathcal{W}, \boldsymbol{\tau} \in \mathcal{T}, \boldsymbol{\lambda} \in \Lambda, \right\}. \tag{2.57}$$

In words, $\mathcal{M}_{\mathcal{H}^L}$ is the set consisting of all matrices whose columns are the outputs of any possible hypothesis applied to the measurements $\boldsymbol{y}_i$. In the compressive sensing scenario, these are the reconstructions from the measurements in the training set, using any possible decoder in our hypothesis space. If the hypothesis space is clear from the context, we write $\mathcal{M}$ instead of $\mathcal{M}_{\mathcal{H}^L}$. In the case $\mathcal{H} = \mathcal{H}^L$, the set $\mathcal{M}$ is parameterized by $(\boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{W}) \in \mathcal{T} \times \Lambda \times \mathcal{W}$ (as $\mathcal{H}^L$ is). In this case, applying Lemma B.4 to (2.56) and rewriting the expression using (2.57), we obtain

$$\begin{aligned}
\mathcal{R}_{\mathcal{S}}(\mathcal{H}^L) &\leq \sqrt{2} \mathbb{E} \sup_{M \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} M_{ik} \\
&= \sqrt{2} \mathbb{E} \sup_{\boldsymbol{\tau} \in \mathcal{T}} \sup_{\boldsymbol{\lambda} \in \Lambda} \sup_{\boldsymbol{W} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{m_L} \varepsilon_{ik} \left(h^L_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y}) \right)_{ik}.
\end{aligned} \tag{2.58}$$

Analogously to (2.20) and thereafter, it can be shown that the Rademacher process under consideration has sub-Gaussian increments, and therefore, we can apply Dudley's inequality. For the set of matrices $\mathcal{M}$ defined in (2.57), the radius can be estimated as

$$\begin{aligned}\Delta(\mathcal{M}) &= \sup_{h\in\mathcal{H}^L}\sqrt{\mathbb{E}\left(\sum_{i=1}^{n}\sum_{k=1}^{m_L}\varepsilon_{ik}h_k(\boldsymbol{x}_i)\right)^2} = \sup_{h\in\mathcal{H}^L}\sqrt{\sum_{i=1}^{n}\sum_{k=1}^{m_L}h_k(\boldsymbol{x}_i)^2}\\ &= \sup_{h\in\mathcal{H}^L}\sqrt{\sum_{i=1}^{n}\|h(\boldsymbol{x}_i)\|_2^2} \leq \sqrt{n}B_{\text{out}},\end{aligned}$$

where the last inequality follows from the properties of the function σ; see (2.35). Dudley's inequality, as stated in [FR13, Theorem 8.23.], then bounds the Rademacher complexity as

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}) \leq \frac{4\sqrt{2}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\mathcal{N}(\mathcal{M},\|\cdot\|_F,\varepsilon)}\mathrm{d}\varepsilon. \tag{2.59}$$

To derive the generalization bound, it essentially suffices to bound the covering numbers of $\mathcal{M}$. All technical details are provided in the next subsection.

2.3.5 Proof

In this subsection, we will prove the main result Theorem 2.11. The proof is an adaption of the proof strategy seen in the previous section to the more general setup studied here. Analogously to the previous section, the proof is split into several steps.

Bounding the output. As a first auxiliary tool, we prove a bound for the output of the network, after any number of (possibly intermediate) layers l, in the next lemma. We state a general version which allows possibly different stepsizes and thresholds for each layer. It is straightforward to obtain special cases such as Lemma 2.2 from this scenario.

Lemma 2.14 *For $l = 1,\dots,L$ and $\boldsymbol{W} = \left(\boldsymbol{w}^{(1)},\dots\boldsymbol{w}^{(J)}\right) \in \mathcal{W}$, $\boldsymbol{\tau} = (\tau_1,\dots,\tau_l) \in \mathcal{T}$ and $\boldsymbol{\lambda} = (\lambda_1,\dots,\lambda_l) \in \Lambda$, we have*

$$\begin{aligned}&\left\|f^l_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y})\right\|_F\\ \leq&\sum_{k=1}^{l}\left(\left\|\tau_k B_k(\boldsymbol{w}^{(j(k))})^\top\boldsymbol{Y}\right\|_F\prod_{i=k}^{l-1}\left\|\boldsymbol{I}_{m_i}-\tau_i B_{i+1}(\boldsymbol{w}^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}^{(j(i+1))})\right\|_{2\to2}\right) \qquad (2.60)\\ \leq&\|\boldsymbol{Y}\|_F\sum_{k=1}^{l}\left(\tau_k\left\|B_k(\boldsymbol{w}^{(j(k))})\right\|_{2\to2}\prod_{i=k}^{l-1}\left\|\boldsymbol{I}_{m_i}-\tau_i B_{i+1}(\boldsymbol{w}^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}^{(j(i+1))})\right\|_{2\to2}\right),\end{aligned}$$

following the usual convention of defining the empty product as one.

Proof. We just prove the first inequality (2.60), from which the last lines then follows immediately. We proceed via induction. For $l = 1$, using the norm contractivity of P_l and of the soft thresholding operator we obtain

$$\left\|f^1_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y})\right\|_F = \left\|P_1 S_{\tau_1,\lambda_1}\left(\tau B_1(\boldsymbol{w}^{(j(1))})^\top\boldsymbol{Y}\right)\right\|_F \leq \left\|\tau_1 B_1(\boldsymbol{w}^{(j(1))})^\top\boldsymbol{Y}\right\|_F.$$

Assuming the statement is true for l, we obtain

$$\left\|f^{l+1}_{\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y})\right\|_F$$

$$
\begin{aligned}
&\leq \left\| \boldsymbol{I}_{m_l} - \tau_{l+1} B_{l+1}(\boldsymbol{w}^{(j(l+1))})^\top B_{l+1}(\boldsymbol{w}^{(j(l+1))}) \right\|_{2\to 2} \left\| f^l_{\mathbf{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y}) \right\|_F + \left\| \tau_{l+1} B_{l+1}(\boldsymbol{w}^{(j(l+1))})^\top \boldsymbol{Y} \right\|_F \\
&\leq \sum_{k=1}^{l} \left(\left\| \tau_k B_k(\boldsymbol{w}^{(j(k))})^\top \boldsymbol{Y} \right\|_F \prod_{i=k}^{l} \left\| \boldsymbol{I}_{m_i} - \tau_{i+1} B_{i+1}(\boldsymbol{w}^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}^{(j(i+1))}) \right\|_{2\to 2} \right) \\
&\qquad + \left\| \tau_{l+1} B_{l+1}(\boldsymbol{w}^{(j(l+1))})^\top \boldsymbol{Y} \right\|_F \\
&\leq \sum_{k=1}^{l+1} \left(\left\| \tau_k B_k(\boldsymbol{w}^{(j(k))})^\top \boldsymbol{Y} \right\|_F \prod_{i=k}^{l} \left\| \boldsymbol{I}_{m_i} - \tau_{i+1} B_{i+1}(\boldsymbol{w}^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}^{(j(i+1))}) \right\|_{2\to 2} \right).
\end{aligned}
$$

Indeed, this is the claimed inequality for $l+1$, completing the proof by induction. ■

We immediately obtain the coming corollary bounding the output of the full network.

Corollary 2.15 *For any $h = h_{\mathbf{W},\boldsymbol{\tau},\boldsymbol{\lambda}} \in \mathcal{H}^L$, the output is bounded with respect to the Frobenius norm by*

$$
\|h(\boldsymbol{Y})\|_F = \left\| \sigma\left(B_{L+1}(\boldsymbol{w}^{(j(L+1))}) f^L_{\mathbf{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y}) \right) \right\|_F \leq B_\infty \left\| f^L_{\mathbf{W},\boldsymbol{\tau},\boldsymbol{\lambda}}(\boldsymbol{Y}) \right\|_F.
$$

Proof. The statement follows immediately from the fact that σ, as being defined in (2.35), is norm-contractive (2.34). ■

Perturbation argument. In this section we prove our main result Theorem 2.11. We consider the most general scenario introduced in Section 2.3.1 with $\mathcal{H} = \mathcal{H}^L$ defined in (2.43). The main ingredient for bounding the covering numbers of $\mathcal{M}$, as is required to continue from (2.59) on, will be Lipschitz estimates of the neural networks with respect to the parameters, *i.e.,* bounds for (again using the compact matrix notation)

$$
\left\| f^L_{\boldsymbol{\tau}^{(1)},\boldsymbol{\lambda}^{(1)},\mathbf{W}_1}(\boldsymbol{Y}) - f^L_{\boldsymbol{\tau}^{(2)},\boldsymbol{\lambda}^{(2)},\mathbf{W}_2}(\boldsymbol{Y}) \right\|_F,
$$

with respect to the differences of the individual involved parameters (for $l = 1, \ldots, L$)

$$
\left| \lambda_l^{(2)} - \lambda_l^{(1)} \right|, \qquad \left| \tau_l^{(2)} - \tau_l^{(1)} \right|, \qquad \left\| B_l(\boldsymbol{w}_1^{(j(l))}) - B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2}.
$$

Here $\boldsymbol{\tau}^{(i)}$, $\boldsymbol{\lambda}^{(i)}$ and $\mathbf{W}_i$ denote the different stepsizes, thresholds and parameters for the B_l functions for $i = 1, 2$. To shorten the notation in the following, we will summarize the respective parameters in a vector $\mathcal{P}$ and write $f^L_{\mathcal{P}}(\boldsymbol{Y})$ and $h^L_{\mathcal{P}}(\boldsymbol{Y})$.

Let us note that the upper bounds provided in Lemma 2.14 do not depend on the threshold λ_l. However, this is not the case anymore when it comes to the perturbation bound. It is easy to verify that $|S_{\tau_2\lambda_2}(x) - S_{\tau_1\lambda_1}(x)| \leq |\tau_2\lambda_2 - \tau_1\lambda_1|$ for arbitrary $x \in \mathbb{R}$ and $\tau_1, \tau_2, \lambda_1, \lambda_2 > 0$. This implies that, for a vector $\boldsymbol{x} \in \mathbb{R}^p$, we have

$$
\|S_{\tau_2\lambda_2}(\boldsymbol{x}) - S_{\tau_1\lambda_1}(\boldsymbol{x})\|_2 \leq \sqrt{p}\, |\tau_2\lambda_2 - \tau_1\lambda_1|
$$

and more generally (see Lemma C.1 in the appendix), for a matrix $\boldsymbol{X} \in \mathbb{R}^{p\times n}$,

$$
\|S_{\tau_2\lambda_2}(\boldsymbol{X}) - S_{\tau_1\lambda_1}(\boldsymbol{X})\|_F \leq \sqrt{np}\, |\tau_2\lambda_2 - \tau_1\lambda_1|. \tag{2.61}
$$

To simplify the notation further, let us introduce the following quantities

$$
\xi_l := \left| \tau_l^{(2)}\lambda_l^{(2)} - \tau_l^{(1)}\lambda_l^{(1)} \right| \leq \tau_\infty \left| \lambda_l^{(2)} - \lambda_l^{(1)} \right| + \lambda_\infty \left| \tau_l^{(2)} - \tau_l^{(1)} \right| \tag{2.62}
$$

$$
\begin{aligned}
\delta_l &:= \left\| \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))}) - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} && (2.63)\\
&\leq B_\infty \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \tau_\infty \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} && (2.64)\\
\gamma_l &:= \Big\| \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) && (2.65)\\
&\qquad - \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) \Big\|_F .
\end{aligned}
$$

The given estimates for ξ_l and δ_l provided immediately after their definition follow easily from the triangle inequality and the definition of τ_∞, λ_∞ and B_∞ in (2.41) and (2.40). The following lemma provides a useful bound also for the quantity γ_l.

Lemma 2.16 *For any $l \in [L]$ and γ_l as being defined in* (2.65), *it holds that*

$$
\begin{aligned}
\gamma_l \leq{}& 2\tau_\infty B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2}\\
&+ B_\infty^2 \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \alpha \left\| f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) - f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F .
\end{aligned}
$$

Proof. We obtain

$$
\begin{aligned}
&\Big\| \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) - \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) \Big\|_F\\
\leq{}& \Big\| \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) - \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y})\\
&+ \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) - \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y})\\
&+ \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) - \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) \Big\|_F\\
\leq{}& \Big\| \Big(\tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \Big\|_F\\
&+ \Big\| \Big(\tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \Big\|_F\\
&+ \Big\| \Big(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \Big) \Big(f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) - f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \Big) \Big\|_F\\
\leq{}& \tau_\infty B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} + \delta_l B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F\\
&+ \left\| \boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} \left\| f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) - f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F\\
\leq{}& \tau_\infty B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} + \alpha \left\| f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) - f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F\\
&+ \Big(B_\infty \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \tau_\infty \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} \Big) B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F\\
={}& 2\tau_\infty B_\infty \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} + \alpha \left\| f_{\mathcal{P}_2}^{l-1}(\boldsymbol{Y}) - f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F\\
&+ B_\infty^2 \left\| f_{\mathcal{P}_1}^{l-1}(\boldsymbol{Y}) \right\|_F \left| \tau_l^{(1)} - \tau_l^{(2)} \right| .
\end{aligned}
$$

Hereby, we have used the estimate (2.64) for δ_l and a simple estimate by α, eq. (2.44). ■

Next we state our main technical result, which will be a key ingredient for the covering number estimate, and thus for deriving the generalization bounds. It bounds the perturbation of the output of a network with respect to changes in the parameters.

Theorem 2.17 *Consider the functions $f_{\tau,\lambda,W}$ as defined in* (2.3.1) *with $L \geq 2$. Then, for any* *two such functions parameterized by* $\left(\boldsymbol{\tau}^{(1)}, \boldsymbol{\lambda}^{(1)}, \boldsymbol{W}_1\right), \left(\boldsymbol{\tau}^{(2)}, \boldsymbol{\lambda}^{(2)}, \boldsymbol{W}_2\right) \in \mathcal{T} \times \Lambda \times \mathcal{W}$ *we have*

$$\begin{aligned} &\left\| f^L_{\boldsymbol{\tau}^{(1)},\boldsymbol{\lambda}^{(1)},\boldsymbol{W}_1}(\boldsymbol{Y}) - f^L_{\boldsymbol{\tau}^{(2)},\boldsymbol{\lambda}^{(2)},\boldsymbol{W}_2}(\boldsymbol{Y}) \right\|_F \\ \leq & K_L \cdot \max_{l\in[L]} \left\| B_l(\boldsymbol{w}_1^{(j(l))}) - B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} + M_L \cdot \left\| \boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(2)} \right\|_\infty + O_L \cdot \left\| \boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)} \right\|_\infty, \end{aligned} \tag{2.66}$$

with K_L, O_L and M_L all being defined before Theorem 2.11 in (2.49), (2.47) *and* (2.46).

Proof. For the sake of avoiding to treat the case $l = 1$ separately, we formally introduce the notation $f^0_{\mathcal{P}_1}(\boldsymbol{Y}) = f^0_{\mathcal{P}_2}(\boldsymbol{Y}) = \boldsymbol{Y}$. As a first step, using that P_l is 1-Lipschitz in the first inequality and basic properties of the involved norms in the second inequality, and applying (2.61) for the third inequality, we obtain

$$\begin{aligned} &\left\| f^l_{\mathcal{P}_1}(\boldsymbol{Y}) - f^l_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F \\ =&\Big\| P_l S_{\tau_l^{(1)}\lambda_l^{(1)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \right] \\ &\quad - P_l S_{\tau_l^{(2)}\lambda_l^{(2)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) + \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top \boldsymbol{Y} \right] \Big\|_F \\ \leq &\Big\| S_{\tau_l^{(1)}\lambda_l^{(1)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathbf{W}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \right] \\ &\quad - S_{\tau_l^{(2)}\lambda_l^{(2)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) + \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top \boldsymbol{Y} \right] \Big\|_F \\ \leq &\Big\| S_{\tau_l^{(1)}\lambda_l^{(1)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \right] \\ &\quad - S_{\tau_l^{(2)}\lambda_l^{(2)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \right] \Big\|_F \\ &\quad + \Big\| S_{\tau_l^{(2)}\lambda_l^{(2)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \right] \\ &\quad - S_{\tau_l^{(2)}\lambda_l^{(2)}} \left[\left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) + \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top \boldsymbol{Y} \right] \Big\|_F \\ \leq &\left| \tau_l^{(2)}\lambda_l^{(2)} - \tau_l^{(1)}\lambda_l^{(1)} \right| \sqrt{m_{l-1}n} \\ &\quad + \Big\| \left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) + \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} \\ &\quad - \left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top \boldsymbol{Y} \Big\|_F \\ \leq &\left| \tau_l^{(2)}\lambda_l^{(2)} - \tau_l^{(1)}\lambda_l^{(1)} \right| \sqrt{m_{l-1}n} \\ &\quad + \left\| \left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top B_l(\boldsymbol{w}_1^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) - \left(\boldsymbol{I}_{m_{l-1}} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top B_l(\boldsymbol{w}_2^{(j(l))}) \right) f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F \\ &\quad + \left\| \tau_l^{(1)} B_l(\boldsymbol{w}_1^{(j(l))})^\top \boldsymbol{Y} - \tau_l^{(2)} B_l(\boldsymbol{w}_2^{(j(l))})^\top \boldsymbol{Y} \right\|_F \\ \leq &\xi_l \sqrt{m_{l-1}n} + \gamma_l + \delta_l \|\boldsymbol{Y}\|_F, \end{aligned}$$

using the abbreviations introduced in (2.62), (2.63) and (2.65). Inserting the estimates for ξ_l, δ_l and γ_l in (2.62), (2.64) and Lemma 2.16, and using $\sqrt{m_{l-1}n} \leq \sqrt{m_\infty n}$, we obtain

$$\begin{aligned} \left\| f^l_{\mathcal{P}_1}(\boldsymbol{Y}) - f^l_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F &\leq \sqrt{m_\infty n}\tau_\infty \left| \lambda_l^{(2)} - \lambda_l^{(1)} \right| + \sqrt{m_\infty n}\lambda_\infty \left| \tau_l^{(2)} - \tau_l^{(1)} \right| \\ &+ 2\tau_\infty B_\infty \left\| f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) \right\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} + B_\infty^2 \left\| f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) \right\|_F \left| \tau_l^{(1)} - \tau_l^{(2)} \right| \\ &+ \alpha \left\| f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) - f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) \right\|_F + B_\infty \|\boldsymbol{Y}\|_F \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \tau_\infty \|\boldsymbol{Y}\|_F \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - \tau_l^{(2)} B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2}. \end{aligned}$$

Recall that by Lemma 2.14 we have, for $\ell = 1, \dots, L$,

$$\begin{aligned}\left\| f^l_{\mathcal{P}_1}(\boldsymbol{Y}) \right\|_F &\leq \|\boldsymbol{Y}\|_F \sum_{k=1}^{l} \left(\tau_k \left\| B_k(\boldsymbol{w}_1^{(j(k))}) \right\|_{2\to2} \prod_{i=k}^{l-1} \left\| \boldsymbol{I}_{m_i} - \tau_i B_{i+1}(\boldsymbol{w}_1^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}_1^{(j(i+1))}) \right\|_{2\to2} \right) \\ &\leq \|\boldsymbol{Y}\|_F \tau_\infty B_\infty \sum_{k=1}^{l} \left(\sup_{i=k,\dots,l-1} \left\| \boldsymbol{I}_{m_i} - \tau_i B_{i+1}(\boldsymbol{w}_2^{(j(i+1))})^\top B_{i+1}(\boldsymbol{w}_2^{(j(i+1))}) \right\|_{2\to2} \right)^{l-k} \\ &\leq \|\boldsymbol{Y}\|_F \tau_\infty B_\infty \sum_{k=1}^{l} \alpha^{l-k} = \|\boldsymbol{Y}\|_F \tau_\infty B_\infty \sum_{k=0}^{l-1} \alpha^k = \|\boldsymbol{Y}\|_F Z_l, \qquad (2.67)\end{aligned}$$

with α as defined in (2.44) and Z_l as in (2.45). This leads to the estimate

$$\begin{aligned}&\left\| f^l_{\mathcal{P}_1}(\boldsymbol{Y}) - f^l_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F \\ &\leq \alpha \left\| f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) - f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) \right\|_F + \tau_\infty \|\boldsymbol{Y}\|_F \left(1 + 2B_\infty Z_{l-1}\right) \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to2} \\ &\quad + \left(\lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F (B_\infty Z_{l-1} + 1)\right) \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \tau_\infty \sqrt{m_\infty n} \left| \lambda_l^{(2)} - \lambda_l^{(1)} \right|.\end{aligned}$$

Introducing the additional quantities

$$\begin{aligned}\beta_l &= \tau_\infty \|\boldsymbol{Y}\|_F \left(1 + 2B_\infty Z_{l-1}\right), \\ \kappa_l &= \left(\lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F (B_\infty Z_{l-1} + 1)\right), \\ \varphi_l &= \tau_\infty \sqrt{m_\infty n},\end{aligned}$$

we can write our estimate more compactly as

$$\begin{aligned}&\left\| f^l_{\mathcal{P}_1}(\boldsymbol{Y}) - f^l_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F \\ \leq& \alpha \left\| f^{l-1}_{\mathcal{P}_1}(\boldsymbol{Y}) - f^{l-1}_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F + \beta_l \left\| B_l(\boldsymbol{w}_2^{(j(l))}) - B_l(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to2} \\ &+ \kappa_l \left| \tau_l^{(1)} - \tau_l^{(2)} \right| + \varphi_l \left| \lambda_l^{(2)} - \lambda_l^{(1)} \right|, \qquad (2.68)\end{aligned}$$

Using our abbreviations, the general formulas for K_L, M_L and O_L for $L \geq 1$ are given by

$$K_L = \sum_{l=1}^{L} \beta_l \alpha^{L-l}, \qquad M_L = \sum_{l=1}^{L} \kappa_l \alpha^{L-l}, \qquad O_L = \sum_{l=1}^{L} \varphi_l \alpha^{L-l}, \qquad L \geq 1, \qquad (2.69)$$

which indeed for $L \geq 2$ is just a compact notation for (2.49), (2.46) and (2.47) in Theorems 2.11 and 2.17. We now prove via induction that (2.66) holds for any number of layers $L \in \mathbb{N}$ with K_L, M_L and O_L as just stated. For $L = 1$, we can directly obtain these factors from the following estimate. Using similar arguments as above, we obtain (with $m_\infty = m_0$, to keep the notation consistent)

$$\begin{aligned}&\left\| f^1_{\mathcal{P}_1}(\boldsymbol{Y}) - f^1_{\mathcal{P}_2}(\boldsymbol{Y}) \right\|_F \\ =& \left\| P_1 S_{\tau_1^{(1)}\lambda_1^{(1)}} \left[\tau_1^{(1)} B_1(\boldsymbol{w}_1^{(j(1))})^\top \boldsymbol{Y} \right] - P_1 S_{\tau_1^{(2)}\lambda_1^{(2)}} \left[\tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))})^\top \boldsymbol{Y} \right] \right\|_F \\ \leq& \left\| S_{\tau_1^{(1)}\lambda_1^{(1)}} \left[\tau_1^{(1)} B_1(\boldsymbol{w}_1^{(j(1))})^\top \boldsymbol{Y} \right] - S_{\tau_1^{(2)}\lambda_1^{(2)}} \left[\tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))})^\top \boldsymbol{Y} \right] \right\|_F\end{aligned}$$

$$
\begin{aligned}
&\leq \left\| S_{\tau_1^{(1)}\lambda_1^{(1)}} \left[\tau_1^{(1)} B_1(\boldsymbol{w}_1^{(j(1))})^\top \boldsymbol{Y}\right] - S_{\tau_1^{(1)}\lambda_1^{(1)}} \left[\tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))})^\top \boldsymbol{Y}\right] \right\|_F \\
&\qquad + \left\| S_{\tau_1^{(1)}\lambda_1^{(1)}} \left[\tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))})^\top \boldsymbol{Y}\right] - S_{\tau_1^{(2)}\lambda_1^{(2)}} \left[\tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))})^\top \boldsymbol{Y}\right] \right\|_F \\
&= \|\boldsymbol{Y}\|_F \left\| \tau_1^{(1)} B_1(\boldsymbol{w}_1^{(j(1))}) - \tau_1^{(2)} B_1(\boldsymbol{w}_2^{(j(1))}) \right\|_F + \sqrt{m_0 n} \left| \tau_1^{(2)}\lambda_1^{(2)} - \tau_1^{(1)}\lambda_1^{(1)} \right| \\
&\leq B_\infty \|\boldsymbol{Y}\|_F \left| \tau_1^{(1)} - \tau_1^{(2)} \right| + \tau_\infty \|\boldsymbol{Y}\|_F \left\| B_1(\boldsymbol{w}_1^{(j(1))}) - B_1(\boldsymbol{w}_2^{(j(1))}) \right\|_F \\
&\qquad + \tau_\infty \sqrt{m_0 n} \left| \lambda_l^{(2)} - \lambda_l^{(1)} \right| + \lambda_\infty \sqrt{m_0 n} \left| \tau_l^{(2)} - \tau_l^{(1)} \right| \\
&= \tau_\infty \|\boldsymbol{Y}\|_F \left\| B_1(\boldsymbol{w}_1^{(j(1))}) - B_1(\boldsymbol{w}_2^{(j(1))}) \right\|_F + (B_\infty \|\boldsymbol{Y}\|_F + \lambda_\infty \sqrt{m_\infty n}) \left| \tau_1^{(1)} - \tau_1^{(2)} \right| \\
&\qquad + \tau_\infty \sqrt{m_\infty n} \cdot \left| \lambda_1^{(1)} - \lambda_1^{(2)} \right|,
\end{aligned}
$$

which by (2.69) gives (2.66) for $L = 1$, since $\beta_1 = \tau_\infty \|\boldsymbol{Y}\|_F$ and $\kappa_1 = \lambda_\infty \sqrt{m_\infty n} + B_\infty \|\boldsymbol{Y}\|_F$ (because $Z_0 = 0$) and $\varphi_1 = \sqrt{m_\infty n}\tau_\infty$. Now we proceed with the induction step, assuming that the claim holds for some $L \in \mathbb{N}$. The estimate (2.68) used for the output after $L+1$ layers, combined with the induction hypothesis give us

$$
\begin{aligned}
&\left\| f_{\mathcal{P}_1}^{L+1}(\boldsymbol{Y}) - f_{\mathcal{P}_2}^{L+1}(\boldsymbol{Y}) \right\|_F \\
&\leq \alpha \left\| f_{\mathcal{P}_1}^{L}(\boldsymbol{Y}) - f_{\mathcal{P}_2}^{L}(\boldsymbol{Y}) \right\|_F + \beta_{L+1} \left\| B_{L+1}(\boldsymbol{w}_2^{(j(l))}) - B_{L+1}(\boldsymbol{w}_1^{(j(l))}) \right\|_{2\to 2} \\
&\qquad + \kappa_{L+1} \left| \tau_{L+1}^{(1)} - \tau_{L+1}^{(2)} \right| + \varphi_{L+1} \left| \lambda_{L+1}^{(2)} - \lambda_{L+1}^{(1)} \right| \\
&\leq (\alpha\beta_L + \beta_{L+1}) \max_{l\in[L]} \left\| B_l(\boldsymbol{w}_1^{(j(l))}) - B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} \\
&\qquad + (\alpha\kappa_L + \kappa_{L+1}) \left\| \boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(2)} \right\|_\infty + (\alpha\varphi_L + \varphi_{L+1}) \cdot \left\| \boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)} \right\|_\infty,
\end{aligned}
$$

so that (2.66) holds with the claimed expression for K_{L+1} and thus finishes the proof, since

$$
K_{L+1} = \alpha K_L + \beta_{L+1} = \alpha \sum_{l=1}^{L} \beta_l \alpha^{L-l} + \beta_{L+1} = \sum_{l=1}^{L+1} \beta_l \alpha^{L+1-l}.
$$

and since similar expressions also hold for O_{L+1} and M_{L+1}. ■

Let us finally provide the Lipschitz bound for the full network in terms of the parameters.

Corollary 2.18 *For two networks $h_{\mathcal{P}_1}, h_{\mathcal{P}_2} \in \mathcal{H}^L$ we have*

$$
\begin{aligned}
&\|h_{\mathcal{P}_1}(\boldsymbol{Y})) - h_{\mathcal{P}_2}(\boldsymbol{Y}))\|_F \\
&\leq \left\| f_{\mathcal{P}_1}^{L}(\boldsymbol{Y}) \right\|_F \left\| B_{L+1}(\boldsymbol{w}_1^{(j(L+1))}) - B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) \right\|_{2\to 2} + \left\| f_{\mathcal{P}_1}^{L}(\boldsymbol{Y}) - f_{\mathcal{P}_2}^{L}(\boldsymbol{Y}) \right\|_F \\
&\leq (B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L) \cdot \max_{l\in[L+1]} \left\| B_l(\boldsymbol{w}_1^{(j(l))}) - B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} \\
&\qquad + M_L \cdot \left\| \boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(2)} \right\|_\infty + O_L \cdot \left\| \boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)} \right\|_\infty
\end{aligned}
$$

with K_L, M_L and O_L as given in (2.49), (2.46) *and* (2.47).

Proof. Using that σ is 1-Lipschitz, and applying the triangle inequality, we obtain

$$
\|h_{\mathcal{P}_1}(\boldsymbol{Y})) - h_{\mathcal{P}_2}(\boldsymbol{Y}))\|_F = \left\| \sigma\left(B_{L+1}(\boldsymbol{w}_1^{(j(L+1))}) f_{\mathcal{P}_1}^{L}(\boldsymbol{Y}) \right) - \sigma\left(B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) f_{\mathcal{P}_2}^{L}(\boldsymbol{Y}) \right) \right\|_F
$$

$$\begin{aligned}
&\leq \left\| B_{L+1}(\boldsymbol{w}_1^{(j(L+1))}) f_{\mathcal{P}_1}^L(\boldsymbol{Y}) - B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) f_{\mathcal{P}_2}^L(\boldsymbol{Y}) \right\|_F \\
&\leq \left\| B_{L+1}(\boldsymbol{w}_1^{(j(L+1))}) f_{\mathcal{P}_1}^L(\boldsymbol{Y}) - B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) f_{\mathcal{P}_1}^L(\boldsymbol{Y}) \right\|_F \\
&+ \left\| B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) f_{\mathcal{P}_1}^L(\boldsymbol{Y}) - B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) f_{\mathcal{P}_2}^L(\boldsymbol{Y}) \right\|_F \\
&\leq \left\| f_{\mathcal{P}_1}^L(\boldsymbol{Y}) \right\|_F \left\| B_{L+1}(\boldsymbol{w}_1^{(j(L+1))}) - B_{L+1}(\boldsymbol{w}_2^{(j(L+1))}) \right\|_{2\to 2} + B_\infty \left\| f_{\mathcal{P}_1}^L(\boldsymbol{Y}) - f_{\mathcal{P}_2}^L(\boldsymbol{Y}) \right\|_F,
\end{aligned}$$

where we used that $\|B_{L+1}(\boldsymbol{w}_1^{(j(L+1))})\|_{2\to 2} \leq B_\infty$ by definition of B_∞, see (2.40). Using the bound (2.66) in Theorem 2.17 for $\left\| f_{\mathcal{P}_1}^L(\boldsymbol{Y}) - f_{\mathcal{P}_2}^L(\boldsymbol{Y}) \right\|_F$ and that $\left\| f_{\mathcal{P}_1}^L(\boldsymbol{Y}) \right\|_F \leq \|\boldsymbol{Y}\|_F Z_L$ by (2.67) yields the claimed estimate. ∎

Covering number estimates and proof of the main result. Finally, we are prepared for the proof of our main result.

Proof of Theorem 2.11. By the assumption (2.32) that B_l is D_l-Lipschitz, and putting $D_\infty := \max_{l=1,\dots,L} D_l$ (see (2.33) for the definition of D_l), Corollary 2.18 implies that

$$\begin{aligned}
&\|h_{\mathcal{P}_1}(\boldsymbol{Y}) - h_{\mathcal{P}_2}(\boldsymbol{Y})\|_F \\
\leq &(B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L) \cdot \max_{l \in [L+1]} \left\| B_l(\boldsymbol{w}_1^{(j(l))}) - B_l(\boldsymbol{w}_2^{(j(l))}) \right\|_{2\to 2} \\
&+ M_L \cdot \left\| \boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(2)} \right\|_\infty + O_L \cdot \left\| \boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)} \right\|_\infty \\
\leq &(B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L) \cdot D_\infty \cdot \|\boldsymbol{W}_1 - \boldsymbol{W}_2\|_{\mathcal{X}} + M_L \cdot \left\| \boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(2)} \right\|_\infty + O_L \cdot \left\| \boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)} \right\|_\infty
\end{aligned}$$

Recalling that $Q_L = (B_\infty K_L + \|\boldsymbol{Y}\|_F Z_L) \cdot D_\infty$, see (2.48), we equip $\mathcal{Y} = \mathcal{T} \times \Lambda \times \mathcal{W}$ with the following norm

$$\|(\boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{W})\|_{\mathcal{Y}} := M_L \|\boldsymbol{\tau}\|_\infty + O_L \|\boldsymbol{\lambda}\|_\infty + Q_L \|\boldsymbol{W}\|_{\mathcal{X}}, \quad (\boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{W}) \in \mathcal{Y}$$

where $\|\cdot\|_{\mathcal{X}}$ was defined in (2.38). Recall from (2.39) that $\mathcal{T} \subset \boldsymbol{\tau_0} + r_1 B^L_{\|\cdot\|_\infty}$ and $\Lambda \subset \boldsymbol{\lambda}_0 + r_2 B^L_{\|\cdot\|_\infty}$, while $\mathcal{W} \subset W_\infty B^K_{\mathcal{X}}$ by (2.40). Using that covering numbers with respect to norms are invariant under translations of the set, Lemma A.3 and Lemma A.2 give

$$\begin{aligned}
&\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon) \leq \mathcal{N}(\mathcal{T} \times \Lambda \times \mathcal{W}, \|\cdot\|_{\mathcal{Y}}, \varepsilon) \\
\leq &\mathcal{N}\left(r_1 B^L_{\|\cdot\|_\infty}, \|\cdot\|_\infty, \varepsilon/(4 \cdot M_L) \right) \cdot \mathcal{N}\left(r_2 B^L_{\|\cdot\|_\infty}, \|\cdot\|_\infty, \varepsilon/(4 \cdot O_L) \right) \\
&\cdot \mathcal{N}\left(W_\infty B^K_{\mathcal{X}}, \|\cdot\|_{\mathcal{X}}, \varepsilon/(4 \cdot Q_L) \right) \\
\leq &\left(1 + \frac{8 r_2 O_L}{\varepsilon} \right)^L \left(1 + \frac{8 r_1 M_L}{\varepsilon} \right)^L \left(1 + \frac{8 W_\infty Q_L}{\varepsilon} \right)^K
\end{aligned}$$

Already preparing its application in Dudley's integral, let us apply the logarithm to obtain

$$\begin{aligned}
&\log\left(\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon) \right) \\
\leq &K \log\left(1 + \frac{8 W_\infty Q_L}{\varepsilon} \right) + L \log\left(1 + \frac{8 r_2 O_L}{\varepsilon} \right) + L \log\left(1 + \frac{8 r_1 M_L}{\varepsilon} \right) \qquad (2.70)
\end{aligned}$$

Plugging the covering number estimate (2.70) into Dudley's integral (see (2.58) and (2.59)) gives

$$
\begin{aligned}
&\mathbb{E}\sup_{M\in\mathcal{M}}\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m_L}\varepsilon_{ik}M_{ik}\leq\frac{4\sqrt{2}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\mathcal{N}(\mathcal{M},\|\cdot\|_F,\varepsilon)}\mathrm{d}\varepsilon\\
\leq&\frac{4\sqrt{2K}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{8W_\infty Q_L}{\varepsilon}\right)}\mathrm{d}\varepsilon+\frac{4\sqrt{2L}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{16r_2O_L}{\varepsilon}\right)}\mathrm{d}\varepsilon\\
&+\frac{4\sqrt{2L}}{n}\int_0^{\sqrt{n}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{16r_1M_L}{\varepsilon}\right)}\mathrm{d}\varepsilon\\
\leq&2\sqrt{2}B_{\text{out}}\left[\sqrt{\frac{K}{n}}\Psi\left(\frac{16W_\infty Q_L}{\sqrt{n}B_{\text{out}}}\right)+\sqrt{\frac{L}{n}}\Psi\left(\frac{8r_2O_L}{\sqrt{n}B_{\text{out}}}\right)+\sqrt{\frac{L}{n}}\Psi\left(\frac{8r_1M_L}{\sqrt{n}B_{\text{out}}}\right)\right].
\end{aligned}
$$

where we applied Lemma A.5 in the last step. The theorem is obtained using Theorem 1.9 and Lemma B.4. ■

2.4 Numerical Experiments

In this section, we report on the numerical experiments performed to practically test our findings in the previous section. Note that we have not aimed at achieving state of the art results in terms of reconstruction, but instead we pursue different goals in this section. Firstly, we would like to give further evidence that the proposed framework is meaningful and captures various interesting examples of practical interest. Secondly, we are interested in the generalization error and its scaling with respect to training parameters. Specifically, we have obtained a sample complexity bound that holds uniformly over the hypothesis space and for any distribution. Although the bound is quite simple and general, it is interesting to see if we expect improvements when it is applied to data from low complexity distributions. ISTA is used mainly in sparse coding and recovery, and therefore we consider a similar scenario. Thirdly, we are interested in the role of sparsity: Recall that in our main results in this Chapter, Theorem 2.11 and Corollary 2.13 (and Theorem 2.1 and Corollary 2.9 for the special case of learning an orthogonal dictionary), we have provided worst-case bounds on the sample complexity that holds uniformly over the hypothesis space and for any arbitrary data distribution. It is interesting to see if this bound can be improved for data distributions limited to low complexity sets distributions, for example over the set of sparse vectors. ISTA is used mainly in sparse coding and recovery tasks, therefore it is reasonable to ask if the generalization error behaves similarly when it is applied to sparse recovery tasks.

We consider both synthetic data as well as the popular MNIST dataset [LeC] using a *Pytorch* implementation and a *Titan XP GPU*. In all the experiments, we have used the *Adam optimizer* [KB14] for training the network with the learning rate 10^{-2}. The objective function for optimization is the *MSE loss* (see equation (1.19)) of the recovered vector with respect to the ground truth. (Note that this slightly differs from the theoretical section; however, thanks to its differentiability it is more convenient from a numerical standpoint.) For all cases, the measurement matrix is a Gaussian random matrix, properly normalized to guarantee convergence of ISTA. The synthetic data is generated for different input and output dimensions, and sparsity levels. The default parameters are the ambient dimension $p = 120$, a number of measurements of $m = 80$ and sparsity s equal to 10. Sparse vectors are generated by choosing their support uniformly randomly and then drawing the non-zero values from the standard normal distribution. The experiments for the synthetic data are repeated at least 50 times, and the results are averaged over the repetitions. For both the MNIST and the synthetic dataset, we sweep over L, p and m to see how the generalization error behaves. For the synthetic data, we use the training data with size 10 000 and the test date with size 50 000. Each model is trained separately and mostly not more than 10 epochs are required to get first promising results, and often times, the loss goes down very slowly after 10 epochs. To generate sparse vectors, the support is chosen uniformly at random. The non-zero values are drawn from the standard normal distribution. We repeat the experiments for the synthetic data between 10 to 100 times to obtain a smoother curve after averaging.

2.4.1 Learning an Orthogonal Dictionary

This subsection considers experiments for our main example of learning implicitly an orthogonal dictionary suitable for reconstruction from compressive measurements, to which the entire Section 2.2 has been devoted; later on, in Subsection 2.3.3 we observed that it is a special case of the more general setup studied in Section 2.3.

Orthogonality Constraint. Firstly, let us comment on the orthogonality constraint for weight matrices. One way to implement it is described in [LCMR19] and uses the fact that the matrix exponential mapping provides an onto mapping from the skew-symmetric matrices onto the special orthogonal group $SO(p)$. However, we use the alternative method of adding a regularization term $\|\boldsymbol{I} - \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\|_F$ (or with another matrix norm) to the loss function, which means to penalize if $\boldsymbol{\Phi}$ is far from being orthogonal during training. We choose a random orthogonal matrix as the ground truth dictionary and initialize the model with a random matrix. Figure 2.1 plots $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ for the learned dictionary for a visual inspection. Indeed, it is approximately the identity matrix, and therefore the learned matrix seems to be approximately orthogonal.

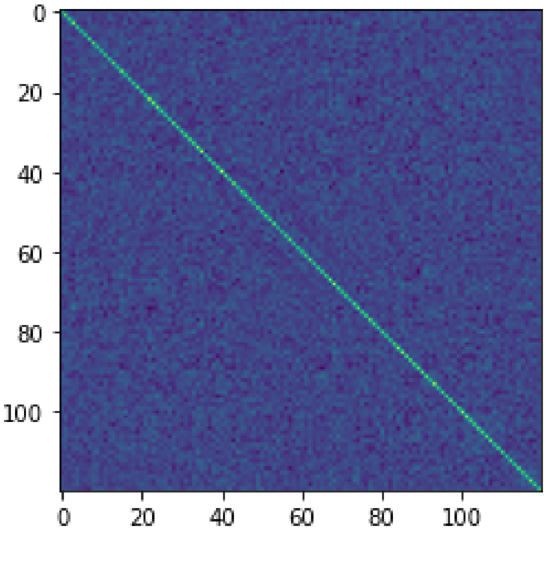
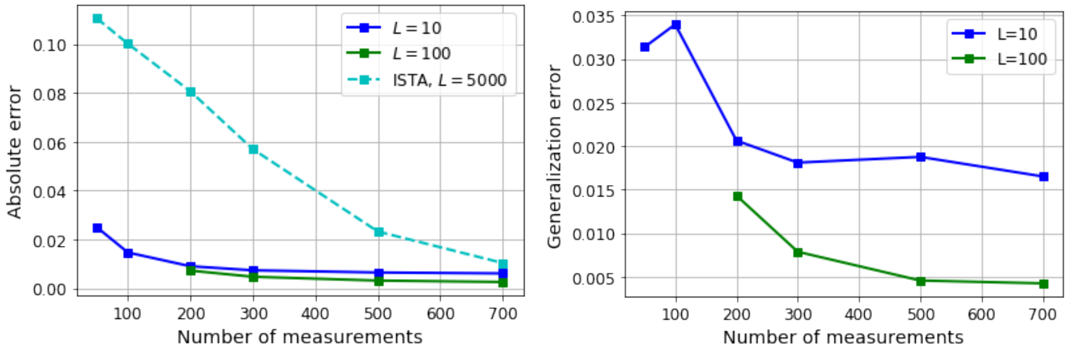


Figure 2.1: Orthogonality of the learned dictionary

Experiments with the MNIST Datset. Before considering the generalization, let us firstly confirm that our model is capable of achieving a low reconstruction error on the MNIST datset. Note that the MNIST images are grayscale images of a pixel size of 28 × 28, that we *vectorize* and represent as a vector in the 784-dimensional space. As can be observed in Figure 2.2(a), even with a comparably small number of layers LISTA outperforms standard ISTA. (Note that the error in the MNIST experiments is the pixel-based error normalized by the image dimension and MNIST pixels are all normalized between 0 and 1.) We have chosen ISTA with a similar structure and 5 000 iterations. The result warrants the applicability of dictionary learning for sparse reconstruction. Figure 2.2(b) shows an decreasing generalization error with an increasing number of measurements on MNIST.



(a) Absolute reconstruction error for different measurements of MNIST

(b) Generalization error for different measurements of MNIST (averaged)

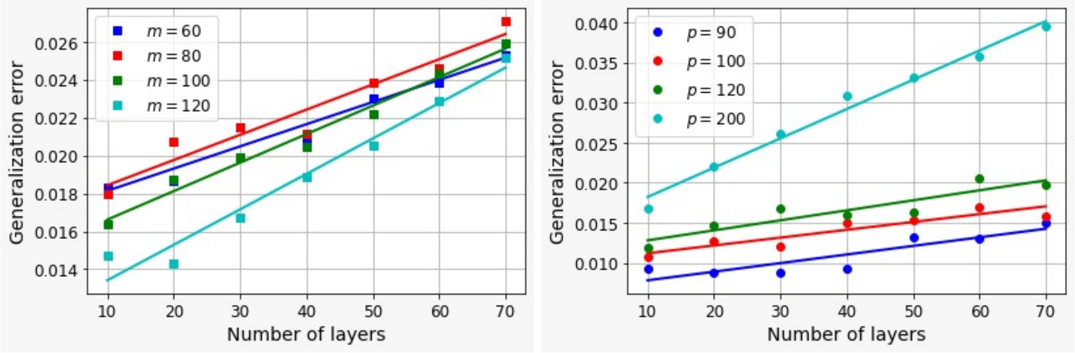
Figure 2.2: MNIST dataset

Experiments with Synthetic Data. Figure 2.3(a) confirms the dependence of the generalization error on the number of layers L. Increasing the number of layers increases the generalization error for a fixed number of measurements m. However, the generalization error decreases by increasing the number of layers for MNIST dataset. For both synthetic and the MNIST dataset, it seems that increasing the number of measurements decreases the generalization error; see also Figures 2.2(b), 2.3(a) and 2.3(b). Besides, Figure 2.3(b) shows that increasing p increases the generalization error. Therefore, our bound scales correctly with the input dimension and the number of layers but incorrectly with the number of measurements. Although not predicted by our theoretical results, this is not unexpected. Note that the number of measurements m is not essential here, since it can always be upper bounded by the dimension p. Therefore, the theoretical bound on the generalization error (see (2.8), and Theorem 2.1 as well as Corollary 2.9 for more details) can be lower and as follows upper bounded via

$$\sqrt{\frac{\log(L)}{n}}p \leq \sqrt{\frac{\log(L)}{n}}(p + \sqrt{pm}) \leq 2\sqrt{\frac{\log(L)}{n}}p.$$

Furthermore, as mentioned above, the sample complexity is supposed to apply to all possible input distributions. Possibly, if we restrict ourselves to distributions over low complexity sets, then various worst-case bounds in our analysis might be improved. The experiments seem to confirm this reasoning. Namely, for the MNIST dataset there is a clear improvement with increasing the number of measurements and the number of layers. This is intuitive from a compressive sensing standpoint, as more number of layers in ISTA leads to better results and more measurements provide more information about the input.

On the other hand, the synthetic dataset shows that the generalization error increases with the input dimension and the number of layers. Note that the bound of this chapter is obtained for a very general setting where nothing is assumed on the data structure. Potentially, additional assumptions on the structure of the problem such as sparsity could be used to improve the current bounds. Nonetheless, the mild logarithmic dimension dependency of the current bound makes it a very good baseline for future comparisons.



(a) Generalization error for different measurements of synthetic data ($p = 120$)

(b) Generalization error for different input dimensions of synthetic data ($m = 80$)

Figure 2.3: Generalization error for synthetic dataset

The model that is used for our experiments shares the weights across layers conforming to our theoretical setup. However, we can improve the performance of this method by using ideas similar to LISTA literature. Many works on LISTA use a different dictionary at each layer, which eases the training procedure and can lead to potentially better results.

Role of Sparsity. Next, we consider similar synthetic data with sparse inputs. In Figure 2.4, the generalization error is plotted for a variation of parameter choices. The input dimension is fixed to $p = 120$. We have used a linear fit between the data points with different numbers of layers. Increasing the number of layers increases the generalization error. Note that increasing sparsity, which can be seen as the effective dimension of the input, increases the generalization error. Also, the observation about dependence on the generalization error on L is compatible with our theoretical results and suggests that the logarithmic scaling in L may not be removed in general. These two points are compatible with findings of our theory. We conjecture that the dependence to input dimension can be relaxed to a potentially smaller effective dimension. On the other hand, in conflict with our theory, the generalization error decreases with the number of measurements m. Larger number of measurements consistently yields better generalization error. Therefore, it is expected as the task becomes easier with more measurements, the generalization error improves. To accommodate this theoretically is an open question.

We run a similar analysis for MNIST dataset. Although MNIST images are themselves sparse, they possess additional structure. The generalization error is plotted in Figure 2.5. First of all, it can be seen that the generalization error decreases with increasing number of measurements. A similar observation is made in the experiments on the synthetic data. But in this case, an additional discrepancy with the theory emerges as increasing the number of layers decreases the generalization error.

While our theoretical bound actually increases with increasing number of layers (and slightly increases with increasing number of measurements [BRS22], although that dependence is swallowed by the constant in (2.55)), the better behavior obtained here may

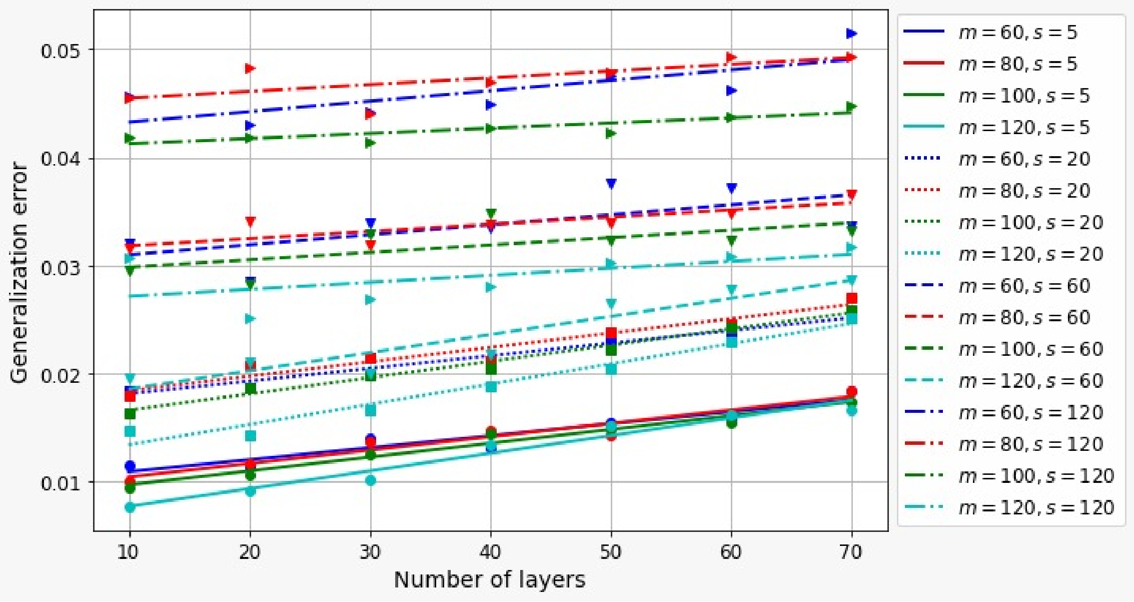


Figure 2.4: Generalization error for $p = 120$

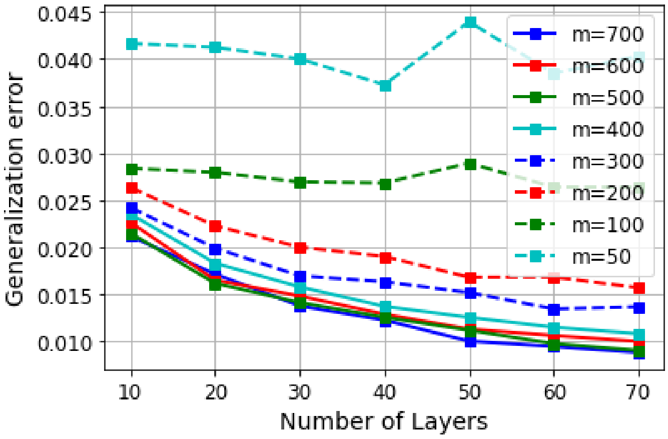


Figure 2.5: Generalization error for different measurements of MNIST

be justified from a compressive sensing standpoint. The reconstruction task becomes easier with more measurements and the quality becomes better with more iterations. Note that the soft-thresholding step only promotes sparsity structure, so additional layers can help recovering more details. Additional assumptions that may take the specific compressive sensing scenario into account are currently not captured by our general worst case result Theorem 2.11, which provides a uniform complexity bound that applies to all possible input distributions. We conjecture that the bound of Theorem 2.11 can be improved by taking into account assumptions like sparsity of the input and properties of the measurement matrix A and the underlying true dictionary $\mathbf{\Phi}_0$ such as a restricted isometry property of $A\mathbf{\Phi}_0$. Presently however, it is not clear how this could potentially be done.

2.4.2 Learning a non-orthogonal dictionary.

Here, we abandon the orthogonality assumption by removing the regularizer mentioned above.

Correlation of Generalization bound and Generalization error In this section, we explore if our bound correlates with the generalization error. We first consider the case where the dictionaries are chosen to be an arbitrary matrix and not necessarily orthogonal. In order to evaluate how close our theoretical bounds are to reality, Figure 2.6 shows the empirically observed generalization error versus our theoretical generalization bound. We clearly observe that our bounds are generally positively correlated with the empirical generalization error. Note that in this experiment, the dictionaries are not orthogonal matrices. The generalization error increases with the number of layers and with with the dimension p. The other dependencies are less clear, since their effect is mixed with other terms in the generalization bound. We have chosen a sparsity $s = 10$ for these experiments and plotted the generalization bound from Theorem 2.11.

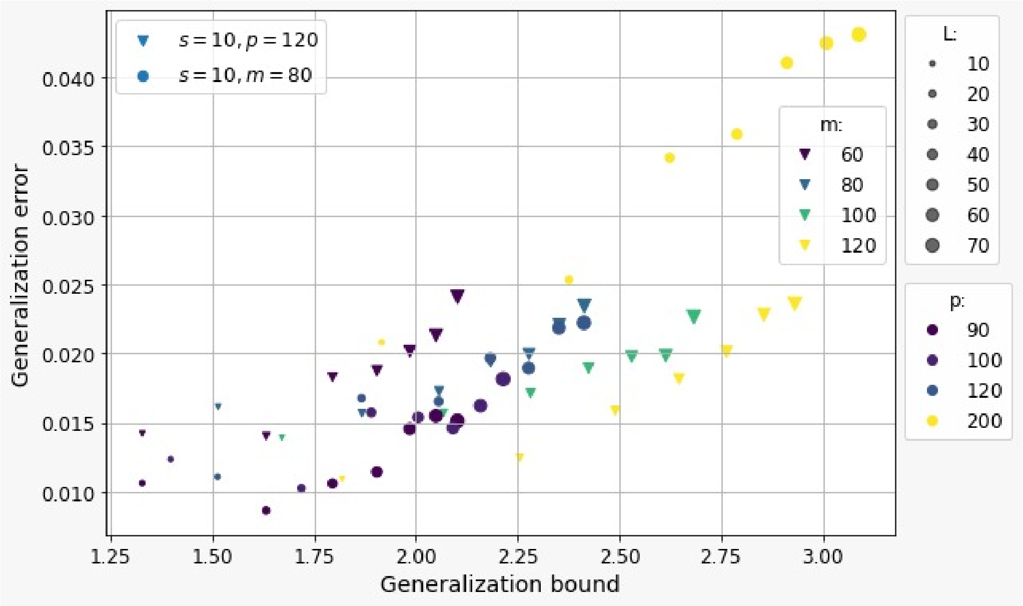


Figure 2.6: Generalization error vs generalization bound

2.5 Related Work and Outlook

In this final section of this chapter, let us conclude by discussing extensions, related work and open questions that are connected with the topics presented in this chapter.

Extensions of Generalization Error Bounds to Other Algorithms. To the best of our knowledge, our papers [BRS22; SBR21] are the first results providing statistical learning guarantees for deep learning with ISTA-inspired neural networks for sparse reconstruction or regression tasks, whereas most work on generalization of deep neural networks focused on classification tasks. Our result has been featured in the survey article [SHRHE22] on theoretical perspectives on deep learning methods in inverse problems. Furthermore, it has proven to be useful beyond the particular setting of ISTA-inspired neural networks and was successfully adapted to other cases, namely in two Master thesis co-supervised by the author of this thesis: firstly, in [Lub21] this approach was extended to the first-order primal-dual algorithm [CP11], and in another Master thesis [Nau22] to ALISTA-like networks [LC19] (both were co-supervised by the author of this thesis). Another extension of our work to deep unfolding network for analysis-sparsity-based compressive sensing is [KP22].

Robustness with Respect to the Measurements. An important part of the proofs in this chapter was to derive perturbation bounds with respect to different choices of the parameters. It is also to consider instead perturbations in the measurements $\boldsymbol{y}$, *i.e.,* we consider, for two different measurements $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^m$ (where we may interpret $\boldsymbol{y}_2$ to be a noisy or perturbed version of $\boldsymbol{y}_1$)

$$\left\| f^L(\boldsymbol{y}_1) - f^L(\boldsymbol{y}_2) \right\|_2 .$$

Robustness is classically studied in compressive sensing, and also from a deep learning perspective with its possible connection to adversarial perturbations [Sze+13], even though they are more common in classification settings. However, also robustness of deep learning for inverse problems has been investigated mostly from an experimental view but remain unsettled, as conflicting results have been reported in the literature [ARPAH20; GAAH20; GMM20]. With regard to robustness to perturbations in the measurements, we would like to find a bound for the quantity above that only mildly depends on $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2$ and the number of layers L. How much can such a perturbation influence the reconstruction error? Under realistic assumptions, one can show that L iterations of ISTA are L-Lipschitz, *i.e.,*

$$\|f^L(\boldsymbol{y}_1) - f^L(\boldsymbol{y}_2)\|_2 \leq L\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2,$$

where f^L simply denotes L iterations of/layers of ISTA. This is shown in the following simple and so far unpublished result, that shows at least a certain degree of robustness with respect to the measurements.

Theorem 2.19 (Rauhut, S., 2020) *Consider L iterations of ISTA and assume that*

$$\left\| \boldsymbol{I} - \tau \boldsymbol{A}^\top \boldsymbol{A} \right\|_{2\to 2} \leq 1, \qquad \tau \|\boldsymbol{A}\|_{2\to 2} \leq 1. \tag{2.71}$$

Then, for any two (different) measurements $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^m$ there is

$$\|f^L(\boldsymbol{y}_1) - f^L(\boldsymbol{y}_2)\|_2 \leq L\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2. \tag{2.72}$$

Note that the result may be sub-optimal, as no assumptions on the data (such as sparsity) or RIP-like conditions on the measurement matrix are being used. Further, let us remark that, given such bounds, it is straightforward to obtain perturbation bounds with respect to perturbations *both in the measurements and the parameters*. Indeed, by the triangle inequality we get

$$\begin{aligned} \|f^L_{\mathcal{P}_1}(\boldsymbol{y}_1) - f^L_{\mathcal{P}_2}(\boldsymbol{y}_2)\|_2 &= \|f^L_{\mathcal{P}_1}(\boldsymbol{y}_1) - f^L_{\mathcal{P}_1}(\boldsymbol{y}_2) + f^L_{\mathcal{P}_1}(\boldsymbol{y}_2) - f^L_{\mathcal{P}_2}(\boldsymbol{y}_2)\|_2 \\ &\leq \|f^L_{\mathcal{P}_1}(\boldsymbol{y}_1) - f^L_{\mathcal{P}_1}(\boldsymbol{y}_2)\|_2 + \|f^L_{\mathcal{P}_1}(\boldsymbol{y}_2) - f^L_{\mathcal{P}_2}(\boldsymbol{y}_2)\|_2, \end{aligned}$$

where the first summand is of the type studied here, and the second summand (perturbations with respect to the parameters, but not measurements) has been studied earlier in this chapter.

Proof. If $\boldsymbol{y}_1 = \boldsymbol{y}_2$, the statement is trivial. Otherwise, we prove this via induction on L. For $L = 2$ layers we obtain by introducing mixed terms the following estimate:

$$\begin{aligned}
&\|f^2(\boldsymbol{y}_1) - f^2(\boldsymbol{y}_2)\|_2 = \|f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_1) - f_2(f_1(\boldsymbol{0},\boldsymbol{y}_2),\boldsymbol{y}_2)\|_2 \\
\leq&\|f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_1) - f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_2)\|_2 \qquad (2.73)\\
&+ \|f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_2) - f_2(f_1(\boldsymbol{0},\boldsymbol{y}_2),\boldsymbol{y}_2)\|_2 \qquad (2.74)\\
=&2\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2.
\end{aligned}$$

For instance, the first term in (2.73) can be estimated as follows, just using basic properties such as 1-Lipschitzness, the assumption (2.71)

$$\begin{aligned}
&\|f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_1) - f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_2)\|_2 \\
=&\left\|S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) + \tau\boldsymbol{A}^\top\boldsymbol{y}_1\right] - S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) + \tau\boldsymbol{A}^\top\boldsymbol{y}_2\right]\right\|_2 \\
\leq&\left\|\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) + \tau\boldsymbol{A}^\top\boldsymbol{y}_1 - \left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) - \tau\boldsymbol{A}^\top\boldsymbol{y}_2\right\|_2 \\
=&\tau\|\boldsymbol{A}\|_{2\to2}\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2 \\
\leq&\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2.
\end{aligned}$$

and a similar computation can be done for the second term in the line (2.74) as follows:

$$\begin{aligned}
&\|f_2(f_1(\boldsymbol{0},\boldsymbol{y}_1),\boldsymbol{y}_2) - f_2(f_1(\boldsymbol{0},\boldsymbol{y}_2),\boldsymbol{y}_2)\|_2 \\
=&\left\|S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) + \tau\boldsymbol{A}^\top\boldsymbol{y}_2\right] - S_{\tau\lambda}\left[\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_2) + \tau\boldsymbol{A}^\top\boldsymbol{y}_2\right]\right\|_2 \\
\leq&\left\|\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) + \tau\boldsymbol{A}^\top\boldsymbol{y}_2 - \left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_2) - \tau\boldsymbol{A}^\top\boldsymbol{y}_2\right\|_2 \\
=&\left\|\left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_1) - \left(\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right)f_1(\boldsymbol{0},\boldsymbol{y}_2)\right\|_2 \\
=&\left\|\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right\|_{2\to2}\|f_1(\boldsymbol{0},\boldsymbol{y}_1) - f_1(\boldsymbol{0},\boldsymbol{y}_2)\|_2 \\
=&\left\|\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right\|_{2\to2}\left\|S_{\tau\lambda}(\tau\boldsymbol{A}^\top\boldsymbol{y}_1) - S_{\tau\lambda}(\tau\boldsymbol{A}^\top\boldsymbol{y}_2)\right\|_2 \\
\leq&\left\|\boldsymbol{I} - \tau\boldsymbol{A}^\top\boldsymbol{A}\right\|_{2\to2}\tau\|\boldsymbol{A}\|_{2\to2}\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2 \\
\leq&\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2.
\end{aligned}$$

where again we used that f_1 is 1-Lipschitz in the last two steps. We proceed with the induction step for the general case as follows. Assume that (2.72) holds for a fixed $L \in \mathbb{N}$. Then, with arguments similar to above, we obtain

$$\begin{aligned}
&\left\|f^{L+1}(\boldsymbol{y}_1) - f^{L+1}(\boldsymbol{y}_2)\right\|_2 \\
=&\left\|f_{L+1}(f^L(\boldsymbol{y}_1),\boldsymbol{y}_1) - f_{L+1}(f^L(\boldsymbol{y}_2),\boldsymbol{y}_2)\right\|_2 \\
\leq&\left\|f_{L+1}(f^L(\boldsymbol{y}_1),\boldsymbol{y}_1) - f_{L+1}(f^L(\boldsymbol{y}_1),\boldsymbol{y}_2)\right\|_2 + \left\|f_{L+1}(f^L(\boldsymbol{y}_1),\boldsymbol{y}_2) - f_{L+1}(f^L(\boldsymbol{y}_2),\boldsymbol{y}_2)\right\|_2 \\
\leq&\left\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2 + \|f^L(\boldsymbol{y}_1) - f^L(\boldsymbol{y}_2)\right\|_2
\end{aligned}$$

$$
\begin{aligned}
&\leq \|y_1 - y_2\|_2 + L\|y_1 - y_2\|_2 \\
&= (L+1)\|y_1 - y_2\|_2,
\end{aligned}
$$

which finishes the proof. ■

Generalization of Deep Neural Networks. In the past decade, deep neural networks have been used with great success in many practical applications, but their theoretical understanding remains limited, despite great efforts. Deep and overparameterized neural networks seem to work very differently to traditional neural networks and tend to generalize surprisingly well. Understanding generalization of neural networks, and the complex interplay of the generalization with an approximation-theoretic or function space viewpoint, and the delicate questions of optimization on the non-convex loss surface etc. remain challenging questions. Note that we have exclusively focused on a generalization viewpoint and omitted the study of the (highly challenging) underlying non-convex optimization problem. Let us briefly review some other works of generalization in the context of deep neural networks. Note that they, in contrast to our work, are mainly devoted to classification (rather then regression) problems and operate in an overparametrized setting.

An early work providing VC dimension bounds for deep neural networks is [AB99]. Later on, bounds on the Rademacher complexity were derived in [BFT17] to obtain norm-based (*i.e.,* norms of involved objects like weight matrices) generalization error bounds for the probability of misclassification of a neural network in a multi-class problem with K classes. A similar, but slightly worse, norm based bound was obtained [NBS18] using a PAC Bayesian approach, which leads to a completely different analysis. A bound with potentially better dimension dependence was obtained in [GRS18].

However, it is doubtful if traditional such as the VC dimension and the Rademacher complexity are suitable tools for explaining generalization in deep learning, as argued based on detailed experiments in [JNMKB20; NK19; ZBHRV17]. For instance, [NK19] shows that such quantities may even grow with an increasing size of the training dataset.

Furthermore, this approach works better with a simple training procedure (for instance, for a convex loss surface with available convergence guarantees, or more generally when the minimizer of the loss function can be expressed implicitly as the solution of a fixed-point equation). On the other hand, it may become intractable in case of highly complicated models, such as the highly non-linear functions represented by deep neural networks, with an inaccessible training procedure (non-convex loss surface, stochastic gradient descent). Nevertheless, some works on (shallow, typically having only one hidden layer) neural networks exist, and using *full batch* training rather than stochastic gradient descent [ASS20; LC18a; LLC18; SMG13].

On the other hand, asymptotic approaches try to determine asymptotically precise behavior, for instance an asymptotically precise generalization, rather than generalization error bounds. In the context of neural networks, this has been investigated for simpler models such as shallow neural with only one hidden layer, using toy models for the data and full batch training [ASS20; LC18a; LLC18; SMG13]. For large and realistic models and using stochastic gradient descent, this approach remains intractable for the time being due to the immense technical difficulties arising from the non-linearity and complicated dependence structure. Another approach that also uses asymptotic techniques, but in an infinite-width limit, is the so-called *neural tangent kernel* approach [JGH18], which has received tremendous interest in recent years. However, it must be pointed out again that, due to the sheer amount of literature on the subject, no complete literature review can be provided. For a recent monograph on the current mathematical understanding of deep learning, let us again refer to [GK22].

Other Related Work The idea of interpreting gradient steps of iterative algorithms such as ISTA [DDDM04] for sparse recovery as layers of neural networks has appeared in [GL10] and has then become an active research topic, e.g., [CLWY18; KM16; LCWY19; MPB15; WGLZ20; XWGWW16]. The present paper is another contribution in this line of work and can be seen as a direct follow-up to our previous work [BRS22]. Both are characterized by studying LISTA-inspired networks from a generalization perspective, which has been neglected in the literature before. Our previous work [BRS22] focusses on a comparably simple problem of learning a dictionary suitable for reconstruction and may serve as an introduction to the topic, containing many related references and also a short introduction to generalization of neural networks for classification problems. Instead, this paper studies a much more general framework aiming to capture many other models of practical interest. It contains the scenario studied in [BRS22] as a special case, but also other models studied before, such as a class of LISTA models that use convolutional dictionaries [SG18a]; see also Section 2.3.3.

To our best knowledge, it provides the first generalization error bounds for all of them, apart from our own previous work [BRS22]. Thus, it will serve as a reference and baseline for comparison with future works. Even though the basic proof methods are very similar to the ones used in [BRS22], the derivations become clearly more involved by taking additional training parameters into accout, as well as the numerical experiments. Instead of novel algorithmic aspects, our contribution is to conduct a generalization analysis for a large class of recovery algorithms, which to the best of our knowledge has not been addressed in the literature before in this particular setting. Furthermore, our setup proposed here also includes general regression tasks apart from reconstruction. In this way, we connect this line of research with recent developments [BFT17; GRS18] in the study of generalization of deep neural networks. Particularly, we use a similar framework to [BFT17] by bounding the Rademacher complexity using Dudley's integral. However, the approach of [BFT17] applies only to the use of neural networks for classification problems. The extension to our problem, which is a regression problem with vector-valued functions, involves additional technicalities requiring the generalized contraction principle for hypothesis classes of vector-valued functions from [Mau16]. Besides, we show linear dependence of the number of training samples with the dimension (number of free parameters), using techniques that are different from the ones in [GRS18]. It is not straightforward to extend the result of [GRS18] to our case because we allow weight sharing between different layers of the thresholding networks.

The unfolded networks we consider here fall into the larger class of proximal neural networks studied in [Has+20; Has+21; HNS21]. Many other related works are in the context of dictionary learning or sparse coding: The central problem of sparse coding is to learn weight matrices for an unfolded version of ISTA. Different works focus on different parametrization of the network for faster convergence and better reconstructions. Learning the dictionary can also be implicit in these works. Some of the examples of these algorithms are recently suggested Ada-LISTA [AGE20], convolutional sparse coding [SG18b] learning efficient sparse and low-rank models [SBS15]. Another line of work considers analytic LISTA (ALISTA) [LCWY19], where only thresholds and step-size parameters are learned. For instance, in neurally augmented ALISTA [BSJ21] step sizes and thresholds are updated based on the output of the previous layers. Like many other related papers, such as ISTA-Net [ZG18], these methods are mainly motivated by applications like inpainting [AGE20]. Sample complexity of dictionary learning has been studied before in the literature [Geo18; GJBKS15; GS10; Sch14; VMB11]. The authors in [VMB11] also use a Rademacher complexity analysis for dictionary learning, but they aim at sparse representation of signals rather than reconstruction from compressed measurements and moreover, they do not use neural network structures. Fundamental limits of dictionary learning from an information-theoretic perspective has been studied in [JEG14; JEG16]. Unique about our perspective and different to the cited papers is our approach for determining the sample complexity based on learning a dictionary (or generally, other parameters to enable good reconstruction) implicitly by training a neural network.

In case of weight sharing between all layers, the networks is a recurrent neural network. The authors of [DS96] derive VC-dimension estimates of recurrent networks for recurrent perceptrons with binary outputs. The VC-dimension of recurrent neural networks for different classes of activation functions has been studied in [KS98]. However, their results do not apply to our setup, since they focus on one-dimensional inputs and outputs, *i.e.,* corresponding to just a single measurement in our compressive sensing scenario. Furthermore, VC dimension bounds are mainly suited for classification tasks, making an application to (and comparison with) our vector-valued regression problem difficult.

3 Sparse Linear Classifiers via ISTA

This chapter revisits one of the most basic models in machine learning, namely simple linear models of the type $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$, *i.e.*, inner products of a weight vector $\boldsymbol{\omega} \in \mathbb{R}^p$ with a data point $\boldsymbol{x} \in \mathbb{R}^p$. Linear models may be used for either regression or classification problems and have been heavily studied in the literature. Examples are classical machine learning algorithms such as least-squares problems, logistic regression, support vector machines (SVMs) etc., some of which we have already briefly encountered in Chapter 1. While being apparently simple, note that often they are building blocks of more powerful, non-linear models such as kernel SVMs that consist in a non-linear feature map (to increase the separability of classes) followed by a linear classifier (hyperplane separation). Furthermore, deep neural networks can also be regarded as highly non-linear, elaborated feature maps that allow a simple classifier in the very final layer, e.g. through logistic regression. This approach is often referred to as *end-to-end* learning, by learning the features and training a classifier simultaneously, rather than independently of each other. Thus, despite their simplicity, such linear models remain relevant also in the context of modern applications of machine learning.

In this chapter, we focus on classification problems where we assume that only a few of the features collected in the data point $\boldsymbol{x}$ characterize its class membership. That means that a good classifier must perform a *feature selection* of those features that are most relevant to obtain good predictions, which can be modelled through a sparsity assumption on the weight vector $\boldsymbol{\omega}$: if it contains only few-nonzero entries, only the few corresponding features will be considered for the task at hand, while all other features will be essentially discarded.

This chapter provides a novel analysis of the performance of sparse linear classifiers obtained through ISTA. It is based on the paper [TSSCV22] which is coauthored by the author of this thesis. Some of the shortcomings of the original paper could have been improved upon while preparing this chapter, notably Section 3.3. Other aspects contained in the original paper [TSSCV22] such as hyperparamter optimization have been left out: even though a very interesting application, we focus purely on a generalization perspective.

We should point out that the paper [TSSCV22], and thus this chapter in general, rely heavily on results developed previously in [LC20; Lou23] on deterministic equivalents and concentration of random equations. This approach has already been applied before to softmax classifiers in [SLCT21]; some parts of our derivation can be regarded as an adaption to our case. Closely related is former work on the asymptotic performance logistic regression model in [EKBBLY13; MLC19]. However, they still consider the data to be Gaussian, while [SLCT21] and the present work consider a more general concentration assumption on the input data. An important tool to break the arising stochastic dependencies is the *leave-one-out approach* [CFMW19; DC18; EKBBLY13], which has been employed also in the aforementioned works.

This chapter is structured as follows. We begin in Section 3.1 with some observations on predicting the accuracy of linear classifiers that are of interest even in more general scenarios as long as the classifications scores $\boldsymbol{\omega}^\top \boldsymbol{x}$ are normally distributed. Then, in Section 3.2 we formally introduce the specific ISTA-based setup we will investigate, along with technical assumptions. Section 3.3 is short but important as we show that the distribution over the hypothesis class has favorable properties: the learned weight vector $\boldsymbol{\omega}$ will be viewed as the solution of a random fixed point equation, and the underlying data distribution induces a tight concentration on $\boldsymbol{\omega}$. This is formulated more rigorously in Theorem 3.13. In Section 3.4 we delve into the laborious derivation of an algorithm to estimate the statistics of $\boldsymbol{\omega}$ or, more precisely, related scalar quantities thereof, and we test the quality of our predictions at the hand of numerical experiments.

Finally, we conclude this chapter in Section 3.5 by discussing our findings, related work, and possible future work.

3.1 Predicting the Classification Accuracy from the Distributions of the Classification Score

We begin this chapter with a few general observations regarding linear classifiers, whose output $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ we refer to as the *classification score*. Note that the results here are not at all specific to the ISTA-based classifiers, so that we formally introduce the latter only below in the next section.

3.1.1 Classification Scores

Firstly, we state a rather elementary but fundamental observation regarding linear models of the type $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ in general. It is relevant far beyond the specific ISTA-based derivation of the weight vector $\boldsymbol{\omega} \in \mathbb{R}^p$ that we consider here, and only assumes a Gaussian behavior of $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ for high-dimensional data, which is not unrealistic due to Theorem 1.11. Formally, we consider a linear binary classifier which allocates a data point $\boldsymbol{x} \in \mathbb{R}^p$ to class $\mathcal{C}_1$ (short: $\boldsymbol{x} \to \mathcal{C}_1$) or to class $\mathcal{C}_2$ (short: $\boldsymbol{x} \to \mathcal{C}_2$) through the test

$$g(\boldsymbol{x}) = \boldsymbol{\omega}^{\star\top} \boldsymbol{x} \underset{\mathcal{C}_1}{\overset{\mathcal{C}_2}{\gtrless}} \eta,$$

with a chosen threshold $\eta \in \mathbb{R}$. We will also shortly write "$\boldsymbol{x} \in \mathcal{C}_\ell$" when we mean that $\boldsymbol{x}$ is a random vector that follows the distribution of class $\mathcal{C}_\ell$ for $\ell = 1, 2$. We will also assume the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ to have class-specific means and covariances

$$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^p, \qquad \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^{p \times p}.$$

(More details and technical assumptions that are not yet required for the purpose of this section are provided later in Section 3.2.) The classification error (*i.e.*, the probability of misclassification) ε of this classifier (for now, simply assuming equal prior class probability, *i.e.*, $c_1 = c_2 = 1/2$) is given by

$$\varepsilon = \frac{1}{2} \mathbb{P}\left(\boldsymbol{x} \to \mathcal{C}_1 | \boldsymbol{x} \in \mathcal{C}_2\right) + \frac{1}{2} \mathbb{P}\left(\boldsymbol{x} \to \mathcal{C}_2 | \boldsymbol{x} \in \mathcal{C}_1\right).$$

It turns out that, assuming that $g(\boldsymbol{x}) = \boldsymbol{\omega}^{\star\top} \boldsymbol{x}$ has (univariate!) normal distributions $\mathcal{N}(\mathfrak{m}_1, \sigma_1^2)$ for $\boldsymbol{x} \in \mathcal{C}_1$ and $\mathcal{N}(\mathfrak{m}_2, \sigma_2^2)$ for $\boldsymbol{x} \in \mathcal{C}_2$, it is possible to derive the precise classification error. Therefore, it is crucial to precisely understand the statistical behavior of $g(\boldsymbol{x}) = \boldsymbol{\omega}^{\star\top} \boldsymbol{x}$, which in turn depends on the statistical properties of the underlying data distribution. Before we delve into this rather technical matter, this section explains how to predict the classification error. The following lemma provides the classification error in the aforementioned setting.

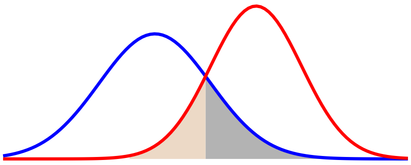


Figure 3.1: Illustration of the one-dimensional Gaussian distributions of the classification score $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}$ for both classes $\mathcal{C}_1$ (blue) and $\mathcal{C}_2$ (red). The smaller the "overlap" (brown/grey area) of the two bell curves, the higher is the classification accuracy. With far distant means, and smaller variances, the classification accuracy becomes higher.

Lemma 3.1 (Classification accuracy: general case) *Let us assume that, for a linear binary classifier $\boldsymbol{\omega}^\star \in \mathbb{R}^p$, the classification score $g(\boldsymbol{x}) = {\boldsymbol{\omega}^\star}^\top \boldsymbol{x}$ has normal distributions $\mathcal{N}(\mathfrak{m}_1, \sigma_1^2)$ for $\boldsymbol{x} \in \mathcal{C}_1$ and $\mathcal{N}(\mathfrak{m}_2, \sigma_2^2)$ for $\boldsymbol{x} \in \mathcal{C}_2$. Then, the classification error* (i.e., *probability of misclassification) is given by, with Q being defined in* (B.4),

$$c_1 Q\left(\frac{\mathfrak{m}_1 - \eta}{\sigma_1}\right) + c_2 Q\left(-\frac{\mathfrak{m}_2 - \eta}{\sigma_2}\right).$$

Thus, for a practical application of this result, one needs to estimate the specific means and variances characterizing both distributions. Note again that this result holds far beyond the specific LASSO-based classification analyzed in this chapter. It makes no assumption on the underlying data distribution and applies whenever the condition of the normally distributed classification scores is satisfied. Even though rather basic, let us give the short proof for completeness.

Proof. The proof is straightforward by computing the conditional probabilities, calculating the tail probabilities using the function Q from (B.4), and applying a substitution therein.

$$\begin{aligned}
\varepsilon =& c_1 \mathbb{P}\left(\boldsymbol{x} \to \mathcal{C}_2 \,|\, \boldsymbol{x} \in \mathcal{C}_1\right) + c_2 \mathbb{P}\left(\boldsymbol{x} \to \mathcal{C}_1 \,|\, \boldsymbol{x} \in \mathcal{C}_2\right) \\
=& c_1 \mathbb{P}\left({\boldsymbol{\omega}^\star}^\top \boldsymbol{x} > \eta \,|\, {\boldsymbol{\omega}^\star}^\top \boldsymbol{x} \sim \mathcal{N}\left(\mathfrak{m}_1, \sigma_1^2\right)\right) + c_2 \mathbb{P}\left({\boldsymbol{\omega}^\star}^\top \boldsymbol{x} < \eta \,|\, {\boldsymbol{\omega}^\star}^\top \boldsymbol{x} \sim \mathcal{N}\left(\mathfrak{m}_2, \sigma_2^2\right)\right) \\
=& c_1 \mathbb{P}\left(X - \eta > 0 \,|\, X \sim \mathcal{N}\left(\mathfrak{m}_1, \sigma_1^2\right)\right) + c_2 \mathbb{P}\left(X - \eta < 0 \,|\, X \sim \mathcal{N}\left(\mathfrak{m}_2, \sigma_2^2\right)\right) \\
=& \frac{c_1}{\sqrt{2\pi}} \int_{-\infty}^{0} \exp\left(-\frac{(z - \eta + \mathfrak{m}_1)^2}{2\sigma_1^2}\right) \mathrm{d}z + \frac{c_2}{\sqrt{2\pi}} \int_{-\infty}^{0} \exp\left(-\frac{(z - \mathfrak{m}_2 + \eta)^2}{2\sigma_2^2}\right) \mathrm{d}z \\
=& c_1 Q\left(\frac{\mathfrak{m}_1 - \eta}{\sigma_1}\right) + c_2 Q\left(-\frac{\mathfrak{m}_2 - \eta}{\sigma_2}\right).
\end{aligned}$$

■

Now it is easy to derive simplified expressions of various special cases of interest.

Corollary 3.2 (Classification accuracy for classes of equal size) *In the aforementioned setting, but with classes of equal size, the classification error* (i.e., *the probability of misclassification) is given by*

$$\varepsilon = \frac{1}{2} Q\left(\frac{\mathfrak{m}_1 - \eta}{\sigma_1} - \frac{\mathfrak{m}_2 - \eta}{\sigma_2}\right).$$

Proof. This result follows immediately by choosing $c_1 = c_2 = 1/2$ in Lemma 3.1. ■

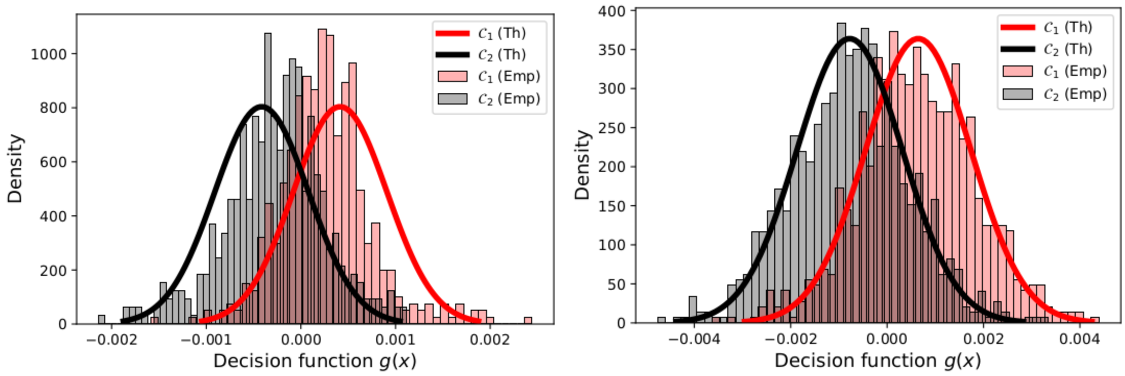
This expression may further simplify in case of $\eta = 0$ or $\sigma_1 = \sigma_2$. For instance, let us consider the case of equal covariance matrices for both classes ($\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$), when the variance of the decision score is the same for class $\mathcal{C}_1$ and $\mathcal{C}_2$, *i.e.*, $\sigma_1 = \sigma_2 = \sigma$ (see (3.2) in Lemma 3.3 below). Then, the classification error is given by $\varepsilon = \frac{1}{2}Q\left(\frac{\mathfrak{m}_1 - \mathfrak{m}_2}{\sigma}\right)$. Furthermore, when additionally the data (assuming equal prior class probability) is centered (*i.e.*, $\mathbb{E}[\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{C}_1] + \mathbb{E}[\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{C}_2] = 0$, that is, $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2$), then also $\mathbb{E}[g(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{C}_1] = -\mathbb{E}[g(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{C}_2]$ so that the optimal threshold is $\eta = 0$ and the decision is given by

$$g(\boldsymbol{x}) \underset{\mathcal{C}_1}{\overset{\mathcal{C}_2}{\gtrless}} 0.$$

Let us briefly consider the trivial extreme case when $\mathfrak{m}_1 = \mathfrak{m}_2$ and $\sigma_1 = \sigma_2$. As the two classes are not distinguishable by the distributions of their corresponding classification scores, an accurate classification is not possible in this case, and indeed we obtain

$$\varepsilon = Q(0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{0} e^{-\frac{x^2}{2}}\, \mathrm{d}x = \frac{1}{2},$$

which corresponds to making "random guesses" in a binary classification problem.



(a) Amazon review dataset ("review to score: positiv vs. negative") for two score classes with dim. $p = 400$ and $n_1 = n_2 = 100$. Histogram of the values of the classification score $g(\boldsymbol{x}) = \boldsymbol{\omega}^{\star\top}\boldsymbol{x}$ generated from 400 test samples.

(b) MNIST dataset: PCA-preprocessed classes corresponding to "4 vs. 9" with $p = 100$, and $n_1 = n_2 = 100$. Histogram of the values of the classification score $g(\boldsymbol{x}) = \boldsymbol{\omega}^{\star\top}\boldsymbol{x}$ generated from 400 test samples.

Figure 3.2: Classifications scores for real datasets

Therefore, the main challenge to apply our findings in this section is to prove that the classification scores are indeed normally distributed, and furthermore to derive the means and variance(s) characterizing their distributions. Before we turn towards this challenge, let us briefly comment on the figures. Figure 3.1 serves illustrative purposes only and exemplarily shows two normal distribution of two classes with respective normal distributions. Figure 3.2 shows the same situation for real datasets and it already anticipates our later findings: it displays both the empirical distribution of the classifications scores and the prediction by our algorithm that will only be derived below in Section 3.4. Note that our prediction is far more accurate for the MNIST dataset (b) than the Amazon review dataset (a); however, let us also remark that the latter is dataset based on text embeddings with unclear statistical properties that may be too far from our assumptions that are given below in Section 3.2.

3.1.2 Decomposing the Classification Score

As demonstrated in the previous section, the classification performance of a linear classifier is fully determined by only few scalar quantities, namely means and variances of the classification score, assuming they are univariate Gaussians. The following result shows how to decompose them in terms of the means and covariances of both the data distribution and the distribution over the hypothesis class, *i.e.,* that of the weight vector $\boldsymbol{\omega}$.

Proposition 3.3 *Let $\boldsymbol{x}, \boldsymbol{\omega} \in \mathbb{R}^p$ be two independent random vectors with means $\mathbb{E}[\boldsymbol{x}] = \bar{\boldsymbol{x}}$ and $\mathbb{E}[\boldsymbol{\omega}] = \bar{\boldsymbol{\omega}}$, and covariance* $\operatorname{Cov}(\boldsymbol{x}) = \boldsymbol{\Sigma}_x$ *and* $\operatorname{Cov}(\boldsymbol{\omega}) = \boldsymbol{\Sigma}_\omega$*, respectively. Then, it holds that*

$$\mathbb{E}[g(\boldsymbol{x})] = \mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{x}] = \bar{\boldsymbol{\omega}}^\top \mathbb{E}[\boldsymbol{x}] = \bar{\boldsymbol{\omega}}^\top \bar{\boldsymbol{x}}, \tag{3.1}$$

$$\operatorname{Var}(g(\boldsymbol{x})) = \mathbb{E}[g(\boldsymbol{x})^2] - \mathbb{E}[g(\boldsymbol{x})]^2 = \operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x) + \operatorname{tr}(\boldsymbol{\Sigma}_\omega \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top) + \operatorname{tr}(\boldsymbol{\Sigma}_x \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top). \tag{3.2}$$

Again, even though only a simple computation, let us give the proof for completeness.

Proof. Firstly, the formula for the expectation (3.1) follows easily by linearity and independence of the random vectors $\boldsymbol{x}$ and $\boldsymbol{\omega}$. Secondly, to show (3.2), by the definition of the covariance we have that $\mathbb{E}[\boldsymbol{\omega}\boldsymbol{\omega}^\top] = \boldsymbol{\Sigma}_\omega + \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top$ and, analogously, $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{\Sigma}_x + \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top$. Using this, together with independence and basic trace properties, we obtain

$$\begin{aligned}\mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\omega}] &= \operatorname{tr}\left(\mathbb{E}[\boldsymbol{\omega}\boldsymbol{\omega}^\top \boldsymbol{x}\boldsymbol{x}^\top]\right) = \operatorname{tr}\left((\boldsymbol{\Sigma}_\omega + \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top)(\boldsymbol{\Sigma}_x + \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top)\right) \\ &= \operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x) + \bar{\boldsymbol{\omega}}^\top \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top \bar{\boldsymbol{\omega}} + \operatorname{tr}(\boldsymbol{\Sigma}_\omega \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top) + \operatorname{tr}(\boldsymbol{\Sigma}_x \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top).\end{aligned}$$

By inserting our findings into the following basic equality,

$$\mathbb{E}[g(\boldsymbol{x})^2] - \mathbb{E}[g(\boldsymbol{x})]^2 = \mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{x} \cdot \boldsymbol{\omega}^\top \boldsymbol{x}] - \left(\bar{\boldsymbol{\omega}}^\top \bar{\boldsymbol{x}}\right)^2 = \mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\omega}] - \bar{\boldsymbol{\omega}}^\top \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top \bar{\boldsymbol{\omega}},$$

the second term cancels out and we obtain

$$\mathbb{E}[g(\boldsymbol{x})^2] - \mathbb{E}[g(\boldsymbol{x})]^2 = \operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x) + \operatorname{tr}(\boldsymbol{\Sigma}_\omega \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top) + \operatorname{tr}(\boldsymbol{\Sigma}_x \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top). \qquad \blacksquare$$

Remark 3.4 Again, Lemma 3.3 as stated above holds under quite general conditions. We will apply it to the weight vector $\boldsymbol{\omega} \in \mathbb{R}^p$ as the solution of LASSO calculated from the *training* data, and $\boldsymbol{x} \in \mathcal{C}_\ell$ being an (independent!) *test* datum from either class, $\ell = 1, 2$. In this case, note that the existence of $\mathbb{E}[\boldsymbol{\omega}]$ and $\operatorname{Cov}(\boldsymbol{\omega})$ is not clear a priori, but will follow from the fact that the distribution of $\boldsymbol{\omega}$ induced by the data distribution is tightly concentrated, as shown in Section 3.3. $\diamondsuit$

Remark 3.5 (Special cases) Depending on the situation, the expression (3.2) may simplify further. For instance, the second and third sumand on the right hand side may be (asymptotically) negligible compared to the first term $\operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x)$. Furthermore, when the data covariance $\boldsymbol{\Sigma}_x$ equals the identity matrix, only the diagonal elements of $\boldsymbol{\Sigma}_\omega$ are required to evaluate the expression $\operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x)$. Indeed, in this case $\operatorname{Var}\left(g(\boldsymbol{x})\right) = \operatorname{tr}(\boldsymbol{\Sigma}_\omega) = \sigma_\omega^\top \mathbb{1}_p$ with $\sigma_\omega = \mathcal{D}(\boldsymbol{\Sigma}_\omega) \in \mathbb{R}^p$ being the vector that consists of the diagonal of the matrix $\boldsymbol{\Sigma}_\omega$. $\diamondsuit$

3.2 Assumptions and Preparations

3.2.1 Setup

We consider ISTA to derive the weight vector $\boldsymbol{\omega}$ of the linear classifier and would like to predict its performance using the results from the previous section. Here, we provide a more detailed account of the basic setup already outlined in the introductory Chapter 1. Suppose we have n data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ gathered as columns in the data matrix

$$\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p \times n}.$$

We study a binary classification problem (the multi-class case can be treated by the *one versus all* technique) where the $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, \ldots, n$, belong to either of the two (nonempty) data classes $\mathcal{C}_1$ (of size n_1) and $\mathcal{C}_2$ (of size n_2) corresponding to the labels ±1, *i.e.,*

$$\boldsymbol{X} = \left[\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}\right], \qquad \boldsymbol{X}^{(\ell)} = \left[\boldsymbol{x}_1^{(\ell)}, \ldots, \boldsymbol{x}_{n_\ell}^{(\ell)}\right], \qquad \ell = 1, 2, \qquad n_1 + n_2 = n,$$

and all the labels $y_i^{(\ell)} \in \{-1, 1\}$ associated to the data points $\boldsymbol{x}_i^{(\ell)} \in \mathcal{C}_l$, $i = 1, \ldots, n_\ell$ are collected in the label vector $\boldsymbol{y} \in \mathbb{R}^n$ given by

$$\boldsymbol{y} = [y_1, \ldots, y_n] = \left[y_1^{(1)}, \ldots, y_{n_1}^{(1)}, y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right]^\top \in \{-1, 1\}^n.$$

Given a test datum $\boldsymbol{x} \in \mathbb{R}^p$, our goal is to predict its associated label $\boldsymbol{y} = \pm 1$. Even though more common in the context of regression or sparse recovery, we employ the LASSO as our loss function (during the training), which means to solve the minimization problem

$$\boldsymbol{\omega}^\star = \underset{\boldsymbol{\omega} \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1. \tag{3.3}$$

While the least-square part simply performs a regression to fit the labels well, the ℓ_1-regularizer promotes the classifier $\boldsymbol{\omega}^\star$ to be sparse, *i.e.,* to select a few features most relevant for the classification task, where the degree of sparsity depends on the hyperparameter $\lambda > 0$. We then consider the simple linear model $\boldsymbol{\omega}^{\star\top} \boldsymbol{x} = \langle \boldsymbol{\omega}^\star, \boldsymbol{x} \rangle$, *i.e.,* the inner product of the *weight vector* $\boldsymbol{\omega}^\star \in \mathbb{R}^p$ with some *feature vector* $\boldsymbol{x} \in \mathbb{R}^p$, geometrically dividing the p-dimensional space, through the *linearly separating hyperplane* $\{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{x} = 0\}$, into two distinct areas where either $\boldsymbol{\omega}^{\star\top} \boldsymbol{x} < 0$ or $\boldsymbol{\omega}^{\star\top} \boldsymbol{x} > 0$. Given linear separability in a binary classification problem, our predictor takes the form (recall (1.4) for the definition of the sign function)

$$h_\omega(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\omega}^\top \boldsymbol{x}),$$

with the so-called *decision boundary* corresponding to the separating hyperplane, that is the set $\{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{x} = 0\}$. To solve (3.3) and find its solution $\boldsymbol{\omega}^\star$, or at least a good approximation thereof, we will once again employ the iterative soft-thresholding algorithm. Even though encountered earlier already, let us recall that ISTA, with the notation specific to this chapter, is the iterative procedure

$$\begin{aligned} \boldsymbol{\omega}^0 &= \boldsymbol{0}_p, \\ \boldsymbol{\omega}^{j+1} &= S_{\tau\lambda}\left[\boldsymbol{\omega}^j + \tau \boldsymbol{X}\left(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}^j\right)\right], \qquad j \geq 1, \end{aligned} \tag{3.4}$$

where $\tau > 0$ denotes the step-size of the (projected) gradient descent step. (We comment below on questions of convergence and the choice of hyperparameters under the assumptions being used here.) However, for inference (*i.e.*, at test time), we are interested in the classification accuracy, which was already a central theme in the preparatory Section 3.1. That is, rather than predicting the loss (3.3) used for training, we employ the *0/1 test loss* given by

$$\ell\left((\boldsymbol{x},y),\boldsymbol{\omega}\right) = \begin{cases} 0, & \hat{y} = y, \\ 1, & \hat{y} \neq y. \end{cases}$$

The main goal of this chapter is to find the asymptotically precise *0/1 test loss*, *i.e.*, under the following framework. Note that the risk - recall (1.13) - is the probability of misclassification, as (for $\mathcal{A}$ being the event of misclassification)

$$\mathbb{E}_{x,y\sim\mathcal{D}}\,\ell_{0/1}(h,\boldsymbol{x},y) = \int_{\mathcal{A}} 1\,\mathrm{d}\mathbb{P} + \int_{\mathcal{A}^{\complement}} 0\,\mathrm{d}\mathbb{P} = \mathbb{P}(\mathcal{A}).$$

3.2.2 Assumptions

Throughout this chapter, we work under the following assumptions.

Assumption 3.6 (Commensurability of n and p) We assume $n > p$ and, as $n_\ell, n, p \to \infty$, asymptotically $p/n \to c_0 \in (0,1)$ and $n_\ell/n \to c_\ell > 0$ for $\ell = 1,2$. Further, we restrict ourselves to tuples with $(p,n) \in \mathbb{N}^2$ such that $c_{\text{lower}} \leq p/n \leq c_{\text{upper}}$, where of course $c_{\text{lower}} \leq c_0 \leq c_{\text{upper}}$.

While this assumption is useful to find asymptotically deterministic behavior of the involved quantities of interest later on, sample sizes and dimensions are finite in practical applications. Thus, in the algorithm derived in Section 3.4 finite quantities are used (such as n_ℓ, rather than normalizing by n to obtain c_ℓ asymptotically). Nevertheless, our experiments show a good match also in a finite-dimensional setting.

Assumption 3.7 (Distribution of $\boldsymbol{X}$ and $\boldsymbol{x}$) The columns of $\boldsymbol{X}$ are independent random vectors, and the columns $\boldsymbol{x}$ of $\boldsymbol{X}$, and $\boldsymbol{X}$ itself, are assumed to follow the concentration

$$\boldsymbol{x} \propto \mathcal{E}_2\left(1 \,|\, \mathbb{R}^p, \|\cdot\|_2\right), \qquad \boldsymbol{X} \propto \mathcal{E}_2\left(1 \,|\, \mathbb{R}^{p\times n}, \|\cdot\|_F\right),$$

following the notation introduced earlier in Definition 1.1 in the introductory Chapter 1.

Remark 3.8 This implies the existence of mean and covariance of $\boldsymbol{x}_i \in \mathcal{C}_\ell$ for $\ell = 1,2$,

$$\boldsymbol{\mu}_\ell = \mathbb{E}\left[\boldsymbol{x}_i\right], \qquad \boldsymbol{\Sigma}_\ell = \operatorname{Cov}\left(\boldsymbol{x}_i\right) = \mathbb{E}\left[\boldsymbol{x}_i\boldsymbol{x}_i^\top\right] - \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\top, \qquad i \in \{1,\ldots,n_\ell\}.$$

Furthermore, let us also introduce $\boldsymbol{C}_\ell$, sometimes called the *generalized covariance matrix*,

$$\boldsymbol{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\top = \mathbb{E}\left[\boldsymbol{x}_i\boldsymbol{x}_i^\top\right] \in \mathbb{R}^{p\times p}, \qquad \ell = 1,2, \tag{3.5}$$

that will also be used throughout the rest of this chapter. Furthermore, it will be convenient to assume that $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$, such that the two classes are only distinguishable by their means. Still, parts of the derivation below will be valid in a more general setting of an arbitrary (or diagonal) covariance, such that we will also make use of the general notation with $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, and pass to the special case when convenient. ◊

Assumption 3.9 (Growth rate of data and stepsize) For the growth rate of the hyperparameter τ and the spectral norm of the data matrix $\boldsymbol{X}$ we assume that

$$\tau = \mathcal{O}(1/n), \qquad \|\boldsymbol{X}\|_{2\to 2} = \mathcal{O}(\sqrt{n}).$$

Making the dependence on n explicit, let us put $\tau_n = \frac{1}{n}$ for the sequence $(\tau_n)_{n\in\mathbb{N}}$.

Remark 3.10 (Consistency with convergence guarantuees) Let us point out that Assumption 3.9 is consistent with the convergence condition (1.11) of ISTA, $\tau\|\boldsymbol{X}\|_{2\to 2}^2 < 2$. $\Diamond$

We have already discussed the convergence of ISTA in Section 1.2 in Chapter 1. Note that under the asymptotic regime employed here, it is not clear in which sense to define convergence of ISTA if not only $j \to \infty$, but in the triple limit $n, p, j \to \infty$ (*i.e.*, the dimension, sample size and the number of iteration tending to infinity). This is different to the notions of convergence under this asymptotic framework we have encountered otherwise, such as in the sense of convergence of measures as seen in Theorem 1.12, or in the sense of deterministic equivalents (see Definition B.5); the latter however boils down to the simpler case of convergence of scalars. For constructing a practical algorithm in Section 3.4 (which of course will always be applied to a case of finite dimension and sample size), we will take the approach of "freezing" the iteration index j and, if required, take only the limit $n, p \to \infty$ to derive deterministic equivalents of interest.

It is interesting to view the solution of ISTA as the as the solution of corresponding random fixed-point equation. A key finding, and explanation for the good theoretical results presented in Section 3.4, is the observation that the concentration assumed on the data (see Assumption 3.7 above) *induces a concentration on the weight vector* $\boldsymbol{\omega}$ (whose performance thus becomes predictable). Similar observations has been made before, like the propagation of data concentration to the weights of the softmax function [SLCT21]. We will show a similar result for the random ISTA fixed point, also establishing more generally a concentration of measure phenomenon appearing in the solution of the LASSO.

3.3 Random Fixed Point Equations: Data Concentration Induces Model Concentration

In this section, we investigate the formulation of ISTA as the random fixed-point equation

$$\boldsymbol{\omega} = \Phi(\boldsymbol{X})(\boldsymbol{\omega}) = S_{\lambda\tau}\left(\boldsymbol{\omega} + \tau\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{\omega})\right), \tag{3.6}$$

where $\Phi(\boldsymbol{X})$ is a random function whose randomness is induced by the random data matrix $\boldsymbol{X}$. Implicitly, this defines a random vector $\boldsymbol{\omega}^\star \in \mathbb{R}^p$, *i.e.*, the solution of the random fixed point equation. (Further below in Remark 3.12, we will comment on questions of measurability of $\Phi(\boldsymbol{X})$ and show that $\boldsymbol{\omega}$ is indeed a Borel-measurable function, and therefore a well-defined random vector taking values in $\mathbb{R}^p$.)

Our major goal in this section is to prove that the assumption of the random data matrix $\boldsymbol{X}$ to be concentrated (in the sense of Definition 1.1), as stated in Assumption 3.7, induces a tight concentration of the random solution of the fixed point equation, that is, of the random vector $\boldsymbol{\omega}^\star \in \mathbb{R}^p$. We will tackle this problem with the help of Theorem 3.11 below, a probabilistic variant and extension of the classical Banach fixed point theorem. It is a slight reformulation of original result [LC20, Theorem 5]; see also [Lou23, Theorem C.6]. Before we state the result, we introduce some necessary notation. For any function

$\Psi \in \mathcal{C}(\mathbb{R}^p) := \{f : \mathbb{R}^p \to \mathbb{R}^p \mid f \text{ is continuous}\}$, let us define a semi-norm on $\mathcal{C}(\mathbb{R}^p)$ by

$$\|\Psi\|_{\mathcal{B}(r)} = \sup_{\omega \in \mathcal{B}(r)} \|\Psi(\omega)\|_2 = \sup_{\|\omega\|_2 \leq r} \|\Psi(\omega)\|_2, \tag{3.7}$$

where $\mathcal{B}(r) = \{\omega \in \mathbb{R}^p : \|\omega\|_2 \leq r\|\}$ is closed the ball of radius r in $(\mathbb{R}, \|\cdot\|_2)$. We consider the situation where Ψ is a random function (more precisely, a random variable taking values in $\mathcal{C}(\mathbb{R}^p)$; for questions of measurability, again we refer to Remark 3.12 below). This evokes the random fixed-point equation $\omega = \Psi(\omega)$, like in the case $\Psi = \Phi(\boldsymbol{X})$ in (3.6). Besides basic question of existence and uniqueness of its solution, also properties like the concentration the of distribution of $\omega^\star$ (induced by the distribution over Ψ) are of interest. The following result provides a general statement in this regard. While existence and uniqueness are a straightforward consequence of Banachs fixed point theorem, under appropriate assumptions it additionally guarantees a concentration of the random solution of the fixed point equation.

Theorem 3.11 ([LC20, Theorem 5]) *Let Ψ be a random function taking values in $\mathcal{C}(\mathbb{R}^p) = \{f : \mathbb{R}^p \to \mathbb{R}^p \mid f \text{ is continuous}\}$, equipped with the semi-norm $\|\cdot\|_{\mathcal{B}(r)}$ from (3.7). Furthermore, let $\varepsilon \in (0,1)$ and $\sigma, \delta > 0$ be some parameters (possibly depending on the dimension p). Then,*

(i) if Ψ is $(1-\varepsilon)$-Lipschitz almost surely, that is, with probability one it holds that

$$\|\Psi(\omega_1) - \Psi(\omega_2)\|_2 \leq (1-\varepsilon)\|\omega_1 - \omega_2\|_2 \qquad \forall \omega_1, \omega_2 \in \mathbb{R}^p,$$

by Banach's fixed point theorem, for almost any realization of Ψ the random fixed point equation $\omega = \Psi(\omega)$ has a unique solution. Further, if additionally the following conditions are satisfied,

(ii) $\|\Psi(\mathbf{0})\|_2 \leq \delta$ almost surely, and

(iii) $\Psi \propto \mathcal{E}_2\left(\sigma \,|\, \mathcal{C}(\mathbb{R}^p), \|\cdot\|_{\mathcal{B}(\delta/\varepsilon)}\right)$,

then, the random vector $\omega^\star \in \mathbb{R}^p$, implicitly defined as the solution of the random fixed point equation $\omega = \Psi(\omega)$, satisfies the concentration $\omega^\star \propto \mathcal{E}_2(\sigma/\varepsilon \,|\, \mathbb{R}^p, \|\cdot\|_2)$.

For convenience, we have formulated the conditions in (i) and (ii) to hold almost surely. If required, the result may be restated such that the conclusion holds conditionally on (the intersection of) the corresponding events. Before we continue to apply this result in Theorem 3.13 to the ISTA-based random fixed point equation (3.6), let us briefly comment on questions of measurability arising in the context of Theorem 3.11 to justify the usage of terms like *random functions* and to show that $\omega^\star$ is actually (measurable) random vector.

Remark 3.12 Let us comment on questions of measurability arising in the context of Theorem 3.11. We will denote by $\mathcal{B}^p$ and $\mathcal{B}^{p\times n}$ the Borel-σ-algebras on $\mathbb{R}^p$ and $\mathbb{R}^{p\times n}$, respectively. Firstly, by definition, $\boldsymbol{X}$ is a random matrix, *i.e.*, it is a measurable mapping

$$\boldsymbol{X} : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathbb{R}^{p\times n}, \mathcal{B}^{p\times n}),$$

where $(\Omega, \mathcal{F}, \mathbb{P})$ denotes the underlying probability space. While generally it is not obvious (and not adressed in [Lou23]) which σ-algebra to consider on $\mathcal{C}(\mathbb{R}^p)$, in the intended case of $\Psi = \Phi(\boldsymbol{X})$, *i.e.*, for $\boldsymbol{\Phi} : \mathbb{R}^{p\times n} \to \mathcal{C}(\mathbb{R}^p), \boldsymbol{X} \mapsto (\omega \mapsto \Phi(\boldsymbol{X})(\omega))$, we employ the following *push-forward* σ-algebra $\mathcal{F}_\Psi$ on $\mathcal{C}(\mathbb{R}^p)$ and the probability measure $\mathbb{P}_\Psi$ given by

$$\mathcal{F}_\Psi = \{\mathcal{A} \subset \mathcal{C}(\mathbb{R}^p) : \Phi^{-1}(\mathcal{A}) \in \mathcal{B}^{p\times n}\},$$

$$\mathbb{P}_\Psi(\mathcal{A}) = \mathbb{P}(\Phi^{-1}(\mathcal{A})), \qquad \mathcal{A} \in \mathcal{F}_\Psi.$$

In this sense, $\Phi(\boldsymbol{X}) : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathbb{R}^{p\times n}, \mathcal{B}^{p\times n}) \to (\mathcal{C}(\mathbb{R}^p), \mathcal{F}_\Psi)$ is indeed measurable, *i.e.*, a $\mathcal{C}(\mathbb{R}^p)$-valued random variable. Furthermore, by the continuity of Φ^k for any $k \in \mathbb{N}$,

$$\begin{aligned} f : (\Omega, \mathcal{F}, \mathbb{P}) &\to (\mathbb{R}^{p\times n}, \mathcal{B}^{p\times n}) \to (\mathbb{R}^p, \mathcal{B}^n) \\ \iota \in \Omega \mapsto \quad \boldsymbol{X} &= \boldsymbol{X}(\iota) \quad \mapsto \lim_{k\to\infty} \Phi^k(\boldsymbol{X}(\iota))(\boldsymbol{\omega}_0) = \boldsymbol{\omega}^\star \quad \text{a.s.}\,, \end{aligned}$$

for any initialization $\boldsymbol{\omega}_0 \in \mathbb{R}^p$ of the fixed point iteration, and therefore, the limit $\boldsymbol{\omega}^\star$ is a (Borel-)measurable random vector taking values in $\mathbb{R}^p$, as it is a pointwise limit of a sequence of measurable functions (recall that continuous functions are also Borel-measurable). ♢

Let us now turn towards our application of the above result to the ISTA-based fixed point equation (3.6), again using the notation introduced in Definition 1.1. The following result has not been published before, and in particular was not contained in [TSSCV22]. From a mathematical perspective, we consider it the main result of this chapter. It states that, under realistic assumptions, the solution of the fixed point equation (3.6) is concentrated with an observable diameter of order $1/\sqrt{p}$ (or $1/\sqrt{n}$, as p, n are commensurable by Assumption 3.6). Intuitively this means that for large p, the solution of the fixed point equation behaves essentially deterministically (by considering φ in Definition 1.1 to be the 1-Lipschitz continuous coordinate projections). This observation also helps to explain our numerical results in the next section.

Theorem 3.13 (S., 2023) *Under the standing assumptions from the previous section, the random vector $\boldsymbol{\omega}^\star \in \mathbb{R}^p$ as implicitly defined as the solution of the random fixed point equation* (3.6), i.e., $\boldsymbol{\omega} = \Phi(\boldsymbol{X})(\boldsymbol{\omega})$ *(and thus, the solution of the corresponding LASSO problem), satisfies, conditionally on the high-probability event $\{\tau\|\boldsymbol{X}\|^2_{2\to2} \le 1-\varepsilon\}$, for $\varepsilon \in (0,1)$, the concentration*

$$\boldsymbol{\omega}^\star \propto \mathcal{E}_2\left(1/\sqrt{p}\,|\,\mathbb{R}^p, \|\cdot\|_2\right).$$

This result is an application of of Theorem 3.11. Compared to the conclusion of that result, note that in the conclusion of Theorem 3.13 we omit the dependency on $\varepsilon \in (0,1)$, which is treated as an absolute constant, as described in the very end of the proof.

Proof. Note that for our application of Theorem 3.11 to the ISTA-based random fixed point equation (3.6), the function $\Phi(\boldsymbol{X})$ plays the role of the (random) function Ψ, with the randomness being induced by the randomness of the data matrix $\boldsymbol{X}$, so that we will write $\Psi = \Phi(\boldsymbol{X})$. We have to check all the conditions from Theorem 3.11. Firstly, with regard to the Lipschitz condition (i), we obtain the following chain of inequalities

$$\begin{aligned} &\|\Phi(\boldsymbol{X})(\boldsymbol{\omega}_1) - \Phi(\boldsymbol{X})(\boldsymbol{\omega}_2)\|_2 \\ \le &\left\| S_{\lambda\tau}\left((\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_1 + \tau\boldsymbol{X}\boldsymbol{y}\right) - S_{\lambda\tau}\left((\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_2 + \tau\boldsymbol{X}\boldsymbol{y}\right)\right\|_2 \\ \le &\left\|(\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_1 + \tau\boldsymbol{X}\boldsymbol{y} - (\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_2 - \tau\boldsymbol{X}\boldsymbol{y}\right\|_2 \\ \le &\left\|(\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_1 - (\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{\omega}_2\right\|_2 \\ \le &\left\|\boldsymbol{I} - \tau\boldsymbol{X}\boldsymbol{X}^\top\right\|_{2\to2} \|\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2\|_2 \\ \le &(1-\varepsilon)\|\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2\|_2, \end{aligned}$$

where the last step holds due to part (ii) of Lemma C.2, conditionally on the high-probability event $\{\tau\|\boldsymbol{X}\|_{2\to2}^2 \leq 1-\varepsilon\}$. In the second step, we will check that condition (ii) in Theorem 3.11 is satisfied. Again replacing Ψ by $\Phi(\boldsymbol{X})$, we obtain the following simple estimate

$$\begin{aligned}\|\Phi(\boldsymbol{X})(\boldsymbol{0})\|_2 =& \|S_{\lambda\tau}(\tau\boldsymbol{X}\boldsymbol{y})\|_2 \leq \|\tau\boldsymbol{X}\boldsymbol{y}\|_2\\ \leq& \tau\|\boldsymbol{X}\|_{2\to2}\|\boldsymbol{y}\|_2 = \tau\|\boldsymbol{X}\|_{2\to2}\sqrt{n}\\ \leq& \delta\end{aligned}$$

with $\delta = \mathcal{O}(1)$ by Assumption 3.9. Finally, let us move to the third condition of Theorem 3.11. Here, we show another Lipschitz condition, namely that the mapping $\boldsymbol{X} \mapsto \Phi(\boldsymbol{X})$ is a Lipschitz continuous mapping from $(\mathbb{R}^{p\times n}, \|\cdot\|_F)$ to $(\mathcal{C}(\mathbb{R}^p), \|\cdot\|_{\mathcal{B}(\delta/\varepsilon)})$, that is

$$\sup_{\boldsymbol{\omega}\in\mathcal{B}(\delta/\varepsilon)} \|\Phi(\boldsymbol{X}_1)(\boldsymbol{\omega}) - \Phi(\boldsymbol{X}_2)(\boldsymbol{\omega})\|_2 \overset{\text{def}}{=} \|\Phi(\boldsymbol{X}_1) - \Phi(\boldsymbol{X}_1)\|_{\mathcal{B}(\delta/\varepsilon)} \lesssim \|\boldsymbol{X}_1 - \boldsymbol{X}_2\|_F,$$

where $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{p\times n}$ denote any two different realizations of the random data matrix. With the simple fact that $\|\boldsymbol{y}\|_2 = \sqrt{n}$ for the label vector $\boldsymbol{y} \in \{-1,1\}^n$, we indeed obtain

$$\begin{aligned}&\|\Phi(\boldsymbol{X}_1)(\boldsymbol{\omega}) - \Phi(\boldsymbol{X}_2)(\boldsymbol{\omega})\|_2\\ =&\left\|S_{\lambda\tau}\left(\boldsymbol{\omega} + \tau\boldsymbol{X}_1(\boldsymbol{y} - \boldsymbol{X}_1^\top\boldsymbol{\omega})\right) - S_{\lambda\tau}\left(\boldsymbol{\omega} + \tau\boldsymbol{X}_2(\boldsymbol{y} - \boldsymbol{X}_2^\top\boldsymbol{\omega})\right)\right\|_2\\ \leq&\left\|\boldsymbol{\omega} + \tau\boldsymbol{X}_1(\boldsymbol{y} - \boldsymbol{X}_1^\top\boldsymbol{\omega}) - \boldsymbol{\omega} - \tau\boldsymbol{X}_2(\boldsymbol{y} - \boldsymbol{X}_2^\top\boldsymbol{\omega})\right\|_2\\ =&\tau\left\|\boldsymbol{X}_1(\boldsymbol{y} - \boldsymbol{X}_1^\top\boldsymbol{\omega}) - \boldsymbol{X}_2(\boldsymbol{y} - \boldsymbol{X}_2^\top\boldsymbol{\omega})\right\|_2\\ =&\tau\left\|\boldsymbol{X}_1\boldsymbol{y} - \boldsymbol{X}_1\boldsymbol{X}_1^\top\boldsymbol{\omega} - \boldsymbol{X}_2\boldsymbol{y} + \boldsymbol{X}_2\boldsymbol{X}_2^\top\boldsymbol{\omega}\right\|_2\\ =&\tau\left\|\boldsymbol{X}_1\boldsymbol{y} - \boldsymbol{X}_2\boldsymbol{y} + \boldsymbol{X}_2\boldsymbol{X}_2^\top\boldsymbol{\omega} - \boldsymbol{X}_1\boldsymbol{X}_1^\top\boldsymbol{\omega}\right\|_2\\ \leq&\tau\left\|(\boldsymbol{X}_1 - \boldsymbol{X}_2)\boldsymbol{y}\right\|_2 + \tau\left\|(\boldsymbol{X}_2\boldsymbol{X}_2^\top - \boldsymbol{X}_1\boldsymbol{X}_1^\top)\boldsymbol{\omega}\right\|_2\\ \leq&\tau\sqrt{n}\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2} + \tau\left\|\boldsymbol{X}_2\boldsymbol{X}_2^\top - \boldsymbol{X}_1\boldsymbol{X}_1^\top\right\|_{2\to2}\|\boldsymbol{\omega}\|_2\\ \leq&\tau\sqrt{n}\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2} + \tau\left\|\boldsymbol{X}_2\boldsymbol{X}_2^\top - \boldsymbol{X}_1\boldsymbol{X}_2^\top + \boldsymbol{X}_1\boldsymbol{X}_2^\top - \boldsymbol{X}_1\boldsymbol{X}_1^\top\right\|_{2\to2}\|\boldsymbol{\omega}\|_2\\ \leq&\tau\sqrt{n}\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2} + \tau\left\|(\boldsymbol{X}_2 - \boldsymbol{X}_1)\boldsymbol{X}_2^\top + \boldsymbol{X}_1(\boldsymbol{X}_2^\top - \boldsymbol{X}_1^\top)\right\|_{2\to2}\|\boldsymbol{\omega}\|_2\\ \leq&\tau\sqrt{n}\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2} + 2\tau\max_{\ell=1,2}\|\boldsymbol{X}_\ell\|_{2\to2}\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2}\|\boldsymbol{\omega}\|_2\\ =&\left(\tau\sqrt{n} + 2\tau\max_{\ell=1,2}\|\boldsymbol{X}_\ell\|_{2\to2}\|\boldsymbol{\omega}\|_2\right)\left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\right\|_{2\to2}.\end{aligned}$$

Next, using the ISTA convergence condition, *i.e.*, $\tau\|\boldsymbol{X}\|_{2\to2}^2 < 2 \iff \|\boldsymbol{X}\|_{2\to2} < 2/\sqrt{\tau}$ (see also Assumption 3.9) and passing to the supremum over $\boldsymbol{\omega}$, and simply moving to the Frobenius norm (by equivalence to the spectral norm), we obtain the inequality

$$\sup_{\boldsymbol{\omega}\in\mathcal{B}(\delta/\varepsilon)} \|\Phi(\boldsymbol{X}_1)(\boldsymbol{\omega}) - \Phi(\boldsymbol{X}_2)(\boldsymbol{\omega})\|_2 \leq (\tau\sqrt{n} + 4\delta\sqrt{\tau}/\varepsilon)\|\boldsymbol{X}_1 - \boldsymbol{X}_2\|_F.$$

Thus, again by Assumption 3.9 on the growth rate of τ, we obtain the Lipschitz condition $\|\Phi(\boldsymbol{X}_1) - \Phi(\boldsymbol{X}_1)\|_{\mathcal{B}(\delta/\varepsilon)} \leq \sigma\|\boldsymbol{X}_1 - \boldsymbol{X}_2\|_F$ with Lipschitz constant $\sigma = \mathcal{O}(1/\sqrt{n})$. By Assumption 3.7 combined with the stability of concentration under Lipschitz continuous mappings (see also the comment below Definition 1.1) we have shown the implication

$$X \propto \mathcal{E}_2\left(1 \,|\, \mathbb{R}^{p\times n}, \|\cdot\|_F\right) \quad \Longrightarrow \quad \Phi(X) \propto \mathcal{E}_2\left(\sigma \,|\, \mathcal{C}(\mathbb{R}^p), \|\cdot\|_{\mathcal{B}(\delta/\varepsilon)}\right),$$

i.e., the (data) concentration of X induces a concentration of $\Phi(X)$. Thus, we have verified the condition in part (iii) of Theorem 3.11. The conclusion of that result, $\omega^\star \propto \mathcal{E}_2\left(\sigma/\varepsilon \,|\, \mathbb{R}^p, \|\cdot\|_2\right)$ with $\varepsilon \in (0,1)$ and $\sigma = \mathcal{O}(1/\sqrt{n})$, gives us the desired result $\omega^\star \propto \mathcal{E}_2\left(1/\sqrt{p} \,|\, \mathbb{R}^p, \|\cdot\|_2\right)$; recall that by Assumption 3.6, n and p are commensurable. ■

Remark 3.14 Note that the proof only makes use of the condition $\|y\|_2 = \mathcal{O}(\sqrt{n})$ on the label vector $y \in \{-1,1\}^n$ which may easily be satisfied beyond the particular classification problem under investigation here, thus potentially opening the door for application such as in regression. However, the applicability is also restricted by Assumption 3.6, *i.e.*, $n > p$ (such that $XX^\top$ is of full rank with high probability/almost surely) and the application of Lemma C.2 (ii); see also the comment below that result. ◊

3.4 Derivation of the Algorithm and Numerical Experiments

Introduction. The goal of this section is to derive, and numerically test, a practical algorithm to predict the classification accuracy of the LASSO-based classifier, based on the results from Section 3.1, in particular Lemma 3.1 and Proposition 3.3. It should be noted that the derivation is not entirely rigorous, and furthermore, there is a lack of convergence guarantees for the resulting algorithm. Nevertheless, some parts of the derivation can be proven (possibly under additional assumptions), and it is confirmed by numerical experiments.

Recall that our goal is to find the mean $\mathbb{E}[\omega^\star]$ and covariance $\mathrm{Cov}(\omega^\star)$ of the classifier $\omega^\star$ from (3.3) in order to employ the techniques from Section 3.1.1. This is a challenging task, and likely there exists no closed-form solution. Our approach is based on the construction of an iteration scheme that computes mean and covariance updates for any ISTA iteration $\omega^j \to \omega^{j+1}$ as in (3.4). This scheme resembles ISTA itself, and while its convergence properties are unclear, it yields very promising numerical results. We begin by rewriting a single iteration of ISTA, simply by introducing an intermediate z^{j+1} and separating the application of the soft-thresholding function $S_{\tau\lambda}$ as follows:

$$z^{j+1} = \omega^j - \tau XX^\top \omega^j + \tau Xy, \qquad \omega^{j+1} = S_{\tau\lambda}(z^{j+1}). \tag{3.8}$$

In the sense of keeping the iteration index j fixed, that is, while considering a single update step of ISTA, we will also omit the index j for simplicity and write in a short way

$$z = \omega - \tau XX^\top \omega + \tau Xy, \qquad \omega = S_{\tau\lambda}(z). \tag{3.9}$$

In the next step, let us rewrite z^{j+1} from (3.8) based on basic rules for matrix computations, and further pass to the mean. Using the short notation z from (3.9), we obtain

$$\begin{aligned} \mathbb{E}[z] &= \mathbb{E}\left[\omega + \tau Xy - \tau XX^\top \omega\right] \\ &= \mathbb{E}[\omega] + \tau\mathbb{E}[Xy] - \tau \sum_{i=1}^n \mathbb{E}\left[(\omega^\top x_i)x_i\right] \\ &= \mathbb{E}[\omega] + \tau \sum_{i=1}^n y_i \mu_{\pi(i)} - \tau \sum_{i=1}^n \mathbb{E}\left[(\omega^\top x_i)x_i\right]. \end{aligned} \tag{3.10}$$

Next, we can express the desired mean and covariance of $\boldsymbol{\omega}^j$ in terms of $\boldsymbol{z}^j$ as follows by

$$\mathbb{E}\left[\boldsymbol{\omega}^j\right] = \mathbb{E}\left[S_{\tau\lambda}(\boldsymbol{z}^j)\right], \tag{3.11}$$

$$\mathrm{Cov}\left(\boldsymbol{\omega}^j\right) = \mathbb{E}\left[S_{\tau\lambda}(\boldsymbol{z}^j)S_{\tau\lambda}(\boldsymbol{z}^j)^\top\right] - \mathbb{E}\left[S_{\tau\lambda}(\boldsymbol{z}^j)\right]\mathbb{E}\left[S_{\tau\lambda}(\boldsymbol{z}^j)\right]^\top. \tag{3.12}$$

It will be convenient to introduce the following functions φ and Γ for computing the expressions in (3.11) and (3.12).

$$\varphi(\lambda,\mu,\sigma) = \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S_\lambda(z)], \tag{3.13}$$

$$\Gamma(\lambda,\mu,\sigma) = \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S_\lambda^2(z)]. \tag{3.14}$$

Note that the usage of the normal distributions in the functions φ and Γ in (3.13) and (3.14) is not justified rigorously, but based on the following reasoning, that assumes an approximately Gaussian behavior of $\boldsymbol{z}^j$. By the concentration of $\boldsymbol{\omega}^\star$ with observable diameter of order $1/\sqrt{p}$ by Theorem 3.13, we may assume a similar tight concentration of $\boldsymbol{\omega}^j$ for sufficiently large j. Then, the normal distribution is justified by Theorem 1.11 in the introductory chapter, see also [TSSCV22, Lemma 2]. Let us use the opportunity to introduce another function ψ which takes a form similar to those of φ and Γ (and where a similar reasoning applies), even though its usage will become apparent later in (3.29),

$$\psi(\lambda,\mu,\sigma) = \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}\left[S_\lambda'(z)\right]. \tag{3.15}$$

(We comment on the derivative of the soft-thresholding function below in Remark 3.15.) Closed-form expressions (even though somewhat technical and requiring numerical integration) for these functions are provided in the Appendix in Section C.2. Note that the functions φ, Γ, ψ are all defined in the setting of a one-dimensional Gaussian distribution $z \sim \mathcal{N}(\mu,\sigma^2)$, but can be extended to the multivariate setting with an elementwise application. More concretely, let us consider a multivariate Gaussian random vector $\boldsymbol{v}$ in $\mathbb{R}^p$, and denote the diagonal of its covariance matrix by $\boldsymbol{\sigma}_v$, *i.e.*, , let us recall our notation

$$\boldsymbol{\sigma}_v := \mathcal{D}(\boldsymbol{\Sigma}_v) \in \mathbb{R}^p, \qquad \text{for} \qquad \boldsymbol{v} \sim \mathcal{N}(\bar{\boldsymbol{v}}, \boldsymbol{\Sigma}_v).$$

With the approximately Gaussian $\boldsymbol{z}$ in the sense of (3.9), the functions φ, ψ will be used as

$$\varphi(\lambda,\bar{\boldsymbol{z}},\boldsymbol{\sigma}_z) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\bar{\boldsymbol{z}},\boldsymbol{\Sigma}_z)}\left[S_\lambda(\boldsymbol{z})\right],$$
$$\psi(\lambda,\bar{\boldsymbol{z}},\boldsymbol{\sigma}_z) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\bar{\boldsymbol{z}},\boldsymbol{\Sigma}_z)}\left[S_\lambda'(\boldsymbol{z})\right],$$

by an elementwise application of the respective function, and the expectation. Let us recall that, by Assumption 3.7 and Remark 3.8, we consider the simple case of $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$, even though parts of the derivation will be carried out in the general setting. Therefore, by Remark 3.5, we only require $\boldsymbol{\sigma}_{\omega^\star} = \mathcal{D}(\boldsymbol{\Sigma}_{\omega^\star}) \in \mathbb{R}^p$, *i.e.*, the diagonal of the covariance $\boldsymbol{\Sigma}_{\omega^\star}$ of the random solution $\boldsymbol{\omega}^\star$, rather than the entire expression $\boldsymbol{\Sigma}_{\omega^\star} \in \mathbb{R}^{p\times p}$. Similarly, for our iterative scheme at the jth iteration, we only compute an approximation of $\boldsymbol{\sigma}_{\omega^j} = \mathcal{D}(\boldsymbol{\Sigma}_{\omega^j}) \in \mathbb{R}^p$ rather than $\boldsymbol{\Sigma}_{\omega^j} \in \mathbb{R}^{p\times p}$. Therefore, to compute the diagonal of the first summand of the right hand side in (3.12), we employ the function Γ from (3.14) as

$$\Gamma(\lambda,\bar{\boldsymbol{z}},\boldsymbol{\sigma}_z) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\bar{\boldsymbol{z}},\boldsymbol{\Sigma}_z)}\left[\mathcal{D}\left(S_\lambda(\boldsymbol{z})S_\lambda(\boldsymbol{z})^\top\right)\right] \in \mathbb{R}^p.$$

Note that indeed we only need to deal with the expressions of the type $S_\lambda(z_k)^2$ for $k = 1, \dots, p$, while the knowledge of the off-diagonal terms $S_\lambda(z_k)S_\lambda(z_l)$ with $k \neq l$ is not required. (They would be needed for the case of general covariances, which would then also require multidimensional numerical integration.) Thus, the one-dimensional form turns out to be sufficient for Γ, given in (3.14), and similar to φ and ψ in (3.13) and (3.15).

Remark 3.15 Regarding the function ψ from (3.15), note that the soft-thresholding function S_λ is differentiable almost everywhere except for the points $\pm\lambda$. We simply put

$$S'_\lambda : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto \begin{cases} 1 & \text{if } x \ \leq -\lambda, \\ 0 & \text{if } |x| \ < \lambda, \\ 1 & \text{if } x \ \geq \lambda. \end{cases} \tag{3.16}$$

Its derivative is piecewise constant, where the points $\pm\lambda$ could have been assigned to the respective other neighboring intervals where S_λ behaves linearly. $\Diamond$

Overview. Before delving into the technical details, let us provide a brief overview on the algorithm to be derived, partially reviewing results already presented, partially providing an outlook and motivating the upcoming derivations. Let us recall that the data distribution on $\boldsymbol{X}$ (characterized by their class-specific means $\boldsymbol{\mu}_\ell$ and covariances $\boldsymbol{\Sigma}_\ell$ for $\ell = 1, 2$) induces a distribution on the classifier $\boldsymbol{\omega}^\star$, which in turn induces distributions on the classification score $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}^\star$. As discussed, we assume normal distributions

$$g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega} \sim \mathcal{N}\left(\mathfrak{m}_\ell, \sigma_\ell^2\right), \qquad \boldsymbol{x} \in \mathcal{C}_\ell, \qquad \ell = 1, 2,$$

with their respective means $\mathfrak{m}_\ell$ and variances σ_ℓ^2. In the special case $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$, simply $\sigma^2 := \sigma_1^2 = \sigma_2^2$ holds. For known means and their joint variance, when we are given

$$\mathfrak{m}_1, \mathfrak{m}_2 \in \mathbb{R}, \qquad \sigma^2 > 0,$$

and for simplicity also assuming that both classes have the same size $n_1 = n_2$, Corollary 3.2 allows us to predict the classification error ε given as follows by

$$\varepsilon = \frac{1}{2} Q\left(\frac{\mathfrak{m}_2 - \mathfrak{m}_1}{\sigma^2}\right), \qquad Q(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}} \, \mathrm{d}x,$$

with Q given in (B.4) in the appendix. With Proposition 3.3 we can decompose the mean and variance of the classification score in terms of the statistics of the underlying data, and the induced distribution of the classifier, which remains to be estimated.

- For the mean updates $\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}$ (and the corresponding mean updates $\mathfrak{m}_\ell^j \to \mathfrak{m}_\ell^{j+1}$ for the classification score) we derive an iterative procedure based on deterministic equivalents in Section 3.4.1. Somewhat simplified (simply using the unknown means for $\bar{\boldsymbol{z}}^j$ as in (3.10), instead of the deterministic equivalents to be used below), this mean update consists of the steps

$$\begin{aligned} \bar{\boldsymbol{z}}^j &= \bar{\boldsymbol{\omega}}^j + \tau \sum_{i=1}^{n} y_i \boldsymbol{\mu}_{\pi(i)} - \tau \sum_{i=1}^{n} \mathbb{E}\left[(\boldsymbol{x}_i^\top \boldsymbol{\omega}^j)\boldsymbol{x}_i\right] \\ \bar{\boldsymbol{\omega}}^{j+1} &= \varphi\left(\lambda\tau, \bar{\boldsymbol{z}}^{j+1}, \sigma_z^j\right), \\ \mathfrak{m}_\ell^{j+1} &= \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j \qquad \ell = 1, 2. \end{aligned}$$

- Similarly, in Section 3.4.2 we will derive an iterative procedure for the covariance updates. Instead of full covariance updates $\mathbf{\Sigma}_{\omega^j} \to \mathbf{\Sigma}_{\omega^{j+1}}$, in the special case of $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \boldsymbol{I}_p$ it is sufficient to compute updates $\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}$ for its diagonal. Indeed, again by Proposition 3.3 we approximate $\sigma^2 = \mathrm{Var}(g(\boldsymbol{x})) = \mathbb{E}[g(\boldsymbol{x})^2] - \mathbb{E}[g(\boldsymbol{x})]^2 \approx \mathrm{tr}(\mathbf{\Sigma}_\omega \boldsymbol{I}) = \boldsymbol{\sigma}_\omega \mathbb{1}_p$, using the dominant first summand from (3.2), leaving us with the task of approximately computing $\boldsymbol{\sigma}_\omega \in \mathbb{R}^p$.

In this way, we obtain an algorithm for the mean and covariance updates, resembling ISTA itself. We begin with the mean updates in the next section.

3.4.1 Mean Updates

The goal of this section is to find at each iteration the mean of $\boldsymbol{z} = \boldsymbol{\omega} + \tau \boldsymbol{X}\left(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}\right)$, as already layed out in (3.10). We consider the more general and easier task of finding a deterministic equivalent $\bar{\boldsymbol{z}}$ of $\boldsymbol{z}$: note that expectations are always a deterministic equivalent, but may be difficult to compute, while other deterministic objects may show a similar behavior under scalar observations. (We refer to Section B.4 for the definition and a small introduction on deterministic equivalents.) Therefore (note that expectations may be replaced by deterministic equivalents later on, *i.e.,* we may sometimes use an expectation and pass to only a deterministic equivalent later on), we approximately compute

$$\bar{\boldsymbol{z}} = \bar{\boldsymbol{\omega}} + \tau \sum_{i=1}^{n} \boldsymbol{\mu}_{\pi(i)} y_i - \tau \sum_{i=1}^{n} \mathbb{E}\left[\left(\boldsymbol{x}_i^\top \boldsymbol{\omega}\right) \boldsymbol{x}_i\right], \tag{3.17}$$

where we recall that $\pi(i) \in \{1, 2\}$ denotes the class ($\mathcal{C}_1$ or $\mathcal{C}_2$, respectively) of the sample i. Then, passing from $\bar{\boldsymbol{z}}$ to $\bar{\boldsymbol{\omega}}$ with (3.11) and (3.13), and iterating this procedure will lead to an iterative algorithm that resembles ISTA itself. The intrinsic difficulty inherent to computing $\bar{\boldsymbol{z}}$ in (3.17) arises from the contained term

$$\mathbb{E}\left[\left(\boldsymbol{x}_i^\top \boldsymbol{\omega}\right) \boldsymbol{x}_i\right], \tag{3.18}$$

due to the non-trivial dependency between $\boldsymbol{\omega}$ (and therefore, also $\boldsymbol{x}_i^\top \boldsymbol{\omega}$), and $\boldsymbol{x}_i$, as $\boldsymbol{\omega}$ itself depends on $\boldsymbol{X}$, and so in particular on $\boldsymbol{x}_i$. To deal with these issues, we will employ the so-called *leave-one-out approach* to first tackle $\boldsymbol{x}_i^\top \boldsymbol{\omega}$ and to "break" the dependence between $\boldsymbol{x}_i$ and $\boldsymbol{\omega}$, followed by an application of Proposition B.2 for the expression in (3.18). Finally, this can be used for the entire sum over the terms of the form in (3.18), enabling us to compute the entire expression from (3.17).

Leave-one-out procedure. To prepare for the *leave-one-out* procedure, let us begin by rewriting the ISTA-based fixed point equations as follows. For any $i = 1, \ldots, n$, note that

$$\begin{aligned}
\boldsymbol{\omega} &= S_{\tau\lambda}\left(\boldsymbol{\omega} + \tau \boldsymbol{X}\left(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}\right)\right) &(3.19)\\
&= S_{\tau\lambda}\left(\boldsymbol{\omega} + \tau \sum_{k=1}^{n}\left(y_k - \boldsymbol{x}_k^\top \boldsymbol{\omega}\right) \boldsymbol{x}_k\right)\\
&= S_{\tau\lambda}\left(\boldsymbol{\omega} + \tau \sum_{k \neq i}^{n}\left(y_k - \boldsymbol{x}_k^\top \boldsymbol{\omega}\right) \boldsymbol{x}_k + \tau\left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\omega}\right) \boldsymbol{x}_i\right)\\
&= S_{\tau\lambda}\left(\boldsymbol{\omega} + \tau \boldsymbol{X}_{-i}\left(\boldsymbol{y}_{-i} - \boldsymbol{X}_{-i}^\top \boldsymbol{\omega}\right) + \tau\left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\omega}\right) \boldsymbol{x}_i\right). &(3.20)
\end{aligned}$$

Here, we splitted up the argument into two parts, one independent of $\boldsymbol{x}_i$ and only the other one involving $\boldsymbol{x}_i$, where $\boldsymbol{X}_{-i}$ and $\boldsymbol{y}_{-i}$ appearing in (3.20) are the data matrix and label vector deprived of the ith data point. Formally, $\boldsymbol{X}_{-i}$ and $\boldsymbol{y}_{-i}$ are defined as

$$\begin{aligned} \boldsymbol{X}_{-i} &:= [\boldsymbol{x}_1, \dots, \boldsymbol{x}_{i-1}, \boldsymbol{0}, \boldsymbol{x}_{i+1}, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{p\times n}, \\ \boldsymbol{y}_{-i} &:= [y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_n]^\top \in \mathbb{R}^n. \end{aligned}$$

Note that by inserting zeros, $\boldsymbol{X}_{-i}$ and $\boldsymbol{y}_{-i}$ keep the same size as their original counterparts $\boldsymbol{X}$ and $\boldsymbol{y}$. Next, by omitting the term in (3.20) depending on $\boldsymbol{x}_i$, we implicitly define a *leave-one-out version* $\boldsymbol{\omega}_{-i}$ of $\boldsymbol{\omega}$ as the solution of the following fixed point equation,

$$\boldsymbol{\omega}_{-i} = S_{\tau\lambda}\left(\boldsymbol{\omega}_{-i} + \tau \boldsymbol{X}_{-i}\left(\boldsymbol{y}_{-i} - \boldsymbol{X}_{-i}^\top \boldsymbol{\omega}_{-i}\right)\right), \tag{3.21}$$

which is independent of $\boldsymbol{x}_i$. Introducing a parameter $t \in [0,1]$ controlling the influence of this term in (3.20) that *does* involve $\boldsymbol{x}_i$ leads to the parameterized fixed point equation

$$\boldsymbol{\omega}_{-i}(t) = S_{\tau\lambda}\left(\boldsymbol{\omega}_{-i}(t) + \tau \boldsymbol{X}_{-i}\left(\boldsymbol{y}_{-i} - \boldsymbol{X}_{-i}^\top \boldsymbol{\omega}_{-i}(t)\right) + \underbrace{\tau t(y_i - \boldsymbol{x}_i^\top \boldsymbol{\omega}_{-i}(t))}_{=:\rho_i(t)} \boldsymbol{x}_i\right), \quad t \in [0,1],$$

which implicitly defines $\boldsymbol{\omega}_{-i}(t)$ for $i = 1, \dots, n$ and $t \in [0,1]$ as the solution of this equation. For any $i = 1, \dots, n$, this defines a path from $\boldsymbol{\omega}_{-i}(0)$ to $\boldsymbol{\omega}_{-i}(1)$ connecting the *leave-one-out* solution $\boldsymbol{\omega}_{-i}$ from (3.21) and the original fixed point $\boldsymbol{\omega}$ from (3.19), as indeed it holds that

$$\boldsymbol{\omega}_{-i}(0) = \boldsymbol{\omega}_{-i}, \qquad \text{and} \qquad \boldsymbol{\omega}_{-i}(1) = \boldsymbol{\omega},$$

and we will be interested in determining the difference between them. By the fundamental theorem of calculus applied entrywise, the difference between $\boldsymbol{\omega}$ and its leave-one-out approximation $\boldsymbol{\omega}_{-i}$ can be expressed as

$$\boldsymbol{\omega}_{\Delta_i} = \boldsymbol{\omega} - \boldsymbol{\omega}_{-i} = \boldsymbol{\omega}_{-i}(1) - \boldsymbol{\omega}_{-i}(0) = \int_0^1 \frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t}\,\mathrm{d}t \in \mathbb{R}^p, \tag{3.22}$$

where $\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t}$ is the derivative of $\boldsymbol{\omega}_{-i}$. Next, by recalling the (scalar-valued) expression

$$\rho_i(t) = \tau t(y_i - \boldsymbol{x}_i^\top \boldsymbol{\omega}_{-i}(t)) \tag{3.23}$$

as already introduced above, this derivative is given (implicitly, like $\boldsymbol{\omega}_{-i}$ itself) by

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} &= \left[\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} - \tau \boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top \frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} + \frac{\partial \rho_i(t)}{\partial t}\boldsymbol{x}_i\right] \\ &\quad \odot \left[S_{\tau\lambda}{}'\left(\boldsymbol{\omega}_{-i}(t) + \tau \boldsymbol{X}_{-i}\left(\boldsymbol{y}_{-i} - \boldsymbol{X}_{-i}^\top \boldsymbol{\omega}_{-i}(t)\right) + \rho_i(t)\boldsymbol{x}_i\right)\right], \qquad t \in [0,1], \end{aligned} \tag{3.24}$$

where $\odot$ is the Hadamard product, *i.e.*, multiplication entrywise; recall (3.16) for the derivative of the soft-thresholding function. In order to pass from this notation to a matrix-vector product, let us define the (parameterized) diagonal random matrix $\boldsymbol{D}_i(t) \in \mathbb{R}^{p\times p}$,

$$\boldsymbol{D}_i(t) := \operatorname{diag}\left[S_{\tau\lambda}{}'\left(\boldsymbol{\omega}_{-i}(t) + \tau \boldsymbol{X}_{-i}\left(\boldsymbol{y}_{-i} - \boldsymbol{X}_{-i}^\top \boldsymbol{\omega}_{-i}(t)\right) + \rho_i(t)\boldsymbol{x}_i\right)\right], \quad t \in [0,1]. \tag{3.25}$$

Note that while $\boldsymbol{D}_i(t)$ may look complicated on first glance, it is a diagonal matrix with entries in $\{0,1\}$; in other words, it can be obtained from the identity matrix of the same size by (possibly) changing some of the diagonal entries to zero. With the help of the matrix $\boldsymbol{D}_i(t)$, we can rewrite the above equation from (3.24) in a more compact form by

$$\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} = \left[\boldsymbol{D}_i(t)\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} - \tau \boldsymbol{D}_i(t)\boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top \frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} + \boldsymbol{D}_i(t)\frac{\partial \rho_i(t)}{\partial t}\boldsymbol{x}_i\right], \qquad t \in [0,1].$$

In the next step, by summarizing terms and rearranging, this can be reformulated as

$$\left[\boldsymbol{I}_p - \boldsymbol{D}_i(t) + \tau \boldsymbol{D}_i(t)\boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top\right]\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} = \boldsymbol{D}_i(t)\frac{\partial \rho_i(t)}{\partial t}\boldsymbol{x}_i, \qquad t \in [0,1].$$

Finally, this can be rewritten once again to obtain the following closed-form solution

$$\frac{\partial \boldsymbol{\omega}_{-i}(t)}{\partial t} = \frac{\partial \rho_i(t)}{\partial t}\boldsymbol{Q}_i(t)\boldsymbol{D}_i(t)\boldsymbol{x}_i, \qquad t \in [0,1], \tag{3.26}$$

with the (parameterized by $t \in [0,1]$) random matrix $\boldsymbol{Q}_i(t) \in \mathbb{R}^{p\times p}$ being defined as

$$\boldsymbol{Q}_i(t) = \left[\boldsymbol{I}_p - \boldsymbol{D}_i(t) + \tau \boldsymbol{D}_i(t)\boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top\right]^{-1}, \qquad t \in [0,1]. \tag{3.27}$$

Note that the invertibility in (3.27) follows from the fact that the matrix is positive definite (with high probability). Next, we will approximate $\boldsymbol{D}_i(t)$ and $\boldsymbol{Q}_i(t)$ from (3.25) and (3.27) by random matrices $\boldsymbol{D}$ and $\boldsymbol{Q}$ taking a simpler form, not depending on t, and derive deterministic equivalents for them, before we are able to derive a deterministic equivalent for $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$, which was the starting point of this section in (3.18).

Definition of $\boldsymbol{D}$ and $\boldsymbol{Q}$ and their deterministic equivalents. We will make the following simplification to approximate the random matrices $\boldsymbol{D}_i(t)$ and $\boldsymbol{Q}_i(t)$ from (3.25) and (3.27) by random matrices $\boldsymbol{D}$ and $\boldsymbol{Q}$ which neither depend on $t \in [0,1]$, nor on the leave-one-out index $i \in \{1,\dots,n\}$. (Note that this step is not rigorously proven.) Concretely, by inserting $t = 1$, firstly $\boldsymbol{D}_i(1)$ becomes

$$\boldsymbol{D} := \operatorname{diag}\left[S_{\tau\lambda}'\left(\boldsymbol{\omega} + \tau \boldsymbol{X}\left(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{\omega}\right)\right)\right], \tag{3.28}$$

including the ith sample and therefore justifying the notation of $\boldsymbol{D}$. A deterministic equivalent for $\boldsymbol{D}$ in (3.28), denoted as $\bar{\boldsymbol{D}}$, can be obtained easily by computing the expectation

$$\bar{\boldsymbol{D}} = \mathbb{E}\left[\boldsymbol{D}\right] = \mathbb{E}\left[\operatorname{diag}\left(S_{\tau\lambda}'(\boldsymbol{z})\right)\right] = \operatorname{diag}\left(\psi(\tau\lambda, \bar{z}, \sigma_z)\right), \tag{3.29}$$

relying on the approximately Gaussian behavior of $\boldsymbol{z}$ and using the function ψ from (3.15). Similar to the procedure for $\boldsymbol{D}_i(t)$, we plug $t = 1$ into $\boldsymbol{Q}_i(t)$ from (3.27) to obtain an expression which however still depends on i as it contains $\boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top$. Replacing this expression simply by $\boldsymbol{X}\boldsymbol{X}^\top$ (which asymptotically has the identical spectral properties), we arrive at, and, as another simplification, replacing the random matrix $\boldsymbol{D}$ by its mean (or deterministic equivalent) $\bar{\boldsymbol{D}} = \mathbb{E}\left[\boldsymbol{D}\right]$ from (3.29),

$$\boldsymbol{Q} = \left[\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \tau \bar{\boldsymbol{D}}\boldsymbol{X}\boldsymbol{X}^\top\right]^{-1}, \tag{3.30}$$

which indeed no longer depends on the leave-one-out index i, which again justifies the notation. Next, we want to derive a deterministic equivalent of $\boldsymbol{Q}$ as defined in (3.30). Let us recall that, by Assumption 3.9, we have $\tau = 1/n$, such that $\tau \boldsymbol{X}\boldsymbol{X}^\top$ coincides with $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top$. Therefore, the task of finding a deterministic equivalent $\bar{\boldsymbol{Q}}$ of $\boldsymbol{Q}$ from (3.30) is related to the task of finding deterministic equivalents of the so-called *resolvent* $(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top + z\boldsymbol{I}_p)^{-1}$ of $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top$, defined for any $z \in \mathbb{C}$ such that $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top + z\boldsymbol{I}_p$ is invertible. (Note that z may be replaced by $-z$ in the definition of the resolvent, but for our purposes this choice is convenient.) This problem has been studied in [LC18b] before, and we will adapt its solution to our situation. As also explained in [LC18b], the naive approach of simply passing to the mean with $(\frac{1}{n}\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top] + z\boldsymbol{I}_p)^{-1}$ generally does *not* provide a deterministic equivalent of the resolvent. Adapting the approach of [LC18b], we instead make the following *ansatz*, for some deterministic matrix $\boldsymbol{S}' \in \mathbb{R}^{p\times p}$ to be determined later on,

$$\boldsymbol{Q}' := \left[\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \boldsymbol{S}'\right]^{-1}. \tag{3.31}$$

Next, we will apply Lemma B.6, a very convenient tool to compute the difference $\boldsymbol{Q}' - \boldsymbol{Q}$ in terms of their respective inverse matrix, leading to favourable simplifications, since

$$\begin{aligned}\boldsymbol{Q}' - \boldsymbol{Q} =& \boldsymbol{Q}(\boldsymbol{Q}^{-1} - \boldsymbol{Q}'^{-1})\boldsymbol{Q}' \\ =& \boldsymbol{Q}\left[(\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \tau\bar{\boldsymbol{D}}\boldsymbol{X}\boldsymbol{X}^\top) - (\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \boldsymbol{S}')\right]\boldsymbol{Q}' \\ =& \boldsymbol{Q}\left(\tau\bar{\boldsymbol{D}}\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{S}'\right)\boldsymbol{Q}'.\end{aligned}$$

Passing to the mean, recalling $\tau = 1/n$ from Assumption 3.9 and further using $\tau\boldsymbol{X}\boldsymbol{X}^\top = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top$ as well as simply $\sum_{i=1}^n \frac{\boldsymbol{S}'}{n} = \boldsymbol{S}'$, we obtain the chain of equalities

$$\begin{aligned}\mathbb{E}[\boldsymbol{Q}' - \boldsymbol{Q}] =& \mathbb{E}\left[\boldsymbol{Q}\left(\tau\bar{\boldsymbol{D}}\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{S}'\right)\boldsymbol{Q}'\right] \\ =& \mathbb{E}\left[\boldsymbol{Q}\left(\frac{1}{n}\bar{\boldsymbol{D}}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i=1}^n \frac{\boldsymbol{S}'}{n}\right)\boldsymbol{Q}'\right] \\ =& \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\boldsymbol{Q}\left(\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \boldsymbol{S}'\right)\boldsymbol{Q}'\right] \\ =& \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\boldsymbol{Q}\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top\boldsymbol{Q}' - \boldsymbol{Q}\boldsymbol{S}'\boldsymbol{Q}'\right].\end{aligned} \tag{3.32}$$

Similar to before, we encounter the problem of the dependence between $\boldsymbol{Q}$ and $\boldsymbol{x}_i$ for each summand in (3.32). To break this dependence, we will next derive equivalent expressions for $\boldsymbol{Q}$ and for $\boldsymbol{Q}\bar{\boldsymbol{D}}\boldsymbol{x}_i$ in (3.32) that are due to the Sherman-Morrison-Woodbury identity from Lemma B.7. To that end, let us rewrite $\boldsymbol{Q}$, and define a leave-one-out variant of $\boldsymbol{Q}_{-i}$,

$$\begin{aligned}\boldsymbol{Q} &= \left[\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \tau\bar{\boldsymbol{D}}\sum_{\substack{k=1\\k\neq i}}^n \boldsymbol{x}_k\boldsymbol{x}_k^\top + \tau\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top\right]^{-1}, \\ \boldsymbol{Q}_{-i} &= \left[\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \tau\bar{\boldsymbol{D}}\sum_{\substack{k=1\\k\neq i}}^n \boldsymbol{x}_k\boldsymbol{x}_k^\top\right]^{-1}.\end{aligned} \tag{3.33}$$

Next, an application of Lemma B.7 in the appendix with $\boldsymbol{b} = \tau\bar{\boldsymbol{D}}\boldsymbol{x}_i \in \mathbb{R}^p$ and $\boldsymbol{c} = \boldsymbol{x}_i \in \mathbb{R}^p$, and further with $\left(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{c}^\top\right)^{-1} = \boldsymbol{Q} \in \mathbb{R}^{p\times p}$ and $\boldsymbol{A}^{-1} = \boldsymbol{Q}_{-i} \in \mathbb{R}^{p\times p}$ from (3.33) gives us

$$\boldsymbol{Q} = \boldsymbol{Q}_{-i} - \tau \frac{\boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}. \tag{3.34}$$

Multiplying (3.34) from the right with $\bar{\boldsymbol{D}}\boldsymbol{x}_i$ and straightforward simplifications yield

$$\begin{aligned}
\boldsymbol{Q}\bar{\boldsymbol{D}}\boldsymbol{x}_i =& \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i - \frac{\tau\boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i} \\
=& \left(\frac{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i - \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}\right) \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i \\
=& \frac{\boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}. \qquad (3.35)
\end{aligned}$$

Finally, we insert the expressions for $\boldsymbol{Q}$ from (3.34), and for $\boldsymbol{Q}\bar{\boldsymbol{D}}\boldsymbol{x}_i$ as provided in (3.35), into their respective appearance in the following expression from (3.32). This then yields

$$\begin{aligned}
&\boldsymbol{Q}\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}' - \boldsymbol{Q}\boldsymbol{S}'\boldsymbol{Q}' \\
=& \boldsymbol{Q}_{-i}\frac{\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}\boldsymbol{Q}' - \left(\boldsymbol{Q}_{-i} - \tau\frac{\boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}\right)\boldsymbol{S}'\boldsymbol{Q}' \\
=& \boldsymbol{Q}_{-i}\left(\frac{\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i} - \boldsymbol{S}'\right)\boldsymbol{Q}' + \tau\boldsymbol{Q}_{-i}\frac{\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}\boldsymbol{S}'\boldsymbol{Q}'.
\end{aligned}$$

Thus, passing again to the mean and once more recalling that $\tau = \frac{1}{n}$ (note that we will both τ and $\frac{1}{n}$ in parallel, depending on what is more convenient), now (3.32) reads as

$$\begin{aligned}
\mathbb{E}[\boldsymbol{Q}' - \boldsymbol{Q}] =& \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\boldsymbol{Q}_{-i}\left(\frac{\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i} - \boldsymbol{S}'\right)\boldsymbol{Q}'\right] \qquad (3.36) \\
&- \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\boldsymbol{Q}_{-i}\frac{\bar{\boldsymbol{D}}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}}{1 + \tau\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i}\boldsymbol{S}'\boldsymbol{Q}'\right]. \qquad (3.37)
\end{aligned}$$

Note that [LC18b, page 8] studies a situation very similar to ours in (3.36) and (3.37), but with $\boldsymbol{I}_p$ instead of $\bar{\boldsymbol{D}}$. We conjecture that the findings from [LC18b] can be extended to our case (recall that $\bar{\boldsymbol{D}}$, as defined in (3.29), takes a form very similar to that of the identity matrix of the same size, with possibly a few of the diagonal entries changed from one to zero), that the second term (3.37) is asymptotically negligible thanks to the additional factor $1/n$ compared to the first term (3.36), and a deterministic equivalent of $\boldsymbol{Q}$ given by

$$\bar{\boldsymbol{Q}} = \left(\boldsymbol{I}_p - \bar{\boldsymbol{D}} + \sum_{\ell=1}^{2}\frac{\tau n_\ell}{1+\kappa_\ell}\boldsymbol{C}_\ell\bar{\boldsymbol{D}}\right)^{-1}, \tag{3.38}$$

where $\kappa_\ell = \kappa_{\pi(i)} \in \mathbb{R}$ is defined, for $\boldsymbol{x}_i$ belonging to class $\mathcal{C}_\ell$ with $\ell = \pi(i) \in \{1,2\}$, by

$$\begin{aligned}\kappa_{\pi(i)} =& \tau\mathbb{E}\left[\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i\right] \approx \tau\,\mathrm{tr}(\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}\bar{\boldsymbol{C}}_\ell) \in \mathbb{R}, \qquad \ell = \pi(i) \in \{1,2\}, \qquad (3.39)\\ \boldsymbol{\kappa} =& \left[\kappa_{\pi(1)}, \dots, \kappa_{\pi(n)}\right]^\top = [\underbrace{\kappa_1, \dots, \kappa_1}_{n_1 \text{ times}}, \underbrace{\kappa_2, \dots, \kappa_2}_{n_2 \text{ times}}]^\top \in \mathbb{R}^n, \qquad n_1 + n_2 = n,\end{aligned}$$

recalling the symmetric matrix $\boldsymbol{C}_\ell$ from (3.5), and the diagonal matrix $\bar{\boldsymbol{D}}$ from (3.29). It should be pointed out that (3.38) is a fixed point equation in $\bar{\boldsymbol{Q}}$ (recall from (3.39) that κ_ℓ on the right-hand side of (3.38) indeed depends on $\bar{\boldsymbol{Q}}$) which we simply attempt to solve numerically via fixed point iterations for the practical experiments. The expression of the deterministic equivalent in (3.38) is based on the reasoning layed out in [LC18b] that $\boldsymbol{S}'$ (remember that this is a suitable deterministic matrix for finding a deterministic equivalent) should be chosen for the right-hand side of (3.36) to vanish, motivating the following choice for $\boldsymbol{S}'$ from (3.31), analogously to [LC18b],

$$\boldsymbol{S}' = \tau \sum_{i=1}^n \frac{\bar{\boldsymbol{D}}\mathbb{E}\left[\boldsymbol{x}_i\boldsymbol{x}_i^\top\right]}{1 + \tau\mathbb{E}\left[\boldsymbol{x}_i^\top \boldsymbol{Q}_{-i}\bar{\boldsymbol{D}}\boldsymbol{x}_i\right]} = \sum_{\ell=1}^2 \frac{\tau n_\ell}{1 + \tau\,\mathrm{tr}(\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}\bar{\boldsymbol{C}}_\ell)} \boldsymbol{C}_\ell \bar{\boldsymbol{D}}.$$

The deterministic equivalents $\bar{\boldsymbol{D}}$ and $\bar{\boldsymbol{Q}}$ from (3.29) and (3.38) will turn out useful for dealing with the expression $\boldsymbol{\omega}_{\Delta_i}$ from (3.22). Plugging (3.26) into the integral in (3.22), and by $\rho_i(0) = 0$ (recall (3.23) for the definition of ρ_i),

$$\begin{aligned}\boldsymbol{\omega}_{\Delta_i} =& \int_0^1 \frac{\partial \rho_i(t)}{\partial t} \boldsymbol{Q}_i(t)\boldsymbol{D}_i(t)\boldsymbol{x}_i \,\mathrm{d}t\\ \approx& \int_0^1 \frac{\partial \rho_i(t)}{\partial t} \boldsymbol{Q}\boldsymbol{D}\boldsymbol{x}_i \,\mathrm{d}t\\ =& \rho_i(1)\boldsymbol{Q}\boldsymbol{D}\boldsymbol{x}_i\\ =& \tau(y_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{Q}\boldsymbol{D}\boldsymbol{x}_i. \qquad (3.40)\end{aligned}$$

Next, recall that $\boldsymbol{\omega}_{\Delta_i} = \boldsymbol{\omega} - \boldsymbol{\omega}_{-i}$ by definition in (3.22). Combining this with our findings,

$$\begin{aligned}\boldsymbol{\omega}^\top \boldsymbol{x}_i =& \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + \boldsymbol{\omega}_\Delta^\top \boldsymbol{x}_i\\ \approx& \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + \tau(y_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i^\top \boldsymbol{D}^\top \boldsymbol{Q}^\top \boldsymbol{x}_i\\ =& \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + \tau y_i \boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i - \tau\boldsymbol{\omega}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i, \qquad (3.41)\end{aligned}$$

where we have also used that both $\boldsymbol{D}$ (as a diagonal matrix) as well as $\boldsymbol{Q}$ (as the inverse of a symmetric matrix) are symmetric (recall (3.28) and (3.30)). We will make another simplification which is not precisely justified, namely to replace the expression $\tau\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i$, which appears twice in (3.41), by its mean $\kappa_{\pi(i)} = \tau\mathbb{E}[\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i]$ from (3.39). This is based on the reasoning that, if $\boldsymbol{D}$ and $\boldsymbol{Q}$ were deterministic, the expression $\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i$ would have a tighter concentration (with an observable diameter of $1/n$) compared to $\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i$ with an observable diameter of $1/\sqrt{n}$ (or equivalently, $1/p$ and $1/\sqrt{p}$, respectively); in other words: the fluctuations in the right-hand side of (3.41) are due to the first summand, whereas the second and third summand are assumed to be essentially constant in comparison. In this way, we will obtain the following equation (3.42) provides a relation between $\boldsymbol{\omega}^\top \boldsymbol{x}_i$ and its leave-one-out version $\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i$. This will be a crucial ingredient to derive a deterministic equivalent of $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$ in the next step. Proceeding in the described manner, by inserting $\kappa_{\pi(i)}$ twice in in (3.41) and rearranging terms after $\boldsymbol{\omega}^\top \boldsymbol{x}_i$ we obtain

$$\boldsymbol{\omega}^\top \boldsymbol{x}_i \approx \frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + y_i \kappa_{\pi(i)}}{1+\kappa_{\pi(i)}}, \qquad \pi(i) \in \{1,2\}. \tag{3.42}$$

Deterministic equivalent of $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$. We return to the challenging task of dealing with the term (3.18), the main obstacle in computing the mean updates as outlined in (3.17) in the beginning of this section. We will take the approach of deriving a deterministic equivalent (again, we refer to Definition B.5 where they are formally introduced), *i.e.*, we are aiming for an expression of the following type (for any deterministic "test vector" $\boldsymbol{v} \in \mathbb{R}^p$)

$$\mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{v}^\top \boldsymbol{x}_i\right] = \boldsymbol{v}^\top \boldsymbol{a}_\ell, \qquad \boldsymbol{x}_i \in \mathcal{C}_\ell. \tag{3.43}$$

Here, the (deterministic) $\boldsymbol{a}_\ell \in \mathbb{R}^p$ provides a deterministic equivalent of $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$, that is: it provides a deterministic expression that behaves, when taking inner products with arbitrary $\boldsymbol{v} \in \mathbb{R}^p$, as if taking inner products of $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$ with $\boldsymbol{v}$, in expectation. Towards obtaining the desired expression (3.43), we insert (3.42) in the first step followed by an application of Steins identity, (B.5) from Proposition B.2, in the second step, and using $\mathbb{E}[\boldsymbol{\omega}] \approx \mathbb{E}[\boldsymbol{\omega}_{-i}]$ for large n (or $\lim_{p\to\infty} \mathbb{E}[\boldsymbol{u}^\top \boldsymbol{\omega}_{-i}] = \mathbb{E}[\boldsymbol{u}^\top \boldsymbol{\omega}]$ asymptotically for any $\boldsymbol{u} \in \mathbb{R}^p$), to obtain that

$$\begin{aligned}
\mathbb{E}\left[\left(\boldsymbol{\omega}^\top \boldsymbol{x}_i\right)\left(\boldsymbol{v}^\top \boldsymbol{x}_i\right)\right] =& \frac{\mathbb{E}_{\boldsymbol{\omega}_{-i}}\mathbb{E}_{\boldsymbol{x}_i}\left[\left(\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + y_i\kappa_\ell\right)\left(\boldsymbol{v}^\top \boldsymbol{x}_i\right)\right]}{1+\kappa_\ell} \\
=& \frac{\mathbb{E}_{\boldsymbol{\omega}_{-i}}\left[\mathbb{E}_{\boldsymbol{x}_i}\left[\left(\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i\right)\left(\boldsymbol{v}^\top \boldsymbol{x}_i\right)\right]\right] + y_i\kappa_\ell \boldsymbol{v}^\top \boldsymbol{\mu}_\ell}{1+\kappa_\ell} \\
=& \frac{\mathbb{E}_{\boldsymbol{\omega}_{-i}}\left[\boldsymbol{v}^\top \boldsymbol{\mu}_\ell \boldsymbol{\omega}_{-i}^\top \boldsymbol{\mu}_\ell + \boldsymbol{v}^\top \boldsymbol{\Sigma}_\ell \boldsymbol{\omega}_{-i}\right] + y_i\kappa_\ell \boldsymbol{v}^\top \boldsymbol{\mu}_\ell}{1+\kappa_\ell} \\
\approx& \frac{\boldsymbol{v}^\top \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}} + \boldsymbol{v}^\top \boldsymbol{\Sigma}_\ell \bar{\boldsymbol{\omega}} + y_i\kappa_\ell \boldsymbol{v}^\top \boldsymbol{\mu}_\ell}{1+\kappa_\ell} \\
=& \boldsymbol{v}^\top \left(\frac{\boldsymbol{C}_\ell \bar{\boldsymbol{\omega}} + y_i\kappa_\ell \boldsymbol{\mu}_\ell}{1+\kappa_\ell}\right),
\end{aligned} \tag{3.44}$$

recalling again $\boldsymbol{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$ from (3.5) and κ_ℓ from (3.39), for each $\ell = 1,2$. Therefore, a deterministic equivalent of $(\boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$ for $\boldsymbol{x}_i$ belonging to class $\mathcal{C}_\ell$ can be read off (compare again (3.43) for this approach to obtain deterministic equivalents) from (3.44) to be

$$\boldsymbol{a}_\ell := \frac{\boldsymbol{C}_\ell \bar{\boldsymbol{\omega}} + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell}{1+\kappa_\ell}, \qquad \ell = 1,2, \tag{3.45}$$

since $y_i = (-1)^\ell$ by the convention for the labels, -1 for class $\mathcal{C}_1$ and 1 for class $\mathcal{C}_2$, respectively. Note that, in the special case of $\boldsymbol{C}_\ell = \boldsymbol{I}_p + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$, the expression for $\boldsymbol{a}_\ell$ becomes

$$\boldsymbol{a}_\ell = \frac{\bar{\boldsymbol{\omega}} + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}} + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell}{1+\kappa_\ell} = \frac{\boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}} + (-1)^\ell \kappa_\ell}{1+\kappa_\ell} \boldsymbol{\mu}_\ell + \frac{\bar{\boldsymbol{\omega}}}{1+\kappa_\ell} \tag{3.46}$$

With the derived deterministic equivalent from (3.45), and returning to (3.17), we obtain

$$\bar{\boldsymbol{z}} = \bar{\boldsymbol{\omega}} + \tau \sum_{i=1}^n y_i \boldsymbol{\mu}_{\pi(i)} - \tau \sum_{i=1}^n \boldsymbol{a}_{\pi(i)}$$

$$
\begin{aligned}
&= \bar{\omega} + \tau \sum_{\ell=1}^{2} (-1)^{\ell} n_\ell \boldsymbol{\mu}_\ell - \tau \sum_{\ell=1}^{2} n_\ell \boldsymbol{a}_\ell \\
&= \bar{\omega} + \tau \sum_{\ell=1}^{2} n_\ell \left((-1)^{\ell} \boldsymbol{\mu}_\ell - \boldsymbol{a}_\ell \right). \qquad (3.47)
\end{aligned}
$$

3.4.2 Covariance Updates

Preparations. Besides (approximately) computing the mean $\mathbb{E}[\boldsymbol{\omega}]$, we also need to derive the covariance $\mathrm{Cov}(\boldsymbol{\omega})$ in order to apply Proposition 3.3, which provides a decomposition of the mean $\mathbb{E}[g(\boldsymbol{x})]$ and the variance $\mathrm{Var}(g(\boldsymbol{x}))$ of the classification score $g(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x}$ in terms of $\mathbb{E}[\boldsymbol{\omega}]$ and $\mathrm{Cov}(\boldsymbol{\omega})$, each for $\boldsymbol{x} \in \mathcal{C}_1$ and $\boldsymbol{x} \in \mathcal{C}_2$. (Recall Section 3.1 and Figure 3.1 for an illustration.) Again, similar as for the mean, we will derive an iterative update scheme for one iteration $\boldsymbol{z} = \boldsymbol{\omega} - \tau \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y})$ of ISTA (3.8); recall again that we discard the iteration index j in the interest of readability. Note that we first deal with $\mathrm{Cov}(\boldsymbol{z})$ in order to pass to $\mathrm{Cov}(\boldsymbol{\omega})$ as in (3.12), with the help of the function Γ introduced in (3.14). However, instead of directly computing the large covariance matrix $\mathrm{Cov}(\boldsymbol{z}) = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] - \bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top$ we consider the following random matrix

$$
\boldsymbol{M} := \boldsymbol{z}\boldsymbol{z}^\top - \bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top, \qquad (3.48)
$$

that satisfies $\mathbb{E}[\boldsymbol{M}] = \mathrm{Cov}(\boldsymbol{z})$ and is a "stochastic approximation" of the original covariance matrix as the sum of the random matrix $\boldsymbol{z}\boldsymbol{z}^\top$ and the deterministic matrix $\bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top$. Again we take the approach of deterministic equivalents, and since by Proposition 3.3 we are interested in expressions of the type $\mathrm{tr}(\boldsymbol{P}\boldsymbol{\Sigma}_\omega)$, we aim for a deterministic equivalent $\bar{\boldsymbol{M}}$ of $\boldsymbol{M}$ in the sense of

$$
\mathrm{tr}\left(\boldsymbol{P}\mathrm{Cov}(\boldsymbol{z})\right) = \mathrm{tr}\left(\boldsymbol{P}\mathbb{E}[\boldsymbol{M}]\right) = \mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{P}\boldsymbol{M}\right)\right] = \mathrm{tr}(\boldsymbol{P}\bar{\boldsymbol{M}}) \qquad (3.49)
$$

for any deterministic matrix $\boldsymbol{P}$, when $\bar{\boldsymbol{M}}$ can be read off to be a deterministic equivalent. (Note that this approach is very similar to that of (3.43) for the mean, where we considered an inner product instead of a trace.) Large parts of the upcoming derivation hold for general data covariances $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. However, in the end we need to return to the simpler setting of $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}$, as then (3.12) can be found more easily with the help of the function Γ introduced in (3.14) (and also the function φ (3.13), introduced earlier on the same occasion), which makes it possible to avoid non-diagonal entries; recall also (3.4). We begin our derivation with the first summand in (3.48) and the simple observation of

$$
\begin{aligned}
\boldsymbol{z}\boldsymbol{z}^\top &= \left[\boldsymbol{\omega} - \tau \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y})\right] \left[\boldsymbol{\omega} - \tau \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y})\right]^\top \\
&= \boldsymbol{\omega}\boldsymbol{\omega}^\top - \tau \boldsymbol{\omega}\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top + \tau \boldsymbol{\omega}\boldsymbol{y}^\top \boldsymbol{X}^\top - \tau \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{\omega}^\top + \tau \boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top \qquad (3.50) \\
&\quad + \tau^2 \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top - \tau^2 \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{y}^\top \boldsymbol{X}^\top - \tau^2 \boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top + \tau^2 \boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}^\top,
\end{aligned}
$$

by straightforward matrix computations. Similarly, now including the expectation, we obtain for $\bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top$, the second summand in (3.48), by the linearity of the expected value

$$
\begin{aligned}
\bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top &= \mathbb{E}\left[\boldsymbol{\omega} - \tau \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y})\right] \mathbb{E}\left[\boldsymbol{\omega} - \tau \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{\omega} - \boldsymbol{y})\right]^\top \\
&= \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top - \tau \bar{\boldsymbol{\omega}} \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top\right] + \tau \bar{\boldsymbol{\omega}} \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right] - \tau \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \bar{\boldsymbol{\omega}}^\top \\
&\quad + \tau \mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\right] \bar{\boldsymbol{\omega}}^\top + \tau^2 \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top\right] - \tau^2 \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right]
\end{aligned}
$$

$$- \tau^2 \mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\right] \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top\right] + \tau^2 \mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\right] \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right]. \tag{3.51}$$

Leave-one-out procedure. Before deriving a deterministic equivalent $\bar{\boldsymbol{M}}$ of $\boldsymbol{M}$, we will need, as an additional preparation, an adaption of our findings regarding the leave-one-out approach, notably of (3.42). Recall that this provided us with a link between $\boldsymbol{\omega}^\top \boldsymbol{x}_i$ and $\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i$, thus enabling to break the dependence between the ith sample and the weight vector. Here, we will need to adapt this connection to an expression of the type $\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i$ for any deterministic matrix $\boldsymbol{P} \in \mathbb{R}^{p\times p}$. We will need to be able to deal with the expression $\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i$. Recalling $\boldsymbol{\omega} = \boldsymbol{\omega}_{-i} + \boldsymbol{\omega}_{\Delta_i}$ from (3.22) and $\boldsymbol{\omega}_{\Delta_i} \approx \tau(y_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{Q}\boldsymbol{D}\boldsymbol{x}_i$ from (3.40), and further using (3.42), as well as the fact that both $\boldsymbol{D}$ (as a diagonal matrix) as well as $\boldsymbol{Q}$ (as the inverse of a symmetric matrix) are symmetric, we obtain

$$\begin{aligned}
\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i =& \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \boldsymbol{\omega}_{\Delta_i}^\top \boldsymbol{P}\boldsymbol{x}_i \\
=& \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \tau(y_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i)\boldsymbol{x}_i^\top \boldsymbol{D}^\top \boldsymbol{Q}^\top \boldsymbol{P}\boldsymbol{x}_i \\
=& \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \tau(y_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i)\,\mathrm{tr}\left(\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{P}\boldsymbol{x}_i\right) \\
=& \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \tau\left(y_i - \frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + y_i \kappa_{\pi(i)}}{1+\kappa_{\pi(i)}}\right)\mathrm{tr}\left(\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{P}\right) \\
=& \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \tau\left(\frac{y_i - \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right)\mathrm{tr}\left(\boldsymbol{P}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\right)
\end{aligned} \tag{3.52}$$

for which we also recall $\kappa_{\pi(i)} = \kappa_\ell$ from (3.39), for $\ell \in \{1,2\}$. Next, we proceed in a way similar to the derivation from (3.41) dealing with $\boldsymbol{\omega}^\top \boldsymbol{x}_i$. Here, assuming the trace expression to be almost constant and thus simply replacing it by its mean again, we obtain

$$\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i = \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \left(\frac{y_i - \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right) K_{\pi(i)}, \qquad \ell = \pi(i) \in \{1,2\}. \tag{3.53}$$

where $K_{\pi(i)}$ is given as follows by, with $\bar{\boldsymbol{D}}$ from (3.29) and $\bar{\boldsymbol{Q}}$ from (3.38) and $\boldsymbol{C}_\ell$ from (3.5),

$$K_\ell = \tau\mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{P}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\right)\right] \approx \tau\,\mathrm{tr}\left(\boldsymbol{P}\boldsymbol{C}_\ell\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}\right) = \mathrm{tr}\left(\boldsymbol{P}\boldsymbol{K}_\ell\right), \quad \ell = \pi(i) \in \{1,2\}, \tag{3.54}$$

where the matrix $\boldsymbol{K}_\ell \in \mathbb{R}^{p\times p}$ appearing in (3.54), and its diagonal $\boldsymbol{k}_\ell \in \mathbb{R}^p$, are defined as

$$\boldsymbol{K}_\ell = \tau\boldsymbol{C}_\ell\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}} \in \mathbb{R}^{p\times p}, \qquad \boldsymbol{k}_\ell = \mathcal{D}(\boldsymbol{K}_\ell) \in \mathbb{R}^p. \tag{3.55}$$

Note that K_ℓ from (3.54) is similar to $\kappa_\ell = \tau\,\mathrm{tr}(\boldsymbol{C}_\ell\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}) = \mathrm{tr}(\boldsymbol{K}_\ell)$ from (3.39), but it does contain the matrix $\boldsymbol{P}$ additionally. Further note that κ_ℓ is simply the sum of the entries of $\boldsymbol{k}_\ell$ for $\ell = 1,2$. These preparations will turn out useful for deriving deterministic equivalents in the next paragraph.

Deterministic equivalents. Now, let us turn to the task of finding a deterministic equivalent $\bar{\boldsymbol{M}}$ as stated in (3.49). We begin by rewriting the left-hand side of (3.49) using (3.50) and (3.51) by rearranging and summarizing the individual terms in the following order,

$$\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{M}\right)\right] = \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{z}\boldsymbol{z}^\top\right)\right] - \operatorname{tr}\left(\boldsymbol{P}\bar{\boldsymbol{z}}\bar{\boldsymbol{z}}^\top\right) = \sum_{k=1}^{6} \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}_k\right)\right], \tag{3.56}$$

with $\boldsymbol{M}$ from (3.48), and the individual terms $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_6$ arising in the sum given by

$$\begin{aligned}
\boldsymbol{A}_1 &= \boldsymbol{\omega}\boldsymbol{\omega}^\top - \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top, \\
\boldsymbol{A}_2 &= \tau(\boldsymbol{B}_2 + \boldsymbol{B}_2^\top), \\
\boldsymbol{A}_3 &= \tau(\boldsymbol{B}_3 + \boldsymbol{B}_3^\top), \\
\boldsymbol{A}_4 &= \tau^2\left(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\boldsymbol{\omega}^\top\boldsymbol{X}\boldsymbol{X}^\top - \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\right]\mathbb{E}\left[\boldsymbol{\omega}^\top\boldsymbol{X}\boldsymbol{X}^\top\right]\right), \\
\boldsymbol{A}_5 &= -\tau^2(\boldsymbol{B}_5 + \boldsymbol{B}_5^\top), \\
\boldsymbol{A}_6 &= \tau^2\left(\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top\boldsymbol{X}^\top - \mathbb{E}[\boldsymbol{X}\boldsymbol{y}]\mathbb{E}\left[\boldsymbol{y}^\top\boldsymbol{X}^\top\right]\right),
\end{aligned}$$

where the random matrices $\boldsymbol{B}_2, \boldsymbol{B}_3, \boldsymbol{B}_5 \in \mathbb{R}^{p\times p}$ from the definitions of $\boldsymbol{A}_2, \boldsymbol{A}_3$ and $\boldsymbol{A}_5$ are

$$\boldsymbol{B}_2 = \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\right]\bar{\boldsymbol{\omega}}^\top - \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\boldsymbol{\omega}^\top, \tag{3.57}$$

$$\boldsymbol{B}_3 = \boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top - \mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\right]\bar{\boldsymbol{\omega}}^\top, \tag{3.58}$$

$$\boldsymbol{B}_5 = \left(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\boldsymbol{y}^\top\boldsymbol{X}^\top - \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\right]\mathbb{E}\left[\boldsymbol{y}^\top\boldsymbol{X}^\top\right]\right). \tag{3.59}$$

We will compute individual deterministic equivalents of the random matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_6$, as their sum then provides us with a deterministic equivalent of the matrix $\boldsymbol{M}$ by (3.56). We will proceed analogously to the approach in (3.49) - only here applied to the individual summands. Note that the deterministic equivalents of $\boldsymbol{A}_1$ and $\boldsymbol{A}_6$ are clearly the easiest to obtain, simply by a straightforward mean computation, since they do not involve both $\boldsymbol{X}$ *and* $\boldsymbol{\omega}$. For the others, we will rely on variants of the leave-one-out approach to break the dependency between the two expressions and be able to derive deterministic equivalents, and we may also rely on asymptotics. Let us emphasize again that for simplicity we drop the iteration index j. For instance, $\bar{\boldsymbol{\omega}}$ is to be read as $\bar{\boldsymbol{\omega}}^j$, and $\boldsymbol{\Sigma}_\omega$ is to be understood as $\boldsymbol{\Sigma}_\omega^j$. Of course, the deterministic equivalents will again depend on the data means and (generalized) covariance $\boldsymbol{\mu}_\ell$, $\boldsymbol{\Sigma}$ and $\boldsymbol{C}_\ell$, $\ell = 1, 2$, which are in practice estimated from the training data. Before we proceed with the proofs, let us also introduce the notation $\boldsymbol{C}_\omega$ (similar to $\boldsymbol{C}_\ell$ as seen in (3.5) earlier) which will be useful in the sequel,

$$\boldsymbol{C}_\omega = \mathbb{E}\left[\boldsymbol{\omega}\boldsymbol{\omega}^\top\right] \in \mathbb{R}^{p\times p}. \tag{3.60}$$

Lemma 3.16 (Deterministic equivalent $\bar{\boldsymbol{A}}_1$.) *A deterministic equivalent $\bar{\boldsymbol{A}}_1$ of $\boldsymbol{A}_1$ is given by*

$$\bar{\boldsymbol{A}}_1 = \boldsymbol{\Sigma}_\omega.$$

Proof. For the summand corresponding to $\boldsymbol{A}_1$ on the right-hand side of (3.56) it holds that

$$\mathbb{E}\left[\operatorname{tr}(\boldsymbol{P}\boldsymbol{A}_1)\right] = \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\left(\boldsymbol{\omega}\boldsymbol{\omega}^\top - \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top\right)\right)\right] = \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\boldsymbol{\omega}\boldsymbol{\omega}^\top - \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top\right]\right) = \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{\Sigma}_\omega\right).$$

Thus, we can read off - compare again (3.49) - a deterministic equivalent $\bar{\boldsymbol{A}}_1 = \boldsymbol{\Sigma}_\omega$. ■

Lemma 3.17 (Deterministic equivalent $\bar{\boldsymbol{A}}_2$.) *A deterministic equivalent $\bar{\boldsymbol{A}}_2$ of $\boldsymbol{A}_2$ is given by*

$$\bar{\boldsymbol{A}}_2 = \tau(\bar{\boldsymbol{B}}_2 + \bar{\boldsymbol{B}}_2^\top),$$

where the deterministic equivalent $\bar{\boldsymbol{B}}_2$ of $\boldsymbol{B}_2$ from (3.57) *is given as follows by*

$$\bar{\boldsymbol{B}}_2 = \sum_{\ell=1}^{2} n_\ell \left(\bar{\boldsymbol{\omega}} \boldsymbol{a}_\ell^\top + \frac{\boldsymbol{C}_\ell \boldsymbol{C}_\omega + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top}{1+\kappa_\ell} + \frac{\kappa_\ell - \operatorname{tr}(\boldsymbol{C}_\omega \boldsymbol{C}_\ell)}{(1+\kappa_\ell)^2} \boldsymbol{K}_\ell \right).$$

Proof. We only have to derive the deterministic equivalent $\bar{\boldsymbol{B}}_2$ of $\boldsymbol{B}_2$ (as the statement for $\bar{\boldsymbol{A}}_2$ then immediately follows from their connection $\boldsymbol{A}_2 = \tau(\boldsymbol{B}_2 + \boldsymbol{B}_2^\top)$), *i.e.*, we consider

$$\mathbb{E}\left[\operatorname{tr}(\boldsymbol{P}\boldsymbol{B}_2)\right] = \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \bar{\boldsymbol{\omega}}^\top\right) - \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{\omega}^\top\right]\right). \tag{3.61}$$

We begin by considering the first summand from the right hand side of (3.61). With the help of the deterministic equivalent $\boldsymbol{a}_\ell$, $\ell = 1, 2$, from (3.45), it is straightforward to obtain

$$\operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \bar{\boldsymbol{\omega}}^\top\right) = \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\sum_{k=1}^{n} (\boldsymbol{\omega}^\top \boldsymbol{x}_k)\boldsymbol{x}_k\right] \bar{\boldsymbol{\omega}}^\top\right) = \sum_{\ell=1}^{2} n_\ell \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{a}_\ell \bar{\boldsymbol{\omega}}^\top\right), \tag{3.62}$$

such that we can immediately find the deterministic equivalent $\sum_{\ell=1}^{2} n_\ell \boldsymbol{a}_\ell \bar{\boldsymbol{\omega}}^\top$ for this part of (3.61). Next, we move on to consider the second summand from (3.61). By inserting both the expression for $\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i$ from (3.53), and the expression for $\boldsymbol{\omega}^\top \boldsymbol{x}_i$ from (3.42), both of which allowing to break the dependency due to the leave-one-out approach, we obtain

$$\begin{aligned}
\operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{\omega}^\top\right]\right) =& \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \\
=& \sum_{i=1}^{n} \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i \boldsymbol{\omega}^\top \boldsymbol{x}_i\right] \\
=& \sum_{i=1}^{n} \mathbb{E}\left[\left(\boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \frac{y_i K_{\pi(i)} - K_{\pi(i)} \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right)\left(\frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + y_i \kappa_{\pi(i)}}{1+\kappa_{\pi(i)}}\right)\right] \\
=& \sum_{i=1}^{n} \mathbb{E}\left[\frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right] + \sum_{i=1}^{n} \mathbb{E}\left[\frac{y_i \kappa_{\pi(i)} \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right] &(3.63) \\
&+ \sum_{i=1}^{n} \mathbb{E}\left[\left(\frac{y_i K_{\pi(i)} - K_{\pi(i)} \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right)\left(\frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i + y_i \kappa_{\pi(i)}}{1+\kappa_{\pi(i)}}\right)\right] &(3.64)
\end{aligned}$$

After developing terms in (3.64) next, the quantity considered in the previous chain of equalities can be written as the sum of all the individual summands arising in (3.63) and (3.64). Towards finding a deterministic equivalent, we will next rewrite them as follows:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbb{E}\left[\frac{\boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right] &= \sum_{\ell=1}^{2} n_\ell \frac{\operatorname{tr}(\boldsymbol{P}\boldsymbol{C}_\ell \boldsymbol{C}_\omega)}{1+\kappa_\ell}, \\
\sum_{i=1}^{n} \mathbb{E}\left[\frac{y_i \kappa_{\pi(i)} \boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right] &= \sum_{\ell=1}^{2} n_\ell \frac{(-1)^\ell \kappa_\ell \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top\right)}{1+\kappa_\ell},
\end{aligned}$$

$$\sum_{i=1}^{n}\mathbb{E}\left[\frac{y_i K_{\pi(i)}\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{\left(1+\kappa_{\pi(i)}\right)^2}\right] = \sum_{\ell=1}^{2} n_\ell \frac{(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell}{(1+\kappa_\ell)^2}K_\ell,$$

$$\sum_{i=1}^{n}\mathbb{E}\left[\frac{y_i^2 K_{\pi(i)}\kappa_{\pi(i)}}{\left(1+\kappa_{\pi(i)}\right)^2}\right] = \sum_{\ell=1}^{2} n_\ell \frac{\kappa_\ell}{(1+\kappa_\ell)^2}K_\ell,$$

$$-\sum_{i=1}^{n}\mathbb{E}\left[\frac{K_{\pi(i)}\left(\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i\right)^2}{\left(1+\kappa_{\pi(i)}\right)^2}\right] = -\sum_{\ell=1}^{2} n_\ell \frac{\operatorname{tr}\left(\boldsymbol{C}_\omega \boldsymbol{C}_\ell\right)}{(1+\kappa_\ell)^2}K_\ell,$$

$$-\sum_{i=1}^{n}\mathbb{E}\left[\frac{y_i \kappa_{\pi(i)} K_{\pi(i)}\boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{\left(1+\kappa_{\pi(i)}\right)^2}\right] = -\sum_{\ell=1}^{2} n_\ell \frac{(-1)^\ell \kappa_\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell}{(1+\kappa_\ell)^2}K_\ell.$$

Let us also recall $K_\ell = \operatorname{tr}(\boldsymbol{P}\boldsymbol{K}_\ell)$ with $\boldsymbol{K}_\ell = \tau \boldsymbol{C}_\ell \bar{\boldsymbol{D}}\bar{\boldsymbol{Q}} \in \mathbb{R}^{p\times p}$ introduced earlier in (3.54) and (3.55) for $\ell = 1, 2$, as well as $\boldsymbol{C}_\omega$ from (3.60). (Note that we pass from $\boldsymbol{C}_{\omega_{-i}}$ to $\boldsymbol{C}_\omega$ as they have asymptotically the same spectral properties; compare passing from $\boldsymbol{X}_{-i}\boldsymbol{X}_{-i}^\top$ to $\boldsymbol{X}\boldsymbol{X}^\top$ just below (3.29)). Combining our findings from (3.62) and the expressions derived for the individual summands in (3.63) and (3.64), we obtain the claimed deterministic equivalent of $\boldsymbol{B}_2$, and thus of $\boldsymbol{A}_2$. ■

Lemma 3.18 (Deterministic equivalent $\bar{\boldsymbol{A}}_3$.) *A deterministic equivalent $\bar{\boldsymbol{A}}_3$ of $\boldsymbol{A}_3$ is given by*

$$\bar{\boldsymbol{A}}_3 = \tau(\bar{\boldsymbol{B}}_3 + \bar{\boldsymbol{B}}_3^\top),$$

where the deterministic equivalent $\bar{\boldsymbol{B}}_3$ of $\boldsymbol{B}_3$ from (3.58) *is given as follows by*

$$\bar{\boldsymbol{B}}_3 = \tau \sum_{\ell=1}^{2} n_\ell \frac{\left(1-(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell\right)}{1+\kappa_\ell} \boldsymbol{C}_\ell \bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}.$$

Proof. We only have to derive the deterministic equivalent $\bar{\boldsymbol{B}}_3$ of $\boldsymbol{B}_3$ (as the statement for $\bar{\boldsymbol{A}}_3$ then immediately follows from their connection $\boldsymbol{A}_3 = \tau(\boldsymbol{B}_3 + \boldsymbol{B}_3^\top)$), *i.e.*, we consider

$$\begin{aligned}\mathbb{E}\left[\operatorname{tr}(\boldsymbol{P}\boldsymbol{B}_3)\right] =& \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\left(\boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top - \mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\right]\bar{\boldsymbol{\omega}}^\top\right)\right)\right] \\ =& \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top\right)\right] - \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{X}\boldsymbol{y}\bar{\boldsymbol{\omega}}^\top\right)\right]. \end{aligned} \tag{3.65}$$

For the second summand in (3.65), we can directly compute the mean as follows by

$$\mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\bar{\boldsymbol{\omega}}^\top\right] = \left(\sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell\right)\bar{\boldsymbol{\omega}}^\top = \sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top.$$

Therefore, we obtain the following expression for the second summand fom (3.65),

$$\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{X}\boldsymbol{y}\bar{\boldsymbol{\omega}}^\top\right)\right] = \operatorname{tr}\left(\boldsymbol{P}\sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top\right). \tag{3.66}$$

Next, we consider the first summand in (3.65), which poses challanges due to the dependency of $\boldsymbol{X}$ and $\boldsymbol{\omega}$, requiring to use (3.53). Here, we obtain (see again (3.54) for K_ℓ)

$$
\begin{aligned}
\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{X}\boldsymbol{y}\boldsymbol{\omega}^\top\right)\right] =& \sum_{i=1}^{n} y_i \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{P}\boldsymbol{x}_i\right] \\
=& \sum_{i=1}^{n} y_i \mathbb{E}\left[\boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i + \left(\frac{y_i - \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right) K_{\pi(i)}\right] \\
=& \sum_{i=1}^{n} y_i \mathbb{E}\left[\boldsymbol{\omega}_{-i}^\top \boldsymbol{P}\boldsymbol{x}_i\right] + \sum_{i=1}^{n} y_i \mathbb{E}\left[\left(\frac{y_i - \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i}{1+\kappa_{\pi(i)}}\right) K_{\pi(i)}\right] \\
=& \sum_{i=1}^{n} \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{x}_i\boldsymbol{\omega}_{-i}^\top\right)\right] + \sum_{i=1}^{n} y_i \mathbb{E}\left[\frac{(y_i^2 - y_i \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i)}{1+\kappa_{\pi(i)}} K_{\pi(i)}\right]. \qquad (3.67)
\end{aligned}
$$

While the first term in (3.67) cancels for large n with (3.66), as indeed it holds that

$$
\lim_{n\to\infty} \sum_{i=1}^{n} y_i \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{x}_i\boldsymbol{\omega}_{-i}^\top\right)\right] = \operatorname{tr}\left(\boldsymbol{P}\sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top\right),
$$

it remains to consider the second only the summand from (3.67) to derive the deterministic equivalent. For this term, we get as $y_i^2 = 1$ and $y_i = (-1)^\ell$ with $\ell = \pi(i)$ as usual,

$$
\lim_{n\to\infty} \sum_{i=1}^{n} \mathbb{E}\left[\frac{(y_i^2 - y_i \boldsymbol{\omega}_{-i}^\top \boldsymbol{x}_i)}{1+\kappa_{\pi(i)}} K_{\pi(i)}\right] = \sum_{\ell=1}^{2} n_\ell \frac{(1-(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell)}{1+\kappa_\ell} K_\ell,
$$

where we recall $K_\ell = \tau \operatorname{tr}(\boldsymbol{P}\boldsymbol{K}_\ell)$ from (3.54), with $\boldsymbol{K}_\ell = \tau \boldsymbol{C}_\ell \bar{\boldsymbol{D}} \bar{\boldsymbol{Q}}$ from (3.55). Therefore

$$
\begin{aligned}
\bar{\boldsymbol{B}}_3 =& \sum_{\ell=1}^{2} n_\ell \frac{(1-(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell)}{1+\kappa_\ell} \boldsymbol{K}_\ell \\
=& \tau \sum_{\ell=1}^{2} n_\ell \frac{(1-(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell)}{1+\kappa_\ell} \boldsymbol{C}_\ell \bar{\boldsymbol{D}} \bar{\boldsymbol{Q}},
\end{aligned}
$$

which is the desired deterministic equivalent of $\boldsymbol{B}_3$, and thus of $\boldsymbol{A}_3$. ■

Lemma 3.19 (Deterministic equivalent $\bar{\boldsymbol{A}}_4$.) *A deterministic equivalent $\bar{\boldsymbol{A}}_4$ of $\boldsymbol{A}_4$ is given by*

$$
\begin{aligned}
\bar{\boldsymbol{A}}_4 =& \tau^2 \sum_{\ell,\ell'=1}^{2} \frac{n_\ell n_{\ell'} \boldsymbol{\Sigma}_{\ell'} \boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell}{(1+\kappa_\ell)(1+\kappa_{\ell'})} \\
&+ \tau^2 \sum_{\ell=1}^{2} n_\ell \frac{\operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell) + 2(-1)^\ell \kappa_\ell \boldsymbol{\omega}^\top \boldsymbol{\mu}_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} \boldsymbol{C}_\ell \\
&+ \sum_{\ell=1}^{2} n_\ell \left(\boldsymbol{a}_\ell \boldsymbol{a}_\ell^\top + \frac{\operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell) + 2\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell (-1)^\ell \kappa_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top\right).
\end{aligned}
$$

Proof. Similar to before, up to normalization by τ^2 the summand with $\boldsymbol{A}_4$ is given by

$$
\begin{aligned}
&\frac{1}{\tau^2}\mathbb{E}\left[\operatorname{tr}(\boldsymbol{P}\boldsymbol{A}_4)\right] \\
=& \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\left(\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top - \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\omega}\right] \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{X}\boldsymbol{X}^\top\right]\right)\right)\right]. \qquad (3.68)
\end{aligned}
$$

We begin by considering the first term in (3.68). By rewriting the expression and splitting up the sum, using basic properties of the trace and, as before, $\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega} = \sum_{k=1}^n(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k$ and a similar expression for its transpose $\boldsymbol{\omega}^\top\boldsymbol{X}\boldsymbol{X}^\top = \sum_{k=1}^n(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k^\top$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\boldsymbol{\omega}^\top\boldsymbol{X}\boldsymbol{X}^\top\right)\right] \\
&= \sum_{k=1}^n \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\boldsymbol{x}_k\boldsymbol{x}_k^\top\right)\right] + \sum_{\substack{k,l=1\\k\neq l}}^n \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}(\boldsymbol{\omega}^\top\boldsymbol{x}_k)(\boldsymbol{\omega}^\top\boldsymbol{x}_l)\boldsymbol{x}_k\boldsymbol{x}_l^\top\right)\right] \\
&= \sum_{k=1}^n \mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\boldsymbol{x}_k^\top\boldsymbol{P}\boldsymbol{x}_k\right] + \sum_{\substack{k,l=1\\k\neq l}}^n \mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)(\boldsymbol{\omega}^\top\boldsymbol{x}_l)\boldsymbol{x}_l^\top\boldsymbol{P}\boldsymbol{x}_k\right]. \qquad (3.69)
\end{aligned}$$

Next, let us move to the second term in (3.68). Again, rewriting the arising expressions and splitting up the sum, and further using the deterministic equivalent (3.45), we obtain

$$\begin{aligned}
&\operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\omega}\right]\mathbb{E}\left[\boldsymbol{\omega}^\top\boldsymbol{X}\boldsymbol{X}^\top\right]\right) \\
&= \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\sum_{k=1}^n(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[\sum_{l=1}^n(\boldsymbol{\omega}^\top\boldsymbol{x}_l)\boldsymbol{x}_l^\top\right]\right) \\
&= \sum_{k,l=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_l)\boldsymbol{x}_l^\top\right]\right) \\
&= \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right) + \sum_{\substack{k,l=1\\k\neq l}}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_l)\boldsymbol{x}_l^\top\right]\right) \\
&= \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right) + \sum_{\substack{k,l=1\\k\neq l}}^n \boldsymbol{a}_{\pi(l)}^\top\boldsymbol{P}\boldsymbol{a}_{\pi(k)}, \qquad (3.70)
\end{aligned}$$

with $\boldsymbol{a}_\ell$ from (3.45) for $\ell = 1, 2$ in the last step. In the next step, we will subtract the first term in (3.70) from the first term in (3.69). With the help of Steins identity - Proposition B.2 in the appendix - and recalling $\boldsymbol{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}\boldsymbol{\mu}^\top$ from (3.5) for either class $\ell = 1, 2$,

$$\begin{aligned}
&\sum_{k=1}^n \mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\boldsymbol{x}_k^\top\boldsymbol{P}\boldsymbol{x}_k\right] - \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right) \\
&= \sum_{k=1}^n \mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\right]\operatorname{tr}(\boldsymbol{P}\boldsymbol{\Sigma}_{\pi(k)}) \\
&\quad + \sum_{k=1}^n \mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\right]\operatorname{tr}\left(\boldsymbol{P}\boldsymbol{\mu}_{\pi(k)}\boldsymbol{\mu}_{\pi(k)}^\top\right) - \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right) \\
&= \sum_{k=1}^n \mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)^2\right]\operatorname{tr}(\boldsymbol{P}\boldsymbol{\Sigma}_{\pi(k)}) \qquad (3.71) \\
&\quad + \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)\boldsymbol{\mu}_{\pi(k)}\left(\boldsymbol{\omega}^\top\boldsymbol{x}_k\right)\boldsymbol{\mu}_{\pi(k)}^\top\right]\right) \\
&\quad - \sum_{k=1}^n \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top\boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right).
\end{aligned}$$

For (3.71), we obtain an expression convenient for finding a deterministic equivalent,

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_i)^2\right]\operatorname{tr}(\boldsymbol{P\Sigma}_{\pi(k)}) =& \frac{1}{n}\sum_{\ell=1}^{2} n_\ell \mathbb{E}\left[\left(\boldsymbol{\omega}^\top \boldsymbol{x}_1^{(\ell)}\right)^2\right]\operatorname{tr}(\boldsymbol{P\Sigma}_\ell)\\ &\to \sum_{\ell=1}^{2} c_\ell \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell\right)+2\bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell(-1)^\ell\kappa_\ell+\kappa_\ell^2}{(1+\kappa_\ell)^2}\operatorname{tr}(\boldsymbol{P\Sigma}_\ell). \quad (3.72)\end{aligned}$$

as $n, p \to \infty$, where we have also made use of the following consequence of (3.42),

$$\begin{aligned}\mathbb{E}\left[\left(\boldsymbol{\omega}^\top \boldsymbol{x}_i\right)^2\right] =& \frac{\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{\omega}_{-i}\boldsymbol{\omega}_{-i}^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top\right)\right]+2\mathbb{E}\left[\boldsymbol{\omega}_{-i}\right]^\top\mathbb{E}\left[\boldsymbol{x}_i\right]y_i\kappa_{\pi(i)}+\kappa_{\pi(i)}^2}{\left(1+\kappa_{\pi(i)}\right)^2}\\ \approx& \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell\right)+2\bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell(-1)^\ell\kappa_\ell+\kappa_\ell^2}{(1+\kappa_\ell)^2}, \qquad \ell=\pi(i). \quad (3.73)\end{aligned}$$

Next, we consider the second and third term from just below (3.71), *i.e.*, the expression

$$\begin{aligned}&\sum_{k=1}^{n}\operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\left(\boldsymbol{\omega}^\top \boldsymbol{x}_k\right)^2\boldsymbol{\mu}_{\pi(k)}\boldsymbol{\mu}_{\pi(k)}^\top\right]\right)-\sum_{k=1}^{n}\operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_k)\boldsymbol{x}_k\right]\mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_k)\boldsymbol{x}_k\right]^\top\right)\\ &=:I-II. \quad (3.74)\end{aligned}$$

For the first term I in (3.74), similar to (3.72) and using once again (3.73), we obtain

$$\begin{aligned}I =& \sum_{k=1}^{n}\operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\left(\boldsymbol{\omega}^\top \boldsymbol{x}_k\right)^2\right]\boldsymbol{\mu}_{\pi(k)}\boldsymbol{\mu}_{\pi(k)}^\top\right)\\ =& \sum_{\ell=1}^{2} n_\ell \operatorname{tr}\left(\boldsymbol{P}\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell\right)+2\bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell(-1)^\ell\kappa_\ell+\kappa_\ell^2}{(1+\kappa_\ell)^2}\boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\top\right),\end{aligned}$$

such that we can read off a deterministic equivalent corresponding to this part, namely

$$\sum_{\ell=1}^{2} n_\ell \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell\right)+2\bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell(-1)^\ell\kappa_\ell+\kappa_\ell^2}{(1+\kappa_\ell)^2}\boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\top. \quad (3.75)$$

Next, let us consider the second term II from (3.74) above, for which we can immediately find a deterministic equivalent using of with $\boldsymbol{a}_\ell$ from (3.45) for $\ell = 1, 2$ as follows,

$$\sum_{\ell=1}^{2} n_\ell \boldsymbol{a}_\ell\boldsymbol{a}_\ell^\top. \quad (3.76)$$

After dealing with the first summands from each (3.69) and (3.70), we will next - keeping in mind (3.68) - subtract the second term in (3.70) from the second term in (3.69). To that end, we first approximate each single summand from the second term in (3.69) analogously to [SLCT21, p. 16] as follows by, for any $k, l \in \{1, \ldots, n\}$ with $k \neq l$,

$$\mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_k)(\boldsymbol{\omega}^\top \boldsymbol{x}_l)\boldsymbol{x}_l^\top\boldsymbol{P}\boldsymbol{x}_k\right]$$

$$
\begin{aligned}
&\approx \left(\frac{\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_{\pi(k)} + y_{\pi(k)}\kappa_{\pi(k)}}{1+\kappa_{\pi(k)}} \boldsymbol{\mu}_{\pi(k)} + \frac{\boldsymbol{\Sigma}_{\pi(k)}\bar{\boldsymbol{\omega}}}{1+\kappa_{\pi(k)}} \right)^\top \boldsymbol{P} \left(\frac{\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_{\pi(l)} + y_{\pi(l)}\kappa_{\pi(l)}}{1+\kappa_{\pi(l)}} \boldsymbol{\mu}_{\pi(l)} + \frac{\boldsymbol{\Sigma}_{\pi(l)}\bar{\boldsymbol{\omega}}}{1+\kappa_{\pi(l)}} \right) \\
&\qquad + \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_{\pi(k)} \boldsymbol{P} \boldsymbol{\Sigma}_{\pi(l)} \boldsymbol{\Sigma}_{\omega} \right)}{\left(1+\kappa_{\pi(k)}\right)\left(1+\kappa_{\pi(l)}\right)} \\
&= \left(\frac{\boldsymbol{C}_{\pi(k)}\bar{\boldsymbol{\omega}} + (-1)^{\pi(k)}\kappa_{\pi(k)}\boldsymbol{\mu}_{\pi(k)}}{1+\kappa_{\pi(k)}} \right)^\top \boldsymbol{P} \left(\frac{\boldsymbol{C}_{\pi(l)}\bar{\boldsymbol{\omega}} + (-1)^{\ell}\kappa_{\pi(l)}\boldsymbol{\mu}_{\pi(l)}}{1+\kappa_{\pi(l)}} \right) \\
&\qquad + \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_{\pi(k)} \boldsymbol{P} \boldsymbol{\Sigma}_{\pi(l)} \boldsymbol{\Sigma}_{\omega} \right)}{\left(1+\kappa_{\pi(k)}\right)\left(1+\kappa_{\pi(l)}\right)}.
\end{aligned}
$$

Next, we pass to the sum and obtain the following expression, also using (3.45),

$$
\sum_{\substack{k,l=1 \\ k\neq l}}^{n} \mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_k)(\boldsymbol{\omega}^\top \boldsymbol{x}_l)\boldsymbol{x}_l^\top \boldsymbol{P} \boldsymbol{x}_k \right] = \sum_{\substack{k,l=1 \\ k\neq l}}^{n} \boldsymbol{a}_{\pi(l)}^\top \boldsymbol{P} \boldsymbol{a}_{\pi(k)} + \sum_{\substack{k,l=1 \\ k\neq l}}^{n} \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_{\pi(k)} \boldsymbol{P} \boldsymbol{\Sigma}_{\pi(l)} \boldsymbol{\Sigma}_{\omega} \right)}{\left(1+\kappa_{\pi(k)}\right)\left(1+\kappa_{\pi(l)}\right)}. \tag{3.77}
$$

Finally, we subtract the second term in (3.70) from the second term in (3.69), as just rewritten in (3.77), which after a straightforward cancellation leaves us with the difference

$$
\begin{aligned}
&\sum_{\substack{k,l=1 \\ k\neq l}}^{n} \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_{\pi(k)} \boldsymbol{P} \boldsymbol{\Sigma}_{\pi(l)} \boldsymbol{\Sigma}_{\omega} \right)}{\left(1+\kappa_{\pi(k)}\right)\left(1+\kappa_{\pi(l)}\right)} \\
=& n_1(n_1-1)\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_1 \boldsymbol{P} \boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_\omega\right)}{(1+\kappa_1)(1+\kappa_1)} + n_1 n_2 \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_1 \boldsymbol{P} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_\omega\right)}{(1+\kappa_1)(1+\kappa_2)} \\
&\quad + n_1 n_2 \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_2 \boldsymbol{P} \boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_\omega\right)}{(1+\kappa_2)(1+\kappa_1)} + n_2(n_2-1)\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_2 \boldsymbol{P} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_\omega\right)}{(1+\kappa_2)(1+\kappa_2)} \\
\approx& \sum_{\ell,\ell'=1}^{2} \frac{n_\ell n_{\ell'} \operatorname{tr}\left(\boldsymbol{\Sigma}_{\ell'} \boldsymbol{P} \boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell\right)}{(1+\kappa_\ell)(1+\kappa_{\ell'})},
\end{aligned} \tag{3.78}
$$

for sufficiently large n_1, n_2 (or equality asymptotically). Finally, combining our findings from (3.72), (3.75), (3.76) and (3.78), and again inserting the factor τ^2 to make up for the normalization in (3.68), we obtain the deterministic equivalent $\bar{\boldsymbol{A}}_4$ of $\boldsymbol{A}_4$,

$$
\begin{aligned}
\bar{\boldsymbol{A}}_4 =& \tau^2 \sum_{\ell,\ell'=1}^{2} \frac{n_\ell n_{\ell'} \boldsymbol{\Sigma}_{\ell'} \boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell}{(1+\kappa_\ell)(1+\kappa_{\ell'})} + \tau^2 \sum_{\ell=1}^{2} n_\ell \mathcal{E}_\ell \boldsymbol{C}_\ell \\
&\quad + \sum_{\ell=1}^{2} n_\ell \left(\boldsymbol{a}_\ell \boldsymbol{a}_\ell^\top + \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell\right) + 2\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell (-1)^\ell \kappa_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \right),
\end{aligned}
$$

where $\mathcal{E}_\ell$ is defined as an abbreviation for the expression from (3.72),

$$
\mathcal{E}_\ell := \mathbb{E}\left[(\boldsymbol{\omega}^\top \boldsymbol{x}_i)^2 \right] = \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell\right) + 2(-1)^\ell \kappa_\ell \boldsymbol{\omega}^\top \boldsymbol{\mu}_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2}, \qquad \ell = \pi(i),
$$

finishing the proof. ■

Lemma 3.20 (Deterministic equivalent $\bar{A}_5$.) *A deterministic equivalent* $\bar{A}_5$ *of* A_5 *is given by*

$$\bar{A}_5 = -\tau^2(\bar{B}_5 + \bar{B}_5^\top),$$

where the deterministic equivalent $\bar{B}_5$ *of* B_5 *from* (3.59) *is given as follows by*

$$\bar{B}_5 = \sum_{\ell=1}^{2} n_\ell(-1)^\ell \left(\frac{\bar{\omega}^\top \mu_\ell + (-1)^\ell \kappa_\ell}{1+\kappa_\ell} C_\ell - \frac{C_\ell \bar{\omega} \mu_\ell^\top + (-1)^\ell \kappa_\ell \mu_\ell \mu_\ell^\top}{1+\kappa_\ell} \right).$$

Proof. We only have to derive the deterministic equivalent $\bar{B}_5$ of B_5 (as the statement for $\bar{A}_5$ then immediately follows from their connection $A_5 = -\tau^2(B_5 + B_5^\top)$), *i.e.*, we consider

$$\begin{aligned}\mathbb{E}\left[\operatorname{tr}(PB_5)\right] =& \mathbb{E}\left[\operatorname{tr}\left(P\left(XX^\top \omega y^\top X^\top - \mathbb{E}\left[XX^\top \omega\right]\mathbb{E}\left[y^\top X^\top\right]\right)\right)\right] \\ =& \mathbb{E}\left[\operatorname{tr}\left(PXX^\top \omega y^\top X^\top\right)\right] - \operatorname{tr}\left(P\mathbb{E}\left[XX^\top \omega\right]\mathbb{E}\left[y^\top X^\top\right]\right). \qquad (3.79)\end{aligned}$$

We begin by rewriting the expression involving the trace in the first summand. With the identity $\operatorname{tr}(uv^\top) = u^\top v$ for any $u, v \in \mathbb{R}^p$, applied to $u = XX^\top \omega$ and $v^\top = y^\top X^\top P$,

$$\begin{aligned}\operatorname{tr}\left(PXX^\top \omega y^\top X^\top\right) &= \operatorname{tr}\left(XX^\top \omega y^\top X^\top P\right) \\ &= \sum_{k=1}^{n} (\omega^\top x_k) x_k^\top \sum_{l=1}^{n} y_l P^\top x_l \\ &= \sum_{k,l=1}^{n} y_l (\omega^\top x_k) x_k^\top P^\top x_l. \qquad (3.80)\end{aligned}$$

Rewrite the first term in (3.79) by splitting up the sum and passing to the mean in (3.80),

$$\begin{aligned}&\mathbb{E}\left[\operatorname{tr}\left(PXX^\top \omega y^\top X^\top\right)\right] \\ &= \sum_{k=1}^{n} y_k \mathbb{E}\left[\omega^\top x_k x_k^\top P^\top x_k\right] + \sum_{\substack{k,l=1 \\ k\neq l}}^{n} y_l \mathbb{E}\left[\omega^\top x_k x_k^\top\right]\mathbb{E}\left[P^\top x_l\right]. \qquad (3.81)\end{aligned}$$

Next, we consider the second term from (3.79). In a similar way, it can be rewritten as

$$\begin{aligned}\operatorname{tr}\left(P\mathbb{E}\left[XX^\top \omega\right]\mathbb{E}\left[y^\top X^\top\right]\right) &= \sum_{k=1}^{n} \mathbb{E}\left[\omega^\top x_k x_k^\top\right] \sum_{l=1}^{n} y_l \mathbb{E}\left[P^\top x_l\right] \\ &= \sum_{k,l=1}^{n} y_l \mathbb{E}\left[\omega^\top x_k x_k^\top\right]\mathbb{E}\left[P^\top x_l\right] \qquad (3.82)\end{aligned}$$

Thus, subtracting (3.82) from (3.81) reduces the task to finding a deterministic equivalent

$$\sum_{k=1}^{n} y_k \mathbb{E}\left[\omega^\top x_k x_k^\top P^\top x_k\right] - \sum_{k=1}^{n} y_k \mathbb{E}\left[\omega^\top x_k x_k^\top\right]\mathbb{E}\left[P^\top x_k\right], \qquad (3.83)$$

our focus from now on. The first term in (3.83) can be treated with Proposition B.2; furthermore also using (3.42) and proceeding in a similar way to (3.44) above, we obtain

$$\sum_{k=1}^{n} y_k \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x}_k \boldsymbol{x}_k^\top \boldsymbol{P}^\top \boldsymbol{x}_k\right] = \sum_{\ell=1}^{2} n_\ell (-1)^\ell \frac{\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell + (-1)^\ell \kappa_\ell}{1+\kappa_\ell} \operatorname{tr}(\boldsymbol{P}\boldsymbol{C}_\ell) \tag{3.84}$$

where we also used that $\operatorname{tr}(\boldsymbol{P}^\top \boldsymbol{C}_\ell) = \operatorname{tr}(\boldsymbol{P}\boldsymbol{C}_\ell^\top) = \operatorname{tr}(\boldsymbol{P}\boldsymbol{C}_\ell)$, thanks to the symmetry of $\boldsymbol{C}_\ell$. Next we consider the second term from (3.83). With the help of the deterministic equivalent (3.45) of $(\boldsymbol{\omega}^\top \boldsymbol{x}_k)\boldsymbol{x}_k$ for $\boldsymbol{x}_k$ belonging to class $\mathcal{C}_\ell$, $\pi(k) = \ell$, we find that

$$\begin{aligned}
\sum_{k=1}^{n} y_k \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x}_k \boldsymbol{x}_k^\top\right] \mathbb{E}\left[\boldsymbol{P}^\top \boldsymbol{x}_k\right] &= \sum_{k=1}^{n} y_k \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x}_k \boldsymbol{x}_k\right] \mathbb{E}\left[\boldsymbol{x}_k^\top \boldsymbol{P}\right]\right) \\
&= \sum_{k=1}^{n} y_k \operatorname{tr}\left(\boldsymbol{P}\mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x}_k \boldsymbol{x}_k\right] \boldsymbol{\mu}_{\pi(k)}^\top\right) \\
&= \sum_{\ell=1}^{2} n_\ell (-1)^\ell \operatorname{tr}\left(\boldsymbol{P}\frac{\boldsymbol{C}_\ell \bar{\boldsymbol{\omega}} + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell}{1+\kappa_\ell} \boldsymbol{\mu}_\ell^\top\right) \\
&= \sum_{\ell=1}^{2} n_\ell (-1)^\ell \operatorname{tr}\left(\boldsymbol{P}\frac{\boldsymbol{C}_\ell \bar{\boldsymbol{\omega}} \boldsymbol{\mu}_\ell^\top + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top}{1+\kappa_\ell}\right)
\end{aligned} \tag{3.85}$$

Thus, with regard to (3.83) and the individual expressions found in (3.84) and (3.85), we obtain the claimed deterministic equivalent of $\boldsymbol{B}_5$, and thus of $\boldsymbol{A}_5$. ■

Lemma 3.21 (Deterministic equivalent $\bar{\boldsymbol{A}}_6$.) *A deterministic equivalent $\bar{\boldsymbol{A}}_6$ of $\boldsymbol{A}_6$ is given by*

$$\bar{\boldsymbol{A}}_6 = \tau^2 \sum_{\ell=1}^{2} n_\ell \boldsymbol{\Sigma}_\ell$$

Proof. For $\boldsymbol{A}_6$, again ignoring the factor τ^2 in the interest of a clearer presentation for now,

$$\begin{aligned}
\frac{1}{\tau^2}\mathbb{E}\left[\operatorname{tr}(\boldsymbol{P}\boldsymbol{A}_6)\right] =& \mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{P}\left(\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}^\top - \mathbb{E}[\boldsymbol{X}\boldsymbol{y}]\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right]\right)\right)\right] \\
=& \operatorname{tr}\left(\boldsymbol{P}\left(\mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}^\top\right] - \mathbb{E}[\boldsymbol{X}\boldsymbol{y}]\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right]\right)\right).
\end{aligned} \tag{3.86}$$

Next, for the second summand within the trace in (3.86) it is straightforward to obtain

$$\begin{aligned}
\mathbb{E}[\boldsymbol{X}\boldsymbol{y}]\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top\right] =& \left(\sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell\right)\left(\sum_{\ell=1}^{2}(-1)^\ell n_\ell \boldsymbol{\mu}_\ell^\top\right) \\
=& n_1^2 \boldsymbol{\mu}_1 \boldsymbol{\mu}_1^\top - n_1 n_2 \boldsymbol{\mu}_1 \boldsymbol{\mu}_2^\top - n_1 n_2 \boldsymbol{\mu}_2 \boldsymbol{\mu}_1^\top + n_2^2 \boldsymbol{\mu}_2 \boldsymbol{\mu}_2^\top.
\end{aligned} \tag{3.87}$$

For the first summand within the trace in (3.86), the following term can be rewritten as

$$\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}^\top = \left(\sum_{k=1}^{n} y_k \boldsymbol{x}_k\right)\left(\sum_{l=1}^{n} y_l \boldsymbol{x}_l^\top\right) = \sum_{k,l=1}^{n} y_k y_l \boldsymbol{x}_k \boldsymbol{x}_l^\top.$$

We pass to the expectation and split up the sum on the right hand side into two parts.

Recalling $\boldsymbol{C}_\ell$ from (3.5) for the first sum, and exploiting independence for the second sum,

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top\boldsymbol{X}^\top\right] =& \mathbb{E}\left[\sum_{k,l=1}^{n} y_k y_l \boldsymbol{x}_k\boldsymbol{x}_l^\top\right] \\
=& \mathbb{E}\left[\sum_{k=1}^{n} y_k^2 \boldsymbol{x}_k\boldsymbol{x}_k^\top\right] + \mathbb{E}\left[\sum_{\substack{k,l=1\\k\neq l}}^{n} y_k y_l \boldsymbol{x}_k\boldsymbol{x}_l^\top\right] \\
=& \sum_{k=1}^{n} \mathbb{E}\left[\boldsymbol{x}_k\boldsymbol{x}_k^\top\right] + \sum_{\substack{k,l=1\\k\neq l}}^{n} y_k y_l \mathbb{E}\left[\boldsymbol{x}_k\right]\mathbb{E}\left[\boldsymbol{x}_l\right]^\top \\
=& \sum_{\ell=1}^{2} n_\ell \boldsymbol{C}_\ell + n_1(n_1-1)\boldsymbol{\mu}_1\boldsymbol{\mu}_1^\top - n_1 n_2 \boldsymbol{\mu}_1\boldsymbol{\mu}_2^\top \\
& \quad - n_1 n_2 \boldsymbol{\mu}_2\boldsymbol{\mu}_1^\top + n_2(n_2-1)\boldsymbol{\mu}_2\boldsymbol{\mu}_2^\top.
\end{aligned} \tag{3.88}$$

Subtracting (3.87) from (3.88), thanks to cancellations we obtain the simplified expression

$$\mathbb{E}\left[\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top\boldsymbol{X}^\top\right] - \mathbb{E}[\boldsymbol{X}\boldsymbol{y}]\mathbb{E}\left[\boldsymbol{y}^\top\boldsymbol{X}^\top\right] = \sum_{\ell=1}^{2} n_\ell \boldsymbol{C}_\ell - n_1\boldsymbol{\mu}_1\boldsymbol{\mu}_1^\top - n_2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^\top = \sum_{\ell=1}^{2} n_\ell \boldsymbol{\Sigma}_\ell.$$

Taking into account the factor τ^2 leads to the desired deterministic equivalent $\bar{\boldsymbol{A}}_6$. ■

Summary: Covariance Updates. Before moving on to considering the special case of $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$ for the covariances (when the classes only differ in their means), let us first provide an overview summarizing our findings for the deterministic equivalents $\bar{\boldsymbol{A}}_1, \ldots, \bar{\boldsymbol{A}}_6$ that we found in the previous lemmas, from Lemma 3.16 to Lemma 3.21. With $\bar{\boldsymbol{B}}_2, \bar{\boldsymbol{B}}_3, \bar{\boldsymbol{B}}_5$ (recall $\boldsymbol{B}_2, \boldsymbol{B}_3, \boldsymbol{B}_5$ from (3.57), (3.58) and (3.59)) appearing in the expressions of $\bar{\boldsymbol{A}}_2, \bar{\boldsymbol{A}}_3, \bar{\boldsymbol{A}}_5$, given as

$$\begin{aligned}
\bar{\boldsymbol{B}}_2 =& \sum_{\ell=1}^{2} n_\ell \left(\bar{\boldsymbol{\omega}}\boldsymbol{a}_\ell^\top + \frac{\boldsymbol{C}_\ell\boldsymbol{C}_\omega + (-1)^\ell \kappa_\ell \boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top}{1+\kappa_\ell} + \frac{\kappa_\ell - \mathrm{tr}(\boldsymbol{C}_\omega\boldsymbol{C}_\ell)}{(1+\kappa_\ell)^2}\boldsymbol{K}_\ell\right), \\
\bar{\boldsymbol{B}}_3 =& \tau \sum_{\ell=1}^{2} n_\ell \frac{(1-(-1)^\ell \bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell)}{1+\kappa_\ell}\boldsymbol{C}_\ell\bar{\boldsymbol{D}}\bar{\boldsymbol{Q}}, \\
\bar{\boldsymbol{B}}_5 =& \sum_{\ell=1}^{2} n_\ell (-1)^\ell \left(\frac{\bar{\boldsymbol{\omega}}^\top\boldsymbol{\mu}_\ell + (-1)^\ell\kappa_\ell}{1+\kappa_\ell}\boldsymbol{C}_\ell - \frac{\boldsymbol{C}_\ell\bar{\boldsymbol{\omega}}\boldsymbol{\mu}_\ell^\top + (-1)^\ell\kappa_\ell\boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\top}{1+\kappa_\ell}\right),
\end{aligned}$$

the deterministic equivalents $\bar{\boldsymbol{A}}_1, \ldots, \bar{\boldsymbol{A}}_6$ are given as follows by

$$\begin{aligned}
\bar{\boldsymbol{A}}_1 =& \boldsymbol{\Sigma}_\omega, \\
\bar{\boldsymbol{A}}_2 =& \tau(\bar{\boldsymbol{B}}_2 + \bar{\boldsymbol{B}}_2^\top), \\
\bar{\boldsymbol{A}}_3 =& \tau(\bar{\boldsymbol{B}}_3 + \bar{\boldsymbol{B}}_3^\top), \\
\bar{\boldsymbol{A}}_4 =& \tau^2 \sum_{\ell,\ell'=1}^{2} \frac{n_\ell n_{\ell'}\boldsymbol{\Sigma}_{\ell'}\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell}{(1+\kappa_\ell)(1+\kappa_{\ell'})} + \tau^2 \sum_{\ell=1}^{2} n_\ell \frac{\mathrm{tr}\left(\boldsymbol{\Sigma}_\omega\boldsymbol{\Sigma}_\ell\right) + 2(-1)^\ell\kappa_\ell\boldsymbol{\omega}^\top\boldsymbol{\mu}_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2}\boldsymbol{C}_\ell
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{\ell=1}^{2} n_\ell \left(\boldsymbol{a}_\ell \boldsymbol{a}_\ell^\top + \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell\right) + 2\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell (-1)^\ell \kappa_\ell + \kappa_\ell^2}{\left(1+\kappa_\ell\right)^2} \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \right),\\
\bar{\boldsymbol{A}}_5 =& -\tau^2 (\bar{\boldsymbol{B}}_5 + \bar{\boldsymbol{B}}_5^\top),\\
\bar{\boldsymbol{A}}_6 =& \tau^2 \sum_{\ell=1}^{2} n_\ell \boldsymbol{\Sigma}_\ell .
\end{aligned}
$$

Note that in case of the covariances being the identity matrices, that is $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$, to compute $\operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell) = \operatorname{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{I}) = \operatorname{tr}(\boldsymbol{\Sigma}_\omega) = \boldsymbol{\sigma}_\omega^\top \mathbb{1}_p$ (3.2) in Proposition 3.3 for the variance of the classification score for each class we just require the diagonal, of $\boldsymbol{\Sigma}_\omega$, denoted by $\boldsymbol{\sigma}_\omega = \mathcal{D}(\boldsymbol{\Sigma}_\omega)$. Further, when passing to this special case, we will use again the approximation $\boldsymbol{C}_\ell \approx \boldsymbol{\Sigma}_\ell = \boldsymbol{I}_p$, and obtain $\mathcal{D}(\boldsymbol{C}_\omega \boldsymbol{C}_\ell) \approx \mathcal{D}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_\ell) = \mathcal{D}(\boldsymbol{\Sigma}_\omega \boldsymbol{I}) = \mathcal{D}(\boldsymbol{\Sigma}_\omega) = \boldsymbol{\sigma}_\omega$. Passing to the diagonals, e.g. $\mathcal{D}(\boldsymbol{\Sigma}_\ell) = \mathcal{D}(\boldsymbol{I}_p) = \mathbb{1}_p$ for $\ell = 1, 2$, passing both from $\boldsymbol{C}_\ell$ to $\mathcal{D}(\boldsymbol{C}_\ell)$ as well as from $\boldsymbol{K}_\ell = \tau \boldsymbol{C}_\ell \bar{\boldsymbol{D}} \bar{\boldsymbol{Q}} \in \mathbb{R}^{p\times p}$ to $\boldsymbol{k}_\ell = \mathcal{D}(\boldsymbol{K}_\ell) \in \mathbb{R}^p$ (both defined in (3.55) already), and summarizing terms in $\bar{\boldsymbol{A}}_2, \bar{\boldsymbol{A}}_3, \bar{\boldsymbol{A}}_5$ (they are the sum of a matrix with its transpose, which can be easily simplified when passing to the diagonal), we obtain the following counterparts $\boldsymbol{\sigma}_k$ to $\bar{\boldsymbol{A}}_k$ for $k = 1, \ldots, 6$:

$$
\begin{aligned}
\boldsymbol{\sigma}_1 =& \boldsymbol{\sigma}_\omega,\\
\boldsymbol{\sigma}_2 =& 2\tau \sum_{\ell=1}^{2} n_\ell \left(\mathcal{D}\left(\bar{\boldsymbol{\omega}} \boldsymbol{a}_\ell^\top\right) + \frac{\boldsymbol{\sigma}_\omega}{1+\kappa_\ell} + \frac{(-1)^\ell \kappa_\ell}{1+\kappa_\ell} \mathcal{D}(\boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top) + \frac{\kappa_\ell - \boldsymbol{\sigma}_\omega \mathbb{1}_p}{(1+\kappa_\ell)^2} \boldsymbol{k}_\ell \right),\\
\boldsymbol{\sigma}_3 =& 2\tau \sum_{\ell=1}^{2} n_\ell \frac{(1 - (-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell)}{1+\kappa_\ell} \boldsymbol{k}_\ell,\\
\boldsymbol{\sigma}_4 =& \tau^2 \sum_{\ell,\ell'=1}^{2} \frac{n_\ell n_{\ell'}}{(1+\kappa_\ell)(1+\kappa_{\ell'})} \boldsymbol{\sigma}_\omega + \tau^2 \sum_{\ell=1}^{2} n_\ell \frac{\boldsymbol{\sigma}_\omega^\top \mathbb{1}_p + 2(-1)^\ell \kappa_\ell \boldsymbol{\omega}^\top \boldsymbol{\mu}_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} \mathcal{D}(\boldsymbol{C}_\ell)\\
&+ \sum_{\ell=1}^{2} n_\ell \left(\mathcal{D}\left(\boldsymbol{a}_\ell \boldsymbol{a}_\ell^\top\right) + \frac{\boldsymbol{\sigma}_\omega^\top \mathbb{1}_p + 2\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell (-1)^\ell \kappa_\ell + \kappa_\ell^2}{\left(1+\kappa_\ell\right)^2} \mathcal{D}\left(\boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top\right) \right),\\
\boldsymbol{\sigma}_5 =& -2\tau^2 \sum_{\ell=1}^{2} n_\ell (-1)^\ell \frac{\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell + (-1)^\ell \kappa_\ell}{1+\kappa_\ell} \mathcal{D}(\boldsymbol{C}_\ell),\\
\boldsymbol{\sigma}_6 =& \tau^2 \sum_{\ell=1}^{2} n_\ell \mathbb{1}_p .
\end{aligned}
$$

Let us summarize some terms arising in $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_6$. The term from σ_1 and one of the summands from σ_2 and σ_4 containing $\boldsymbol{\sigma}_\omega$ are collected in $\boldsymbol{\varsigma}_1$, given as follows by

$$
\boldsymbol{\varsigma}_1 = \sum_{\ell=1}^{2} \left(\frac{1}{2} + \frac{2\tau n_\ell}{1+\kappa_\ell} + \sum_{\ell'=1}^{2} \frac{\tau^2 n_\ell n_{\ell'}}{(1+\kappa_\ell)(1+\kappa_{\ell'})} \right) \boldsymbol{\sigma}_\omega .
$$

Similarly, we summarize σ_3 and one of the summands from σ_2 containing $\boldsymbol{k}_\ell$ in $\boldsymbol{\varsigma}_2$,

$$
\begin{aligned}
\boldsymbol{\varsigma}_2 =& 2\tau \sum_{\ell=1}^{2} n_\ell \left(\frac{1 - (-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell}{1+\kappa_\ell} + \frac{\kappa_\ell - \boldsymbol{\sigma}_\omega \mathbb{1}_p}{(1+\kappa_\ell)^2} \right) \boldsymbol{k}_\ell\\
=& 2\tau \sum_{\ell=1}^{2} n_\ell \left(\frac{1 + 2\kappa_\ell - (1+\kappa_\ell)(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell - \boldsymbol{\sigma}_\omega \mathbb{1}_p}{(1+\kappa_\ell)^2} \right) \boldsymbol{k}_\ell .
\end{aligned}
$$

A similar procedure for the terms corresponding to $\mathcal{D}(\boldsymbol{C}_\ell)$, *i.e.*, σ_5 and one of the remaining summands from σ_4, leads to ς_3, which is given by

$$\begin{aligned}\varsigma_3 =&\tau^2 \sum_{\ell=1}^{2} n_\ell \left(\frac{\boldsymbol{\sigma}_{\omega}^\top \mathbb{1}_p + 2(-1)^\ell \kappa_\ell \boldsymbol{\omega}^\top \boldsymbol{\mu}_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} - 2\frac{(-1)^\ell \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell + \kappa_\ell}{1+\kappa_\ell} \right) \mathcal{D}(\boldsymbol{C}_\ell)\\ =&\tau^2 \sum_{\ell=1}^{2} n_\ell \left(\frac{\boldsymbol{\sigma}_{\omega}^\top \mathbb{1}_p - 2(-1)^\ell \kappa_\ell \boldsymbol{\omega}^\top \boldsymbol{\mu}_\ell - \kappa_\ell}{(1+\kappa_\ell)^2} \right) \mathcal{D}(\boldsymbol{C}_\ell).\end{aligned}$$

All the remaining terms are collected as follows in ς_4 and ς_5,

$$\begin{aligned}\varsigma_4 =&2\tau \sum_{\ell=1}^{2} n_\ell \left(\frac{\tau}{2}\mathbb{1}_p + \mathcal{D}\left(\bar{\boldsymbol{\omega}} \boldsymbol{a}_\ell^\top\right) + \frac{(-1)^\ell \kappa_\ell}{1+\kappa_\ell} \mathcal{D}\left(\boldsymbol{\mu}_\ell \bar{\boldsymbol{\omega}}^\top\right) \right),\\ \varsigma_5 =& \sum_{\ell=1}^{2} n_\ell \left(\mathcal{D}\left(\boldsymbol{a}_\ell \boldsymbol{a}_\ell^\top\right) + \frac{\boldsymbol{\sigma}_{\omega}^\top \mathbb{1}_p + 2\bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell (-1)^\ell \kappa_\ell + \kappa_\ell^2}{(1+\kappa_\ell)^2} \mathcal{D}\left(\boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top\right) \right).\end{aligned}$$

Note that by construction it holds that $\sum_{k=1}^{6} \sigma_k^j = \sum_{k=1}^{5} \varsigma_k^j$.

3.4.3 Algorithm and Numerical Experiments

Finally, let us combine our findings and state the overall algorithm which will then be tested numerically. Recall that we assume a normal distribution of the classification scores

$$g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega} \sim \mathcal{N}\left(\mathfrak{m}_\ell, \sigma_\ell^2\right), \qquad \boldsymbol{x} \in \mathcal{C}_\ell,$$

for each $\ell = 1, 2$ with their respective means $\mathfrak{m}_\ell \in \mathbb{R}$ and variances $\sigma_\ell^2 > 0$. Note however that in the special case $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$ and by Proposition 3.3, the variances are the same, *i.e.*, $\sigma_1^2 = \sigma_2^2$. Notably, again by Proposition 3.3 we approximate $\sigma_1^2 = \sigma_2^2 = \mathrm{Var}(g(\boldsymbol{x})) = \mathbb{E}[g(\boldsymbol{x})^2] - \mathbb{E}[g(\boldsymbol{x})]^2 \approx \mathrm{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{\Sigma}_x) = \mathrm{tr}(\boldsymbol{\Sigma}_\omega \boldsymbol{I}) = \boldsymbol{\sigma}_\omega \mathbb{1}_p$, using the dominant first summand from (3.2), leaving us with the task of approximately computing $\boldsymbol{\sigma}_\omega \in \mathbb{R}^p$; see also Remark 3.5. Based on our findings from the previous section, we state an iterative scheme

$$\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}, \qquad \boldsymbol{\sigma}_\omega^j \to \boldsymbol{\sigma}_\omega^{j+1}$$

which will then provide us, thanks to Proposition 3.3, with corresponding updates on the means and variances of the classification scores $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}$ for $\boldsymbol{x}$ belonging to class $\mathcal{C}_\ell$,

$$\mathfrak{m}_\ell^j \to \mathfrak{m}_\ell^{j+1}, \qquad (\sigma_\ell^2)^j \to (\sigma_\ell^2)^{j+1}, \qquad \ell = 1, 2.$$

This algorithm resembles ISTA itself, and while the convergence remains unclear from a theoretical perspective, the numerical results are promising. Note further that now for stating the algorithm we will make the iteration index visible again as an upper j, which we left out in the previous derivations in the interest of readability. Recalling (3.1), again from Proposition 3.3, we have that for $\boldsymbol{x}$ belonging to class $\mathcal{C}_\ell$ the mean of the corresponding classification score is given as

$$\mathfrak{m}_\ell = \mathbb{E}[g(\boldsymbol{x})] = \mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{x}] = \bar{\boldsymbol{\omega}}^\top \mathbb{E}[\boldsymbol{x}] = \bar{\boldsymbol{\omega}}^\top \bar{\boldsymbol{x}} = \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell,$$

where the class mean $\boldsymbol{\mu}_\ell$ is estimated from the training data. As we compute mean updates $\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}$ according to our findings from Section 3.4.1, the classification score means are obtained accordingly as follows (again making the iteration index j explicit),

$$\mathfrak{m}_\ell^j = \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j. \tag{3.89}$$

Input and Initializations.

For $\ell = 1, 2$, we first precompute and initialize some of the involved objects as follows, before running the subsequent iteration until a stopping criterion is met, when updates for all involved parameters fall below a certain threshold (measured by the ℓ_2- norm for vectors, and by the absolute values for scalar quantities), or simply for a fixed number of iterations.

- Hyperparameters $\tau, \lambda > 0$; training data for classes $\mathcal{C}_\ell$ of size n_ℓ with $n = n_1 + n_2$.
- Estimate $\boldsymbol{\mu}_\ell$ from the training data and put $\boldsymbol{C}_\ell = \boldsymbol{I}_p + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$ (since $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$).
- Initialize $\boldsymbol{k}_\ell^0 = \bar{\boldsymbol{\omega}}^0 = \boldsymbol{\sigma}_\omega^0 = \boldsymbol{\sigma}_z^0 = \boldsymbol{0}_p \in \mathbb{R}^p$ and $\mathfrak{m}_\ell^0 = \kappa_\ell^0 = 0 \in \mathbb{R}$.

Mean Updates $\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}$ and $\mathfrak{m}_\ell^j \to \mathfrak{m}_\ell^{j+1}$

The mean update $\bar{\boldsymbol{\omega}}^j \to \bar{\boldsymbol{\omega}}^{j+1}$ is straightforward to obtain by the deterministic equivalent for $\bar{\boldsymbol{z}}$ (or $\bar{\boldsymbol{z}}^j$) from (3.47), containing $\boldsymbol{a}_\ell$ (or $\boldsymbol{a}_\ell^j$) from (3.45), which we will use in the special case as provided in (3.46). We pass from $\boldsymbol{z}^j$ to $\boldsymbol{\omega}^j$ as in (3.11) with the help of φ from (3.13). Finally, the mean update $\mathfrak{m}_\ell^j \to \mathfrak{m}_\ell^{j+1}$ for the classification score is computed via (3.89).

$$\begin{aligned}
\boldsymbol{a}_\ell^j &= \frac{\mathfrak{m}_\ell^j + (-1)^\ell \kappa_\ell^j}{1 + \kappa_\ell^j} \boldsymbol{\mu}_\ell + \frac{1}{1 + \kappa_\ell^j} \bar{\boldsymbol{\omega}}^j, \qquad \ell = 1, 2 \\
\bar{\boldsymbol{z}}^{j+1} &= \bar{\boldsymbol{\omega}}^j - \tau \sum_{\ell=1}^{2} n_\ell \left(\boldsymbol{a}_\ell^j + (-1)^\ell \boldsymbol{\mu}_\ell \right), \\
\bar{\boldsymbol{\omega}}^{j+1} &= \varphi\left(\lambda\tau, \bar{\boldsymbol{z}}^{j+1}, \boldsymbol{\sigma}_z^j \right), \\
\mathfrak{m}_\ell^{j+1} &= \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j, \qquad \ell = 1, 2.
\end{aligned} \tag{3.90}$$

Covariance Update $\boldsymbol{\sigma}_\omega^j \to \boldsymbol{\sigma}_\omega^{j+1}$ and Computing the Variance $\sigma^2 = \sigma_1^2 = \sigma_2^2$

For the covariance updates, we recall that in (3.48) in Section 3.4.2 we have considered the random matrix $\boldsymbol{M}^j = \boldsymbol{z}^j \boldsymbol{z}^{j^\top} - \bar{\boldsymbol{z}}^j \bar{\boldsymbol{z}}^{j^\top} \in \mathbb{R}^{p \times p}$ (now again with the index j). For $\boldsymbol{M}^j$, we have derived a deterministic equivalent $\bar{\boldsymbol{M}}^j = \sum_{k=1}^{6} \bar{\boldsymbol{A}}_k^j$ that behaves similar under taking traces, like the traces in Proposition 3.3. Again for the special case $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$,

$$\begin{aligned}
\varsigma_1^j =& \sum_{\ell=1}^{2} \left(\frac{1}{2} + \frac{2\tau n_\ell}{1 + \kappa_\ell^j} + \sum_{\ell'=1}^{2} \frac{\tau^2 n_\ell n_{\ell'}}{(1 + \kappa_\ell^j)(1 + \kappa_{\ell'}^j)} \right) \boldsymbol{\sigma}_\omega^j, \\
\varsigma_2^j =& 2\tau \sum_{\ell=1}^{2} n_\ell \left(\frac{1 + 2\kappa_\ell - (1 + \kappa_\ell)(-1)^\ell \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j - \mathbb{1}_p^\top \boldsymbol{\sigma}_\omega^j}{(1 + \kappa_\ell^j)^2} \right) \boldsymbol{k}_\ell^j,
\end{aligned}$$

$$\begin{aligned}
\varsigma_3^j =& \tau^2 \sum_{\ell=1}^{2} n_\ell \left(\frac{\mathbb{1}_p^\top \boldsymbol{\sigma}_\omega^j - 2(-1)^\ell \kappa_\ell^j \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j - \kappa_\ell^j}{(1+\kappa_\ell^j)^2} \right) \mathcal{D}(\boldsymbol{C}_\ell), \\
\varsigma_4^j =& 2\tau \sum_{\ell=1}^{2} n_\ell \left(\frac{\tau}{2}\mathbb{1}_p + \mathcal{D}\left(\bar{\boldsymbol{\omega}}^j {\boldsymbol{a}_\ell^j}^\top \right) + \frac{(-1)^\ell \kappa_\ell^j}{1+\kappa_\ell^j} \mathcal{D}\left(\boldsymbol{\mu}_\ell {\bar{\boldsymbol{\omega}}^j}^\top \right) \right), \\
\varsigma_5^j =& \sum_{\ell=1}^{2} n_\ell \left(\mathcal{D}\left(\boldsymbol{a}_\ell^j {\boldsymbol{a}_\ell^j}^\top \right) + \frac{\mathbb{1}_p^\top \boldsymbol{\sigma}_\omega^j + 2(-1)^\ell \kappa_\ell^j \boldsymbol{\mu}_\ell^\top \bar{\boldsymbol{\omega}}^j + {\kappa_\ell^j}^2}{\left(1+\kappa_\ell^j\right)^2} \mathcal{D}\left(\boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \right) \right).
\end{aligned}$$

Next we pass to the diagonal of the covariance; recall (3.12) and (3.4) for the function Γ, which is sufficient to handle diagonal covariances as in (3.14). With $\varsigma_1^j, \ldots, \varsigma_5^j$ as above,

$$\begin{aligned}
\boldsymbol{\sigma}_z^j &= \mathcal{D}\left(\bar{\boldsymbol{M}}^j \right) = \sum_{k=1}^{5} \varsigma_k^j, \\
\boldsymbol{\sigma}_\omega^{j+1} &= \Gamma\left(\lambda\tau, \bar{\boldsymbol{z}}^{j+1}, \boldsymbol{\sigma}_z^{j+1} \right) - \bar{\boldsymbol{\omega}}^{j+1} {\bar{\boldsymbol{\omega}}^{j+1}}^\top.
\end{aligned}$$

Next, we compute the updates for the other involved quantities such as κ_ℓ from (3.39), $\bar{\boldsymbol{Q}}$ from (3.38), $\bar{\boldsymbol{D}}$ from (3.29) and κ_ℓ from (3.55), now inserting the iteration index j:

$$\begin{aligned}
\bar{\boldsymbol{D}}^{j+1} &= \psi\left(\lambda\tau, \bar{\boldsymbol{z}}^{j+1}, \boldsymbol{\sigma}_z^{j+1} \right), \\
\bar{\boldsymbol{Q}}^{j+1} &= \left(\boldsymbol{I}_p - \bar{\boldsymbol{D}}^{j+1} + \sum_{\ell=1}^{2} \frac{\tau n_\ell}{1+\kappa_\ell^j} \boldsymbol{C}_\ell \bar{\boldsymbol{D}}^{j+1} \right)^{-1}, \\
\boldsymbol{k}_\ell^{j+1} &= \tau \mathcal{D}\left(\boldsymbol{C}_\ell \bar{\boldsymbol{D}}^{j+1} \bar{\boldsymbol{Q}}^{j+1} \right), \qquad \ell = 1, 2, \\
\kappa_\ell^{j+1} &= \operatorname{tr}\left(\tau \boldsymbol{C}_\ell \bar{\boldsymbol{D}}^{j+1} \bar{\boldsymbol{Q}}^{j+1} \right) = \mathbb{1}_p^\top \boldsymbol{k}_\ell^j, \qquad \ell = 1, 2.
\end{aligned}$$

Output: Predicted Classification Error

After running the mean and covariance updates as described above for $J \in \mathbb{N}$ iterations, after termination we obtain the estimates for the classification score means and variances,

$$\mathfrak{m}_1 := \mathfrak{m}_1^J, \qquad \mathfrak{m}_2 := \mathfrak{m}_2^J, \qquad \sigma^2 := \sigma_1^2 := \sigma_2^2 := \boldsymbol{\sigma}_\omega^\top \mathbb{1}_p.$$

Then, Corollary 3.2 allows us (in case of balanced classes, *i.e.*, with the same number of samples $n_1 = n_2$ from both classes $\mathcal{C}_1$ and $\mathcal{C}_2$; for the general case, we refer instead to Lemma 3.1) to predict the classification error ε (or the accuracy $1 - \varepsilon$) given as follows by

$$\varepsilon = \frac{1}{2} Q\left(\frac{\mathfrak{m}_2 - \mathfrak{m}_1}{\sigma^2} \right), \qquad Q(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}} \, \mathrm{d}x,$$

where for the function Q we also refer to (B.4) in the appendix. We summarize the algorithm in Algorithm 1 below.

We conclude this section by summarizing weaknesses in the derivation of the algorithm and open questions that are of interest for future work.

- The derivation is somewhat hindered due to the fact that the soft-thresholding function S_λ is not differentiable everywhere; recall Remark 3.15; a more rigorous attempt would arguably require to work with smooth approximations.

- Open questions and gaps revolving around $\boldsymbol{D}$ and $\boldsymbol{Q}$ from (3.28) and (3.30), in particular justifying to obtain them as an approximation by inserting $t = 1$ into $\boldsymbol{D}_i(t)$ and $\boldsymbol{Q}_i(t)$ from (3.25) and (3.27). Furthermore, regarding $\boldsymbol{D}$ and $\boldsymbol{Q}$ obtained this way, we have simplified the derivation by assuming a tight concentration of $\boldsymbol{x}_i^\top \boldsymbol{D}\boldsymbol{Q}\boldsymbol{x}_i$ around its mean in (3.41); and similar for (3.52).

- Rigorous convergence guarantees for the iterative scheme developed would be desirable, beyond merely a verification with numerical experiments.

- Possible simplifications of the algorithm by identifying asymptotically negligible terms, as suggested by numerical experiments.

- Of great interest for practical applications beyond the experiments performed here is an extension of the derivation to cover the more general case of generic covariance matrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^{p\times p}$, beyond the simple case of $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$ considered here. While large parts of the derivation hold in a more general setting, for the computation of the covariance (3.12) we would require multi-dimensional numerical integration, beyond the one-dimensional solution for the diagonal entries with the function Γ from (3.14).

Algorithm 1 Predicting accuracy of LASSO-based classification

```
Input: Parameters $\lambda, \tau$; estimated means $\boldsymbol{\mu}_\ell$ of classes of size $n_\ell$, $\ell = 1, 2$.
        Generalized covariance $\boldsymbol{C}_\ell = \boldsymbol{I}_p + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top \in \mathbb{R}^{p\times p}$ for $\ell = 1, 2$.
Initialize: For $\ell = 1, 2$, initialize $\boldsymbol{k}_\ell, \bar{\boldsymbol{\omega}}, \boldsymbol{\sigma}_\omega, \boldsymbol{\sigma}_z = \boldsymbol{0}_p$ and $\mathfrak{m}_\ell, \kappa_\ell = 0$;
        compute initial $\boldsymbol{a}_\ell$ by (3.90)
repeat
    Compute $\bar{\boldsymbol{z}} = \bar{\boldsymbol{\omega}} - \sum_{\ell=1}^{2} \tau n_\ell \left(\boldsymbol{a}_\ell + (-1)^\ell \boldsymbol{\mu}_\ell\right)$.
    Compute the ridge-less variance $\boldsymbol{\sigma}_z = \boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_3 + \boldsymbol{\sigma}_4 + \boldsymbol{\sigma}_5 + \boldsymbol{\sigma}_6$.
    Update $\bar{\boldsymbol{D}}$ and $\bar{\boldsymbol{Q}}$ and $\boldsymbol{k}_\ell$ and $\kappa_\ell = \tau \operatorname{tr}\left(\boldsymbol{C}_\ell \bar{\boldsymbol{D}} \bar{\boldsymbol{Q}}\right)$ for $\ell = 1, 2$
    Update $\bar{\boldsymbol{\omega}} = \varphi(\lambda, \bar{\boldsymbol{z}}, \boldsymbol{\sigma}_z)$ and $\boldsymbol{\sigma}_\omega = \Gamma(\lambda, \bar{\boldsymbol{z}}, \boldsymbol{\sigma}_z) - \bar{\boldsymbol{\omega}}\bar{\boldsymbol{\omega}}^\top$
    Update $\mathfrak{m}_\ell = \bar{\boldsymbol{\omega}}^\top \boldsymbol{\mu}_\ell$ and $\boldsymbol{a}_\ell$ for $\ell = 1, 2$.
until Convergence criterion met.
Output: Classification error $\varepsilon = \frac{1}{2} Q\left(\frac{\mathfrak{m}_2 - \mathfrak{m}_1}{\boldsymbol{\sigma}_\omega^\top \mathbb{1}_p}\right)$.
```

3.4.4 Numerical Experiments

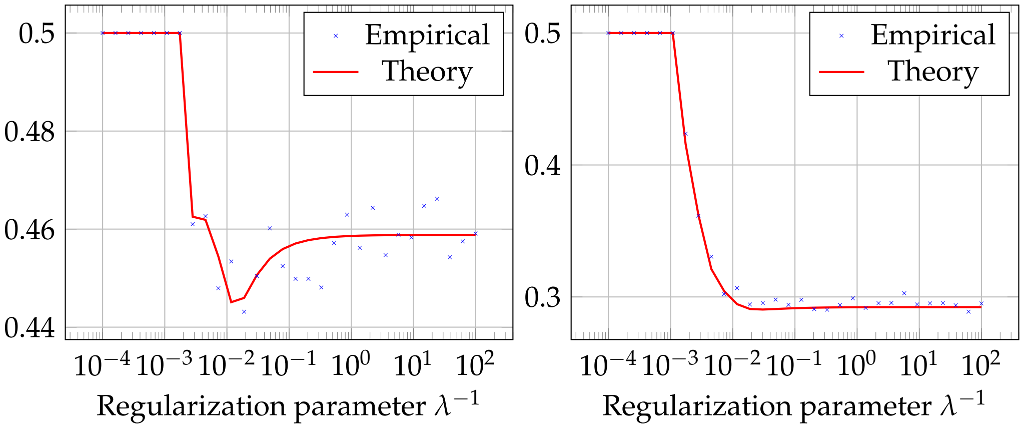


Figure 3.3: Theoretical versus empirical classification error as function of the regularization parameter. $\boldsymbol{\mu}_1$ drawn once from $\boldsymbol{\mu}_1 \sim \mathcal{N}(\mathbf{0}_p, \boldsymbol{I}_p)$ and sparsified by putting its entries to zero with $\alpha = 0.95$ **(left)** and $\alpha = 0.5$ **(right)**; furthermore, simply $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1$ for the means.

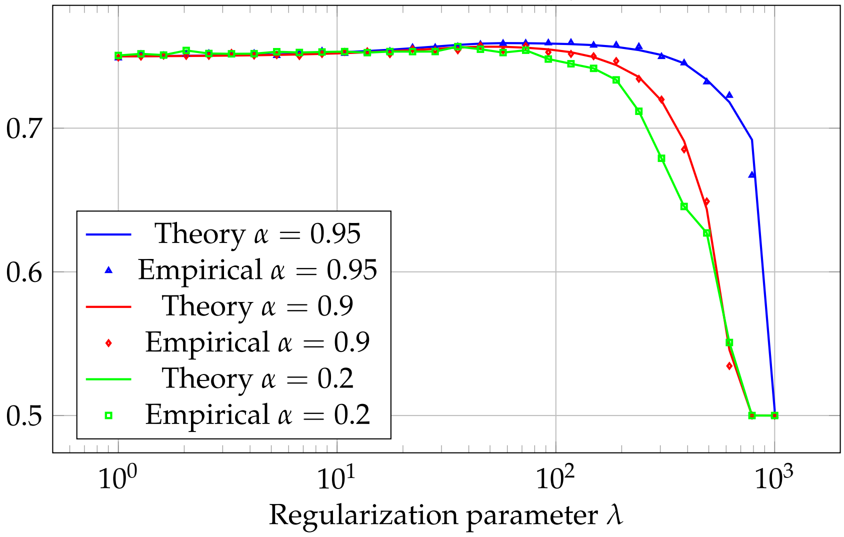


Figure 3.4: Close fit between the theoretical and empirical (averaged over 1 000 test samples) classification accuracy (as a function of the regularization parameter λ), for three different values of α (sparsity level). We consider Gaussian mixture model with class sizes $n_1, n_2 = 500$ and $\boldsymbol{x}_i^{(\ell)} \sim \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{I}_p)$, for $\ell = 1, 2$, with mean $\boldsymbol{\mu}_\ell = (-1)^\ell \boldsymbol{b} \odot \boldsymbol{m}$, where $\boldsymbol{m} \sim \mathcal{N}(\mathbf{0}_p, \frac{1}{p}\boldsymbol{I}_p)$, and where $\boldsymbol{b}$ is a Bernoulli random vector that puts each single entry to zero with probability α/p, where $p = 100$ is the dimension (number of features), and where $\alpha \in \{0.2, 0.9, 0.95\}$ is a parameter controlling the sparsity.

We test Algorithm 1 with experiments on synthetic data to predict the classification accuracy of the linear classifier found through ISTA. Recall that we have already plotted the classification score for practical datasets in Figure 3.2. Here, we test the algorithm for different values of λ and observe a very good agreement between the theoretical prediction and the empirical performance. This opens up new ways of hyperparameter optimization by grid search, as an alternative to the established method of cross validation. In both figures, we observe that for large λ, when the regularization term outweighs the least square loss, the accuracy tends to $1/2$, corresponding to random guesses.

3.5 Related Work and Outlook

Related Work. A large body of literature on the LASSO and related topics exist; for an extensive treatment of the subject, we refer to [BVDG11] and the references therein. From a technical point of view, the work presented in this chapter is similar to the analysis of machine learning algorithms such as a high dimensional analysis of logistic regression [EKBBLY13; MLC19], support vector machines [MC18; ML19], and more recently of the softmax classifier [SLCT21]. However, in contrast the aforementioned works, the difficulty of the LASSO lies in the non-differentiability of the loss function and the complex iterative procedure used to solve the minimization problem. Sparse linear classifiers have been studied from the statistical learning perspective before, based on VC dimension bounds, previously in [SSSSHZ15] and, for sparse logistic regression in [AG18]. Besides, previous work often focused on a regression rather than a classification setup. Based on techniques from approximate message passing, [BM11; CMW20; GAK20; Hua20] derive exact asymptotic expressions for the reconstruction error. These works have been complemented by an analysis using the Convex Gaussian Min-max Theorem [ASKAAN20; TOH15]. The present paper is part of this line of work employing an asymptotic analysis of the LASSO. However, unlike previous works, we have derived the error in the different setting of classification, *i.e.,* , the classification accuracy. Furthermore, we use different tools, namely the powerful *leave-one-out approach* [CFMW19; DC18; EKBBLY13].

Fixed point based methods are a classical in mathematics and being used for instance in numerical analysis and in the study of differential equations. Random fixed point equations have been studied before, but often in the context of functional analysis and operator theory with applications to stochastic differential equations [Ito79; Pap86]. Fixed point methods receive increasing attention from an machine learning viewpoint. Similar to ISTA, many iterative algorithms such as gradient descent methods allow an interpretation as fixed point equations [Jun17]; a comprehensive survey article on fixed point strategies in data science is [CP21]. Of great interest are connections to stochastic approximation, a field that more systematically studies random iterative methods [BMP12; Duf13; HKY97].

Outlook. This chapter has seen an approach to study generalization in machine learning that is different from classical VC dimension or Rademacher complexity bounds. A main difference is that, in contrast to uniform convergence bounds, the data distribution needs to be taken into account and notably how it induces a distribution over the hypothesis class *through a specific algorithm*. If it is possible to overcome the technical difficulties, this may result in a highly accurate prediction of the classification accuracy *even before training the model*, that is just based on the statistical properties of the training dataset. Interesting future work could be a more elaborate study to compare such different approaches. On the one hand, in the case of simple linear classifiers, sharp VC dimension and Rademacher complexity estimates exist. On the other hand, Proposition 3.3 allows us to accurately describe the classification accuracy under a Gaussian distribution of the classification score. However, it requires estimates of the (first and second moments) of the data distribution, or, more precisely, just scalar observations thereof. Here, we have simply relied on arithmetic means and sample covariances. An interesting extension would be to take into account a more rigorous study of the estimation quality, possibly allowing a comparison with classical results based on the Rademacher complexity or VC dimension in terms of the estimation accuracy.

A Covering Numbers and Dudley's Integral

A.1 Covering Numbers

Given a metric space $(\mathcal{S}, d)$, its *covering numbers* $\mathcal{N}(\mathcal{S}, d, \varepsilon)$ at level $\varepsilon > 0$ is the smallest number $n_\varepsilon \in \mathbb{N}$, such that there exists a subset $\mathcal{S}_\varepsilon \subset \mathcal{S}$ of cardinality n_ε that *covers* $\mathcal{S}$, i.e.

$$\bigcup_{x \in \mathcal{S}_\varepsilon} B_\varepsilon(x) = \mathcal{S},$$

where $B_\varepsilon(x) = \{y \in \mathcal{S} : d(x, y) \leq \varepsilon\}$ denotes the closed balls of radius $\varepsilon > 0$ centered at x. For a normed space $(\mathcal{S}, \|\,.\,\|)$, with a slight abuse of notation we write $\mathcal{N}(\mathcal{S}, \|\cdot\|, \varepsilon)$.

Lemma A.1 *Let $(\mathcal{S}, d)$ be a metric space. Then, the following statement holds.*

(i) Covering numbers are monotone, i.e. for a subset $\mathcal{U} \subset \mathcal{S}$, there is $\mathcal{N}(\mathcal{U}, d, \varepsilon) \leq \mathcal{N}(\mathcal{S}, d, \varepsilon)$.

(ii) If $(\mathcal{S}, d)$ is a subset of a normed space $(\mathcal{V}, \|\,.\,\|)$, with d induced by $\|\,.\,\|$, then, for all $\alpha > 0$,

$$\mathcal{N}(\alpha \cdot \mathcal{S}, \|\,.\,\|, \varepsilon) = \mathcal{N}(\mathcal{S}, \alpha \cdot d, \varepsilon) = \mathcal{N}(\mathcal{S}, d, \varepsilon/\alpha).$$

(iii) If d' is another metric on $\mathcal{S}$ with $d'(x, y) \leq d(x, y)$ for all $x, y \in \mathcal{S}$, then it holds that

$$\mathcal{N}(\mathcal{S}, d', \varepsilon) \leq \mathcal{N}(\mathcal{S}, d, \varepsilon).$$

The next lemma provides covering numbers estimates for subsets of the unit balls (and by rescaling through Lemma A.1 (ii), for any bounded subsets). Again we omit simple proof, which can be found in various sources; as a reference, see [FR13, Proposition C.3].

Lemma A.2 *Let $\varepsilon > 0$ and let $\|\cdot\|$ be a norm on a n-dimensional vector space $\mathcal{V}$. Then, for any subset $\mathcal{S} \subseteq B_{\mathcal{V}} := \{x \in V : \|x\| \leq 1\}$ contained in the unit ball of $\mathcal{V}$, there is*

$$\mathcal{N}(\mathcal{S}, \|\cdot\|, \varepsilon) \leq \left(1 + \frac{2}{\varepsilon}\right)^n.$$

The next lemma provides a bound for the covering numbers of product spaces, based on the individual covering numbers of each single metric space that is part of the product.

Lemma A.3 *Consider p metric spaces $(\mathcal{S}_1, d_1), \ldots, (\mathcal{S}_p, d_p)$, and positive numbers $c_1, \ldots, c_p$. We define the product space $\mathcal{S}$, equipped with the metric d by*

$$\mathcal{S} = (\mathcal{S}_1 \times \cdots \times \mathcal{S}_p, d), \qquad d(x, y) = \sum_{k=1}^{p} c_k d_k(x_k, y_k),$$

where $x = (x_1, \ldots, x_p), y = (y_1, \ldots, y_p) \in \mathcal{S}$. Then, we have the covering number estimate

$$\mathcal{N}(\mathcal{S}, d, \varepsilon) \leq \prod_{k=1}^{p} \mathcal{N}(\mathcal{S}_k, d_k, \varepsilon/(c_k \cdot p)).$$

Proof. Suppose that, for any $k \in [p]$, we have individual coverings of $\mathcal{S}_k$ at level $\varepsilon/(c_k p)$ of cardinality $\mathcal{N}(\mathcal{S}_k, d_k, \varepsilon/(c_k p))$. We will show that the product of all these $\varepsilon/(c_k p)$-nets is an ε-net for the product space S. Indeed let $x = (x_1, \dots, x_p) \in \mathcal{S}$, i.e. $x_k \in \mathcal{S}_k$. Then, for each $x_k \in \mathcal{S}_k$, there exists some element y_k in the $\varepsilon/(c_k \cdot p)$-net of $\mathcal{S}_k$, i.e. $d_k(x_k, y_k) \leq \varepsilon/(c_k \cdot p)$. Then, $y = (y_1, \dots, y_p)$ is an element of the product of all nets, and by the definition of the metric d there is $d(x,y) \leq c_1(\varepsilon/(c_1 \cdot p)) + \cdots + c_p(\varepsilon/(c_p \cdot p) = \varepsilon$. ∎

A.2 Dudley's Inequality

A *stochastic process* $(X_t)_{t\in\mathcal{T}}$ is a family of random variables indexed by an index set $\mathcal{T}$. While for many practical applications $\mathcal{T}$ is interpreted as the *time* (e.g. by considering a time interval as a subset of the real numbers), in some applications more complicated index sets of higher dimensions appear. In particular, let us point out the case of matrix sets, frequently met in compressive sensing [KMR14] and statistical learning theory [BBL03]. In many of these applications (compare e.g. Rademacher complexity, Definition 1.4), one is interested in upper bounding the expression

$$\mathbb{E} \sup_{t\in T} X_t := \sup \left\{ \mathbb{E} \max_{t\in F} X_t : F \subset T, F \text{ finite} \right\}. \tag{A.1}$$

(Note that considering finite subsets F is necessary to ensure measurability of $\sup_{t\in\mathcal{T}} X_t$.) Expressions of the type of (A.1) notably appear in areas like compressive sensing and statistical learning theory for bounding the Rademacher complexity, as also seen in Chapter 2 in this thesis, where Dudley's inequality often provides tight bounds.

Furthermore, we define the *pseudometric* on the index set $\mathcal{T}$ associated to $(X_t)_{t\in\mathcal{T}}$ by

$$d(s,t) := \left(\mathbb{E}|X_s - X_t|^2\right)^{1/2} \qquad \forall\, s,t \in \mathcal{T}. \tag{A.2}$$

$(X_t)_{t\in T}$ is called a *sub-Gaussian process*, if it is centered (i.e. $\mathbb{E}X_{t_0} = 0$ for all $t_0 \in \mathcal{T}$) and

$$\mathbb{E}\exp(\theta(X_s - X_t)) \leq \exp(\theta^2 d(s,t)^2/2) \qquad \forall\, s,t \in \mathcal{T},\, \theta \in \mathbb{R}. \tag{A.3}$$

Now we are ready to state Dudley's inequality, which, under certain conditions, provides an upper bound of expressions of the type given in (A.1) as follows.

Theorem A.4 (Dudley's inequality) *Let $(X_t)_{t\in\mathcal{T}}$ be a sub-Gaussian process with radius $\Delta(T) := \sup_{t\in T}(\mathbb{E}|X_t|^2)^{1/2}$ and associated pseudometric d. Then,*

$$\mathbb{E}\sup_{t\in\mathcal{T}} X_t \leq 4\sqrt{2} \int_0^{\Delta(\mathcal{T})/2} \sqrt{\log \mathcal{N}(\mathcal{T}, d, \varepsilon)}\, \mathrm{d}\varepsilon. \tag{A.4}$$

Remarkably, Dudley's inequality bounds a probabilistic quantity (on the left-hand side of the inequality) by a geometric property (based on the covering numbers of the index set of the involved stochastic process on the right-hand side of (A.4)). By integrating over different levels of the covering in the so-called *Dudley's integral*, Dudley's inequality is an example for the method of *chaining*, which typically refers to *multiscale coverings* of sets. In some sources Dudley's inequality is stated with ∞ as an upper bound of the integral; however, for bounded sets and sufficiently large levels ε, the covering number of $\mathcal{T}$ equals one, so that the integrand vanishes then, and the integral is becomes definite.

The following Lemma is used to bound an integral arising from Dudley's inequality in Chapter 2. This estimate is a refinement of the similar result [FR13, Lemma C.9], which is too crude for β close to zero.

Lemma A.5 *For $\alpha, \beta > 0$ and the function Ψ being defined in* (2.50), *it holds*

$$\int_0^\alpha \sqrt{\log\left(1+\frac{\beta}{t}\right)}\mathrm{d}t \leq \alpha\Psi(\beta/\alpha), \tag{A.5}$$

where

$$\Psi(t) := \sqrt{\log(1+t) + t(\log(1+t) - \log(t))}.$$

The function Ψ satisfies $\lim_{t\to 0}\Psi(t) = 0$ and $\Psi(t) \leq \sqrt{\log(e(1+t))}$ for all $t \in \mathbb{R}$.

Note that by setting $\Psi(0) = 0$ the above estimates is trivially true also for $\beta = 0$.

Proof. We proceed similarly to the proof of [FR13, Lemma C.9] and first apply the Cauchy-Schwarz inequality to obtain

$$\int_0^\alpha \sqrt{\log(1+\beta t^{-1})}\mathrm{d}t \leq \sqrt{\int_0^\alpha 1\mathrm{d}t \cdot \int_0^\alpha \log\left(1+\beta t^{-1}\right)\mathrm{d}t}$$

For the second integral on the right hand side above we apply a change of variable and integration by parts to obtain

$$\begin{aligned}
&\int_0^\alpha \log(1+\beta t^{-1})\mathrm{d}t = \beta\int_{\beta/\alpha}^\infty u^{-2}\log(1+u)\mathrm{d}u \\
=&\beta\ -u^{-1}\log(1+u)\Big|_{\beta/\alpha}^\infty + \beta\int_{\beta\alpha}^\infty u^{-1}\frac{1}{1+u}\mathrm{d}u \\
=&\alpha\log(1+\beta/\alpha) + \beta\lim_{z\to\infty}\left[\int_{\beta/\alpha}^z \frac{1}{u}\mathrm{d}u - \int_{\beta/\alpha}^z \frac{1}{1+u}\mathrm{d}u\right] \\
=&\alpha\log(1+\beta/\alpha) + \beta\left(\log(1+\beta/\alpha) - \log(\beta/\alpha)\right) \\
=&\alpha\Psi(\beta/\alpha)
\end{aligned}$$

by the definition of Ψ. A combination with the inequality derived shows inequality (A.5). Since $\lim_{t\to 0} t\log(t) = 0$ it follows easily that also $\lim_{t\to 0}\Psi(t) = 0$. Moreover, by the mean-value theorem, there exists some $\xi \in [t, 1+t]$,

$$\log(1+t) - \log(t) = \frac{1}{\xi} \leq \frac{1}{t}.$$

Hence, $\Psi(t) \leq \sqrt{\log(1+t)+1} = \sqrt{\log(e(1+t))}$. ∎

B High-Dimensional Probability Theory

B.1 The Normal Distribution

Here, we briefly recall some basic technical notions connected with the normal distribution, and collect some necessary notations. Firstly, a real-valued random variable x is (unitary) normally distributed or Gaussian with mean μ and standard deviation σ (or variance σ^2), if it has the density function $f : \mathbb{R} \to \mathbb{R}$ given by

$$f_{\mu,\sigma^2}(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right).$$

We shortly write $x \sim \mathcal{N}(\mu, \sigma^2)$ and there is $\mathbb{E}[x] = \mu$ and $\operatorname{Var}(x) = \sigma^2$. If $x \sim \mathcal{N}(0,1)$ (*i.e.*, when $\mu = 0$ and $\sigma^2 = 1$), then x is called *standard normally distributed*. Let us further recall the *error function* erf and the *cumulative distribution function* $F_{\mu,\sigma^2}(y)$ of the univariate normal distribution $\mathcal{N}(\mu, \sigma^2)$ and their various properties that we need.

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2}\,\mathrm{d}\tau, \qquad \operatorname{erf}'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}, \qquad \operatorname{erf}(-x) = -\operatorname{erf}(x),$$

$$\operatorname{erf}(a,b) = \frac{2}{\sqrt{\pi}} \int_a^b e^{-\tau^2}\,\mathrm{d}\tau, \qquad \operatorname{erf}(a,b) = \operatorname{erf}(b) - \operatorname{erf}(a),$$

$$F_{\mu,\sigma^2}(y) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{y-\mu}{\sqrt{2\sigma^2}}\right)\right), \tag{B.1}$$

$$\operatorname{erf}(0) = 0, \qquad \lim_{x\to\infty} \operatorname{erf}(x) = 1, \qquad \lim_{x\to-\infty} \operatorname{erf}(x) = -1.$$

Furthermore, in the sequel we will require the anti-derivative H_{μ,σ^2} of the function $y \mapsto y \cdot f_{\mu,\sigma^2}(y)$ as well as the anti-derivative $G_{\mu,\sigma^2}(y)$ of the function $y \mapsto y^2 \cdot f_{\mu,\sigma^2}(y)$. They are given by given by

$$H_{\mu,\sigma^2}(y) = \frac{\sigma}{2}\left(-\frac{\mu}{\sigma}\operatorname{erf}\left(\frac{\mu-y}{\sqrt{2}\sigma}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)\right), \tag{B.2}$$

$$G_{\mu,\sigma^2}(y) = -\frac{\mu^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu-y}{\sqrt{2\sigma^2}}\right) - \frac{\sigma\,(\mu+y)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu-y)^2}{2\sigma^2}\right). \tag{B.3}$$

For $x \sim \mathcal{N}(0,1)$ we can calculate tail probabilities via the Gaussian Q-function (or, closely related, the cumulative distribution function $F_{0,1}$ of the normal distribution),

$$\mathbb{P}(x \geq t) = 1 - F_{0,1}(t) = Q(t) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}}\,\mathrm{d}x \tag{B.4}$$

B.2 Stein's Lemma

For the following classical result, see Lemma 1 (and the comment thereafter) in [Ste81].

Lemma B.1 (Stein's Lemma) *Let $x \sim \mathcal{N}(\mu, \sigma^2)$ be normally distributed. Furthermore, suppose that $f : \mathbb{R} \to \mathbb{R}$ is a differentiable function for which the two expectations $\mathbb{E}[f(x)(x-\mu)]$ and $\mathbb{E}[f'(x)]$ both exist. Then,*

$$\mathbb{E}[f(x)(x-\mu)] = \sigma^2 \mathbb{E}[f'(x)].$$

In particular, in case of the identity function $f(x) = x$, we get $\mathbb{E}[x^2 - \mu x] = \sigma^2 \mathbb{E}[x] = \sigma^2 \mu$, which can also be derived through the properties of the Chi-square distribution. We need the following consequence of Stein's Lemma involving inner products and quadratic forms. It is formulated for Gaussian random vectors, but can be extended to concentrated random vectors $\boldsymbol{x} \propto \mathcal{E}_2\left(1 \,|\, \mathbb{R}^p, \|\cdot\|_2\right)$ in the sense of Definition 1.1, and as used also in Assumption 3.7. As references, we refer to [SLCT21, Proposition 1.8 and Remark 1.9.].

Proposition B.2 *Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian random vector in $\mathbb{R}^p$. Furthermore, let $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^p$ be two (deterministic) vectors, and $\boldsymbol{A} \in \mathbb{R}^{p \times n}$ be a deterministic matrix. Then, for any function $f : \mathbb{R} \to \mathbb{R}$ which is twice differentiable, it holds that*

$$\begin{aligned} \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x}) \boldsymbol{v}^\top \boldsymbol{x}] &= \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x})] \boldsymbol{v}^\top \boldsymbol{\mu} + \mathbb{E}[f'(\boldsymbol{\omega}^\top \boldsymbol{x})] \boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{\omega}, \qquad \text{(B.5)} \\ \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x}) \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}] &= \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x})] \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}] \\ \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x}) \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}] &= \mathbb{E}[f(\boldsymbol{\omega}^\top \boldsymbol{x})] \operatorname{tr}(\boldsymbol{A}\boldsymbol{\Sigma}) + \mathcal{O}(p^{-1/2}). \end{aligned}$$

When f is the identity function, *i.e.*, $f(x) = x$, then $f'(x) = 1$ such that (B.5) simplifies to

$$\mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x} \boldsymbol{v}^\top \boldsymbol{x}\right] = \mathbb{E}\left[\boldsymbol{\omega}^\top \boldsymbol{x}\right] \boldsymbol{v}^\top \boldsymbol{\mu} + \boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{\omega}.$$

B.3 Contraction Principles

Let us recall the classical contraction principle due to Talagrand [LT91, Corollary 3.17]; see also [SSBD14b, Lemma 26.9].

Theorem B.3 (Talagrand's Contraction Principle) *Let $\mathcal{H}$ be a class of functions $h : \mathcal{X} \to \mathbb{R}$ and let $f : \mathbb{R} \to \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant K. Then, for any $(x_1, \ldots, x_n) \in \mathcal{X}^n$, it holds that*

$$\mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^n \varepsilon_i f(h(x_i)) \leq K \mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^n \varepsilon_i h(x_i),$$

where $(\varepsilon_i)_{i=1}^n$ is a finite i.i.d. Rademacher sequence.

In the context of machine learning, this statement is useful to estimate the (empirical) Rademacher complexity. Here, we think of $h \in \mathcal{H}$ as an element of the hypothesis class applied to some data $x_1, \ldots, x_n$, and we typically interpret h as the loss function. It is natural to ask (and it can be, like in this thesis, be motivated through regression problems requiring vector-valued hypothesis classes, for instance) for a generalization of the above result to vector-valued functions $h : \mathcal{X} \to \mathbb{R}^d$, and thus a (Lipschitz) function $f : \mathbb{R}^d \to \mathbb{R}$.

It turns out that the seemingly natural conjecture (with $C > 0$ being a universal constant)

$$\mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \varepsilon_i f(h(x_i)) \leq CK \mathbb{E} \sup_{h \in \mathcal{H}} \left\| \sum_{i=1}^{n} \varepsilon_i h(x_i) \right\|_2 ,$$

i.e. simply applying a norm on the right hand side in order to obtain a scalar there so that the supremum is well-defined, is *false*, as shown with a counterexample by Maurer [Mau16]. Instead, he proves the following result [Mau16, Corollary 4]. (In fact it is a corollary of a more general result shown in [Mau16], but the following formulation is useful for our purposes).

Lemma B.4 *Suppose that $\mathcal{H}$ is a set of functions $h : \mathcal{X} \to \mathbb{R}^d$ and that $f : \mathbb{R}^d \to \mathbb{R}^d$ is K-Lipschitz with respect to the ℓ_2-norm. Let $\mathcal{S} = (\boldsymbol{x}_i)_{i \in [n]}$ be the training sequence. Then*

$$\mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \varepsilon_i f \circ h(\boldsymbol{x}_i) \leq \sqrt{2} K \mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sum_{k=1}^{d} \varepsilon_{ik} h_k(\boldsymbol{x}_i),$$

where (ε_i) and (ε_{ik}) are both Rademacher sequences.

B.4 Random Matrix Theory

Deterministic equivalents. Random matrix theory studies studies matrices whose entries are real or complex-valued random variables (or equivalently, matrix-valued random variables) and in particular their spectral properties (eigenvalues and eigenvectors). For a monograph on non-asymptotic random matrix theory we refer to [Ver10]. A mathematical work on asymptotic random matrix theory is given [Tao12], and [CL22] provides a treatment with applications in machine learning.

Definition B.5 (Deterministic equivalents) We say that (the sequence; omitting the index n) a matrix $\bar{\boldsymbol{Q}} \in \mathbb{R}^{n \times n}$ is a deterministic equivalent for the symmetric random matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ (again, more precisely: in the limit $n \to \infty$) if, for (sequences of) deterministic matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ of unit spectral norm *i.e.*, $\|\boldsymbol{A}\|_{2 \to 2} = 1$, it holds that as $n \to \infty$,

$$\frac{1}{n} \operatorname{tr} \boldsymbol{A}(\boldsymbol{Q} - \bar{\boldsymbol{Q}}) \to 0,$$

with convergence in probability or almost surely.

Note that there exist alternative definitions of deterministic equaivalents, for instance with respect to quadratic forms (*i.e.*, conditions like $\boldsymbol{a}^\top(\boldsymbol{Q} - \bar{\boldsymbol{Q}})\boldsymbol{b} \to 0$ rather, or additionally to, taking the trace as in our definition above). For our purposes, the definition given above is convenient as in Chapter 3 we are interested in deterministic equivalents with respect to the trace, as this behaviour helps to characterize the performance of the classifiers studied there. Further note that deterministic equivalents are not unique, so that typically one aims to find a simple one. Deterministic equivalents are related to the mean: indeed, if the expectation can be easily calculated, one may simply choose $\bar{\boldsymbol{Q}} = \mathbb{E}\boldsymbol{Q}$. However, the entrywise computation of the mean could be elaborated or not feasible, while its asymptotic scalar behavior (under the trace) may be accessible. Therefore, in practice (and as done in Chapter 3) we try to compute the mean of the trace

$$\mathbb{E}\left[\frac{1}{n} \operatorname{tr} \boldsymbol{A}\boldsymbol{Q}\right] = \frac{1}{n} \operatorname{tr} \boldsymbol{A}\bar{\boldsymbol{Q}}$$

and try to rewrite in a way that $\bar{\boldsymbol{Q}}$ can be read off. Furthermore, the following two results are useful in Chapter 3 for computing deterministic equivalents.

Lemma B.6 *For any two invertible square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the following identity is satisfied,*

$$\boldsymbol{B} - \boldsymbol{A} = \boldsymbol{A}(\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1})\boldsymbol{B}.$$

Proof. This identity is immediately verified by multiplying both sides of the equation with $\boldsymbol{A}^{-1}$ from the left, and with $\boldsymbol{B}^{-1}$ from the right. ∎

Lemma B.7 (Sherman-Morrison-Woodbury) *For any square matrix $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ and any $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^p$ with $1 - \boldsymbol{c}\boldsymbol{A}^{-1}\boldsymbol{b}^\top \neq 0$, the rank-one perturbation $\boldsymbol{A} + \boldsymbol{b}\boldsymbol{c}^\top$ of $\boldsymbol{A}$ is invertible with*

$$\left(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{c}^\top\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{c}^\top\boldsymbol{A}^{-1}}{1 - \boldsymbol{c}^\top\boldsymbol{A}^{-1}\boldsymbol{b}}.$$

Proof. Again, it is immediately verified by a straightforward computation that the right-hand-side is indeed the inverse of $\boldsymbol{A} + \boldsymbol{b}\boldsymbol{c}^\top$. ∎

C ISTA and the Soft-Thresholding Operator

C.1 Basics and Perturbation Results

Definition of the soft-thresholding operator. The soft-thresholding function appears in ISTA as the proximal mapping of the absolute value function (or of the ℓ_1-norm in the vector-valued case) and plays a central role in this thesis. In this part of the appendix, we collect a few technical statements that are used in this thesis. Firstly, us recall that it is, for any threshold $\lambda \geq 0$, defined as

$$S_\lambda : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto \begin{cases} 0 & \text{if } |x| \ \leq \lambda, \\ x - \lambda \operatorname{sign}(x) & \text{if } |x| \ > \lambda, \end{cases}$$

which can also be expressed in closed form as $S_\lambda(x) = \operatorname{sign}(x) \cdot \max(0, |x| - \lambda)$. As a side remark, let us not that S_λ can be expressed as the sum of two *rectified linear units* via $S_\lambda(x) = \operatorname{ReLU}(x - \lambda) - \operatorname{ReLU}(-x - \lambda)$, even though we do not make us of it. (The function $\operatorname{ReLU}(x) = \max(0, x)$ is one of the most popular activation functions used by deep learning practitioners.)

Lipschitzness and perturbations bounds. Firstly, it is easy to see that the soft-thresholding function is 1-Lipschitz, *i.e.*, $|S_\lambda(x_1) - S_\lambda(x_2)| \leq |x_1 - x_2|$ for any $x_1, x_2 \in \mathbb{R}$. Similarly, this holds for higher-dimensional objects, *i.e.*,

$$\|S_\lambda(\boldsymbol{M}_1) - S_\lambda(\boldsymbol{M}_2)\|_F \leq \|\boldsymbol{M}_1 - \boldsymbol{M}_2\|_F$$

for any matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ of the same size. We also require bounds on perturbations with respect to the thresholding parameters; even though a Lipschitz-like condition also holds in this case, it depends on the dimensionality of the involved objects, as shown in the following lemma.

Lemma C.1 *Let $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ be a matrix and $\lambda_1, \lambda_2 \geq 0$ thresholding parameters. Then it holds*

$$\|S_{\lambda_1}(\boldsymbol{M}) - S_{\lambda_2}(\boldsymbol{M})\|_F \leq \sqrt{d_1 d_2} |\lambda_1 - \lambda_2|.$$

Proof. If $\lambda_1 = \lambda_2$, the inequality is trivially satisfied. Otherwise, it is easy to verify that

$$|S_{\lambda_1}(x) - S_{\lambda_2}(x)| \leq |\lambda_1 - \lambda_2| \qquad \forall x \in \mathbb{R},$$

i.e., the statement holds in the scalar case. Using this, we obtain the general statement as

$$\|S_{\lambda_1}(\boldsymbol{M}) - S_{\lambda_2}(\boldsymbol{M})\|_F = \sqrt{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \left|S_{\lambda_1}(m_{ij}) - S_{\lambda_2}(m_{ij})\right|^2} \leq \sqrt{d_1 d_2} |\lambda_1 - \lambda_2|,$$

when applying the soft-thresholding function entrywise to some matrix $\boldsymbol{M}$. ■

Spectral norm of $\boldsymbol{I} - \tau \boldsymbol{M}^\top \boldsymbol{M}$. In ISTA, an expression of the type $\boldsymbol{I} - \tau \boldsymbol{M}^\top \boldsymbol{M}$ appears in every iteration. The following Lemma provides useful statements in this context.

Lemma C.2 *For $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ some matrix, $\|\boldsymbol{I}_{d_2} - \tau \boldsymbol{M}^\top \boldsymbol{M}\|_{2\to 2}$ can be bounded as follows.*

(i) For $\tau \|\boldsymbol{M}\|_{2\to 2}^2 \leq 1$ and $d_1 < d_2$, it holds that $\left\|\boldsymbol{I}_{d_2} - \tau \boldsymbol{M}^\top \boldsymbol{M}\right\|_{2\to 2} = 1$.

(ii) For $\tau \|\boldsymbol{M}\|_{2\to 2}^2 < 1$ and $\boldsymbol{M}$ of full rank, it holds that $\left\|\boldsymbol{I}_{d_2} - \tau \boldsymbol{M}^\top \boldsymbol{M}\right\|_{2\to 2} < 1$.

In this thesis, the two parts of this Lemma are relevant in the following two scenarios.

(i) In Chapter 2, the matrix $\boldsymbol{M}$ plays the role of the measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times p}$, with m being the number of measurements and p the ambient dimension, and $d_1 = m < p = d_2$ (or even $m \ll p$) in the typical compressive sensing setup.

(ii) In Chapter 3 (see also Section 3.3 for the discussion on random fixed point equations), the matrix $\boldsymbol{M}$ plays the role of (the transpose of) the data matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ with the data dimension p and the number of samples n. Under reasonable assumptions on the data distribution, and if additionally the sample size exceeds the dimension ($d_2 = n > p = d_1$), then $\boldsymbol{X}\boldsymbol{X}^\top \in \mathbb{R}^p$ has full rank with high probability; furthermore, by choosing τ small enough we can ensure that $\tau \|\boldsymbol{X}\|_{2\to 2}^2 < 1$, as by high probability its singular values are bounded (or almost surely asymptotically by Theorem 1.12).

Proof. Assume that $\boldsymbol{M}$ is of rank k with $k \leq \min\{d_1, d_2\}$ and denote the singular values of $\boldsymbol{M}$ by $\sigma_1, \dots, \sigma_k$. Since $\boldsymbol{I}_{d_2}$ is a diagonal matrix and $\boldsymbol{M}^\top \boldsymbol{M}$ is symmetric and therefore diagonizable, the singular values of the $(d_2 \times d_2)$-matrix $\boldsymbol{I}_{d_2} - \tau \boldsymbol{M}^\top \boldsymbol{M}$ are given by

$$\underbrace{1, \dots, 1}_{d_2 - k}, |1 - \tau \sigma_1^2|, \dots, |1 - \tau \sigma_k^2|.$$

Next, recall that the singular values of $\boldsymbol{M}^\top \boldsymbol{M}$ are the squared singular values of $\boldsymbol{M}$ and further, that the spectral norm of a matrix agrees with its largest singular value. Now, in the first case the condition $d_1 < d_2$ means that $\boldsymbol{M}^\top \boldsymbol{M}$ is rank-deficient and guarantees the existence of $d_2 - d_1$ singular values of 1, while the condition $\tau \left\|\boldsymbol{M}^\top \boldsymbol{M}\right\|_{2\to 2} = \tau \left\|\boldsymbol{M}\right\|_{2\to 2}^2 \leq 1$ makes sure that $0 < \tau \sigma_i^2 \leq 1$ and therefore $|1 - \tau \sigma_i^2| \leq 1$ for $i = 1, \dots, k$, which proves (i). In the case (ii) when $\boldsymbol{M}^\top \boldsymbol{M}$ is of full rank $n = d_2 \neq d_1$, only the singular values $1 - \tau \sigma_i^2$ for $i = 1, \dots, d_2$ remain, which are strictly smaller than 1, so that in this case the strict inequality $\left\|\boldsymbol{I}_{d_2} - \tau \boldsymbol{M}^\top \boldsymbol{M}\right\|_{2\to 2} < 1$ holds. ■

C.2 A few Integrals

In the proof of Chapter 3, we used the three help functions φ, ψ and Γ. The goal of this section is to obtain precise and simplified expressions for those functions that enable a practical computation, even though requiring numerical integration. Even though already introduced before in equations (3.13), (3.14) and (3.15), let us recall the functions for convenience and in the interest of better readability.

$$\begin{aligned}
\varphi(\lambda, \mu, \sigma) &= \mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[S_\lambda(z)], \\
\psi(\lambda, \mu, \sigma) &= \mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[S'_\lambda(z)], \\
\Gamma(\lambda, \mu, \sigma) &= \mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma^2)}[S_\lambda(z)^2].
\end{aligned}$$

Note that all three functions are here defined in a one-dimensional setting $\varphi, \psi, \Gamma : \mathbb{R}_{>0} \times \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}$ (that is, corresponding to a univarite normal distribution $\mathcal{N}(\mu, \sigma^2)$, but this can be easily extended by entrywise application in case of Gaussian random vectors or matrices). In the sequel, the upcoming Lemmas C.3, C.4 and C.5 will provide the desired formulas for the three help functions φ, ψ and Γ that are used in Chapter 3.

Lemma C.3 (Mean of $S_\lambda(z)$.) *Let $z \sim \mathcal{N}(\mu, \sigma^2)$ and S_λ be the soft-thresholding operator with $\lambda > 0$. Furthermore, denote by f_{μ,σ^2} the density function of $\mathcal{N}(\mu, \sigma^2)$. Then, the $\varphi(\lambda, \mu, \sigma) = \mathbb{E}[S_\lambda(z)]$ is given by*

$$\begin{aligned}\varphi(\lambda, \mu, \sigma) =& \mu + \frac{\sigma}{\sqrt{2\pi}} \left[\exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2} \right) - \exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2} \right) \right] \\ &+ \frac{(\mu-\lambda)}{2} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) - \frac{(\mu+\lambda)}{2} \operatorname{erf}\left(\frac{(\mu+\lambda)}{\sqrt{2}\sigma} \right).\end{aligned}$$

Note that $\lim_{\lambda \to 0} \mathbb{E}[S_\lambda(z)] = \mu$, and furthermore $\lim_{\lambda \to \infty} \mathbb{E}[S_\lambda(z)] = 0$. Indeed, note that the summands containing the erf function can be rewritten as

$$\begin{aligned}&\frac{(\mu-\lambda)}{2} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) - \frac{(\mu+\lambda)}{2} \operatorname{erf}\left(\frac{(\mu+\lambda)}{\sqrt{2}\sigma} \right) \\ =& \frac{\mu}{2} \left(\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2}\sigma} \right) - \operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2}\sigma} \right) \right) - \frac{\lambda}{2} \left(\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2}\sigma} \right) + \operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2}\sigma} \right) \right)\end{aligned}$$

By passing to the limit for $\lambda \to \infty$, using basic properties of the erf function and using the rule of de L'Hospital for the second summand, we obtain

$$\lim_{\lambda \to \infty} \left[\frac{\mu}{2} \left(\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2}\sigma} \right) - \operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2}\sigma} \right) \right) - \frac{\lambda}{2} \left(\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2}\sigma} \right) + \operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2}\sigma} \right) \right) \right] = -\mu,$$

which cancels with the other summand μ, while the exponentials vanish in the limit $\lambda \to \infty$.

Proof. Since S_λ is a piecewise linear (or even constant zero) function on the intervals $(-\infty, -\lambda]$, $[-\lambda, -\lambda]$ and $[\lambda, \infty)$, the mean $\mathbb{E}_{z \sim \mathcal{N}(\mu,\sigma^2)}[S_\lambda(z)]$ can be easily obtained by integration via

$$\begin{aligned}\int_{-\infty}^{\infty} S_\lambda(y) f_{\mu,\sigma^2}(y) \,\mathrm{d}y &= \int_{-\infty}^{-\lambda} (y+\lambda) f_{\mu,\sigma^2}(y) \,\mathrm{d}y + \int_{-\lambda}^{\lambda} 0 \cdot f_{\mu,\sigma^2}(y) \,\mathrm{d}y + \int_{\lambda}^{\infty} (y-\lambda) f_{\mu,\sigma^2}(y) \,\mathrm{d}y \\ &= \int_{-\infty}^{-\lambda} (y+\lambda) f_{\mu,\sigma^2}(y) \,\mathrm{d}y + \int_{\lambda}^{\infty} (y-\lambda) f_{\mu,\sigma^2}(y) \,\mathrm{d}y \\ &= \int_{-\infty}^{0} y f_{\mu,\sigma^2}(y-\lambda) \,\mathrm{d}y + \int_{0}^{\infty} y f_{\mu,\sigma^2}(y+\lambda) \,\mathrm{d}y \\ &= \int_{-\infty}^{0} y f_{\mu+\lambda,\sigma^2}(y) \,\mathrm{d}y + \int_{0}^{\infty} y f_{\mu-\lambda,\sigma^2}(y) \,\mathrm{d}y.\end{aligned}$$

Let us first focus on the second summand and use (B.2) (replacing μ by $\mu - \lambda$, and using basic properties of the involved functions):

$$\int_{0}^{\infty} y f_{\mu-\lambda,\sigma^2}(y) \,\mathrm{d}y = \left[H_{\mu-\lambda,\sigma^2}(y) \right]_{0}^{\infty}$$

$$
\begin{aligned}
&= \left[\frac{\sigma}{2} \left(-\frac{(\mu-\lambda)}{\sigma} \operatorname{erf}\left(-\frac{y-(\mu-\lambda)}{\sqrt{2}\sigma} \right) - \sqrt{\frac{2}{\pi}} \exp\left(-\frac{(y-(\mu-\lambda))^2}{2\sigma^2} \right) \right) \right]_0^\infty \\
&= \left[\frac{\sigma}{2} \frac{(\mu-\lambda)}{\sigma} \right] - \left[\frac{\sigma}{2} \left(-\frac{(\mu-\lambda)}{\sigma} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) - \sqrt{\frac{2}{\pi}} \exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2} \right) \right) \right] \\
&= \frac{\sigma}{2} \left[\frac{(\mu-\lambda)}{\sigma} + \frac{(\mu-\lambda)}{\sigma} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) + \sqrt{\frac{2}{\pi}} \exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2} \right) \right] \\
&= \frac{(\mu-\lambda)}{2} + \frac{(\mu-\lambda)}{2} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) + \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2} \right).
\end{aligned}
$$

Next, we deal with the first summand above and again use (B.2) (this time replacing μ by $\mu + \lambda$); similar to above, we obtain

$$
\begin{aligned}
&\int_{-\infty}^{0} y f_{\mu+\lambda,\sigma^2}(y)\,\mathrm{d}y = \Big[H_{\mu+\lambda,\sigma^2}(y) \Big]_{-\infty}^{0} \\
&= \left[\frac{\sigma}{2} \left(-\frac{(\mu+\lambda)}{\sigma} \operatorname{erf}\left(-\frac{y-(\mu+\lambda)}{\sqrt{2}\sigma} \right) - \sqrt{\frac{2}{\pi}} \exp\left(-\frac{(y-(\mu+\lambda))^2}{2\sigma^2} \right) \right) \right]_{-\infty}^{0} \\
&= \left[\frac{\sigma}{2} \left(-\frac{(\mu+\lambda)}{\sigma} \operatorname{erf}\left(\frac{(\mu+\lambda)}{\sqrt{2}\sigma} \right) - \sqrt{\frac{2}{\pi}} \exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2} \right) \right) \right] + \left[\frac{(\mu+\lambda)}{2} \right] \\
&= -\frac{(\mu+\lambda)}{2} \operatorname{erf}\left(\frac{(\mu+\lambda)}{\sqrt{2}\sigma} \right) - \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2} \right) + \frac{(\mu+\lambda)}{2}.
\end{aligned}
$$

Altogether, we obtain the closed-form solution of $\varphi(\lambda, \mu, \sigma)$,

$$
\begin{aligned}
\int_{-\infty}^{\infty} S_\lambda(y) f_{\mu,\sigma^2}(y)\,\mathrm{d}y =& \mu + \frac{\sigma}{\sqrt{2\pi}} \left[\exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2} \right) - \exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2} \right) \right] \\
&+ \frac{(\mu-\lambda)}{2} \operatorname{erf}\left(\frac{(\mu-\lambda)}{\sqrt{2}\sigma} \right) - \frac{(\mu+\lambda)}{2} \operatorname{erf}\left(\frac{(\mu+\lambda)}{\sqrt{2}\sigma} \right).
\end{aligned}
$$

finishing the proof. ■

Lemma C.4 *[Mean of $S'_\lambda(z)$.] Let $z \sim \mathcal{N}(\mu, \sigma^2)$ and S_λ be the soft-thresholding operator with $\lambda > 0$. Furthermore, denote by f_{μ,σ^2} the density function of $\mathcal{N}(\mu, \sigma^2)$. Then, the mean $\mathbb{E}[S'_\lambda(z)]$ is given by*

$$
\psi(\lambda, \mu, \sigma) = \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S'_\lambda(z)] = 1 + \frac{1}{2} \left(\operatorname{erf}\left(-\frac{\lambda+\mu}{\sqrt{2\sigma^2}} \right) - \operatorname{erf}\left(\frac{\lambda-\mu}{\sqrt{2\sigma^2}} \right) \right).
$$

By the properties of the erf function, we immediately obtain $\lim_{\lambda\to\infty} \mathbb{E}[S'_\lambda(z)] = 0$.

Proof. Since S'_λ is a piecewise linear (or even constant function) on the intervals $(-\infty, -\lambda)$, $(-\lambda, -\lambda)$ and (λ, ∞). Even though not differentiable at $z = \pm\lambda$, we can calculate the mean $\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S'_\lambda(z)]$ by piecewise computation of the corresponding integrals. (More formally, one could smoothly approximate $S_\lambda(z)$. More generally, let us recall that the approach of smooth approximations [Sad+19] could be an interesting alternative to avoid the technical problems due to the non-smoothness of the soft-thresholding function.)

$$\begin{aligned}
\int_{-\infty}^{\infty} S_\lambda(y) f_{\mu,\sigma^2}(y)\,\mathrm{d}y =& \int_{-\infty}^{-\lambda} 1\cdot f_{\mu,\sigma^2}(y)\,\mathrm{d}y + \int_{-\lambda}^{\lambda} 0\cdot f_{\mu,\sigma^2}(y)\,\mathrm{d}y + \int_{\lambda}^{\infty} 1\cdot f_{\mu,\sigma^2}(y)\,\mathrm{d}y \\
=& \int_{-\infty}^{-\lambda} f_{\mu,\sigma^2}(y)\,\mathrm{d}y + \int_{\lambda}^{\infty} f_{\mu,\sigma^2}(y)\,\mathrm{d}y \\
=& \int_{-\infty}^{0} f_{\mu,\sigma^2}(y-\lambda)\,\mathrm{d}y + \int_{0}^{\infty} f_{\mu,\sigma^2}(y+\lambda)\,\mathrm{d}y \\
=& \int_{-\infty}^{0} f_{\mu-\lambda,\sigma^2}(y)\,\mathrm{d}y + \int_{0}^{\infty} f_{\mu+\lambda,\sigma^2}(y)\,\mathrm{d}y \\
=& \int_{-\infty}^{0} f_{\mu-\lambda,\sigma^2}(y)\,\mathrm{d}y + 1 - \int_{-\infty}^{0} f_{\mu+\lambda,\sigma^2}(y)\,\mathrm{d}y \\
=& \frac{1}{2}\left(1+\operatorname{erf}\left(-\frac{\lambda+\mu}{\sqrt{2\sigma^2}}\right)\right) + 1 - \frac{1}{2}\left(1+\operatorname{erf}\left(\frac{\lambda-\mu}{\sqrt{2\sigma^2}}\right)\right) \\
=& 1+\frac{1}{2}\left(\operatorname{erf}\left(-\frac{\lambda+\mu}{\sqrt{2\sigma^2}}\right) - \operatorname{erf}\left(\frac{\lambda-\mu}{\sqrt{2\sigma^2}}\right)\right),
\end{aligned}$$

finishing the proof. ■

Lemma C.5 *[Variance of $S_\lambda(z)$.] Let $z \sim \mathcal{N}(\mu,\sigma^2)$ and S_λ be the soft-thresholding operator with $\lambda > 0$. Furthermore, denote by f_{μ,σ^2} the density function of $\mathcal{N}(\mu,\sigma^2)$. Then, the variance $\Gamma(\lambda,\mu,\sigma) = \operatorname{Var}(S_\lambda(z))$ is given by*

$$\begin{aligned}
\Gamma(\lambda,\mu,\sigma) =& \mu^2+\lambda^2+\sigma^2 + \frac{(\mu+\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2\sigma^2}}\right) + \frac{\sigma(\mu+\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2}\right) \\
&- \frac{(\mu-\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2\sigma^2}}\right) - \frac{\sigma(\mu-\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2}\right) - \mathbb{E}[S_\lambda(z)]^2,
\end{aligned}$$

with $\mathbb{E}[S_\lambda(z)]$ given by Lemma C.3.

Proof. The mean $\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S_\lambda^2(z)]$ can be easily obtained by integration via

$$\begin{aligned}
&\int_{-\infty}^{\infty} S_\lambda(y)^2 f_{\mu,\sigma^2}(y)\,\mathrm{d}y \\
=& \int_{-\infty}^{-\lambda} (y+\lambda)^2 f_{\mu,\sigma^2}(y)\,\mathrm{d}y + \int_{-\lambda}^{\lambda} 0\cdot f_{\mu,\sigma^2}(y)\,\mathrm{d}y + \int_{\lambda}^{\infty} (y-\lambda)^2 f_{\mu,\sigma^2}(y)\,\mathrm{d}y \\
=& \int_{-\infty}^{0} y^2 f_{\mu+\lambda,\sigma^2}(y)\,\mathrm{d}y + \int_{0}^{\infty} y^2 f_{\mu-\lambda,\sigma^2}(y)\,\mathrm{d}y. \qquad \text{(C.1)}
\end{aligned}$$

Using the formula for the anti-derivative (B.3) allows to retrieve for the first summand in (C.1)

$$\begin{aligned}
&\int_{-\infty}^{0} y^2 f_{\mu+\lambda,\sigma^2}(y)\,\mathrm{d}y = \left[G_{\mu+\lambda,\sigma^2}(y)\right]_{-\infty}^{0} \\
=& \left[-\frac{(\mu+\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu+\lambda-y}{\sqrt{2\sigma^2}}\right) - \frac{\sigma(\mu+\lambda+y)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu+\lambda-y)^2}{2\sigma^2}\right)\right]_{-\infty}^{0} \\
=& -\frac{(\mu+\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2\sigma^2}}\right) - \frac{\sigma(\mu+\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2}\right) + \frac{(\mu+\lambda)^2+\sigma^2}{2}.
\end{aligned}$$

For the second summand in (C.1), we obtain in a similar way

$$\int_0^\infty y^2 f_{\mu-\lambda,\sigma^2}(y)\,\mathrm{d}y = \Big[G_{\mu-\lambda,\sigma^2}(y)\Big]_0^\infty$$

$$= \left[-\frac{(\mu-\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu-\lambda-y}{\sqrt{2\sigma^2}}\right) - \frac{\sigma(\mu-\lambda+y)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu-\lambda-y)^2}{2\sigma^2}\right)\right]_0^\infty$$

$$=\frac{(\mu-\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2\sigma^2}}\right) + \frac{\sigma(\mu-\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2}\right) + \frac{(\mu-\lambda)^2+\sigma^2}{2}.$$

Therefore, combining our findings finally yields

$$\begin{aligned}
&\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[S_\lambda^2(z)]\\
=&\mu^2+\lambda^2+\sigma^2+\frac{(\mu+\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu+\lambda}{\sqrt{2\sigma^2}}\right)+\frac{\sigma(\mu+\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu+\lambda)^2}{2\sigma^2}\right)\\
&\quad-\frac{(\mu-\lambda)^2+\sigma^2}{2}\operatorname{erf}\left(\frac{\mu-\lambda}{\sqrt{2\sigma^2}}\right)-\frac{\sigma(\mu-\lambda)}{\sqrt{2\pi}}\exp\left(-\frac{(\mu-\lambda)^2}{2\sigma^2}\right).
\end{aligned}$$

We can then deduce the result by using $\mathrm{Var}_{z\sim\mathcal{N}(\mu,\sigma^2)}(S_\lambda(z)) = \mathbb{E}\left[S_\lambda^2(z)\right] - \mathbb{E}\left[S_\lambda(z)\right]^2$. ■

Bibliography

[AB99] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. DOI: `10.1017/CBO9780511624216`.

[Ada05] R. Adamczak. "Logarithmic Sobolev inequalities and concentration of measure for convex functions and polynomial chaoses". *arXiv preprint math/0505175* (2005).

[Ada15] R. Adamczak. "A note on the Hanson-Wright inequality for random vectors with dependencies". *Electronic Communications in Probability* 20 (2015), 1–13.

[AG18] F. Abramovich and V. Grinshtein. "High-dimensional classification by sparse logistic regression". *IEEE Transactions on Information Theory* 65.5 (2018), 3068–3079.

[AGE20] A. Aberdam, A. Golts, and M. Elad. "Ada-LISTA: Learned Solvers Adaptive to Varying Models". *Preprint arXiv:2001.08456* (2020).

[ARPAH20] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. "On instabilities of deep learning in image reconstruction and the potential costs of AI". *Proceedings of the National Academy of Sciences* 117.48 (2020), 30088–30095.

[ASKAAN20] A. M. Alrashdi, H. Sifaou, A. Kammoun, M.-S. Alouini, and T. Y. Al-Naffouri. "Precise error analysis of the lasso under correlated designs". *arXiv preprint arXiv:2008.13033* (2020).

[ASS20] M. S. Advani, A. M. Saxe, and H. Sompolinsky. "High-dimensional dynamics of generalization error in neural networks". *Neural Networks* 132 (2020), 428–446.

[BBL03] O. Bousquet, S. Boucheron, and G. Lugosi. "Introduction to statistical learning theory". *Summer school on machine learning*. Springer. 2003, 169–207.

[BCDH10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. "Model-based compressive sensing". *IEEE Transactions on information theory* 56.4 (2010), 1982–2001.

[BD09] T. Blumensath and M. E. Davies. "Iterative hard thresholding for compressed sensing". *Applied and computational harmonic analysis* 27.3 (2009), 265–274.

[BDDW08] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. "A simple proof of the restricted isometry property for random matrices". *Constructive Approximation* 28.3 (2008), 253–263.

[Ber09] D. Bertsekas. *Convex optimization theory*. Vol. 1. Athena Scientific, 2009.

[BFT17] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. "Spectrally-normalized margin bounds for neural networks". *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. 2017, 6240–6249.

[BHSN10] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. "Compressed channel sensing: A new approach to estimating sparse multipath channels". *Proceedings of the IEEE* 98.6 (2010), 1058–1076.

[BJPD17] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. "Compressed sensing using generative models". *International Conference on Machine Learning*. PMLR. 2017, 537–546.

[BM02] P. L. Bartlett and S. Mendelson. "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results". *Journal of Machine Learning Research* 3.Nov (2002), 463–482. ISSN: ISSN 1533-7928. (Visited on 03/21/2021).

[BM11] M. Bayati and A. Montanari. “The LASSO risk for Gaussian matrices”. *IEEE Transactions on Information Theory* 58.4 (2011), 1997–2017.

[BM98] P. S. Bradley and O. L. Mangasarian. “Feature selection via concave minimization and support vector machines.” *ICML*. Vol. 98. 1998, 82–90.

[BMP12] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*. Vol. 22. Springer Science & Business Media, 2012.

[BQL21] J. Bassey, L. Qian, and X. Li. “A survey of complex-valued neural networks”. *arXiv preprint arXiv:2101.12249* (2021).

[BRMVO] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez. “Merits of Complex-Valued Neural Networks for PolSAR image segmentation”.

[BRMVO22] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez. “Comparison Between Equivalent Architectures of Complex-valued and Real-valued Neural Networks-Application on Polarimetric SAR Image Segmentation”. *Journal of Signal Processing Systems* (2022), 1–10.

[BRS22] A. Behboodi, H. Rauhut, and E. Schnoor. “Compressive Sensing and Neural Networks from a Statistical Learning Perspective”. *Compressed Sensing in Information Processing* (2022), 247–277.

[BS07] R. Baraniuk and P. Steeghs. “Compressive radar imaging”. *2007 IEEE radar conference*. IEEE. 2007, 128–133.

[BSJ21] F. Behrens, J. Sauder, and P. Jung. “Neurally Augmented ALISTA”. *International Conference on Learning Representations*. 2021.

[BVDG11] P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[BW09] R. G. Baraniuk and M. B. Wakin. “Random projections of smooth manifolds”. *Foundations of computational mathematics* 9.1 (2009), 51–77.

[CBG16] R. Couillet and F. Benaych-Georges. “Kernel spectral clustering of large dimensional data”. *Electronic Journal of Statistics* 10.1 (2016), 1393–1454.

[CD11] R. Couillet and M. Debbah. *Random Matrix Methods for Wireless Communications*. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511994746.

[CDD09] A. Cohen, W. Dahmen, and R. DeVore. “Compressed sensing and best k-term approximation”. *Journal of the American mathematical society* 22.1 (2009), 211–231.

[CDS01] S. S. Chen, D. L. Donoho, and M. A. Saunders. “Atomic decomposition by basis pursuit”. *SIAM review* 43.1 (2001), 129–159.

[CEG15] M. Chiani, A. Elzanaty, and A. Giorgetti. “Analysis of the restricted isometry property for Gaussian random matrices”. *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2015, 1–6.

[CFMW19] Y. Chen, J. Fan, C. Ma, and K. Wang. “Spectral method and regularized MLE are both optimal for top-K ranking”. *Annals of statistics* 47.4 (2019), 2204.

[CL20] R. Couillet and C. Louart. “Concentration of solutions to random equations with concentration of measure hypotheses” (2020).

[CL22] R. Couillet and Z. Liao. *Random Matrix Methods for Machine Learning*. Cambridge University Press, 2022.

[CLMPV22] A. Caragea, D. G. Lee, J. Maly, G. Pfander, and F. Voigtlaender. “Quantitative approximation results for complex-valued neural networks”. *SIAM Journal on Mathematics of Data Science* 4.2 (2022), 553–580.

[CLWY18] X. Chen, J. Liu, Z. Wang, and W. Yin. “Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds”. *Advances in Neural Information Processing Systems*. 2018, 9061–9071.

[CMW20] M. Celentano, A. Montanari, and Y. Wei. "The Lasso with general Gaussian designs with applications to hypothesis testing". *arXiv preprint arXiv:2007.13716* (2020).

[Con+17] T. O. Conrad, M. Genzel, N. Cvetkovic, N. Wulkow, A. Leichtle, J. Vybiral, G. Kutyniok, and C. Schütte. "Sparse Proteomics Analysis–a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data". *BMC bioinformatics* 18.1 (2017), 1–20.

[CP11] A. Chambolle and T. Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". *Journal of mathematical imaging and vision* 40.1 (2011), 120–145.

[CP21] P. L. Combettes and J.-C. Pesquet. "Fixed point strategies in data science". *IEEE Transactions on Signal Processing* 69 (2021), 3878–3905.

[CT05] E. J. Candès and T. Tao. "Decoding by linear programming". *IEEE transactions on information theory* 51.12 (2005), 4203–4215.

[DC18] L. Ding and Y. Chen. "Leave-one-out approach for matrix completion: Primal and dual analysis". *arXiv preprint arXiv:1803.07554* (2018).

[DDDM04] I. Daubechies, M. Defrise, and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57.11 (2004), 1413–1457.

[Don06] D. L. Donoho. "Compressed sensing". *IEEE Transactions on information theory* 52.4 (2006), 1289–1306.

[DR16] M. A. Davenport and J. Romberg. "An overview of low-rank matrix recovery from incomplete observations". *IEEE Journal of Selected Topics in Signal Processing* 10.4 (2016), 608–622.

[DS89] D. L. Donoho and P. B. Stark. "Uncertainty principles and signal recovery". *SIAM Journal on Applied Mathematics* 49.3 (1989), 906–931.

[DS96] B. DasGupta and E. Sontag. "Sample complexity for learning recurrent perceptron mappings". *IEEE Transactions on Information Theory* 42.5 (Sept. 1996), 1479–1487.

[Dud67] R. M. Dudley. "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes". *Journal of Functional Analysis* 1.3 (1967), 290–330.

[Duf13] M. Duflo. *Random iterative models*. Vol. 34. Springer Science & Business Media, 2013.

[EK09] N. El Karoui. "Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond". *The Annals of Applied Probability* 19.6 (2009), 2362–2405.

[EKBBLY13] N. El Karoui, D. Bean, P. J. Bickel, C. Lim, and B. Yu. "On robust regression with high-dimensional predictors". *Proceedings of the National Academy of Sciences* 110.36 (2013), 14557–14562.

[Ela10] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Vol. 2. 1. Springer, 2010.

[FGP07] B. Fleury, O. Guédon, and G. Paouris. "A stability result for mean width of Lp-centroid bodies". *Advances in Mathematics* 214.2 (2007), 865–877.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. New York, NY: Springer New York, 2013. (Visited on 06/18/2016).

[GAAH20] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen. "The troublesome kernel: why deep learning for inverse problems is typically unstable". *Preprint arXiv:2001.01258* (2020).

[GAK20] C. Gerbelot, A. Abbara, and F. Krzakala. "Asymptotic errors for high-dimensional convex penalized linear regression beyond gaussian matrices". *Conference on Learning Theory*. PMLR. 2020, 1682–1713.

[Gau87] C. F. Gauss. *Abhandlungen zur Methode der kleinsten Quadrate*. P. Stankiewicz, 1887.

[Geo18] A. Georgogiannis. "The Generalization Error of Dictionary Learning with Moreau Envelopes". *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, 1617–1625.

[GJBKS15] 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* 61.6 (2015), 3469–3486.

[GK22] P. Grohs and G. Kutyniok. *Mathematical Aspects of Deep Learning*. Cambridge University Press, 2022.

[GL10] K. Gregor and Y. LeCun. "Learning fast approximations of sparse coding". *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010, 399–406.

[GMM20] M. Genzel, J. Macdonald, and M. März. "Solving Inverse Problems With Deep Neural Networks – Robustness Included?" *arXiv:2011.04268* (Nov. 2020).

[Goo+20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial networks". *Communications of the ACM* 63.11 (2020), 139–144.

[GQ14] D. Goldfarb and Z. Qin. "Robust low-rank tensor recovery: Models and algorithms". *SIAM Journal on Matrix Analysis and Applications* 35.1 (2014), 225–253.

[GRS18] N. Golowich, A. Rakhlin, and O. Shamir. "Size-Independent Sample Complexity of Neural Networks". *Conference On Learning Theory*. July 2018, 297–299.

[GS10] R. Gribonval and K. Schnass. "Dictionary identification - sparse matrix-factorisation via ℓ_1-minimisation". *IEEE Transactions on Information Theory* 56.7 (2010), 3523–3539.

[GSWTY21] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye. "A review on generative adversarial networks: Algorithms, theory, and applications". *IEEE Transactions on Knowledge and Data Engineering* (2021).

[Has+20] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. "Parseval Proximal Neural Networks". en. *Journal of Fourier Analysis and Applications* 26.4 (July 2020), 59. ISSN: 1531-5851. (Visited on 03/24/2021).

[Has+21] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. "Correction to: Parseval Proximal Neural Networks". *Journal of Fourier Analysis and Applications* 27.3 (2021), 1–2.

[HHHV21] W. Huang, P. Hand, R. Heckel, and V. Voroninski. "A provably convergent scheme for compressive sensing under random generative priors". *Journal of Fourier Analysis and Applications* 27.2 (2021), 1–34.

[HKY97] J Harold, G Kushner, and G. Yin. "Stochastic approximation and recursive algorithm and applications". *Application of Mathematics* 35.10 (1997).

[HLN07] W. Hachem, P. Loubaton, and J. Najim. "Deterministic equivalents for certain functionals of large random matrices". *The Annals of Applied Probability* 17.3 (2007), 875–930.

[HMGW14] B. Huang, C. Mu, D. Goldfarb, and J. Wright. "Provable low-rank tensor recovery". *Optimization-Online* 4252.2 (2014), 455–500.

[HNS21] J. Hertrich, S. Neumayer, and G. Steidl. "Convolutional proximal neural networks and plug-and-play algorithms". *Linear Algebra and its Applications* 631 (2021), 203–234.

[HS06] G. E. Hinton and R. R. Salakhutdinov. "Reducing the dimensionality of data with neural networks". *science* 313.5786 (2006), 504–507.

[Hua20] H. Huang. "Asymptotic risk and phase transition of l_{1}-penalized robust estimator". *The Annals of Statistics* 48.5 (2020), 3090–3111.

[HV18] P. Hand and V. Voroninski. "Global guarantees for enforcing deep generative priors by empirical risk". *Conference On Learning Theory*. PMLR. 2018, 970–978.

[HV19] P. Hand and V. Voroninski. "Global guarantees for enforcing deep generative priors by empirical risk". *IEEE Transactions on Information Theory* 66.1 (2019), 401–418.

[Ito79] S. Itoh. "Random fixed point theorems with an application to random differential equations in Banach spaces". *Journal of Mathematical Analysis and Applications* 67.2 (1979), 261–273.

[JEG14] A. Jung, Y. C. Eldar, and N. Görtz. "Performance limits of dictionary learning for sparse coding". *2014 22nd European Signal Processing Conference (EUSIPCO)*. 2014, 765–769.

[JEG16] A. Jung, Y. C. Eldar, and N. Görtz. "On the Minimax Risk of Dictionary Learning". *IEEE Transactions on Information Theory* 62.3 (2016), 1501–1515.

[JGH18] A. Jacot, F. Gabriel, and C. Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". *Advances in neural information processing systems* 31 (2018).

[JNMKB20] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. "Fantastic Generalization Measures and Where to Find Them". *International Conference on Learning Representations*. 2020.

[Joh84] W. B. Johnson. "Extensions of Lipschitz mappings into a Hilbert space". *Contemp. Math.* 26 (1984), 189–206.

[Jun17] A. Jung. "A fixed-point of view on gradient methods for big data". *Frontiers in Applied Mathematics and Statistics* 3 (2017), 18.

[Jun22] A. Jung. *Machine Learning: The Basics*. Springer Nature, 2022.

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014).

[Kla07] B. Klartag. "A central limit theorem for convex sets". *Inventiones mathematicae* 168.1 (2007), 91–131.

[KM16] U. S. Kamilov and H. Mansour. "Learning optimal nonlinearities for iterative thresholding algorithms". *IEEE Signal Processing Letters* 23.5 (2016), 747–751.

[KMR14] F. Krahmer, S. Mendelson, and H. Rauhut. "Suprema of chaos processes and the restricted isometry property". *Communications on Pure and Applied Mathematics* 67.11 (2014), 1877–1904.

[KP22] V. Kouni and Y. Panagakis. "DECONET: an Unfolding Network for Analysis-based Compressed Sensing with Generalization Error Estimates". *arXiv preprint arXiv:2205.07050* (2022).

[KR17] M. Kabanava and H. Rauhut. "Masked Toeplitz covariance estimation". *arXiv preprint arXiv:1709.09377* (2017).

[KS98] P. Koiran and E. D. Sontag. "Vapnik-Chervonenkis dimension of recurrent neural networks". *Discrete Applied Mathematics*. Vapnik-Chervonenkis dimension 86.1 (Aug. 1998), 63–79. ISSN: 0166-218X. (Visited on 06/05/2020).

[KV17] A. Kolleck and J. Vybíral. “Non-Asymptotic Analysis of ℓ_1-Norm Support Vector Machines”. *IEEE Transactions on Information Theory* 63.9 (2017), 5461–5476.

[KW11] F. Krahmer and R. Ward. “New and improved Johnson–Lindenstrauss embeddings via the restricted isometry property”. *SIAM Journal on Mathematical Analysis* 43.3 (2011), 1269–1281.

[KW13] D. P. Kingma and M. Welling. “Auto-encoding variational bayes”. *arXiv preprint arXiv:1312.6114* (2013).

[LC18a] Z. Liao and R. Couillet. “The dynamics of learning: A random matrix approach”. *International Conference on Machine Learning*. PMLR. 2018, 3072–3081.

[LC18b] C. Louart and R. Couillet. “Concentration of Measure and Large Random Matrices with an application to Sample Covariance Matrices”. *arXiv preprint arXiv:1805.08295* (2018).

[LC19] J. Liu and X. Chen. “ALISTA: Analytic weights are as good as learned weights in LISTA”. *International Conference on Learning Representations (ICLR)*. 2019.

[LC20] C. Louart and R. Couillet. “Concentration of solutions to random equations with concentration of measure hypotheses”. *arXiv preprint arXiv:2010.09877* (2020).

[LCMR19] M. Lezcano-Casado and D. Martınez-Rubio. “Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group”. *International Conference on Machine Learning*. PMLR. 2019, 3794–3803.

[LCWY19] J. Liu, X. Chen, Z. Wang, and W. Yin. “ALISTA: Analytic Weights Are As Good As Learned Weights in LISTA”. *International Conference on Learning Representations*. 2019.

[LeC] Y. LeCun. “The MNIST database of handwritten digits”. *http://yann.lecun.com/exdb/mnist/* ().

[Led01] M. Ledoux. *The concentration of measure phenomenon*. 89. American Mathematical Soc., 2001.

[LHG22] C. Lee, H. Hasegawa, and S. Gao. “Complex-Valued Neural Networks: A Comprehensive Survey”. *IEEE/CAA Journal of Automatica Sinica* 9.8 (2022), 1406–1426.

[LLC18] C. Louart, Z. Liao, and R. Couillet. “A random matrix approach to neural networks”. *The Annals of Applied Probability* 28.2 (2018), 1190–1248.

[Lou23] C. Louart. “Concentration of the measure and random matrices to study data processessing algorithms”. PhD thesis. 2023.

[LSLDP05] M. Lustig, J. M. Santos, J.-H. Lee, D. L. Donoho, and J. M. Pauly. “Application of compressed sensing for rapid MR imaging”. *SPARS,(Rennes, France)* (2005).

[LT11] M. Ledoux and M. Talagrand. *Probability in Banach spaces: isoperimetry and processes*. Classics in mathematics. Berlin ; London: Springer, 2011. ISBN: 978-3-642-20211-7 978-3-642-20212-4.

[LT91] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer, 1991.

[Lub21] S. Lubjuhn. “Neural Networks motivated by Primal-Dual Algorithms for Sparse Reconstruction”. *Master Thesis* (2021).

[Mal99] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[Mau16] A. Maurer. “A vector-contraction inequality for rademacher complexities”. *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*. Springer. 2016, 3–17.

[MC18] X. Mai and R. Couillet. “Statistical analysis and improvement of large dimensional svm”. *private communication* (2018).

[McD] C McDiarmid. *Surveys in Combinatorics, Chapter On the methods of bounded differences, 148–188, 1989*.

[MF13] A. Makhzani and B. Frey. "K-sparse autoencoders". *arXiv preprint arXiv:1312.5663* (2013).

[ML19] X. Mai and Z. Liao. "High Dimensional Classification via Regularized and Unregularized Empirical Risk Minimization: Precise Error and Optimal Loss". *arXiv preprint arXiv:1905.13742* (2019).

[MLC19] X. Mai, Z. Liao, and R. Couillet. "A large scale analysis of logistic regression: Asymptotic performance and new insights". *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, 3357–3361.

[MP67] V. A. Marčenko and L. A. Pastur. "Distribution of eigenvalues for some sets of random matrices". *Mathematics of the USSR-Sbornik* 1.4 (1967), 457.

[MPB15] A. Mousavi, A. B. Patel, and R. G. Baraniuk. "A deep learning approach to structured signal recovery". *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2015, 1336–1343.

[MRT18] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[MS12] M. Meckes and S. Szarek. "Concentration for noncommutative polynomials in random matrices". *Proceedings of the American Mathematical Society* 140.5 (2012), 1803–1813.

[Nat95] B. K. Natarajan. "Sparse approximate solutions to linear systems". *SIAM journal on computing* 24.2 (1995), 227–234.

[Nau22] B. Naumova. "Neural Networks via Unfolded Iterative Optimization Algorithms fof Compressive Sensing". *Master Thesis* (2022).

[NBS18] B. Neyshabur, S. Bhojanapalli, and N. Srebro. "A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks". *International Conference on Learning Representations*. 2018.

[Ng+11] A. Ng et al. "Sparse autoencoder". *CS294A Lecture notes* 72.2011 (2011), 1–19.

[Nit97] T. Nitta. "An extension of the back-propagation algorithm to complex numbers". *Neural Networks* 10.8 (1997), 1391–1415.

[NJW01] A. Ng, M. Jordan, and Y. Weiss. "On spectral clustering: Analysis and an algorithm". *Advances in neural information processing systems* 14 (2001).

[NK19] V. Nagarajan and J. Z. Kolter. "Uniform convergence may be unable to explain generalization in deep learning". *Advances in Neural Information Processing Systems*. 2019, 11611–11622.

[Pap86] N. S. Papageorgiou. "Random fixed point theorems for measurable multifunctions in Banach spaces". *Proceedings of the American Mathematical Society* 97.3 (1986), 507–514.

[PAW07] J. L. Paredes, G. R. Arce, and Z. Wang. "Ultra-wideband compressed sensing: Channel estimation". *IEEE Journal of Selected Topics in Signal Processing* 1.3 (2007), 383–395.

[Rau] H. Rauhut. "Compressive sensing and structured random matrices". *Theoretical foundations and numerical methods for sparse recovery* 9.1 (), 92.

[Ros58] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6 (1958), 386.

[RSS17] H. Rauhut, R. Schneider, and Ž. Stojanac. "Low rank tensor recovery via iterative hard thresholding". *Linear Algebra and its Applications* 523 (2017), 220–262.

[Sad+19] M. Sadeghi, F. Ghayem, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten. "LOSoft: ℓ_0 Minimization via Soft Thresholding". *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE. 2019, 1–5.

[SBR21] E. Schnoor, A. Behboodi, and H. Rauhut. "Generalization Error Bounds for Iterative Recovery Algorithms Unfolded as Neural Networks". *arXiv preprint arXiv:2112.04364* (2021).

[SBS15] P. Sprechmann, A. M. Bronstein, and G. Sapiro. "Learning efficient sparse and low rank models". *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), 1821–1833.

[SC08] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[Sch14] K. Schnass. "On the Identifiability of Overcomplete Dictionaries via the Minimisation Principle Underlying K-SVD". *Applied and Computational Harmonic Analysis* 3 (2014), 37.

[SG18a] H. Sreter and R. Giryes. "Learned convolutional sparse coding". *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, 2191–2195.

[SG18b] H. Sreter and R. Giryes. "Learned convolutional sparse coding". *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, 2191–2195.

[Sha48] C. E. Shannon. "A mathematical theory of communication". *The Bell system technical journal* 27.3 (1948), 379–423.

[SHRHE22] J. Scarlett, R. Heckel, M. R. Rodrigues, P. Hand, and Y. C. Eldar. "Theoretical perspectives on deep learning methods in inverse problems". *arXiv preprint arXiv:2206.14373* (2022).

[Sil+16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". *nature* 529.7587 (2016), 484–489.

[Sil+17a] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. "Mastering chess and shogi by self-play with a general reinforcement learning algorithm". *arXiv preprint arXiv:1712.01815* (2017).

[Sil+17b] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. "Mastering the game of go without human knowledge". *nature* 550.7676 (2017), 354–359.

[Sil+18] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". *Science* 362.6419 (2018), 1140–1144.

[SLCT21] M. E. A. Seddik, C. Louart, R. Couillet, and M. Tamaazousti. "The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers". *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, 1045–1053.

[SLTC20] M. E. A. Seddik, C. Louart, M. Tamaazousti, and R. Couillet. "Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures". *International Conference on Machine Learning*. PMLR. 2020, 8573–8582.

[SM00] J. Shi and J. Malik. "Normalized cuts and image segmentation". *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), 888–905.

[SMG13] A. M. Saxe, J. L. McClelland, and S. Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". *arXiv preprint arXiv:1312.6120* (2013).

[SSBD14a] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[SSBD14b] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: from theory to algorithms*. New York, NY, USA: Cambridge University Press, 2014.

[SSSSHZ15] S. Sabato, S. Shalev-Shwartz, N. Srebro, D. J. Hsu, and T. Zhang. "Learning sparse low-threshold linear classifiers." *J. Mach. Learn. Res.* 16 (2015), 1275–1304.

[Ste81] C. M. Stein. "Estimation of the mean of a multivariate normal distribution". *The annals of Statistics* (1981), 1135–1151.

[Sze+13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". *arXiv preprint arXiv:1312.6199* (2013).

[Tal14] M. Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics. Springer Berlin Heidelberg, 2014. ISBN: 9783642540752.

[Tao12] T. Tao. *Topics in random matrix theory*. Vol. 132. American Mathematical Soc., 2012.

[TB20] A. Tsigler and P. L. Bartlett. "Benign overfitting in ridge regression". *arXiv preprint arXiv:2009.14286* (2020).

[Tib96] R. Tibshirani. "Regression selection and shrinkage via the lasso". *Journal of the Royal Statistical Society Series B* 58.1 (1996), 267–288.

[TOH15] C. Thrampoulidis, S. Oymak, and B. Hassibi. "Regularized linear regression: A precise analysis of the estimation error". *Conference on Learning Theory*. PMLR. 2015, 1683–1709.

[TSSCV22] M. Tiomoko, E. Schnoor, M. E. A. Seddik, I. Colin, and A. Virmaux. "Deciphering lasso-based classification through a large dimensional analysis of the iterative soft-thresholding algorithm". *International Conference on Machine Learning*. PMLR. 2022, 21449–21477.

[VC15] V. N. Vapnik and A. Y. Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". *Measures of Complexity: Festschrift for Alexey Chervonenkis*. Ed. by V. Vovk, H. Papadopoulos, and A. Gammerman. 2015, 11–30.

[Ver10] R. Vershynin. "Introduction to the non-asymptotic analysis of random matrices". *arXiv preprint arXiv:1011.3027* (2010).

[Ver18] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Vol. 47. Cambridge University Press, 2018.

[VL07] U. Von Luxburg. "A tutorial on spectral clustering". *Statistics and computing* 17.4 (2007), 395–416.

[VMB11] D. Vainsencher, S. Mannor, and A. M. Bruckstein. "The sample complexity of dictionary learning". *Journal of Machine Learning Research* 12.Nov (2011), 3259–3281.

[VW15] V. Vu and K. Wang. "Random weighted projections, random quadratic forms and random eigenvectors". *Random Structures & Algorithms* 47.4 (2015), 792–821.

[Wai19] M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. DOI: 10.1017/9781108627771.

[WGLZ20] K. Wu, Y. Guo, Z. Li, and C. Zhang. "Sparse Coding with Gated Learned ISTA". *International Conference on Learning Representations*. 2020.

[Wig55] E. P. Wigner. "Characteristic vectors of bordered matrices with infinite dimensions". *Ann. of Math.* 62 (1955), 548–564.

[Wis28] J. Wishart. “The generalised product moment distribution in samples from a normal multivariate population”. *Biometrika* (1928), 32–52.

[WRL19] Y. Wu, M. Rosca, and T. Lillicrap. “Deep Compressed Sensing”. *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, 6850–6860. URL: `https://proceedings.mlr.press/v97/wu19d.html`.

[Wu+19] S. Wu, A. Dimakis, S. Sanghavi, F. Yu, D. Holtmann-Rice, D. Storcheus, A. Rostamizadeh, and S. Kumar. “Learning a Compressed Sensing Measurement Matrix via Gradient Unrolling”. *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, 6828–6839. URL: `https://proceedings.mlr.press/v97/wu19b.html`.

[XWGWW16] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang. “Maximal sparsity with deep networks?” *Advances in Neural Information Processing Systems*. 2016, 4340–4348.

[ZBHRV17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. “Understanding deep learning requires rethinking generalization”. *International Conference on Learning Representations*. 2017.

[ZG18] J. Zhang and B. Ghanem. “ISTA-Net: interpretable optimization-inspired deep network for image compressive sensing”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 1828–1837.

[Zha11] T. Zhang. “Sparse recovery with orthogonal matching pursuit under RIP”. *IEEE transactions on information theory* 57.9 (2011), 6215–6221.

[ZRTH03] J. Zhu, S. Rosset, R. Tibshirani, and T. Hastie. “1-norm support vector machines”. *Advances in neural information processing systems* 16 (2003).

[ZWZM19] X. Zhang, D. Wang, Z. Zhou, and Y. Ma. “Robust low-rank tensor recovery with rectification and alignment”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2019), 238–255.