Consistency, Robustness and Sparsity for Learning Algorithms

Konsistenz, Robustheit und Dünnbesetztheit von Lern-Algorithmen

Der Naturwissenschaftlichen Fakultät
der
Friedrich-Alexander-Universität Erlangen-Nürnberg

zur Erlangung des Doktorgrades
Dr. rer. nat.

vorgelegt von
Tim Roith
aus
Amberg

Als Dissertation genehmigt von der Naturwissenschaftlichen Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	—
Vorsitzender des Promotionsorgans:	—
Gutachter*in:	Martin Burger
	Dejan Slepčev
	Franca Hoffmann

Acknowledgement

Coming soon…

Contents

List of Figures

Preface

This work is structured into two main parts, Part I the presentation and explanation of the topics and results presented in **??**, the peer-reviewed articles.

Part I: Exposition	**??**: **??**
Chapter 2: Learning Paradigms	—
Chapter 3: Consistent Semi-Supervised Learning on Sparse Graphs	**????**
Chapter 4: Robust and Sparse Supervised Learning	**??????**

Part I consists of five chapters, of which the first two give an introduction and explain the paradigms, *unsupervised*, *semi-supervised* and *supervised* learning. The next two chapters are split up thematically, concerning the topics semi-supervised and supervised learning respectively. Here, a short overview provides the necessary framework allowing us to explain the main contributions. The last chapter presents the conclusion. In **??** the following publications are reprinted:

[LIP-I] T. Roith and L. Bungert. “Continuum limit of Lipschitz learning on graphs.” In: *Foundations of Computational Mathematics* (2022), pp. 1–39.

[LIP-II] L. Bungert, J. Calder, and T. Roith. “Uniform convergence rates for Lipschitz learning on graphs.” In: *IMA Journal of Numerical Analysis* (Sept. 2022).

[CLIP] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. “CLIP: Cheap Lipschitz training of neural networks.” In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Proceedings*. Springer. 2021, pp. 307–319.

[BREG-I] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. “A Bregman learning framework for sparse neural networks.” In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[FNO] S. Kabri, T. Roith, D. Tenbrinck, and M. Burger. “Resolution-Invariant Image Classification based on Fourier Neural Operators.” In: *Scale Space and Variational Methods in Computer Vision: 9th International Conference, SSVM 2023, Proceedings*. Springer. 2023, pp. 307–319.

The following two works that are not part of this thesis but provide an additional insight.

[LIP-III] L. Bungert, J. Calder, and T. Roith. *Ratio convergence rates for Euclidean first-passage percolation: Applications to the graph infinity Laplacian*. 2022. arXiv: `2210.09023 [math.PR]`.

[BREG-II] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. "Neural Architecture Search via Bregman Iterations." In: (2021). arXiv: `2106.02479 [cs.LG]`.

TR's Contribution

Here we list TR's contribution to the publications included in the thesis.

[LIP-I]: This work builds upon the findings in TR's masters thesis [Roi21]. It is however important to note that the results constitute a significant extension and are conceptually stronger than the ones in [Roi21], see Section 3.3. TR adapted the continuum limit framework to the L^∞ case, worked out most of the proofs and wrote a significant part of the paper. In collaboration with LB, he identified the crucial domain assumptions that allow to work on non-convex domains and proved convergence for approximate boundary conditions.

[LIP-II]: In collaboration with LB, TR worked on the convergence proofs building upon the ideas of JC. Together with LB and JC he proved the main convergence result and the various lemmas leading up to it. Here, he was especially concerned with the adaptation of the theory for AMLEs to the graph case, with is a crucial element for the whole work. Furthermore, he contributed to the design and implementation of the numerical examples conducted in the paper.

[CLIP]: TR worked out the main algorithm proposed in the paper together with LB, based on LB's idea. Together with LS, RR and DT he conducted the numerical examples and also wrote large parts of the source code. Furthermore, he wrote significant parts of the paper, where DT proofread and clarified the final document.

[BREG-I]: TR expanded LB's ideas of employing Bregman iteration for sparse training, conceptualized by DT. Together with MB and LB he worked out the convergence analysis of stochastic Bregman iterations. Here, he also proposed a profound sparse initialization strategy. Furthermore, he conducted the numerical examples and wrote most of the source code.

[FNO]: This work is based on SK's masters thesis, employing the initial ideas of MB for resolution invariance with FNOs. In the paper TR worked out the proofs for well-definedness and Fréchet-differentiability, together with SK. He wrote large parts of the paper and the source code, where DT helped with proofreading of the published version. Here, he conducted the numerical studies in collaboration with SK.

Part I.

Exposition

Chapter 1
Introduction

The field of *machine learning* emerged in the 1950s [Sam59; Ros58], motivated by the idea of letting a computer discover algorithms and patterns without having to explicitly arrange them by hand. After the initial phase and multiple “AI-winters” [SG96], numerous important developments—e.g. the rediscovery of the backpropagation algorithm, originally due to [Kel60; Ros+62] and then popularized in [RHW86], see e.g. [Sch22]—contributed to the relevance of learning methods. The advances in computer hardware together with the availability of large amounts of data, finally allowed the machine learning enthusiasm of the recent years to spark. While “deep” learning methods—i.e. techniques involving many stacked neural layers as originally proposed in [Ros58]—are the most prominent examples, there is a whole zoo of learning-based strategies that are actively applied in fields like computer vision [Cha+21], natural language processing [Khu+23] or healthcare [She+22]. In this work we mainly focus on data-driven approaches, applied to classification tasks, where the concrete modality of the given data determines our approach. Namely, we focus on supervised—the dataset consists only of input-output pairs, i.e. is fully labeled—and semi-supervised—the data is only partially labeled—learning tasks.

For both regimes especially the last 20 years have seen great success of these data-driven methods. However, the sometimes purely heuristic learning strategies also exhibit serious drawbacks. In the supervised setting one is usually interested in the generalization behavior of a learned classifier, i.e. how good is the performance on unseen inputs which are not part of the given training data. Unfortunately in [GSS14] it was discovered, that this performance can be completely corrupted, by small, seemingly invisible perturbations known as *adversarial attacks*. More generally this phenomenon leads us to the issue of *input robustness*. Given some input x, suppose that a human and some machine would classify this input to be of type c. In a rather vague but demonstrative formulation, the key implication we want to obtain for an input $\bar{x}$ is

$$\left.\begin{array}{r}\bar{x} \text{ is close to } x,\\ \bar{x} \text{ is still classified as } c \text{ by a human}\end{array}\right\} \Rightarrow \text{the machine classifies } \bar{x} \text{ as } c.$$

Next to adversarial examples this also includes resolution changes of images, which do not change the classification by a human, if they are reasonably small. In any case, the existence of these perturbations exhibit critical flaws of learning methods and call for a better theoretical understanding of the employed models. This is where the mathematical foundation of the field becomes more relevant and properties apart from the classification performance come into play, which are discussed within this thesis.

For the semi-supervised setting we consider graph-based algorithms as originally proposed in [ZGL03] with the graph Laplacian. The main problem we highlight in this thesis was first observed in [NSZ09], namely that the classification performance deters significantly with increasing dimensionality of the data. In fact it turned out that solutions obtained by the standard graph Laplacian tend to be constant over the whole dataset, whenever the dimension is larger than two, which can be related to the Sobolev embedding theorem [AF03]. This issue is prevalent in the infinite data-limit, where a priori we consider the case, when the amount of unlabeled data points goes to infinity, which leads us to the question of *consistency* for semi-supervised algorithms.

An issue that is shared across the supervised and semi-supervised setting is the high demand for computational resources. Training a neural network usually involves the use of GPUs for long amounts of time. On the one hand this makes the process infeasible for less powerful machines or even mobile devices and on the other hand generates questionable amounts of CO_2 emissions [Hoe+21]. For graph-based semi-supervised learning one first needs to compute distances between many different data points, to obtain edge weights, which itself is a costly task. Furthermore, the computational complexity of various tasks on a given graph, scales with the number of edges. For example, the run time for Dijkstra's algorithm to compute shortest paths on a graph, already scales linearly with the amount of edges [Dij22]. In this thesis, the keyword to reduce the computational load in both cases, is *sparsity*. The concept of sparse matrices routes deeply into the field of numerical linear algebra [Lan52; GV13] and basically consists of exploiting zeros in a matrix to speed up the computation time. For neural networks this can be incorporated by enforcing the weight matrices of the layers to be sparse. For graphs, sparsity of the connectivity matrix simply means that we have only a small amount of active edges, which also reduces computational cost.

Contributions in This Work Taking up the previously mentioned subjects this thesis is concerned with *consistency*, *robustness* and *sparsity* of supervised and semi-supervised learning algorithms.

For the latter we mainly consider the so-called Lipschitz learning task [NSZ09] for which we prove convergence and convergence rates for discrete solutions to their continuum counterpart in the infinite data limit. Here, we always work in a framework that allows for very sparse and therefore computationally feasible graphs.

In the supervised regime we deal with input-robustness w.r.t. adversarial attacks and resolution changes. In the first case we propose an efficient algorithm, penalizing the Lipschitz constant [Lip77] of a neural network, which trains an adversarially robust network. In the multi-resolution setting we analyze the role of Fourier neural operators as proposed in [Li+20] and their connection to standard convolutional neural layers [Fuk80]. Concerning the computational complexity of neural network training, we propose an algorithm based on Bregman iterations [Osh+05] that allows for sparse weight matrices throughout the training. We also provide the convergence analysis for the stochastic adaption of the original Bregman iterations.

Structure of The Exposition In Chapter 2 we introduce the learning paradigms and basic notions used throughout this thesis. We then present the topics on consistency for semi-supervised learning on graphs in Chapter 3. After an explanatory introduction we then highlight the main contributions of [LIP-I; LIP-II]. Here, we try to have as little redundancy to the prints in **??** as possible, while still allowing for an understandable context. In Chapter 4 we then comment on the supervised part of this thesis. After an additional introduction the chapter contains three section presenting the works [FNO; CLIP; BREG-I] individually. Finally, in **??** we summarize the contents of the whole thesis and provide possible future directions.

Chapter 2

Learning Paradigms

Throughout this thesis, we assume to be given data $\mathcal{X}_n \subset \mathcal{X} \subset \mathbb{R}^d$ consisting of n data points. We consider the task of *learning* a function $f : \tilde{\mathcal{X}} \to \mathcal{Y}$ from the given data, where $\mathcal{Y}$ denotes the output space. In our case the set $\tilde{\mathcal{X}} \subset \mathcal{X}$ is usually chosen either as the set of data points $\mathcal{X}_n$ or as the whole space $\mathcal{X}$. The two most important cases for us are listed below.

- **Classification**: The function f assigns a label to each $x \in \tilde{\mathcal{X}}$ out of $C \in \mathbb{N}$ possible classes, i.e. $\mathcal{Y} = \{1, \dots, C\}$. In some architectures the last layer of the neural network is given as a vector $y \in \mathbb{R}^C$. Typically, this vector is a probability vector, i.e.

  $$y \in \Delta^C := \left\{ z \in [0,1]^C : \sum_{i=1}^{C} z_i = 1 \right\}.$$

  This can be enforced via the softmax function [Bri90] $\mathrm{softmax} : \mathbb{R}^C \to \mathbb{R}^C$

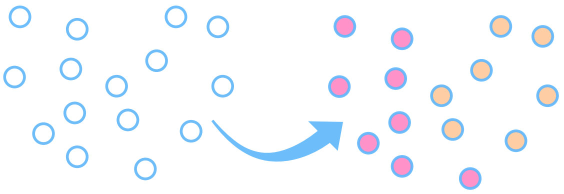
  $$\mathrm{softmax}(z)_i := \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}$$

  which was actually introduced by Boltzman in [Bol68]. This allows the interpretation that the ith entry of $f_\theta(x) \in \Delta^C$ models the probability that x belongs to class i. In order to obtain a label one can simply choose the maximum entry, i.e. $\mathrm{argmax}_{i=1,\dots,C}\, f_\theta(x)_i$.

- **Image denoising**: The function f outputs a denoised version of an input image. Here we have $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{K \times N \times M}$, where
  - $K \in \mathbb{N}$ is the number of color channels,
  - N, M denote the width and height of the image.

The learning paradigms we consider in this thesis, differ by their usage of labeled data. We review the concepts in the following.

2.1. Unsupervised Learning

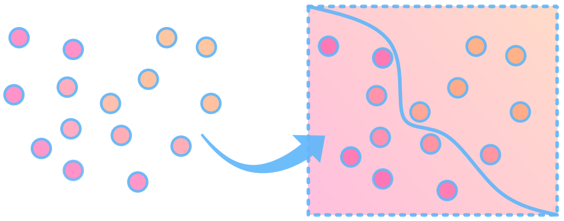


In the case of unsupervised learning we are not given any labeled data. In our context the most important application is data clustering. Other tasks involve dimensionality reduction or density estimation, see [ST14]. The clustering task consists of grouping data based on some similarity criterion. In this sense, clustering can also be interpreted as classification, i.e., the desired function is a mapping $f : \tilde{\mathcal{X}} \to \{1, \dots, C\}$ where $C \in \mathbb{N}$ denotes the number of clusters. Typically, one wants to obtain a clustering of the given data set, i.e., $\tilde{\mathcal{X}} = \mathcal{X}_n$. We list some of the typical clustering methods below:

- K-means algorithm [Ste+56],
- expectation maximization [DLR77],
- Cheeger cuts [GS15; SB09; Gar+16; GMT22],
- spectral clustering [GS18; THH21; Hof+22].

Unsupervised learning is not the main focus of this present work. However, we note that especially the concepts developed in [GS15] for Cheeger cuts are crucial for the continuum limit framework in Section 3.3.

2.2. Supervised Learning



In this setting, each data point $x \in \mathcal{X}_n$ is labeled, via a given function $g : \mathcal{X}_n \to \mathcal{Y}$ such that we have a finite training set $\mathcal{T} = \{(x, g(x)) : x \in \mathcal{X}_n\}$. The task is to infer a function defined on the underlying space, $f : \mathcal{X} \to \mathcal{Y}$, i.e., we want to assign a label to unseen inputs $x \in \mathcal{X}$ that are not necessarily part of the given data. Often, one models the problem via a joint probability function $P_{\mathcal{X},\mathcal{Y}}$ and assumes that the training data are independent and identically distributed (i.i.d) with respect to $P_{\mathcal{X},\mathcal{Y}}$. In this interpretation, the aim is to approximate the conditional $P_{\mathcal{X},\mathcal{Y}}(y|x)$ for an input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}$.

In order to *learn* the function f from the given data, one needs to choose a parameterized class of functions, where typically each element can be describe by a finite number of parameters. Among others, common methods or parametrizations include

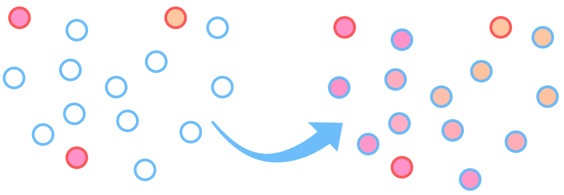
- support vector machines [CV95; SS05],

- decision trees [MS63; Bre+84],
- neural networks [Tur04; Ros58; MP69].

We refer to [Sch15] for an exhaustive historical overview.

In Chapter 4 we exclusively focus on supervised learning algorithms employing neural networks, where the concrete setting is given at the start of the chapter.

2.3. Semi-Supervised Learning



In the semi-supervised setting we assume that only a fraction of the data $\mathcal{X}_n$ is labeled, i.e., we are given a a function $g : \mathcal{O}_n \to \mathcal{Y}$ where $\mathcal{O}_n \subset \mathcal{X}_n$ is the non-empty set of labeled data. Typically this constitutes only a small fraction of all available points, i.e. $|\mathcal{O}_n| << |\mathcal{X}_n|$. In this thesis we restrict ourselves to the *transductive setting*, i.e. we want to infer a function acting only on the data $f : \mathcal{X}_n \to \mathcal{Y}$. This is opposed to the inductive setting, where f also classifies unseen points $x \in \mathcal{X}$, [Zhu05]. Common algorithms and methods include

- expectation maximization and mixture models [DLR77; CCC+03],
- self-training and co-training [BM98],
- graph-based learning [Zhu05].

Mostly, we consider the extension task with $\mathcal{Y}$ being chosen as $\mathbb{R}$. In application this can be seen as a binary classification task, where for $o \in \mathcal{O}_n$ we have $g(o) = 1$ if o belongs to a some class and $g(o) = 0$ otherwise. The function $f : \mathcal{X}_n \to \mathbb{R}$ then determines the probability that any vertex $x \in \mathcal{X}_n$ belongs to this class, where we can binarize the output via some thresholding, e.g.,

$$x \text{ belongs to the class } \Leftrightarrow f(x) > 1/2.$$

This methodology can be extended to classification tasks beyond the binary case, via the so-called one-vs-all technique [ZGL03]. Given a classification problem with $C \in \mathbb{N}$ possible classes, we assume that the labeling function $g : \mathcal{O}_n \to \Delta^C$ outputs one-hot vectors, i.e. $g(o)_c = 1$ if o belongs to class c and $g(o)_c = 0$ otherwise, for every $c = 1, \dots, C$. We then perform the binary classification problem "x belongs to class c" for every $c = 1, \dots, C$, by considering the extension task of

$$g_c : \mathcal{O}_n \to \mathbb{R} \qquad g_c(o) = g(o)_c,$$

which yield functions $f_c : \mathcal{X}_n \to [0, 1], c = 1, \dots, C$. The final output can then either obtained by emplyoing a Bayes classifier [DGL13], i.e. $f : \mathcal{X}_n \to \{1, \dots, C\}$

$$f(x) := \operatorname{argmax}_{c=1,\dots,C} f_c(x)$$

or by applying a softmax to obtain a probability vector, i.e. $f : \mathcal{X}_n \to \Delta^C$

$$f(x) := \operatorname{softmax}\left(f_1(x), \dots, f_c(x)\right).$$

In Chapter 3 we focus on graph-based learning algorithms, however we refer to [Zhu05] for an overview of semi-supervised learning algorithms.