Consistency, Robustness and Sparsity for Learning Algorithms

Konsistenz, Robustheit und Dünnbesetztheit von Lern-Algorithmen

zur Erlangung des Doktorgrades

Dr. rer. nat.

im Studiengang Mathematik

am Department Mathematik der
Friedrich-Alexander-Universität Erlangen-Nürnberg

vorgelegt am **27.Juni 2023**

von **Tim Roith**

Prüfer:	Martin Burger
Betreuer:	M.Sc. C

Für eine besondere Person,
ohne sie hätte ich es nie geschafft.

Acknowledgement

I would like to thank my supervisors for helping me finish this thesis and supporting me with their profound knowledge..

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 Part II, the peer-reviewed articles.

Part I: Exposition	Part II: Prints
??: ??	
??: ??	????
Chapter 1: Robust and Sparse Supervised Learning	????

Part I consists of three chapters, of which the first explains the paradigms, *unsupervised*, *semi-supervised* and *supervised* learning. The other chapters are the split up thematically, concerning the topics semi-supervised and supervised learning respectively. In each of these chapters a short introduction provides the necessary framework allowing us to explain the main contributions. The following publications are reprinted in Part II:

[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). DOI: 10.1093/imanum/drac048.

[CLIP] L. Bungert et al. "CLIP: Cheap Lipschitz training of neural networks." In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Virtual Event, May 16–20, 2021, Proceedings*. Springer. 2021, pp. 307–319.

[BREG-I] L. Bungert et al. "A bregman learning framework for sparse neural networks." In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[FNO] S. Kabri et al. "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.

[BREG-II] L. Bungert et al. "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 master thesis [Roi22]. It is however important to note that the results constitute a significant extension and are conceptually stronger than the ones in [Roi22], see **??**. 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. He contributed to both the numeric and the analysis 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 and RR he conducted the numerical examples and also wrote most of the source code. Furthermore, he wrote large parts of the paper.

[BREG-I]: TR expanded LB's ideas of employing Bregman iteration for sparse training. 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. Here, he conducted the numerical studies in collaboration with SK.

Part I.

Exposition

Chapter 1

Robust and Sparse Supervised Learning

In this chapter we now focus on supervised learning as described in **??**. [CLIP; BREG-II; FNO; BREG-I]

1.1. Setting

We are given a finite training set $\mathcal{T} \subset \Omega \times \Upsilon$. For a family of functions $f_\theta : \Omega \to \Upsilon$ parameterized by $\theta \in \theta$ we consider the empirical minimization

$$\min_{\theta \in \theta} \mathcal{L}(\theta)$$

where for a function $\ell : \Upsilon \times \Upsilon \to \mathbb{R}$ we define

$$\mathcal{L}(\theta) := \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y).$$

Assuming that $\mathcal{T}$ is sampled from a joint distribution π on $\Omega \times \Upsilon$ this approximates the infeasible population risk minimization

$$\int_{\Omega\times\Upsilon} \ell(f_\theta(x), y) d\pi(x, y).$$

In this thesis we focus on feed-forward neural networks, i.e., we consider layers of the form

$$\Phi(w, W, b)(z) := wz + \sigma(Wz + b)$$

where $w \in \mathbb{R}$ models a residual connection, $W \in \mathbb{R}^{n\times n}$ is a weight matrix, $b \in \mathbb{R}^n$ a bias vector and $z \in \mathbb{R}^m$. We consider a concatenation of $L \in \mathbb{N}$ such layers, which then forms a neural network

$$f_\theta = \Phi^L \circ \ldots \circ \Phi^1$$

with parameters $\theta = (w_1, \ldots, w_L, W_1, \ldots, W_L, b_1, \ldots, b_L) \in \theta$ and layers $\Phi^i := \Phi(w_i, W_i, b_i)$.

MLP

Convolutions

ResNets

1.1.1. Gradient Computation and Stochastic Gradient Descent

Training a neural network requires to solve a optimization problem w.r.t. to the parameters $\theta \in \Theta$. In this work we only focus on first order methods, however both zero [**Riedl**] and second order methods [**Hessian**] have been successfully applied in this context. Employing first order methods, requires to evaluate the gradient $\nabla_\theta \mathcal{L}$, however in this scenario it is not common to compute the full gradient but rather to have a gradient estimator. This estimator is usually obtained by randomly dividing the train set $\mathcal{T}$ into disjoint minibatches $B_1 \cup \ldots \cup B_b = \mathcal{T}$ and then successively computing the gradient of the minibatch loss

$$\frac{1}{|B_i|} \sum_{(x,y)\in B_i} \ell(f_\theta(x), y).$$

Iterating over all batches $i = 1, \ldots, b$ is referred to as one epoch. From a mathematical point of view this yields stochastic optimization methods, since in each step the true gradient is replaced by an estimator. In the abstract setting we let $(\Omega, F, \mathbb{P})$ be a probability space and consider a function $g : \Theta \times \Omega \to \Theta$ as an unbiased estimator of $\nabla \mathcal{L}$, i.e.

$$\mathbb{E}\left[g(\theta; \omega)\right] = \nabla \mathcal{L}(\theta) \text{ for all } \theta \in \Theta.$$

Most notably this method transforms the standard gradient descent update [Cau+47]

$$\theta^{(k+1)} = \theta^{(k)} - \tau^{(k)} \nabla \mathcal{L}(\theta^{(k)})$$

to *stochastic* gradient descent [RM51]

$$\begin{aligned} &\text{draw } \omega^{(k)} \text{ from } \Omega \text{ using the law of } \mathbb{P}, \\ g^{(k)} &:= g(\theta^{(k)}; \omega^{(k)}), \\ \theta^{(k+1)} &:= \theta^{(k)} - \tau^{(k)} g^{(k)}. \end{aligned}$$

1.2. Adversarial Stability

1.3. Sparsity via Bregman Iterations: [BREG-I]

Intro sparsity blah blah

1. efficency
2. robustness
3. generalization

1.3.1. Preliminaries on Convex Analysis

We first review some necessary concepts from convex analysis that allow us to introduce the framework in [BREG-I]. We refer to [BB18; Roc97; BC11] for a more exhaustive introduction to the topics. The functional J is called lower semicontinuous if $J(u) \leq \liminf_{n\to\infty} J(u_n)$ holds for all sequences $(u_n)_{n\in\mathbb{N}} \subset \Theta$ converging to u.

Definition 1.1. Given a Hilbert space Θ and a functional $J : \Theta \to (-\infty, \infty]$.

1. The functional J is called convex, if

$$J(\lambda\overline{\theta} + (1-\lambda)\theta) \leq \lambda J(\overline{\theta}) + (1-\lambda)J(\theta), \quad \forall\lambda \in [0,1], \overline{\theta}, \theta \in \Theta. \tag{1.1}$$

2. The effective domain of J is defined as $\mathrm{dom}(J) := \{\theta \in \Theta : J(\theta) \neq \infty\}$ and J is called proper if $\mathrm{dom}(J) \neq \emptyset$.

In the following we want to consider functionals J that are convex, but not necessarily differentiable. Therefor, we define the subdifferential.

Definition 1.2. of a convex and proper functional $J : \Theta \to (-\infty, \infty]$ at a point $\theta \in \Theta$ as

$$\partial J(\theta) := \left\{p \in \Theta \,:\, J(\theta) + \langle p, \overline{\theta} - \theta\rangle \leq J(\overline{\theta}),\ \forall\overline{\theta} \in \Theta\right\}. \tag{1.2}$$

If J is differentiable, then the subdifferntial coincides with the classical gradient (or Fréchet derivative). We denote $\mathrm{dom}(\partial J) := \{\theta \in \Theta : \partial J(\theta) \neq \emptyset\}$ and observe that $\mathrm{dom}(\partial J) \subset \mathrm{dom}(J)$.

The main algorithm in this section are so-called Bregman iterations, for which we first define the Bregman distance.

Definition 1.3 (Bregman Distance). Let $J : \Theta \to (-\infty, \infty]$ be a proper, convex functional. Then we define for $\theta \in \mathrm{dom}(\partial J), \overline{\theta} \in \Theta$

$$D_J^p(\overline{\theta}, \theta) := J(\overline{\theta}) - J(\theta) - \langle p, \overline{\theta} - \theta\rangle, \quad p \in \partial J(\theta). \tag{1.3}$$

For $p \in \partial J(\theta)$ and $\overline{p} \in \partial J(\overline{\theta})$ we define the *symmetric* Bregman distance as

$$D_J^{\mathrm{sym}}(\overline{\theta}, \theta) := D_J^p(\overline{\theta}, \theta) + D_J^{\overline{p}}(\theta, \overline{\theta}). \tag{1.4}$$

Intuitively, the Bregman distance $D_J^p(\overline{\theta}, \theta)$, measures the distance of J to its linearization around θ, see Fig. 1.1. If J is differentiable, then the subdifferential is single valued—we can suppress the sup script p—and we have

$$D_J(\overline{\theta}, \theta) = J(\overline{\theta}) - J(\theta) - \langle \nabla J(\theta), \overline{\theta} - \theta\rangle.$$

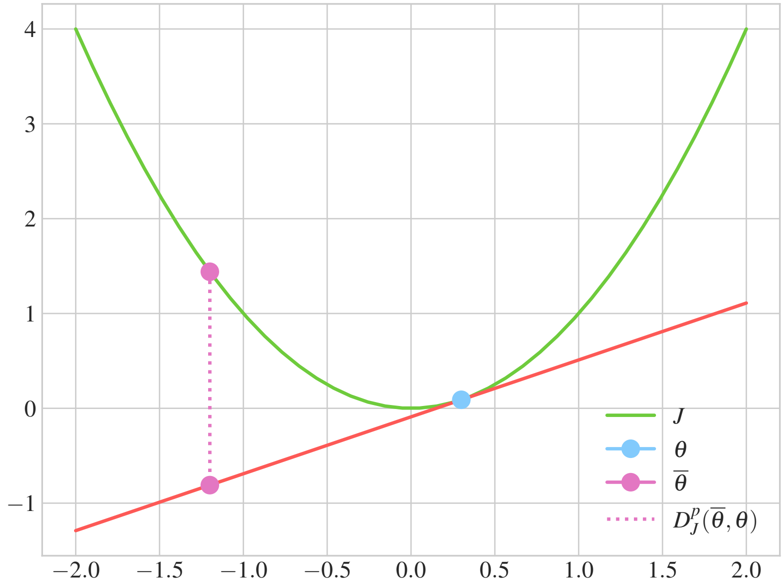


Figure 1.1.: Visualization of the Bregman distance.

Example 1.4. For $\Theta = \mathbb{R}^n$ and $J = \frac{1}{2}\|\cdot\|_2^2$ we see that $\partial J(\theta) = \{\theta\}$ and therefore

$$\begin{aligned} D_J^p(\overline{\theta},\theta) &= \frac{1}{2}\langle\overline{\theta},\overline{\theta}\rangle - \frac{1}{2}\langle\theta,\theta\rangle - \langle\theta,\overline{\theta}-\theta\rangle \\ &= \frac{1}{2}\langle\overline{\theta},\overline{\theta}\rangle + \frac{1}{2}\langle\theta,\theta\rangle - \langle\theta,\overline{\theta}\rangle \\ &= \frac{1}{2}\|\overline{\theta}-\theta\|_2^2 = J(\overline{\theta}-\theta). \end{aligned}$$

We can easily see that in general it is neither definite, symmetric nor fulfills the triangle inequality, hence it is not a metric. However, it fulfills the two distance axioms

$$D_J^p(\overline{\theta},\theta) \geq 0, \quad D_J^p(\theta,\theta) = 0, \quad \forall\overline{\theta}\in\Theta, \theta\in\operatorname{dom}(\partial J). \tag{1.5}$$

The same holds for the symmetric Bregman distance, where additionally—as the name suggests—the symmetry property is fulfilled. The last concept that is crucial in [BREG-I] is the so-called proximal operator.

Definition 1.5. Let $J : \Theta \to (-\infty,\infty]$ be convex, proper and lower semicontinuous functional, then we define the *proximal operator* as

$$\operatorname{prox}_J(\overline{\theta}) := \operatorname{argmin}_{\theta\in\Theta} \frac{1}{2}\|\theta-\overline{\theta}\|^2 + J(\theta).$$

The the optimality conditions yield for $\theta = \operatorname{prox}_J(\overline{\theta})$

$$\theta - \overline{\theta} \in \partial J(\theta).$$

If J is differentiable, we then have

$$\theta - \nabla J(\theta) = \overline{\theta} \Leftrightarrow \theta = (I + \nabla J)^{-1}(\overline{\theta}).$$

check this!

1.3.2. Bregman Iterations

We first consider the following implicit Euler scheme, for a step size $\tau > 0$

$$\theta^{(k+1)} = \operatorname{argmin}_{\theta \in \Theta} D_J^{p^{(k)}}\left(\theta, \theta^{(k)}\right) + \tau \mathcal{L}(\theta), \tag{1.6a}$$

$$p^{(k+1)} = p^{(k)} - \tau \nabla \mathcal{L}(\theta^{(k+1)}) \in \partial J(\theta^{(k+1)}) \tag{1.6b}$$

which is known as the *Bregman iteration* [Osh+05]. The intuitive interpretation here is, that in each step we want to minimize $\mathcal{L}$ while also being close to the previous iterate in terms of the Bregman distance induced by J. Therefore, the very nature of Bregman iterations means starting with a iterate $\theta^{(0)}$ that has a low value in J—preferably $J(\theta^{(0)}) = 0$—and only increase $J(\theta^{(k)})$ gradually as k increases.

Remark 1.6. Originally, the iterations were employed for solving inverse problems. Here, we are given a forward operator $A : \Theta \to \tilde{\Theta}$ and a noisy measurement $f = A\theta + \delta$ where $\delta \in \tilde{\Theta}$ is additive noise. The loss function is then of the form

$$\mathcal{L} = \frac{1}{2}\|A \cdot - f\|_2^2$$

for which one can show that the Bregman iterations converge to

$$\operatorname{argmin}_{\theta : \mathcal{L}(\theta) = 0} J(\theta), \tag{1.7}$$

see e.g. [**??**]. In comparison the concept of adding a regularizing term with parameter $\lambda > 0$, i.e. considering the problem

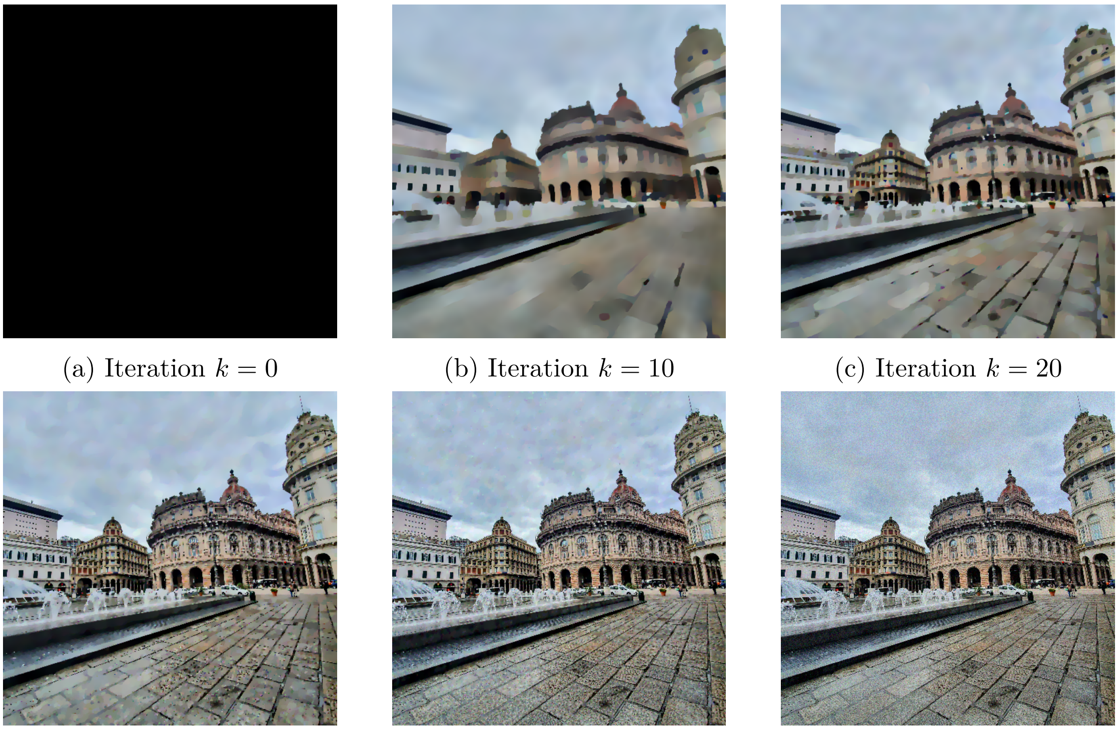
$$\min_\theta \mathcal{L}(\theta) + \lambda J(\theta)$$

actually modifies the minimizers. In this sense Bregman iterations do not introduce a bias. △

Example 1.7. In order to get an intuition about the behavior of Bregman iterations, we consider an image denoising task. I.e. we are given a noisy image $\mathbb{R}^{n \times m} \ni f = u + \delta$ where $\delta \in \mathbb{R}^{n \times m}$ is additive noise. In order to obtain $u \in \mathbb{R}^{n \times n}$ from f we employ the TV functional

$$J(u) = TV(u) := MISSING,$$

together with the loss function $\mathcal{L}(u) := \frac{1}{2}\|u - f\|_2^2$. We start with an image $u^{(0)}$ such that $TV(u^{(0)}) = 0$, i.e. a constant image. In Fig. 1.2 we visualize the iteration. At



(a) Iteration $k = 0$ (b) Iteration $k = 10$ (c) Iteration $k = 20$

(d) Iteration $k = 50$ (e) Iteration $k = 100$ (f) Iteration $k = 500$

Figure 1.2.: Bregman iterations for image denoising in Example 1.7

lower iterations $u^{(k)}$ only displays features on a larger scale, while at the end, the iteration converges back to smallest possible scale, the noisy data. In order to obtain an appropriate denoising, one needs to employ a early stopping here. This fits well to the insight from Eq. (1.7) since here the forward operator is the identity. I.e.

$$\{u : \frac{1}{2}\|u - f\|^2 = 0\} = \{f\}.$$

It should also be noted that this example only serves a explanatory purpose. In practice directly applying Eq. (1.6) for $J = TV$ can become infeasible since the first minimization problem is expensive.

If $J = \frac{1}{2}\|\cdot\|_2^2$ as in Example 1.4 then this amounts to the step

$$\theta^{(k+1)} = \text{argmin}_{\theta\in\Theta} \frac{1}{2}\|\theta - \theta^{(k)}\|_2^2 + \tau\mathcal{L}(\theta),$$

where the optimality conditions then yield

$$\theta^{(k+1)} - \theta^{(k)} + \tau\nabla\mathcal{L}(\theta^{(k+1)}) = 0 \Leftrightarrow \theta^{(k+1)} = \theta^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k+1)})$$

which is a standard implicit Euler iteration. The time continuous flow for $\tau \to 0$ is known as the *inverse scale space* flow [Bur+06; Bur+07],

$$\begin{cases} \dot{p}_t = -\nabla\mathcal{L}(\theta_t), \\ p_t \in \partial J(\theta_t), \end{cases}$$

where again for $J = \frac{1}{2}\|\cdot\|_2^2$ we obtain that $\partial J(\theta_t) = \theta_t$ and therefore obtain the standard gradient flow. Hence we see, that the inverse scale space flow is a generalization of the standard gradient flow.

1.3.3. Linearized Bregman Iterations

The minimization step in Eq. (1.6) is infeasible for large scale applications, especially in our setting of neural networks. Therefore, we employ the idea introduced in [Yin+08; COS09]. Here, we first linearize the loss function around the previous iterate,

$$\mathcal{L}(\theta) \approx \mathcal{L}(\theta^{(k)}) + \left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta - \theta^{(k)} \right\rangle.$$

The next step is to replace J with the strongly convex elastic net regularization $J_\delta := J + \frac{1}{2\delta}\|\cdot\|_2^2$. The minimization step then transforms to

$$\begin{aligned} &\text{argmin}_{\theta\in\Theta} D_{J_\delta}^{p^{(k)}}\left(\theta, \theta^{(k)}\right) + \tau \left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta \right\rangle \\ &\qquad = \text{argmin}_{\theta\in\Theta}\, J(\theta) + \frac{1}{2\delta}\|\theta\|_2^2 - \langle p^k, \theta\rangle + \tau \left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta \right\rangle \\ &\qquad = \text{argmin}_{\theta\in\Theta}\, J(\theta) + \frac{1}{2\delta}\|\theta - \delta(p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\|_2^2 - \underbrace{\|p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\|_2^2}_{\text{constant in } \theta} \\ &\qquad = \text{prox}_{\delta J}\left(\delta\left(p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\right)\right). \end{aligned} \tag{1.8}$$

Note that here $p^{(k)}$ is a subgradient of J_δ at θ therefore we derive the subgradient update rule

$$p^{(k+1)} := p^{(k)} - \tau\mathcal{L}(\theta^{(k)}).$$

This finally yields the linearized Bregman iterations

$$p^{(k+1)} = p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)}), \tag{1.9a}$$

$$\theta^{(k+1)} = \text{prox}_{\delta J}(\delta p^{(k+1)}). \tag{1.9b}$$

The last line is equivalent to $p^{(k+1)} \in \partial J_\delta(\theta^{(k+1)})$ for which we obtain the continuous linearized flow

$$\begin{aligned} \dot{p}_t &= -\nabla\mathcal{L}(\theta_t), \\ p_t &\in \partial J_\delta(\theta_t). \end{aligned}$$

A lot of citations missing here

1.3.4. Connection to Mirror Descent

As already noticed by [**??**] linearized Bregman iteration are equivalent to mirror descent in some situations. We show the equivalence in the following, where we employ similar arguments as in [**Duchi**]. One assumes to be given a differentiable and strongly convex function $h : \Theta \to \mathbb{R}$, i.e.,

$$h(\overline{\theta}) - h(\theta) - \langle \nabla h(\theta), \overline{\theta} - \theta \rangle \geq \frac{1}{2} \|\overline{\theta} - \theta\|_2^2$$

for all $\theta, \overline{\theta} \in \Theta$. The mirror descent update then reads ([NY83; BT03])

$$\theta^{(k+1)} = \nabla h^* \left(\nabla h \left(\theta^{(k)} \right) - \tau \mathcal{L}(\theta^{(k)}) \right) \tag{1.10}$$

where h^* denotes the Fenchel conjugate

$$h^*(p) = \sup_\theta \langle p, \theta \rangle - h(\theta)$$

with the gradient

$$\nabla h^*(p) = \text{argmax}_\theta \left\{ \langle p, \theta \rangle - h(\theta) \rangle \right\}.$$

Therefore, we see that Eq. (1.10) can be written as

$$\begin{aligned} \theta^{(k+1)} &= \text{argmax}_\theta \left\{ \left\langle \nabla h \left(\theta^{(k)} \right) - \tau \mathcal{L}(\theta^{(k)}), \theta \right\rangle - h(\theta) \right\} \\ &= \text{argmax}_\theta \left\{ -D_h(\theta, \theta^{(k)}) - \tau \left\langle \mathcal{L}(\theta^{(k)}), \theta \right\rangle \right\} \\ &= \text{argmin}_\theta \left\{ D_h(\theta, \theta^{(k)}) + \tau \left\langle \mathcal{L}(\theta^{(k)}), \theta \right\rangle \right\} \end{aligned}$$

which was our starting point to derive linearized Bregman iterations for $h = J_\delta$ in Eq. (1.8). In fact, we can always find a convex functional $J : \Theta \to \mathbb{R}$ such that $h = J + \frac{1}{2} \| \cdot \|_2^2$ for which we see, that Eq. (1.9) is a more general formulation of Eq. (1.10).

1.3.5. Convergence of Stochastic Bregman Iterations

We want to employ linearized Bregman iterations to train a neural network. As mentioned in Section 1.1.1 we therefore do not compute the full gradient of $\mathcal{L}$ but rather a minibatched variant. This yields stochastic Bregman Iterations

$$\begin{aligned} &\text{draw } \omega^{(k)} \text{ from } \Omega \text{ using the law of } \mathbb{P}, && (1.11a) \\ g^{(k)} &:= g(\theta^{(k)}; \omega^{(k)}), && (1.11b) \\ v^{(k+1)} &:= v^{(k)} - \tau^{(k)} g^{(k)}, && (1.11c) \\ \theta^{(k+1)} &:= \text{prox}_{\delta J}(\delta v^{(k+1)}). && (1.11d) \end{aligned}$$

While various previous works proof convergence of linearized Bregman iterations (see e.g. [Osh+05; COS09]), the stochastic setting requires special treatment. In [BREG-I] the first guarantees for the algorithm in Eq. (1.11) were proven. Other work on convergence of stochastic Bregman iterations [DEH21; HR21; ZH18; DOr+21] requires a differentiable functional J. However, since our main motivation is to a functional in the flavor of the ℓ^1 norm this is not applicable. Therefore, we present the novel convergence analysis of [BREG-I].

Assumptions We first state some basic assumptions on the loss function $\mathcal{L}$. We require it to be bounded from below and differentiable, which are both standard assumptions. Additionally, we require Lipschitz continuity of the gradient, which also commonly employed in optimization literature.

Assumption 1.8 (Loss function). We assume the following conditions on the loss function:

- The loss function $\mathcal{L}$ is bounded from below and without loss of generality we assume $\mathcal{L} \geq 0$.
- The function $\mathcal{L}$ is continuously differentiable.
- The gradient of the loss function $\theta \mapsto \nabla\mathcal{L}(\theta)$ is L-Lipschitz for $L \in (0, \infty)$:

$$\|\nabla\mathcal{L}(\tilde{\theta}) - \nabla\mathcal{L}(\theta)\| \leq L\|\tilde{\theta} - \theta\|, \quad \forall \theta, \tilde{\theta} \in \Theta. \tag{1.12}$$

If the loss function $\mathcal{L}$ fulfills the previous assumptions we are able to prove loss decay of the iterates in Theorem 1.11. However, in order to show convergence of the iterates we additionally need a convexity assumption.

Assumption 1.9 (Strong convexity). For a proper convex function $H : \Theta \to \mathbb{R}$ and $\nu \in (0, \infty)$, we say that the loss function $\theta \mapsto \mathcal{L}(\theta)$ is ν-strongly convex w.r.t. H, if

$$\mathcal{L}(\overline{\theta}) \geq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta)\overline{\theta} - \theta \rangle + \nu D_J^p(\overline{\theta}, \theta), \quad \forall \theta, \overline{\theta} \in \Theta, p \in \partial H(\theta). \tag{1.13}$$

Remark 1.10. We have two relevant cases for the choice of H. For $H = \frac{1}{2}\|\cdot\|^2$ **??** 1.9 reduces to standard strong ν-convexity. The other relevant case, is $H = J_\delta$, i.e. we consider convexity w.r.t. to the functional J_δ. △

Main Convergence Results

Theorem 1.11 (Loss decay). Assume that **??** 1.8 and **????** hold true, let $\delta > 0$, and let the step sizes satisfy $\tau^{(k)} \leq \frac{2}{\delta L}$. Then there exist constants $c, C > 0$ such that for every $k \in \mathbb{N}$ the iterates of (1.11) satisfy

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] + \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}}\mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] \\ \leq \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \tau^{(k)}\delta\frac{\sigma^2}{2c},\end{aligned} \tag{1.14}$$

1.4. Resolution Stability

Part II.

Prints

Bibliography

Books

[Roc97] R. Rockafellar. *Convex analysis.* Princeton, N.J: Princeton University Press, 1997.

[BC11] H. Bauschke and P. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces.* New York: Springer, 2011.

Articles

[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). DOI: 10.1093/imanum/drac048.

[BREG-I] L. Bungert et al. “A bregman learning framework for sparse neural networks.” In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[BREG-II] L. Bungert et al. “Neural Architecture Search via Bregman Iterations.” In: (2021). arXiv: 2106.02479 [cs.LG].

[Cau+47] A. Cauchy et al. “Méthode générale pour la résolution des systemes d’équations simultanées.” In: *Comp. Rend. Sci. Paris* 25.1847 (1847), pp. 536–538.

[RM51] H. Robbins and S. Monro. “A stochastic approximation method.” In: *The annals of mathematical statistics* (1951), pp. 400–407.

[BB18] M. Benning and M. Burger. “Modern regularization methods for inverse problems.” In: *Acta Numerica* 27 (2018), pp. 1–111.

[Osh+05] S. Osher et al. “An iterative regularization method for total variation-based image restoration.” In: *Multiscale Modeling & Simulation* 4.2 (2005), pp. 460–489.

[Bur+06] M. Burger et al. “Nonlinear inverse scale space methods.” In: *Communications in Mathematical Sciences* 4.1 (2006), pp. 179–212.

[Bur+07] M. Burger et al. “Inverse total variation flow.” In: *Multiscale Modeling & Simulation* 6.2 (2007), pp. 366–395.

[Yin+08] W. Yin et al. “Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing.” In: *SIAM Journal on Imaging sciences* 1.1 (2008), pp. 143–168.

[COS09] J.-F. Cai, S. Osher, and Z. Shen. “Convergence of the linearized Bregman iteration for ℓ_1-norm minimization.” In: *Mathematics of Computation* 78.268 (2009), pp. 2127–2136.

[NY83] A. S. Nemirovskij and D. B. Yudin. “Problem complexity and method efficiency in optimization.” In: (1983).

[BT03] A. Beck and M. Teboulle. “Mirror descent and nonlinear projected subgradient methods for convex optimization.” In: *Operations Research Letters* 31.3 (2003), pp. 167–175.

[HR21] F. Hanzely and P. Richtárik. “Fastest rates for stochastic mirror descent methods.” In: *Computational Optimization and Applications* 79 (2021), pp. 717–766.

[ZH18] S. Zhang and N. He. “On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization.” In: *arXiv preprint arXiv:1806.04781* (2018).

[DOr+21] R. D’Orazio et al. “Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize.” In: *arXiv preprint arXiv:2110.15412* (2021).

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Erlangen, den 27.Juni 2023 ....

Tim Roith