The Trade-offs of Incremental Linearization Algorithms for Nonsmooth Composite Problems

Krishna Pillutla[*] PILLUTLA@CS.WASHINGTON.EDU
Vincent Roulet VROULET@UW.EDU
University of Washington

Sham Kakade SHAM@SEAS.HARVARD.EDU
Harvard University

Zaid Harchaoui ZAID@UW.EDU
University of Washington

Abstract

Gauss-Newton methods and their stochastic version have been widely used in machine learning. Their nonsmooth counterparts, modified Gauss-Newton or prox-linear algorithms, can lead to contrasted outcomes when compared to gradient descent in large scale settings. We explore the contrasting performance of these two classes of algorithms in theory on a stylized statistical example, and experimentally on learning problems including structured prediction.

1. Introduction

Arising from the literature on non-linear least squares [1, 17], the Gauss-Newton method was proposed to tackle generic compositional problems of the form $\min_{w\in\mathbb{R}^d} f(\phi(w))$ by linearizing the inner function ϕ around the current iterate and solving the resulting subproblem [2].

The Gauss-Newton method and its variants such as the Levenberg-Marquardt method [15, 16] have been applied successfully in phase retrieval [5, 11, 20], nonlinear control [22, 24], and non-negative matrix factorization [12]. Modern machine learning problems such as deep learning possess a similar compositional structure, which makes Gauss-Newton-like algorithms potential good candidates [8, 26, 30]. However, in such problems, we are often interested in the generalization performance evaluated on unseen data. It is unclear a priori whether the additional cost of solving the subproblems can be amortized by the superior efficiency of Gauss-Newton-like algorithms.

In this paper, we investigate whether modified Gauss-Newton methods or prox-linear algorithms with incremental gradient inner loops are superior to direct stochastic subgradient algorithms for nonsmooth problems with a compositional objective and a finite-sum structure. We present a statistical example and quantify when the quadratic convergence of the exact prox-linear method is not active before hitting the noise level of the problem. We present synthetic experiments that delineate the regimes where the stochastic subgradient methods outperform the prox-linear method. We also compare these algorithms on a structured prediction problem with a convolutional neural network (end-to-end path planning). Experimental results suggest that modified Gauss-Newton methods or prox-linear algorithms offer marginal gains in some settings, and confirm the versatility of direct stochastic subgradient algorithms to tackle complex learning problems.

. [*]Now at Google Research.

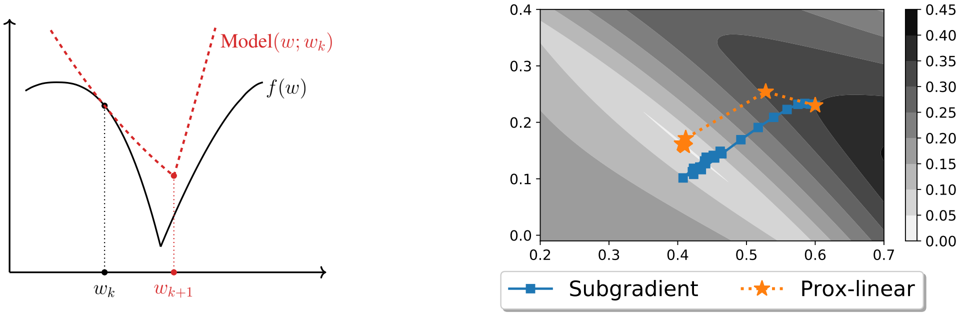

Figure 1: **Left**: The prox-linear method builds a convex model of the objective $F(w)$ around w_k, and finds w_{k+1} by minimizing this model. **Right**: A 2D illustration of the prox-linear and subgradient methods.

2. Problem Setting and Optimization Algorithms

Given $\phi_i : \mathbb{R}^d \to \mathbb{R}^k$ smooth and nonconvex, and $f : \mathbb{R}^k \to \mathbb{R}$ convex and Lipschitz, we consider finite-sum compositional minimization problems of the form

$$F(w) := \frac{1}{n}\sum_{i=1}^{n} f(\phi_i(w)) \tag{1}$$

For *multi-output regression* of input $x_i \in \mathbb{R}^p$ to output $y_i \in \mathbb{R}^k$, we take $\phi_i(w) = \varphi(x_i; w) - y_i$ as the residual of a predictor $\varphi(\cdot\,; w)$, and f. We take a nonsmooth loss function such as $f(u) = \|u\|_2$ (ℓ_2 norm without the square), which is applicable in robust regression problems. A more sophisticated example is *structured prediction*, the prediction of a combinatorial object such as a sequence. Here, $\phi_i(w)$ is a score for each structured output, and f is the structural hinge loss [6, 25, 27], computed efficiently using dynamic programming [28]. For applications, see e.g. [9, 19, 23].

We compare two families of optimization algorithms, which are stochastic and nonsmooth versions of gradient descent and the Gauss-Newton algorithm. The **stochastic subgradient method** (abbreviated *SGD*) is the nonsmooth and stochastic analogue of gradient descent. In each iteration k, SGD samples an element i_k from the available n uniformly at random and takes a step in the direction of its subgradient $v_k \in \partial(f \circ \phi_{i_k})(w_k)$:

$$w_{k+1} = w_k - \gamma v_k\,, \tag{2}$$

where γ is the learning rate and $\partial(f \circ \phi_i$ denotes the regular (or Fréchet) subdifferential. For (1), it takes a simple form $\partial(f \circ \phi_i)(w) = \nabla\phi_i(w)^\top \partial f\big(\phi_i(w)\big)$, where ∂f refers to the convex subdifferential of f and $\nabla\phi_i$ refers to the Jacobian of ϕ_i [21, Theorem 10.6]. In deep learning, the subgradient $v \in \partial(f \circ \phi_i(w))$ requires the computation of the vector-Jacobian product, readily given by reverse-mode automatic differentiation implemented in software such as PyTorch.

The **modified Gauss-Newton** method [16], also known as the **prox-linear** method [2, 3, 8], applied to (1) proceeds by finding approximate solutions of a partially linearized approximation of the objective with an additional regularization term. It computes iterates of the form

$$w_{k+1} = \operatorname*{arg\,min}_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f(\phi_i(w_k) + \nabla\phi_i(w_k)^\top(w - w_k)) + \frac{\kappa}{2}\|w - w_k\|_2^2 \tag{3}$$

As explained in Fig. 1 (left), the prox-linear method creates a *convex model* of F around w_k by linearizing the inner function as $\phi_i(w) \approx \phi_i(w_k) + \nabla\phi_i(w_k)^\top(w - w_k)$. The next iterate (3) is the minimizer of the model plus a proximal term. Compare this with the subgradient method, where the model is $F(w_k)+v^\top(w-w_k)$, where $v \in \partial F(w_k)$; see also Fig. 1 (right). In practice, it might not be possible to solve the subproblem (3) exactly. We consider using accelerated incremental algorithms such as Casimir-SVRG [18] in this case and use the abbreviation *PLI*. Computationally, this requires computing a Jacobian-vector product $\nabla\phi_i(w)v$ for some vector v. This is most efficiently computed via forward-mode automatic differentiation, which enjoys limited support in PyTorch.

3. Tradeoffs of the Prox-linear Method in Statistical Settings

Gauss-Newton methods and its variants are known to enjoy quadratic local convergence, provided the subproblems are solved exactly. In statistical learning problems, it is not meaningful to optimize beyond the noise level of the problem. If the noise level of the problem is large, then the local quadratic convergence may not be useful. We formalize this in a stylized example.

We start with a typical quadratic local convergence result of the prox-linear method in the overparameterized regime $d > nk$. We show the quadratic convergence under the assumption that the minimal singular value $\sigma_{\min}(\nabla\phi(w)^\top)$ of the transposed Jacobian of $\phi = (\phi_1; \cdots; \phi_n)$ is strictly positive, or that the Jacobian $\nabla\phi$ is surjective. The proof is given in App. A.

Proposition 1 *Consider problem* (1) *where f is ℓ-Lipschitz, convex, and μ-sharp for some $\mu > 0$ (see App. A for a precise definition). Suppose the function $\phi(w) = (\phi_1(w); \ldots; \phi_n(w)) \in \mathbb{R}^{nk}$ is L-smooth and satisfies $\sigma_{\min}(\nabla\phi(w)^\top) \ge \nu > 0$ for any $w \in \mathbb{R}^d$. Then, the sequence $(w_k)_{k=0}^\infty$ produced by the exact prox-linear algorithm* (3) *with $\kappa = L\ell$ starting from arbitrary $w_0 \in \mathbb{R}^d$ converges globally to its minimum value $F(w_k) \to F^* := \min F$. Further, as soon as an iterate w_j satisfies $F(w_j) - F^* \le (\mu\nu)^2/(L\ell n^{3/2})$, the subsequence $(w_k)_{k=j}^\infty$ converges quadratically as*

$$F(w_{k+1}) - F^* \le \frac{L\ell n^{3/2}}{2(\mu\nu)^2}\big(F(w_k) - F^*\big)^2 . \tag{4}$$

Statistical Setting. Suppose we are given n input-output pairs (x_i, y_i) where $y_i = \varphi(x_i; \overline{w}) + \xi_i$ from a parameterized nonlinear function $\varphi(\,\cdot\,; \overline{w}) : \mathbb{R}^p \to \mathbb{R}^k$ with parameters $\overline{w} \in \mathbb{R}^d$, and $\xi_i \sim \mathcal{N}(0, \sigma^2 I_k)$ is i.i.d. Gaussian noise. We wish to recover the true signal $\overline{w}$'s predictions, i.e. find $\hat{w}$ such that $\varphi(x_i; \hat{w}) \approx \varphi(x_i; \overline{w})$ for each i. We instantiate (1) with $\phi_i(w) = \varphi(x_i; w) - y_i$ and $f = \|\cdot\|_2$ in order to solve this problem. This is different from the related choice of $f = \|\cdot\|_2^2$, which corresponds to non-linear least squares regression in the fixed design setting.

We consider *early stopping* of the optimization once we reach the noise level. That is, we stop the optimization once the objective value $F(w_k)$ in iteration k falls below $F(\overline{w})$. In this setting, we now show that the prox-linear method can enjoy quadratic local convergence only when the noise level σ of the problem is small enough. To this end, we make a general assumption on the radius R of local quadratic convergence; Prop. 1 provides a concrete lower bound on R.

Proposition 2 *Fix some $\delta \in (0, 1)$ and consider problem* (1) *with ϕ_i and f as defined above with the output dimension $k \ge 4\log(2n/\delta)$. Suppose that (i) $w \mapsto \varphi(x_i; w)$ is L-smooth for each $i \in [n]$,*

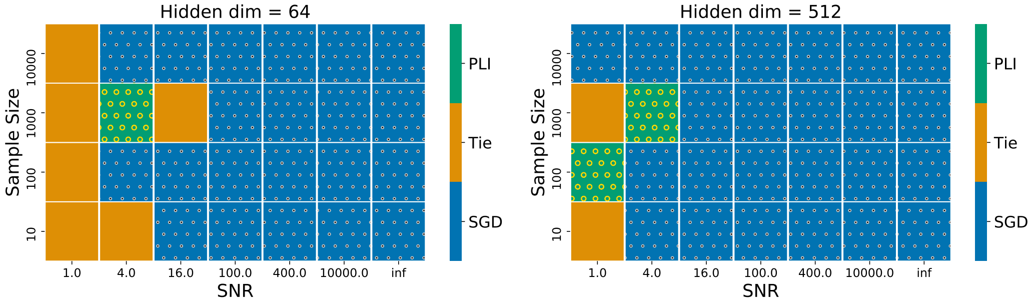

Figure 2: Synthetic multi-output regression: stochastic subgradient method (SGD) vs. the prox-linear with incremental gradient inner-loop (PLI) while varying the number of samples n and the signal-to-noise ratio (SNR). We highlight the method which finds the smallest test ℓ_2 loss.

and, (ii) φ can interpolate the data so that $\varphi(x_i; w^*) = y_i$ for each $i \in [n]$ for some $w^* \in \mathbb{R}^d$. Suppose there exists a scalar $R > 0$ and an integer j such that for all integers $k \geq j$, we have

$$F(w_k) - \min F \leq R\,, \quad \textit{and} \quad F(w_{k+1}) - \min F \leq \frac{1}{2R}\big(F(w_k) - \min F\big)^2\,. \tag{5}$$

Then, we have the following with probability at least $1 - \delta$. If the noise level satisfies $\sigma > \tilde{O}\big(R/(k^{1/2} - k^{1/4})\big)$, then the first iterate w_k enjoying quadratic convergence (5) satisfies $F(w_k) < F(\overline{w})$. Conversely, if the noise level satisfies $\sigma < \tilde{O}\big(R/(k^{1/2} + k^{1/4})\big)$, then the first iterate w_k enjoying quadratic convergence (5) satisfies $F(w_k) > F(\overline{w})$.

Prop. 2 shows that the potential advantages of the prox-linear method in terms of local quadratic convergence may not be relevant in some statistical problems where the noise level is large.

4. Experiments

We consider two setups: multi-output regression and structured prediction. We give more experiments with stochastic nonlinear equations in App. B. All hyper-parameters are tuned by grid search.

Synthetic Multi-output Regression. We consider a regression task of predicting output $y \in \mathbb{R}^k$ from input $x \in \mathbb{R}^p$, given a synthetic dataset $\{(x_i, y_i)\}_{i=1}^n$ of input-output pairs of varying size n where $p = 128$ and $k = 10$. We sample the each input $x_i \sim \mathcal{N}(0, \Sigma)$, where the covariance Σ exhibits a $1/j^2$ spectral decay. The output is generated as $y_i = \varphi^\star(x_i; w^\star) + \sigma\xi_i$, where $\varphi^\star(\cdot\,; w^\star)$ is a multilayer perceptron (MLP) with one hidden layer of width 256, and standard normal weights $w^\star$, while ξ_i is sampled from a standard Laplace distribution in $\mathbb{R}^k$ and σ is the noise scale, which we vary. We define the signal-to-noise ratio (SNR) of a problem instance as $\text{SNR} = \|w^\star\|^2/\sigma^2 \propto 1/\sigma^2$. Finally, the loss and evaluation measure we use is the nonsmooth ℓ_2 loss.

We vary the number of samples n and the SNR (equivalently, σ) and compare the two methods introduced in Sec. 2: the stochastic subgradient method (SGD) and prox-linear with incremental gradient inner loop (PLI). We tune hyperparameters to achieve the smallest loss on a held-out validation dataset in 100 epochs and report the test loss. We run the experiment in two regimes: (a) under-parameterized, where the model is a MLP with 64 hidden units, and, (b) over-parameterized, where the MLP has 512 hidden units, compared to the 256 hidden units of $\varphi^\star(\cdot\,; w^\star)$. We see in

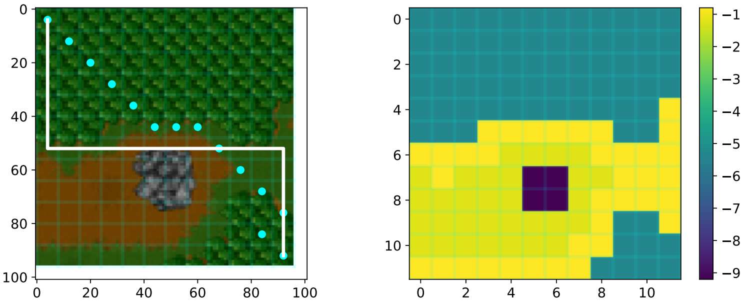

Figure 3: Planning example. **Left**: a map and its best path in (solid) white. **Right**: corresponding rewards.

Fig. 2 that **SGD tends to outperform PLI overall**, especially in high SNR regime. In the low SNR regime, PLI and SGD are mostly tied in their performance, exhibiting very similar test errors.

Path Planning as Structured Prediction. Among all monotonic paths from the top left corner to the bottom right corner of a grid, our task is to find the path that maximizes the rewards collected on each tile it passes through. Specifically, we consider images generated in the game Warcraft II [10]; see Fig. 3. Each tile corresponds to some terrain such as water, desert, grass, or rock with a fixed reward (grass $>$ desert $>$ water $>$ rock). As long as the rewards can directly be observed, this task can be solved by dynamic programming. In this experiment, the rewards are not directly observed; they are computed as the transformation of the raw pixels of each tile by a convolutional neural network. Our goal is to learn the reward function from a dataset of random maps with their associated optimal path. Given a map x with associated best path y, let denote by $\psi(x, y, y'; w)$ the score of a path y' parameterized by w. Our objective is

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \max_{y'\in\mathcal{Y}} \psi(x_i, y_i, y'; w) + \frac{\mu}{2}\|w\|_2^2, \tag{6}$$

where $(x_i)_{i=1}^n$ are the maps, $(y_i)_{i=1}^n$ are their best paths, and $\mu \geq 0$ is a regularization parameter.

The methods we consider are (i) stochastic subgradient methods [7], denoted SGD, with various learning rates strategies $\gamma_t = \gamma_0$, $\gamma_t = \gamma_0/\sqrt{t}$ and $\gamma_t = \gamma_0/t$, (ii) a variance reduced stochastic sub-gradient method, denoted SVRG [13], (iii) an accelerated algorithm on the Moreau-envelope of the objective as described in [18], denoted Casimir-SVRG, (iv) a modified Gauss-Newton algorithm as described in (3), denoted PL. For SGD, subgradients of (6) can be computed by estimating the highest reward path y' associated with a given sample (x_i, y_i) for a feature map parameterized by the current parameters w. Both Casimir-SVRG and PL require smoothing the objective (6). We take inf-convolution of the max by a squared ℓ_2 norm, which can be approximated by returning the top-K shortest paths for the given score function [18].

In Fig. 4, we observe that SGD with constant step-size carefully tuned can perform as well as more sophisticated methods such as the modified Gauss-Newton method. Most importantly, for a small regularization parameter ($\mu = 10^{-4}/n$), **SGD yields the best test Hamming loss**, which in this task is the target metric.

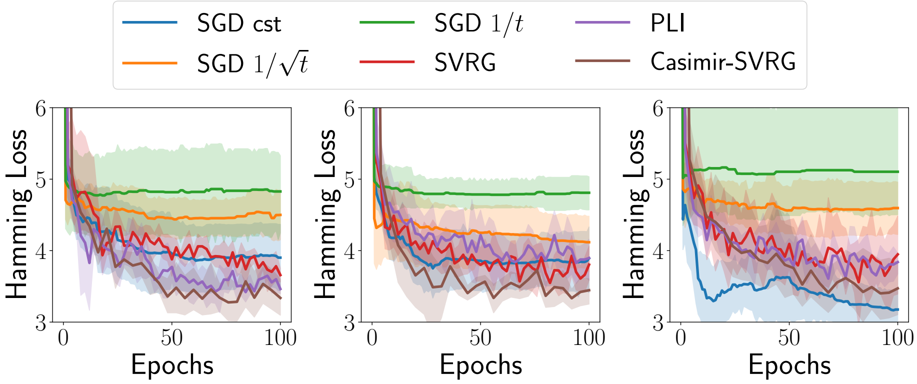

Figure 4: Planning as a structured prediction problem. We plot the Hamming loss, which measures how good the predicted path is to the actual shortest path on unseen grids. From left to right: $\mu = 1/n, 10^{-2}/n, 10^{-4}/n$

References

[1] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[2] James V Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, 1985.

[3] James V Burke and Michael C Ferris. A gauss—newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] Andrzej Cichocki and Shun-ichi Amari. *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002.

[6] Koby Crammer and Yoram Singer. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2(Dec):265–292, 2001.

[7] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[8] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019.

[9] Mark John Francis Gales, Shinji Watanabe, and Eric Fosler-Lussier. Structured Discriminative Models for Speech Recognition: An Overview. *IEEE Signal Processing Magazine*, 29(6): 70–81, 2012.

[10] Jean Guyomarch. Warcraft II open-source map-editor, 2017. URL http://github.com/war2/war2edit.

[11] James Lincoln Herring, James Nagy, and Lars Ruthotto. Gauss–Newton Optimization for Phase Recovery from the Bispectrum. *IEEE Transactions on Computational Imaging*, 6:235–247, 2019.

[12] Kejun Huang and Xiao Fu. Low-Complexity Proximal Gauss-Newton Algorithm for Nonnegative Matrix Factorization. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

[14] Beatrice Laurent and Pascal Massart. Adaptive Estimation of a Quadratic Functional by Model Selection. *Annals of Statistics*, pages 1302–1338, 2000.

[15] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[16] Yurii Nesterov. Modified Gauss–Newton scheme with worst case guarantees for global performance. *Optimisation methods and software*, 22(3):469–483, 2007.

[17] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[18] Krishna Pillutla, Vincent Roulet, Sham M. Kakade, and Zaid Harchaoui. A Smoother Way to Train Structured Prediction Models. In *NeurIPS*, 2018. URL `https://arxiv.org/pdf/1902.03228.pdf`.

[19] Nathan D. Ratliff, J. Andrew Bagnell, and Martin Zinkevich. Maximum Margin Planning. In *International Conference Machine Learning*, volume 148, pages 729–736, 2006.

[20] Audrey Repetti, Emilie Chouzenoux, and Jean-Christophe Pesquet. A nonconvex regularized approach for phase retrieval. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1753–1757. IEEE, 2014.

[21] R Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

[22] Vincent Roulet, Dmitriy Drusvyatskiy, Siddhartha S. Srinivasa, and Zaïd Harchaoui. Iterative Linearized Control: Stable Algorithms and Complexity Guarantees. In *ICML*, volume 97, pages 5518–5527, 2019.

[23] Alexander Rush. Torch-struct: Deep structured prediction library. In *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342. Association for Computational Linguistics, July 2020. doi: 10.18653/v1/2020.acl-demos.38.

[24] Athanasios Sideris and James E Bobrow. An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2275–2280. IEEE, 2005.

[25] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*, pages 25–32, 2004.

[26] Quoc Tran-Dinh, Nhan Pham, and Lam Nguyen. Stochastic Gauss-Newton Algorithms for Nonconvex Compositional Optimization. In *International Conference on Machine Learning*, pages 9572–9582. PMLR, 2020.

[27] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, page 104, 2004.

[28] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269, 1967. doi: 10.1109/TIT.1967.1054010.

[29] Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic Compositional Gradient Descent: Algorithms for Minimizing Compositions of Expected-value Functions. *Mathematical Programming*, 161(1-2):419–449, 2017.

[30] Junyu Zhang and Lin Xiao. Stochastic Variance-Reduced Prox-Linear Algorithms for Nonconvex Composite Optimization. *arXiv preprint arXiv:2004.04357*, 2020.

Appendix A. Proofs

A.1. Proof of Quadratic Convergence (Prop. 1)

Here, we give the full statement and a simple proof of Prop. 1, following standard techniques [16, Theorem 3]. We prove the proposition for the case $n = 1$ first by considering the problem $\min_{w\in\mathbb{R}^d} f\circ\phi(w)$. We then generalize to $n > 1$ for a proof of Prop. 1 in full generality.

We are interested primarily in the overparameterized case where $k \le d$. Below, we denote $\nabla\phi(w) \in \mathbb{R}^{k\times d}$ as the Jacobian of ϕ at w. We impose the assumption that its minimal singular value is bounded away from 0 as $\sigma_{\min}(\nabla\phi(w)^\top) \ge \nu > 0$ for any $w \in \mathbb{R}^d$, This assumption implies the surjectivity of the Jacobian at each w. That is, for every $u \in \mathbb{R}^k$, there exists a $v \in \mathbb{R}^d$ such that $\nabla\phi(w)v = u$.

We also assume that the following minimum values are bounded from below:

$$f^* = \min_{u\in\mathbb{R}^k} f(u), \quad \text{and,} \quad (f\circ\phi)^* = \min_{w\in\mathbb{R}^d} f\big(\phi(w)\big).$$

We have the following statement.

Proposition 3 *Consider the compositional problem $\min_{w\in\mathbb{R}^d} f\circ\phi(w)$ with following assumptions:*

(a) f is ℓ-Lipschitz continuous, convex and μ-sharp, i.e., $f(u) - f^* \ge \mu\operatorname{dist}(u, U^*)$ for any $u \in \mathbb{R}^k$ with $\mu > 0$ and $\operatorname{dist}(u, U^*)$ the Euclidean distance of u to $U^* = \arg\min_{u\in\mathbb{R}^k} f(u) \ne \emptyset$.

(b) ϕ is L-smooth and satisfies $\sigma_{\min}(\nabla\phi(w)^\top) \ge \nu > 0$ for any $w \in \mathbb{R}^d$.

The sequence $(w_k)_{k=0}^\infty$ produced by the prox-linear algorithm (3) with $M = L\ell$ starting from arbitrary $w_0 \in \mathbb{R}^d$ converges globally as $(f\circ\phi)(w_k) \to (f\circ\phi)^* = f^*$. Furthermore, as soon as an iterate w_j satisfies $f(\phi(w_k)) - (f\circ\phi)^* \le (\mu\nu)^2/(L\ell)$, the subsequence $(w_k)_{k=j}^\infty$ convergences quadratically as

$$f(\phi(w_{k+1})) - (f\circ\phi)^* \le \frac{L\ell}{2(\mu\nu)^2}(f(\phi(w_k)) - (f\circ\phi)^*)^2 \le \frac{1}{2}(f(\phi(w_k)) - (f\circ\phi)^*).$$

Proof For an iterate w_k of the prox-linear algorithm, denote $u_k^* = \mathrm{Proj}_{U^*}(\phi(w_k))$ the Euclidean projection of $\phi(w_k)$ onto the set of minimizers of f such that $\mathrm{dist}(\phi(w_k), U^*) = \|\phi(w_k) - u_k^*\|_2$.

Since the Jacobian $\nabla \phi(w_k)^\top$ is surjective, there exists v_k^* be such that $\nabla\phi(w_k) v_k^* = u_k^* - \phi(w_k)$. Furthermore, from the minimum singular value condition, there exists a choice of v_k^* such that $\|v_k^*\| \leq \|u_k^* - \phi(w)\|_2/\nu$ (see [16, Lemma 6] for a proof).

If $M \geq L\ell$, then the iterates of the prox-linear algorithm satisfy [8]

$$\begin{aligned} f(\phi(w_{k+1})) &\leq \min_{v\in\mathbb{R}^d} \left\{ f(\phi(w_k) + \nabla\phi(w_k)v) + \frac{M}{2}\|v\|_2^2 \right\} \\ &\overset{\text{(i)}}{\leq} \min_{t\in[0,1]} \left\{ f(\phi(w_k) + t\nabla\phi(w_k)v_k^*) + \frac{Mt^2}{2}\|v_k^*\|_2^2 \right\} \\ &\overset{\text{(ii)}}{\leq} \min_{t\in[0,1]} \left\{ f(\phi(w_k) + t(u_k^* - \phi(w_k)) + \frac{Mt^2}{2\nu^2}\|u_k^* - \phi(w_k)\|_2^2 \right\} \\ &\overset{\text{(iii)}}{\leq} \min_{t\in[0,1]} \left\{ tf^* + (1-t)f(\phi(w_k)) + \frac{Mt^2}{2(\nu\mu)^2}(f(\phi(w_k)) - f^*)^2 \right\}. \end{aligned} \tag{7}$$

Here, we (i) restricted the domain of the minimization to $v = tv_k^*$ with $t \in [0,1]$, (ii) plugged in the definition of v_k^* and the bound on $\|v_k^*\|$, and, (iii) used the convexity and sharpness of f. Next, by subtracting f^* from both sides, we get

$$f(\phi(w_{k+1})) - f^* \leq \min_{t\in[0,1]} \left\{ (1-t)(f(\phi(w_k)) - f^*) + \frac{t^2 M}{2(\nu\mu)^2}(f(\phi(w_k)) - f^*)^2 \right\}.$$

If $f(\phi(w_k)) - f^* \leq (\mu\nu)^2/M$, the minimum in (7) is reached at $t = 1$ and we get

$$f(\phi(w_{k+1})) - f^* \leq \frac{M}{2(\nu\mu)^2}(f(\phi(w_k)) - f^*)^2 \leq \frac{1}{2}(f(\phi(w_k)) - f^*).$$

This is the quadratic convergence phase. On the other hand, if $f(\phi(w_k)) - f^* \geq (\mu\nu)^2/M$, then the minimum in (7) is reached at $t = (\nu\mu)^2/\left(M\left(f(\phi(w_k)) - f^*\right)\right)$, and we have the bound

$$f(\phi(w_{k+1})) - f^* \leq f(\phi(w_k)) - f^* - \frac{(\mu\nu)^2}{2M}.$$

Since f is bounded from below, the sequence $f(\phi(w_k))$ converges to f^*. Hence, the minimum of the composite objective matches the minimum of the outer function, i.e., $f^* = (f \circ \phi)^*$. ■

We can now prove Prop. 1 as a corollary of Prop. 3.

Proof [Proof of Prop. 1] Consider the reduction $\phi : \mathbb{R}^d \to \mathbb{R}^{kn}$ and $\bar{f} : \mathbb{R}^{kn} \to \mathbb{R}$ given by

$$\phi(w) = (\phi_1(w); \ldots; \phi_n(w)), \quad \text{and,} \quad \bar{f}\big(u_1; \ldots; u_n\big) = \frac{1}{n}\sum_{i=1}^n f_i(u_i), \tag{8}$$

where we use semi-colons to denote the concatenation of vectors. The finite-sum problem (1) now reduces to $\min_w \bar{f} \circ \phi(w)$.

We have by definition that $\bar{f}$ is convex. Next, $\bar{f}$ is $\bar{\ell}$-Lipschitz with $\bar{\ell} = \ell/\sqrt{n}$ since

$$\begin{aligned}|\bar{f}(u) - \bar{f}(u')| &\le \frac{1}{n}\sum_{i=1}^{n} |f(u_i) - f(u_i')| \le \frac{\ell}{n}\sum_{i=1}^{n} \left\|u_i - u_i'\right\|_2 \\ &\le \frac{\ell}{\sqrt{n}}\left(\sum_{i=1}^{n}\left\|u_i - u_i'\right\|_2^2\right)^{1/2} = \frac{\ell}{\sqrt{n}}\left\|u - u'\right\|_2 .\end{aligned}$$

Further, we argue that $\bar{f}$ is $\bar{\mu}$-sharp with $\bar{\mu} = \mu/n$. Note that $(U^*)^n$ is the argmin set of $\bar{f}$ where U^* is the argmin set of f. Further, their minimum values satisfy $\bar{f}^* := \min \bar{f} = \min f = f^*$. Therefore, we have,

$$\bar{f}(u) - \bar{f}^* = \frac{1}{n}\sum_{i=1}^{n}(f(u_i) - f^*) \ge \frac{\mu}{n}\sum_{i=1}^{n}\mathrm{dist}(u_i, U^*) \ge \frac{\mu}{n}\mathrm{dist}(u, (U^*)^n),$$

where we used

$$\mathrm{dist}(u, (U^*)^n) = \sqrt{\sum_{i=1}^{n}\mathrm{dist}(u_i, U^*)^2} \le \sum_{i=1}^{n}\mathrm{dist}(u_i, U^*).$$

All the assumptions of Prop. 3 are met, and invoking it now completes the proof. ■

A.2. Proof of the Statistical Setting (Prop. 2)

We give the full statement of Prop. 2 and its proof.

Proposition 4 *Fix some $\delta \in (0, 1)$ and consider problem (1) with g_i and f as defined above with the output dimension $k \ge 4\log(2n/\delta)$. Suppose that (i) $w \mapsto \varphi(x_i; w)$ is L-smooth for each $i \in [n]$, and, (ii) the function φ can interpolate the data so that $\varphi(x_i; w^*) = y_i$ for each $i \in [n]$ for some $w^* \in \mathbb{R}^d$. Suppose there exists a scalar $R > 0$ and an integer j such that for all integers $k \ge j$, we have*

$$F(w_k) - \min F \le R\,, \quad \textit{and} \quad F(w_{k+1}) - \min F \le \frac{1}{2R}\big(F(w_k) - \min F\big)^2 . \tag{9}$$

Then, we have the following with probability at least $1 - \delta$:

(a) If the noise level satisfies

$$\sigma > \frac{R}{\sqrt{k}}\left(1 - \left(\frac{4}{k}\log(2n/\delta)\right)^{1/4}\right)^{-1},$$

then the first iterate w_k enjoying quadratic convergence (9) satisfies $F(w_k) < F(\overline{w})$.

(b) Conversely, if the noise level satisfies

$$\sigma < \frac{R}{\sqrt{k}}\left(1 + \left(\frac{16}{k}\log(2n/\delta)\right)^{1/4}\right)^{-1},$$

then the first iterate w_k enjoying quadratic convergence (9) satisfies $F(w_k) > F(\overline{w})$.

Proof First, under the interpolation assumption, we have that $0 \leq \min F \leq F(w^*) = 0$, so $\min F = 0$. Therefore, under this setting, quadratic convergence holds before the noise level if and only if $F(\overline{w}) \geq R$. To complete the proof, we show below that with probability at least $1-\delta$ that

$$\sigma\sqrt{k}\left(1-\left(\frac{4}{k}\log(2n/\delta)\right)^{1/4}\right) \leq F(\overline{w}) \leq \sigma\sqrt{k}\left(1+\left(\frac{16}{k}\log(2n/\delta)\right)^{1/4}\right). \tag{10}$$

To this end, we simplify

$$F(\overline{w}) = \frac{1}{n}\sum_{i=1}^{n}\|\varphi(x_i;\overline{w}) - y_i\|_2 = \frac{1}{n}\sum_{i=1}^{n}\|\xi_i\|_2\,. \tag{11}$$

Noting that $\|\xi_i\|_2^2$ follows a χ^2 distribution with k degrees of freedom, a standard concentration argument shows that (see example Laurent and Massart [14, Lemma 1])

$$\mathbb{P}\left(\sigma^2 k\left(1-2\sqrt{\frac{\lambda}{k}}\right) \leq \|\xi_i\|_2^2 \leq \sigma^2 k\left(1+2\sqrt{\frac{\lambda}{k}}+\frac{2\lambda}{k}\right)\right) \geq 1-2\exp(-\lambda)$$

for any $\lambda > 0$. Next, we plug in $\lambda = \log(2n/\delta)$. Noting that $\lambda/k \leq 1/4$, we use the bound $\lambda/k \leq \sqrt{\lambda/k}$ and $\sqrt{1-x} \geq 1-\sqrt{x}$ to get that

$$\sigma\sqrt{k}\left(1-(4\lambda/k)^{1/4}\right) \leq \|\xi_i\|_2 \leq \sigma\sqrt{k}\sqrt{1+4\sqrt{\lambda/k}} \leq \sigma\sqrt{k}\left(1+(16\lambda/k)^{1/4}\right)$$

with probability at least $1-\delta/n$. Invoking the union bound over $i = 1, \cdots, n$, we have with probability at least $1-\delta$ that

$$\sigma\sqrt{k}\left(1-\left(\frac{4}{k}\log(2n/\delta)\right)^{1/4}\right) \leq \|\xi_i\|_2 \leq \sigma\sqrt{k}\left(1+\left(\frac{16}{k}\log(2n/\delta)\right)^{1/4}\right)$$

holds simultaneously for each $i \in [n]$. Plugging this into (11) completes the proof. ■

Appendix B. Additional Experiments: Stochastic Nonlinear Equations

Gauss-Newton-type methods can be applied to stochastic non-linear equations of the form

$$\min_{w\in\mathbb{R}^d} f\left(\frac{1}{n}\sum_{i=1}^{n}\phi_i(w)\right) \tag{12}$$

where f is a convex, possibly nonsmooth, Lipschitz function such as $\|\cdot\|_1$ and the inner mappings are smooth and typically of the form $\phi_i(w) = \phi(x_i, w) - y_i$ [26, 30]. Problem (12) can be interpreted as ensuring that, on average, the non-linear mapping $\phi(\cdot, w)$ maps the inputs x_i to the targets y_i.
Algorithms. Denoting $\phi(w) = \frac{1}{n}\sum_{i=1}^{n}\phi_i(w)$, a natural baseline algorithm is to compute iterates as

$$w_{k+1} = w_k - \gamma\widehat{\nabla}\phi(w)^\top\nabla f(\widehat{\phi}(w_k))), \tag{13}$$

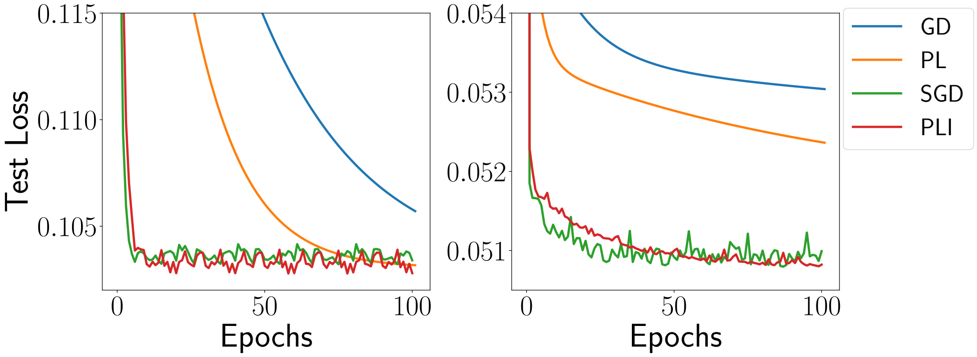

Figure 5: Solving stochastic nonlinear equations. We plot the "test loss", which is the objective value on a separate testing set. **Left**: ijccn1 dataset. **Right**: covtype dataset

where $\widehat{\nabla}\phi(w)$ and $\widehat{\phi}(w)$ are approximations of $\nabla\phi(w)$ and $\phi(w)$ respectively that can be approximated from a mini-batch [29]; we call this "SGD". Note that the minibatch subgradient estimates can be biased since the outer function f can be non-linear. A modified Gauss-Newton or Prox-Linear method adapted to the inner finite-sum performs the iterations

$$w_{k+1} = \underset{w \in \mathbb{R}^d}{\arg\min} \left\{ f\left(\widehat{\phi}(w_k) + \widehat{\nabla}\phi(w_k)(w - w_k)\right) + \frac{M}{2}\|w - w_k\|_2^2 \right\} \tag{14}$$

where each sub-problem can be solved by incremental algorithms such as the accelerated dual proximal gradient ascent [26, 30].

Experiment. We consider the experimental setting of [26]. The objective is to solve (12) where f is the Huber loss, a smooth surrogate of the nonsmooth ℓ_1 norm, and inner mappings ϕ are the concatenation of four different losses, i.e., $\phi_i(w) = (\ell_1(x_i^\top w, y_i), \ldots, \ell_4(x_i^\top w, y_i))$, where the formulations of the losses can be found in [26]. The samples (x_i, y_i) are drawn from the datasets `ijcnn1` or `covtype` from the LIBSVM repository [4].

We consider (i) a gradient descent denoted GD, (ii) a modified Gauss-Newton or prox-linear method denoted PL, (iii) a baseline of the form (13), denoted SGD, (iv) an incremental Gauss-Newton or prox-linear method as described in (14), denoted PLI for consistency. In Fig. 5, we observe that PL outperforms GD as expected in the batch setting. However, this advantage is longer present in the incremental setting. The analogue of gradient descent, namely the SGD baseline (13), often performs better than its Gauss-Newton counterpart PLI.