SIAM J. OPTIM.
Vol. 29, No. 4, pp. 2916–2948

A STOCHASTIC SEMISMOOTH NEWTON METHOD FOR NONSMOOTH NONCONVEX OPTIMIZATION*

ANDRE MILZAREK†, XIANTAO XIAO‡, SHICONG CEN§, ZAIWEN WEN¶, AND MICHAEL ULBRICH‖

Abstract. In this work, we present a globalized stochastic semismooth Newton method for solving stochastic optimization problems involving smooth nonconvex and nonsmooth convex terms in the objective function. We assume that only noisy gradient and Hessian information of the smooth part of the objective function is available via calling stochastic first and second order oracles. The proposed method can be seen as a hybrid approach combining stochastic semismooth Newton steps and stochastic proximal gradient steps. Two inexact growth conditions are incorporated to monitor the convergence and the acceptance of the semismooth Newton steps and it is shown that the algorithm converges globally to stationary points in expectation and almost surely. We present numerical results and comparisons on $l1$-regularized logistic regression and nonconvex binary classification that demonstrate the efficiency of the algorithm.

Key words. nonsmooth stochastic optimization, stochastic approximation, semismooth Newton method, stochastic second order information, global convergence

AMS subject classifications. 49M15, 65C60, 65K05, 90C06

DOI. 10.1137/18M1181249

1. Introduction.

In this paper, we propose and analyze a stochastic semismooth Newton framework for solving general nonsmooth, nonconvex optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \ \psi(x) := f(x) + r(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a (twice) continuously differentiable but possibly nonconvex function and $r : \mathbb{R}^n \to (-\infty, +\infty]$ is a convex, lower semicontinuous, and proper mapping. Although the function f is smooth, we assume that a full evaluation of f and an exact computation of the gradient and Hessian values $\nabla f(x)$ and $\nabla^2 f(x)$ is either not completely possible or too expensive in practice. Instead, we suppose that only noisy gradient and Hessian information is available which can be accessed via

*Received by the editors April 16, 2018; accepted for publication (in revised form) September 9, 2019; published electronically November 19, 2019.
https://doi.org/10.1137/18M1181249

Funding: The first author was supported by the Beijing International Center for Mathematical Research (BICMR, Peking University), the Boya Postdoctoral Fellowship Program, and the Shenzhen Institute for Artificial Intelligence and Robotics for Society (AIRS). The second author's research was supported by NSFC grant 11871135. The fourth author's research was supported in part by NSFC grants 11831002 and 11421101 and by the Beijing Academy of Artificial Intelligence.

†Institute for Data and Decision Analytics (iDDA), Shenzhen Institute for Artificial Intelligence and Robotics for Society (AIRS), Chinese University of Hong Kong, Shenzhen, China (andremilzarek@cuhk.edu.cn).

‡School of Mathematical Sciences, Dalian University of Technology, Dalian, China (xtxiao@dlut.edu.cn).

§School of Mathematical Sciences, Peking University, Beijing, China (tsen9731@pku.edu.cn).

¶Beijing International Center for Mathematical Research, Center for Data Science, National Engineering Laboratory for Big Data Analysis and Applications, Peking University, Beijing, 100871, China (wenzw@pku.edu.cn).

‖Chair of Mathematical Optimization, Department of Mathematics, Technical University of Munich, 85748 Garching b. München, Germany (mulbrich@ma.tum.de).

calls to *stochastic first* ($\mathcal{SFO}$) and *second* ($\mathcal{SSO}$) *order oracles*. Composite problems of the type (1.1) arise frequently in statistics and in large-scale statistical learning (see, e.g., [34, 42, 6, 64, 14]) and in many other applications. In these examples and problems, the smooth mapping f is typically of the form

$$f(x) := \mathbb{E}[F(x,\xi)] = \int_\Omega F(x,\xi(\omega))\,\mathrm{d}\mathbb{P}(\omega) \quad \text{or} \quad f(x) := \frac{1}{N}\sum_{i=1}^N f_i(x), \tag{1.2}$$

where $\xi : \Omega \to W$ is a random variable defined on a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$, W is a measurable space, and $F : \mathbb{R}^n \times W \to \mathbb{R}$ and the component functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, N$, correspond to certain loss models. More specifically, in the latter case, when the nonsmooth term $r \equiv 0$ vanishes, then problem (1.1) reduces to the so-called and well-studied empirical risk minimization problem

$$\min_{x\in\mathbb{R}^n} \; f(x), \quad f(x) := \frac{1}{N}\sum_{i=1}^N f_i(x). \tag{1.3}$$

Since the distribution $\mathbb{P}$ in (1.2) might not be fully known and the number of components N in (1.3) can be extremely large, stochastic approximation techniques, such as the mentioned stochastic oracles, have become an increasingly important tool in the design of efficient and computationally tractable numerical algorithms for problems (1.1) and (1.3) [50, 28, 64, 29, 30, 73, 72]. Moreover, in various interesting problems such as deep learning, dictionary learning, training of neural networks, and classification tasks with nonconvex activation functions [43, 42, 6, 22, 37, 61], the loss function f is nonconvex, which represents another major challenge for stochastic optimization approaches. For further applications and additional connections to simulation-based optimization, we refer to [28, 30].

1.1. Contents and contributions. In this work, we develop a stochastic second order framework for the general problem (1.1). Our basic idea is to apply a semismooth Newton method [55, 54] to approximately solve the nonsmooth fixed point–type equation

$$F^\Lambda(x) := x - \operatorname{prox}_r^\Lambda(x - \Lambda^{-1}\nabla f(x)) = 0, \quad \Lambda \in \mathbb{S}^n_{++}, \tag{1.4}$$

which represents a reformulation of the associated first order optimality conditions of problem (1.1). Specifically, we will consider stochastic variants of the nonsmooth residual (1.4) and of the semismooth Newton method, in which the gradient and Hessian of f are substituted by stochastic oracles. Motivated by deterministic Newton-type approaches [45, 44, 78], our proposed method combines stochastic semismooth Newton steps, stochastic proximal gradient steps, and a globalization strategy that is based on controlling the acceptance of the Newton steps via growth conditions. In this way, the resulting stochastic algorithm can be guaranteed to converge globally in expectation and almost surely, i.e., for a generated stochastic process of iterates $(\mathsf{X}^k)_k$, we have

$$\mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] \to 0 \quad \text{and} \quad F^\Lambda(\mathsf{X}^k) \to 0 \ \text{ almost surely}, \quad k \to \infty.$$

Furthermore, we investigate a reduced globalization mechanism that is especially well suited for stochastic optimization problems and large-scale applications. We show that this simplified approach still converges globally in the above sense in certain situations. To the best of our knowledge, rigorous extensions of existing stochastic

second order methods to the nonsmooth, nonconvex setting considered in this work do not seem to be available so far. We now briefly summarize some of the main challenges and contributions.

- In this paper, we provide a general global convergence theory which covers different aspects of the stochastic semismooth Newton method and of the proposed globalization scheme. In contrast to many other works, convexity of the smooth function f or of the objective function ψ and specific uniform assumptions on the chosen stochastic Newton step are not required in our analysis.
- In order to ensure global convergence and based on an acceptance test, the algorithm is allowed to switch between Newton and proximal gradient steps. Hence, a priori, it is not clear whether the generated iterates correspond to measurable random variables or to a stochastic process. This structural mechanism is significantly different from other existing stochastic approaches and will be discussed in detail in section 3.
- Our algorithmic approach and theoretical results are applicable for general stochastic oracles. Consequently, a large variety of approximation schemes, such as basic subsampling strategies or more elaborate variance reduction techniques [36, 77, 57, 73], can be used within our framework. In particular, in our numerical experiments, we consider a variance reduced version of our method. Similar to [73], the numerical results indicate that the combination of second order information and variance reduction techniques is also very effective in the nonsmooth setting. We note that the proposed method (using different stochastic oracles) performs quite well in comparison with other state-of-the-art algorithms in general.

1.2. Related work. The pioneering idea of utilizing stochastic approximations and the development of the associated, classical stochastic gradient descent (SGD) method for problem (1.3) and other stochastic programs can be traced back to the seminal work of Robbins and Monro [58]. Since then a plethora of stochastic optimization methods, strategies, and extensions has been studied and proposed for different problem formulations and under different basic assumptions. In the following, we give a brief overview of related research directions and related work.

First order methods. Advances in the research of stochastic first order methods for the smooth empirical risk problem (1.3) are numerous and we will only name a few recent directions here. Lately, based on the popularity and flexible applicability of the basic SGD method, a strong focus has been on the development and analysis of more sophisticated stochastic first order oracles to reduce the variance induced by gradient sampling and to improve the overall performance of the underlying SGD method. Examples of algorithms that exploit such variance reduction techniques include SVRG [36], SAG [62], and SAGA [21]. In [26], Friedlander and Schmidt analyze the convergence of a mini-batch stochastic gradient method for strongly convex f, in which the sampling rates are increased progressively. Incorporating different acceleration strategies, the first order algorithms Catalyst [41] and Katyusha [3] further improve the iteration complexity of the (proximal) SGD method. Several of the mentioned algorithms can also be extended to the nonsmooth setting $r \not\equiv 0$ by using the proximity operator of r and associated stochastic proximal gradient steps; see, e.g., prox-SVRG [77] and prox-SAGA [21]. AdaGrad [23] is another extension of the classical SGD method that utilizes special adaptive step size strategies.

The methods discussed so far either require convexity of f or of each of the component functions f_i or even stronger assumptions. Ghadimi and Lan [28, 29] generalize

the basic and accelerated SGD method to solve nonconvex and smooth minimization problems. Allen-Zhu and Hazan [5] and Reddi et al. [56] analyze stochastic variance reduction techniques for the nonconvex version of problem (1.3). Moreover, Reddi et al. [57] and Allen-Zhu [4] further extend existing stochastic first order methods to find approximate stationary points of the general nonconvex, nonsmooth model (1.1). In [30], Ghadimi, Lan, and Zhang discuss complexity and convergence results for a mini-batch stochastic projected gradient algorithm for problem (1.1). Xu and Yin [82] present and analyze a block stochastic gradient method for convex, nonconvex, and nonsmooth variants of the problem (1.1).

Quasi-Newton and second order methods. Recently, in order to accelerate and robustify the convergence of first order algorithms, stochastic second order methods have gained much attention. So far, the majority of stochastic second order methods are designed for the smooth problem (1.3) and are based on variants of the subsampled Newton method in which approximations of the gradient and Hessian of f are generated by selecting only a subsample or mini-batch of the components ∇f_i and $\nabla^2 f_i$, $i = 1, \dots, N$. In [15, 16], assuming positive definiteness of the subsampled Hessians, the authors analyze the convergence of a subsampled Newton-CG method and discuss strategies for selecting the sample sets. Erdogdu and Montanari [24] derive convergence rates of a projected subsampled Newton method with rank thresholding. In [60], Roosta-Khorasani and Mahoney establish nonasymptotic, probabilistic global and local convergence rates for subsampled Newton methods by applying matrix concentration inequalities. Xu et al. [81] present convergence and complexity results for a subsampled Newton-type approach with nonuniform sampling. Bollapragada, Byrd, and Nocedal [12] consider a subsampled Newton method for problems with the more general loss function given in (1.2) and derive r-superlinear convergence rates in expectation. In [72], Wang and Zhang propose an algorithm that combines the advantages of variance reduction techniques and subsampled Newton methods. Convergence properties are studied under the assumption that f is strongly convex and the Hessians $\nabla^2 f_i$ are Lipschitz continuous (with a uniform constant). Based on the existence of a suitable square-root decomposition of the Hessian, Pilanci and Wainwright [53] propose a Newton sketch method for general convex, smooth programs. In [9], the numerical performance of the Newton sketch method and different subsampled Newton approaches is compared. Furthermore, based on unbiased estimators of the inverse Hessian, a stochastic method called LiSSA is studied in [1]. A recent discussion of different stochastic second order algorithms can also be found in [84].

Stochastic quasi-Newton methods represent another large and important class of stochastic numerical algorithms for problem (1.3). Typically, these methods combine specific subsampling schemes for ∇f with randomized BFGS or BFGS-type updates to approximate the Hessian $\nabla^2 f$. In [63], a stochastic quasi-Newton algorithm for quadratic loss functions is proposed. Bordes, Bottou, and Gallinari [13] present a quasi-Newton approach that is based on diagonal curvature estimation. Mokhtari and Ribeiro [47] investigate a regularized stochastic BFGS method for solving strongly convex problems. In [17], Byrd et al. consider a stochastic limited-memory BFGS (L-BFGS) algorithm that incorporates exact Hessian information of the functions f_i to build the BFGS-type updates. The stochastic L-BFGS method discussed in [49] uses variance reduction techniques to improve its performance. Moreover, Gower, Goldfarb, and Richtarik [32] establish linear convergence of a stochastic block L-BFGS method if the functions f_i are strongly convex.

In contrast, the number of stochastic second order algorithms for smooth but nonconvex problems seems to be still quite limited. Based on a damping strategy for

BFGS-type updates and using general stochastic first order oracles, Wang et al. [73] propose a stochastic L-BFGS method for smooth problems. Under the assumption that the full gradient of the objective function is available, Xu, Roosta-Khorasani, and Mahoney [79] derive worst-case optimal iteration complexity results for an adaptive cubic regularization method with inexact or subsampled Hessian information. Generalizations and further aspects of this approach have been considered recently in [80, 83].

Finally, in [72], Wang and Zhang mention an extension of their hybrid method to the nonsmooth setting. A similar and related idea has also been presented in [65]. In particular, these approaches can be interpreted as stochastic variants of the proximal Newton method [39] for the general problem (1.1). Nevertheless, strong and uniform convexity assumptions are still required to guarantee convergence and well-definedness of the inner steps and subproblems.

1.3. Organization. This paper is organized as follows. Our specific stochastic setup, a derivation of (1.4), and the main algorithm are stated in section 2. The global convergence results are presented in section 3. Finally, in section 4, we report and discuss our numerical comparisons and experiments in detail.

1.4. Notation. For any $n \in \mathbb{N}$, we set $[n] := \{1, \dots, n\}$. By $\langle \cdot, \cdot \rangle$ and $\|\cdot\| := \|\cdot\|_2$ we denote the standard Euclidean inner product and norm. The set of symmetric and positive definite $n \times n$ matrices is denoted by $\mathbb{S}^n_{++}$. For a given matrix $\Lambda \in \mathbb{S}^n_{++}$, we define the inner product $\langle x, y \rangle_\Lambda := \langle x, \Lambda y \rangle = \langle \Lambda x, y \rangle$ and $\|x\|_\Lambda := \sqrt{\langle x, x \rangle_\Lambda}$. For a given set $S \subset \mathbb{R}^n$, the set cl S denotes the closure of S and $\mathbb{1}_S : \mathbb{R}^n \to \{0, 1\}$ is the associated characteristic function of S. For $p \in (0, \infty)$ the space ℓ^p_+ consists of all sequences $(x_n)_{n \geq 0}$ satisfying $x_n \geq 0$, $n \geq 0$, and $\sum x_n^p < \infty$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a given probability space. The space $L^p(\Omega) := L^p(\Omega, \mathbb{P})$, $p \in [1, \infty]$, denotes the standard L^p space on Ω. We write $\mathsf{X} \in \mathcal{F}$ for "X is $\mathcal{F}$-measurable." Moreover, we use $\sigma(\mathsf{X}^1, \dots, \mathsf{X}^k)$ to denote the σ-algebra generated by the family of random variables $\mathsf{X}^1, \dots, \mathsf{X}^k$. For a random variable $\mathsf{X} \in L^1(\Omega)$ and a sub-σ-algebra $\mathcal{H} \subseteq \mathcal{F}$, the conditional expectation of X given $\mathcal{H}$ is denoted by $\mathbb{E}[\mathsf{X} \mid \mathcal{H}]$. We use the abbreviation "a.s." for "almost surely."

2. A stochastic semismooth Newton method.

2.1. Probabilistic setting and preliminaries. In this section, we introduce several basic definitions and preparatory results. We start with an overview of the stochastic setting and the sampling strategy.

2.1.1. Stochastic setup. Although the function f is smooth, we assume that an exact or full evaluation of the gradient ∇f and Hessian $\nabla^2 f$ is not possible or is simply too expensive. Hence, we will work with $\mathcal{SFO}$ and $\mathcal{SSO}$,

$$\mathcal{G} : \mathbb{R}^n \times \Xi \to \mathbb{R}^n, \quad \mathcal{H} : \mathbb{R}^n \times \Xi \to \mathbb{S}^n,$$

to approximate gradient and Hessian information. Specifically, given an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a measurable space $(\Xi, \mathcal{X})$, we generate two mini-batches of random samples

$$s^k := \left\{ s^k_1, \dots, s^k_{\mathbf{n}^{\mathsf{g}}_k} \right\} \quad \text{and} \quad t^k := \left\{ t^k_1, \dots, t^k_{\mathbf{n}^{\mathsf{h}}_k} \right\}$$

and calculate the stochastic approximations $\mathcal{G}(x, s^k_i) \approx \nabla f(x)$ and $\mathcal{H}(x, t^k_j) \approx \nabla^2 f(x)$ in each iteration. Here, $\mathbf{n}^{\mathsf{g}}_k$ and $\mathbf{n}^{\mathsf{h}}_k$ denote the chosen sample sizes of the mini-batches

s^k and t^k. Moreover, each of the samples s_i^k, t_j^k, $i \in [\mathbf{n}_k^{\mathsf{g}}]$, $j \in [\mathbf{n}_k^{\mathsf{h}}]$, corresponds to a specific realization of a $(\mathcal{F}, \mathcal{X})$-measurable random mapping $\mathsf{S}_i^k, \mathsf{T}_j^k : \Omega \to \Xi$ and the respective collections of these random mappings are accordingly denoted by S^k and T^k. Throughout this work, we will use uppercase letters and a sans serif letterform to describe random variables or random objects, while lowercase letters or letters with serifs are typically reserved for realizations of a random variable and deterministic parameters. We suppose that the space Ω is sufficiently rich allowing us to model the mini-batches S^k, T^k and other associated stochastic processes in a unified way. Similar to [20, 82, 29, 30, 73], we then construct a mini-batch-type stochastic gradient $G_{s^k}(x)$ and Hessian $H_{t^k}(x)$ as follows:

$$G_{s^k}(x) := \frac{1}{\mathbf{n}_k^{\mathsf{g}}} \sum_{i=1}^{\mathbf{n}_k^{\mathsf{g}}} \mathcal{G}(x, s_i^k), \qquad H_{t^k}(x) := \frac{1}{\mathbf{n}_k^{\mathsf{h}}} \sum_{i=1}^{\mathbf{n}_k^{\mathsf{h}}} \mathcal{H}(x, t_i^k). \tag{2.1}$$

In the following, we use the terms G_k and H_k to denote the stochastic versions of G_{s^k} and H_{t^k}. Further, we assume that the stochastic oracles $\mathcal{G}$ and $\mathcal{H}$ are *Carathéodory functions*.[1] Additional assumptions on the stochastic setting are introduced later in subsection 3.1.

We will also sometimes drop the index k from the mini-batches S^k, T^k and sample sizes $\mathbf{n}_k^{\mathsf{g}}$, $\mathbf{n}_k^{\mathsf{h}}$ when we consider a general pair of batches S and T and their realizations s and t.

2.1.2. Definitions and first order optimality. In this subsection, we derive first order optimality conditions for the composite problem (1.1). Suppose that $x^* \in \operatorname{dom} r$ is a local solution of problem (1.1). Then, x^* satisfies the mixed-type variational inequality

$$\langle \nabla f(x^*), x - x^* \rangle + r(x) - r(x^*) \geq 0 \quad \forall\, x \in \mathbb{R}^n. \tag{2.2}$$

By definition, the latter condition is equivalent to $-\nabla f(x^*) \in \partial r(x^*)$, where ∂r denotes the convex subdifferential of r. We now introduce the well-known *proximal mapping* $\operatorname{prox}_r^\Lambda : \mathbb{R}^n \to \mathbb{R}^n$ of r. For an arbitrary parameter matrix $\Lambda \in \mathbb{S}_{++}^n$, the proximity operator $\operatorname{prox}_r^\Lambda(x)$ of r at x is defined as

$$\operatorname{prox}_r^\Lambda(x) := \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \; r(y) + \frac{1}{2}\|x - y\|_\Lambda^2. \tag{2.3}$$

The proximity operator is a Λ-firmly nonexpansive mapping, i.e., it satisfies

$$\|\operatorname{prox}_r^\Lambda(x) - \operatorname{prox}_r^\Lambda(y)\|_\Lambda^2 \leq \langle \operatorname{prox}_r^\Lambda(x) - \operatorname{prox}_r^\Lambda(y), x - y \rangle_\Lambda \quad \forall\, x, y \in \mathbb{R}^n.$$

Consequently, $\operatorname{prox}_r^\Lambda$ is Lipschitz continuous with modulus 1 with respect to the norm $\|\cdot\|_\Lambda$. We refer to [48, 19, 6, 7] for more details and (computational) properties. Let us further note that the proximity operator can also be uniquely characterized by the optimality conditions of the underlying optimization problem (2.3), i.e.,

$$\operatorname{prox}_r^\Lambda(x) \in x - \Lambda^{-1} \cdot \partial r(\operatorname{prox}_r^\Lambda(x)). \tag{2.4}$$

Using this characterization, condition (2.2) can be equivalently rewritten as follows:

$$F^\Lambda(x^*) = 0, \quad \text{where} \quad F^\Lambda(x) := x - \operatorname{prox}_r^\Lambda(x - \Lambda^{-1}\nabla f(x)). \tag{2.5}$$

[1] A mapping $F : \mathbb{R}^n \times \Xi \to \mathbb{R}$ is called a *Carathéodory function* if $F(\cdot, z) : \mathbb{R}^n \to \mathbb{R}$ is continuous for all $z \in \Xi$ and if $F(x, \cdot) : \Xi \to \mathbb{R}$ is measurable for all $x \in \mathbb{R}^n$.

We call $x \in \mathbb{R}^n$ a *stationary point* of problem (1.1) if it is a solution of the nonsmooth equation (2.5). If the problem is convex, e.g., if f is a convex function, then every stationary point is automatically a local and global solution of (1.1). The fixed point–type equation (2.5) forms the basis of the proximal gradient method [27, 19, 51], which has been studied intensively during the last decades.

For an arbitrary sample s, the corresponding stochastic residual is given by

$$F_s^\Lambda(x) := x - \mathrm{prox}_r^\Lambda(x - \Lambda^{-1}G_s(x)).$$

We will also use $u_s^\Lambda(x) := x - \Lambda^{-1}G_s(x)$ and $p_s^\Lambda(x) := \mathrm{prox}_r^\Lambda(u_s^\Lambda(x))$ to denote the stochastic (proximal) gradient steps.

2.2. Algorithmic framework. Next, we describe our algorithmic approach in detail. The overall idea is to use a stochastic semismooth Newton method to calculate an approximate solution of the optimality system

$$F^\Lambda(x) = 0.$$

The corresponding stochastic semismooth Newton step d^k at iteration k is then given by the linear system of equations

$$M_k d^k = -F_{s^k}^{\Lambda_k}(x^k), \quad M_k \in \mathcal{M}_{s^k,t^k}^{\Lambda_k}(x^k). \tag{2.6}$$

Here, we consider the following set of generalized derivatives:

$$\mathcal{M}_{s,t}^\Lambda(x) := \{M \in \mathbb{R}^{n\times n} : M = (I - D) + D\Lambda^{-1}H_t(x),\ D \in \partial\mathrm{prox}_r^\Lambda(u_s^\Lambda(x))\}, \tag{2.7}$$

where $\partial\mathrm{prox}_r^\Lambda(u_s^\Lambda(x))$ denotes the Clarke subdifferential of prox_r^Λ at the point $u_s^\Lambda(x)$. The set $\mathcal{M}_{s,t}^\Lambda(x)$ depends on the stochastic gradient and on the stochastic Hessian defined in (2.1). Moreover, the samples s^k, t^k and the matrix Λ_k used in (2.6) may change in each iteration; see also Remark 3.6. Let us note that, in practice, the system (2.6) can be solved inexactly via iterative approaches such as the CG or other Krylov subspace methods. We refer to Remark 3.16 for further information on the direction d^k.

In the deterministic setting, the set $\mathcal{M}_{s,t}^\Lambda(x)$ reduces to $\mathcal{M}^\Lambda(x) := \{M = (I - D) + D\Lambda^{-1}\nabla^2 f(x), D \in \partial\mathrm{prox}_r^\Lambda(u^\Lambda(x))\}$ with $u^\Lambda(x) = x - \Lambda^{-1}\nabla f(x)$. In general, $\mathcal{M}^\Lambda(x)$ does not coincide with Clarke's subdifferential $\partial F^\Lambda(x)$. As shown in [18], we can only guarantee $\partial F^\Lambda(x)h \subseteq \mathrm{co}(\mathcal{M}^\Lambda(x)h)$ for $h \in \mathbb{R}^n$. However, the set-valued mapping $\mathcal{M}^\Lambda : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n\times n}$ defines a so-called (strong) linear Newton approximation at x if the proximity operator prox_r^Λ is (strongly) semismooth at $u^\Lambda(x)$. In particular, $\mathcal{M}^\Lambda$ is upper semicontinuous and compact-valued. More details can be found in [25, Chapter 7] and [52]. We also note that the chain rule for semismooth functions implies that $F^\Lambda(x)$ is semismooth at x with respect to $\mathcal{M}^\Lambda(x)$ if prox_r^Λ is semismooth at $u^\Lambda(x)$. Furthermore, in various important examples including, e.g., ℓ_1- or nuclear norm-regularized optimization, group sparse problems, or semidefinite programming, the associated proximal mapping prox_r^Λ can be shown to be (strongly) semismooth and there exist explicit and computationally tractable representations of the generalized derivatives $D \in \partial\mathrm{prox}_r^\Lambda(x)$; see [67, 35] and [52, 44, 68] for a detailed discussion. In general, further structural information about the proximity operator prox_r^Λ is needed to explicitly construct $\partial\mathrm{prox}_r^\Lambda(x)$ and choose an appropriate set of generalized derivatives $\mathcal{M}_{s,t}^\Lambda(x)$. In the following, we assume that a suitable realization of $\mathcal{M}_{s,t}^\Lambda(x)$ is always available.

Algorithm 1: A stochastic semismooth Newton method.

Initialization: Choose an initial point $x^0 \in \operatorname{dom} r$, $\rho_0 \in \mathbb{R}_+$, and mini-batches s^0, t^0. Select sample sizes $(\mathbf{n}_k^{\mathsf{g}})_k$, $(\mathbf{n}_k^{\mathsf{h}})_k$, parameter matrices $(\Lambda_k)_k \subset \mathbb{S}^n_{++}$, and step sizes $(\alpha_k)_k$. Choose $\eta, p \in (0,1)$, $\beta > 0$, and $(\nu_k)_k$, $(\varepsilon_k^1)_k$, $(\varepsilon_k^2)_k$. Set iteration $k := 0$.

while *did not converge* **do**

- Compute $F_{s^k}^{\Lambda_k}(x^k)$ and choose $M_k \in \mathcal{M}_{s^k,t^k}^{\Lambda_k}(x^k)$ according to (2.7). For all $i = 1, \dots, \mathbf{n}_{k+1}^{\mathsf{g}}$ and $j = 1, \dots, \mathbf{n}_{k+1}^{\mathsf{h}}$ select new samples s_i^{k+1}, t_j^{k+1}.
- Compute the Newton step d^k by (approximately) solving
$$M_k d^k = -F_{s^k}^{\Lambda_k}(x^k).$$
- Set $z_{\mathsf{n}}^k = x^k + d^k$. If the conditions $z_{\mathsf{n}}^k \in \operatorname{dom} r$, (2.8), and (2.9) are satisfied, skip step 6 and set $x^{k+1} = z_{\mathsf{n}}^k$, $\rho_{k+1} = \|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|$. Otherwise go to step 6.
- Set $v^k = -F_{s^k}^{\Lambda_k}(x^k)$, $x^{k+1} = x^k + \alpha_k v^k$, and $\rho_{k+1} = \rho_k$.
- Set $k \leftarrow k + 1$.

In order to control the acceptance of the Newton steps and to achieve global convergence of our algorithm, we introduce the following growth conditions for the trial step $z_{\mathsf{n}}^k = x^k + d^k$:

$$\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\| \le (\eta + \nu_k) \cdot \rho_k + \varepsilon_k^1, \tag{2.8}$$

$$\psi(z_{\mathsf{n}}^k) \le \psi(x^k) + \beta \cdot \rho_k^{1-p} \|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^p + \varepsilon_k^2. \tag{2.9}$$

If the trial point z_{n}^k satisfies both conditions and is feasible, i.e., if $z_{\mathsf{n}}^k \in \operatorname{dom} r$, we accept it and compute the new iterate x^{k+1} via $x^{k+1} = z_{\mathsf{n}}^k$. The parameter sequences $(\nu_k)_k$, $(\varepsilon_k^1)_k$, and $(\varepsilon_k^2)_k$ are supposed to be nonnegative and summable and can be chosen during the initialization or during the iteration process. Furthermore, the parameter ρ_k keeps track of the norm of the residual $F_{s^i}^{\Lambda_i}(x^i)$ of the last *accepted* Newton iterate x^i, $i < k$, and is updated after a successful Newton step. The parameters $\beta > 0$, $\eta, p \in (0,1)$ are given constants. If the trial point z_{n}^k does not satisfy the conditions (2.8) and (2.9), we reject it and perform an alternative proximal gradient step using the stochastic residual $F_{s^k}^{\Lambda_k}(x^k)$ as an approximate descent direction. We also introduce a step size α_k to damp the proximal gradient step and to guarantee sufficient decrease in the objective function ψ. A precise bound for the step sizes α_k is derived in Lemma 3.8. The details of the method are summarized in Algorithm 1.

Our method can be seen as a hybrid of the semismooth Newton method and the standard proximal gradient method generalizing the deterministic Newton approaches presented in [45, 44] to the stochastic setting. Our globalization technique is inspired by [45], where a filter globalization strategy was proposed to control the acceptance of the Newton steps. Similar to [45, 44], we add condition (2.8) to monitor the behavior and convergence of the Newton steps. The second condition (2.9) (together with the feasibility condition $z_{\mathsf{n}}^k \in \operatorname{dom} r$) is required to bound the possible ψ-ascent of intermediate Newton steps. In contrast to smooth optimization problems, descent-based damping techniques or step size selections, as used in, e.g., [15, 12, 17, 60, 73], can not always guarantee sufficient ψ-descent of the semismooth Newton steps due to the nonsmooth nature of problem (1.1). This complicates the analysis and globalization

of semismooth Newton methods in general. In practice, the second growth condition (2.9) can be restrictive since an evaluation of the full objective function is required. However, similar descent conditions also appeared in other globalization strategies for smooth problems [60, 79, 80, 83]. In the next section, we verify that Algorithm 1 using the proposed growth conditions (2.8)–(2.9) converges globally in expectation. Moreover, in Theorems 3.12 and 3.13, we discuss global convergence of Algorithm 1 without condition (2.9) in different settings. Let us also note that the growth conditions (2.8)–(2.9) are checked using a new sample mini-batch s^{k+1}. Thus, only one gradient evaluation is required per iteration if the Newton step is accepted. Finally, we notice that the feasibility condition $z_{\mathsf{n}}^k \in \operatorname{dom} r$ can be circumvented by setting $x^{k+1} = \mathcal{P}_{\operatorname{dom} r}(z_{\mathsf{n}}^k)$, where $\mathcal{P}_{\operatorname{dom} r}$ is the projection onto $\operatorname{dom} r$.

Let us mention that an alternative globalization is analyzed in [52, 66], where the authors propose the so-called forward-backward envelope as a smooth merit function for problem (1.1). Since this framework requires an additional proximal gradient step (and thus, an additional gradient evaluation) after each iteration, we do not consider this approach here.

3. Global convergence. In this section, we analyze the global convergence behavior of Algorithm 1. We first present and summarize our main assumptions.

3.1. Assumptions. Throughout this paper, we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on $\mathbb{R}^n$ and $r : \mathbb{R}^n \to (-\infty, +\infty]$ is convex, lower semicontinuous, and proper. As already mentioned, we also assume that the oracles $\mathcal{G}, \mathcal{H} : \mathbb{R}^n \times \Xi \to \mathbb{R}$ are Carathéodory functions. In the following, we further specify the assumptions on the functions f and r.

Assumption 3.1. Let $f : \mathbb{R}^n \to \mathbb{R}$ be given. We assume

(A.1) the gradient mapping ∇f is Lipschitz continuous on $\mathbb{R}^n$ with modulus $L > 0$,
(A.2) the objective function ψ is bounded from below on $\operatorname{dom} r$,
(A.3) there exist parameters $\mu_f \in \mathbb{R}$ and $\mu_r \geq 0$ with $\bar{\mu} := \mu_f + \mu_r > 0$ such that the shifted functions $f - \frac{\mu_f}{2}\|\cdot\|^2$ and $r - \frac{\mu_r}{2}\|\cdot\|^2$ are convex,
(A.4) there exists a constant $\bar{g}_r > 0$ such that for all $x \in \operatorname{dom} r$ there exists $\lambda \in \partial r(x)$ with $\|\lambda\| \leq \bar{g}_r$.

Assumption (A.3) implies that the function ψ is strongly convex with convexity parameter $\bar{\mu}$. Furthermore, if both assumptions (A.1) and (A.3) are satisfied, then the parameter μ_f is bounded by the Lipschitz constant L, i.e., we have $|\mu_f| \leq L$. The assumptions (A.3)–(A.4) are only required for a variant of Algorithm 1 that uses a modified globalization strategy; see Theorem 3.13. A concrete example for r that satisfies (A.4) is given in Remark 3.14. We continue with the assumptions on the parameters used within our algorithmic framework.

Assumption 3.2. Let $(\Lambda_k)_k \subset \mathbb{S}^n_{++}$ be a family of symmetric, positive definite parameter matrices and let $(\nu_k)_k$, $(\varepsilon_k^1)_k$, and $(\varepsilon_k^2)_k$ be given sequences. Then, for some given parameter $p \in (0, 1)$ we assume

(B.1) there exist $0 < \lambda_m \leq \lambda_M < \infty$ such that $\lambda_M I \succeq \Lambda_k \succeq \lambda_m I$ for all $k \in \mathbb{N}$,
(B.2) it holds, $(\nu_k)_k$, $(\varepsilon_k^2)_k \in \ell_+^1$, and $(\varepsilon_k^1)_k \in \ell_+^p$.

In the following sections, we study the convergence properties of the stochastic process $(\mathsf{X}^k)_k$ generated by Algorithm 1 with respect to the filtrations

$$\mathcal{F}_k := \sigma(\mathsf{S}^0, \ldots, \mathsf{S}^k, \mathsf{T}^0, \ldots, \mathsf{T}^k) \quad \text{and} \quad \hat{\mathcal{F}}_k := \sigma(\mathsf{S}^0, \ldots, \mathsf{S}^k, \mathsf{S}^{k+1}, \mathsf{T}^0, \ldots, \mathsf{T}^k).$$

The filtration $\mathcal{F}_k$ represents the information that is collected up to iteration k and

that is used to compute the trial point z_{n}^k or a proximal gradient step

$$z_{\mathsf{p}}^k := x^k + \alpha_k v^k = x^k - \alpha_k F_{s^k}^{\Lambda_k}(x^k).$$

The filtration $\hat{\mathcal{F}}_k$ has a similar interpretation, but it also contains the information produced by deciding whether the Newton step z_{n}^k should be accepted or rejected, i.e., it holds that $\hat{\mathcal{F}}_k = \sigma(\mathcal{F}_k \cup \sigma(\mathsf{S}^{k+1}))$. The filtrations $\{\mathcal{F}_k, \hat{\mathcal{F}}_k\}$ naturally describe the aggregation of information generated by Algorithm 1. Following our convention introduced in subsection 2.1.1, we will use X^k, $\mathsf{Z}_{\mathsf{n}}^k$, and $\mathsf{Z}_{\mathsf{p}}^k$ to denote the random variables associated with the iterates and realizations x^k, z_{n}^k, and z_{p}^k. Next, we introduce our main stochastic assumptions.

Assumption 3.3. We assume

(C.1) for all $k \in \mathbb{N}_0$, the generalized derivative M_k, chosen in step 3 of Algorithm 1, is the realization of an $\mathcal{F}_k$-measurable mapping $\mathsf{M}_k : \Omega \to \mathbb{R}^{n\times n}$, i.e., there exists an $\mathcal{F}_k$-measurable selection M_k of the multifunction $\mathcal{M}_k : \Omega \rightrightarrows \mathbb{R}^{n\times n}$, $\mathcal{M}_k(\omega) := \mathcal{M}^{\Lambda_k}_{\mathsf{S}^k(\omega),\mathsf{T}^k(\omega)}(\mathsf{X}^k(\omega))$ and $\omega \in \Omega$ with $\mathsf{M}_k(\omega) = M_k$,

(C.2) the variance of the individual stochastic gradients is bounded, i.e., for all $k \in \mathbb{N}$ there exists $\sigma_k \geq 0$ such that $\mathbb{E}[\|\nabla f(\mathsf{X}^k) - \mathsf{G}_k(\mathsf{X}^k)\|^2] \leq \sigma_k^2$.

The second condition is common in stochastic programming; see, e.g., [28, 82, 12, 17, 30, 73]. Since the generalized derivative M_k is generated iteratively and depends on the random process $(\mathsf{X}^k)_k$ and on the mini-batches $(\mathsf{S}^k)_k$, $(\mathsf{T}^k)_k$, condition (C.1) is required to guarantee that the selected matrices M_k actually correspond to realizations of $\mathcal{F}_k$-measurable random operators. A similar assumption was also used in [73]. Furthermore, applying the techniques and theoretical results presented in [71] for infinite-dimensional nonsmooth operator equations, we can ensure that the multifunction $\mathcal{M}_k$ admits at least one measurable selection $\mathsf{M}_k : \Omega \to \mathbb{R}^{n\times n}$. We discuss this important observation together with a proof of Fact 3.4 in Appendix A.1. Let us note that it is also possible to generalize the assumptions and allow $\mathcal{F}_k$-measurable parameter matrices Λ_k. However, in order to simplify our analysis, we focus on a deterministic choice of $(\Lambda_k)_k$ and do not consider this extension here.

As a consequence of condition (C.1) and of the assumptions on f, we can infer that the random processes $(\mathsf{Z}_{\mathsf{n}}^k)_k$, $(\mathsf{Z}_{\mathsf{p}}^k)_k$, and $(\mathsf{X}^k)_k$ are adapted to the filtrations $\mathcal{F}_k$ and $\hat{\mathcal{F}}_k$.

FACT 3.4. *Under assumption* (C.1)*, it holds that* $\mathsf{Z}_{\mathsf{n}}^k, \mathsf{Z}_{\mathsf{p}}^k \in \mathcal{F}_k$ *and* $\mathsf{X}^{k+1} \in \hat{\mathcal{F}}_k$ *for all* $k \in \mathbb{N}_0$.

Since the choice of X^{k+1} depends on various criteria, the properties stated in Fact 3.4 are not immediately obvious. In particular, we need to verify that the decision of accepting or rejecting the step z_{n}^k is an $\hat{\mathcal{F}}_k$-measurable action. The proof of Fact 3.4 is given in Appendix A.1.

3.2. Properties of F^{Λ}. In this subsection, we discuss several useful properties of the nonsmooth function F^{Λ} and of its stochastic version F_s^{Λ}. The next statement shows that $\|F_s^{\Lambda}(x)\|$ does not grow too much when the parameter matrix Λ changes. This result was first established by Tseng and Yun in [70].

LEMMA 3.5. *Let* $\Lambda_1, \Lambda_2 \in \mathbb{S}^n_{++}$ *be two arbitrary matrices. Then, for all* $x \in \mathbb{R}^n$*, for all samples* s*, and for* $W := \Lambda_2^{-\frac{1}{2}}\Lambda_1\Lambda_2^{-\frac{1}{2}}$*, it follows that*

$$\|F_s^{\Lambda_1}(x)\| \leq \frac{1+\lambda_{\max}(W)+\sqrt{1-2\lambda_{\min}(W)+\lambda_{\max}(W)^2}}{2}\frac{\lambda_{\max}(\Lambda_2)}{\lambda_{\min}(\Lambda_1)}\|F_s^{\Lambda_2}(x)\|.$$

Proof. The proof is identical to the proof of [70, Lemma 3] and will be omitted here. ∎

Remark 3.6. Let $\Lambda \in \mathbb{S}^n_{++}$ be given and let $(\Lambda_k)_k \subset \mathbb{S}^n_{++}$ be a family of symmetric, positive definite matrices satisfying assumption (B.1). Then, it easily follows that

$$\frac{\lambda_{\max}(\Lambda)}{\lambda_m} I \succeq \Lambda_k^{-\frac{1}{2}}\Lambda\Lambda_k^{-\frac{1}{2}} \succeq \frac{\lambda_{\min}(\Lambda)}{\lambda_M} I \quad \text{and} \quad \frac{\lambda_M}{\lambda_{\min}(\Lambda)} I \succeq \Lambda^{-\frac{1}{2}}\Lambda_k\Lambda^{-\frac{1}{2}} \succeq \frac{\lambda_m}{\lambda_{\max}(\Lambda)} I$$

for all $k \in \mathbb{N}$, and, due to Lemma 3.5, we obtain the bounds

$$\underline{\lambda} \cdot \|F_s^{\Lambda}(x)\| \leq \|F_s^{\Lambda_k}(x)\| \leq \overline{\lambda} \cdot \|F_s^{\Lambda}(x)\| \quad \forall\, k \in \mathbb{N}, \tag{3.1}$$

and for all mini-batches s, $x \in \mathbb{R}^n$. The constants $\underline{\lambda}$, $\overline{\lambda} > 0$ do not depend on k, Λ_k, or s. Thus, the latter inequalities imply

$$F^{\Lambda}(x^k) \to 0 \quad \iff \quad F^{\Lambda_k}(x^k) \to 0, \quad k \to \infty.$$

As indicated in the last section, this can be used in the design of our algorithm. In particular, adaptive schemes or other techniques can be applied to update Λ.

The following result is a simple extension of [70, Theorem 4]; see also [77, Lemma 3.7] and [82] for comparison.

LEMMA 3.7. *Suppose that the assumptions* (A.1), (A.3) *are satisfied and let* $\Lambda \in \mathbb{S}^n_{++}$ *be given with* $\lambda_M I \succeq \Lambda \succeq \lambda_m I$. *Furthermore, let* x^* *denote the unique solution of the problem* $\min_x \psi(x)$ *and for any* $\tau > 0$ *let us set* $b_1 := L - 2\lambda_m - \mu_r$, $b_2 := (\lambda_M + \mu_r)/\bar{\mu}$, *and* $B_1(\tau) := (1+\tau)(\sqrt{b_1 + b_2 + \tau} + \sqrt{b_2})^2/\bar{\mu}$. *Then, there exists some positive constant* $B_2(\tau)$ *that only depends on* τ *such that*

$$\|x - x^*\|^2 \leq B_1(\tau) \cdot \|F_s^{\Lambda}(x)\|^2 + B_2(\tau) \cdot \|\nabla f(x) - G_s(x)\|^2 \tag{3.2}$$

for all $x \in \mathbb{R}^n$ *and for every sample* s. *If the full gradient is used, the term* $\|\nabla f(x) - G_s(x)\|$ *vanishes for all* x *and* (3.2) *holds with* $B_1(\tau) \equiv B_1(0)$.

Proof. The proof of Lemma 3.7 is presented in Appendix A.2. ∎

3.3. Convergence analysis. In the following, we first verify that a stochastic proximal gradient step yields approximate ψ-descent whenever the step size α_k in step 6 of Algorithm 1 is chosen sufficiently small. We also give a bound for the step sizes α_k. Let us note that similar results were shown in [82, 30, 29] and that the proof of Lemma 3.8 mainly relies on the well-known descent lemma

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \quad \forall\, x, y \in \mathbb{R}^n, \tag{3.3}$$

which is a direct consequence of assumption (A.1).

LEMMA 3.8. *Let* $x \in \operatorname{dom} r$ *and* $\Lambda \in \mathbb{S}^n_{++}$ *be arbitrary and suppose that conditions* (A.1) *and* (B.1) *(for* Λ*) are satisfied. Moreover, let* $\gamma \in (0,1)$, $\rho \in (1, \gamma^{-1})$, *and the mini-batch* s *be given and set* $\overline{\alpha} := \min\{1, 2(1-\gamma\rho)\lambda_m L^{-1}\}$. *Then, for all* $\alpha \in [0, \overline{\alpha}]$ *it holds that*

$$\psi(x + \alpha v) - \psi(x) \leq -\alpha\gamma\|v\|^2_{\Lambda} + \frac{\alpha}{4\gamma(\rho-1)\lambda_m}\|\nabla f(x) - G_s(x)\|^2, \tag{3.4}$$

where $v := -F_s^{\Lambda}(x)$.

Proof. We first define $\Delta := \langle G_s(x), v\rangle + r(x+v) - r(x)$. Then, applying the optimality condition of the proximity operator (2.4), it follows that

$$\Delta \leq \langle G_s(x), v\rangle + \langle -\Lambda v - G_s(x), v\rangle = -\|v\|_\Lambda^2.$$

Using the descent lemma (3.3), the convexity of r, and Young's inequality, we now obtain

$$\begin{aligned}
&\psi(x+\alpha v) - \psi(x) + \alpha\gamma\|v\|_\Lambda^2 \\
&\qquad \leq \alpha(\langle \nabla f(x), v\rangle + r(x+v) - r(x)) + 0.5L\alpha^2\|v\|^2 + \alpha\gamma\|v\|_\Lambda^2 \\
&\qquad \leq 0.5L\alpha^2\|v\|^2 - \alpha(1-\gamma)\|v\|_\Lambda^2 + \alpha\langle \nabla f(x) - G_s(x), v\rangle \\
&\qquad \leq \alpha\left[\frac{L\alpha}{2} - (1-\gamma\rho)\lambda_m\right]\|v\|^2 + \frac{\alpha}{4\gamma(\rho-1)\lambda_m}\|\nabla f(x) - G_s(x)\|^2.
\end{aligned}$$

Since the first term is nonpositive for all $\alpha \leq \overline{\alpha}$, this establishes (3.4). ∎

In the special case $\lambda_m \geq L$ and $\rho \leq (2\gamma)^{-1}$, Lemma 3.8 implies that the approximate descent condition (3.4) holds for all $\alpha \in [0,1]$. The next technical lemma is one of our key tools to analyze the stochastic behavior of the Newton iterates and to bound the associated residual terms $\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|$.

LEMMA 3.9. *Let $(y_k)_k$ be an arbitrary binary sequence in $\{0,1\}$ and let $a_0 \geq 0$, $\eta \in (0,1)$, $p \in (0,1]$, and $(\nu_k)_k \in \ell_+^1$, $(\varepsilon_k)_k \in \ell_+^p$ be given. Let the sequence $(a_k)_k$ be defined by*

$$a_{k+1} := (\eta + \nu_k)^{y_k} a_k + y_k\varepsilon_k \quad \forall\ k \in \mathbb{N}_0.$$

Then, for all $K \geq 1$, $k \geq 0$, and all $q \in [p,1]$, it holds that

$$a_{k+1} \leq C_\nu\left[a_0 + \sum_{k=0}^{\infty}\varepsilon_k\right] \quad \text{and} \quad \sum_{k=0}^{K-1} y_k a_{k+1}^q \leq \frac{C_\nu^q}{1-\eta^q}\left[(\eta a_0)^q + \sum_{k=0}^{\infty}\varepsilon_k^q\right],$$

where $C_\nu := \exp\left(\eta^{-1}\sum_{i=0}^{\infty}\nu_i\right)$.

Proof. Using an induction, we can derive an explicit representation for a_{k+1}

$$a_{k+1} = \left\{\prod_{i=0}^{k}(\eta+\nu_i)^{y_i}\right\} a_0 + \sum_{j=0}^{k-1}\left\{\prod_{i=j+1}^{k}(\eta+\nu_i)^{y_i}\right\} y_j\varepsilon_j + y_k\varepsilon_k \quad \forall\ k \geq 0. \tag{3.5}$$

Next, using $y_i \in \{0,1\}$, $i \in \mathbb{N}$, and $\log(1+\nu_i\eta^{-1}) \leq \nu_i\eta^{-1}$, we obtain the estimate

$$\prod_{i=\ell}^{k}(\eta+\nu_i)^{y_i} \leq \left\{\prod_{i=\ell}^{k}\eta^{y_i}\right\} \cdot \exp\left(\eta^{-1}\sum_{i=\ell}^{k} y_i\nu_i\right) \leq C_\nu \cdot \eta^{\sum_{i=\ell}^{k} y_i}, \quad \ell \geq 0. \tag{3.6}$$

The bound on a_{k+1} then follows from (3.5), (3.6), and $\eta \leq 1$. Let us now define the set $K_\ell := \{i \in \{\ell, \dots, K-1\} : y_i = 1\}$. Then, it holds that

$$\sum_{k=\ell}^{K-1} y_k\eta^{\sum_{i=\ell}^{k} q y_i} = \sum_{k\in K_\ell}\eta^{\sum_{i\in K_\ell, i\leq k} q} = \sum_{j=1}^{|K_\ell|}\eta^{jq} \leq \sum_{k=\ell}^{K-1}\eta^{(k-\ell+1)q}, \quad \ell \in \{0,\dots,K-1\}.$$

Combining the last results and using the subadditivity of $x \mapsto x^q$, $q \in [p,1]$, we have

$$\sum_{k=0}^{K-1} y_k a_{k+1}^q \leq C_\nu^q \sum_{k=0}^{K-1} y_k \eta^{\sum_{i=0}^k q y_i} a_0^q + C_\nu^q \sum_{k=0}^{K-1}\sum_{j=0}^{k-1} \eta^{\sum_{i=j+1}^k q y_i} y_k y_j \varepsilon_j^q + \sum_{k=0}^{K-1} y_k \varepsilon_k^q$$
$$\leq C_\nu^q \sum_{k=0}^{K-1} \eta^{q(k+1)} a_0^q + C_\nu^q \sum_{j=0}^{K-1} \left\{ \sum_{k=j}^{K-1} \eta^{\sum_{i=j}^k q y_i} y_k \right\} \eta^{-q y_j} y_j \varepsilon_j^q$$
$$\leq \frac{C_\nu^q}{1-\eta^q} \cdot (\eta a_0)^q + C_\nu^q \sum_{j=0}^{K-1} \left\{ \sum_{k=j}^{K-1} \eta^{q(k-j)} \right\} \varepsilon_j^q \leq \frac{C_\nu^q}{1-\eta^q} \left[(\eta a_0)^q + \sum_{j=0}^{\infty} \varepsilon_j^q \right]$$

as desired. Let us also note that the inclusion $\ell_+^q \subset \ell_+^p$ is used in the last step. ∎

We are now in position to establish global convergence of Algorithm 1 in the sense that the expectation $\mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2]$ converges to zero as $k \to \infty$. We first show convergence of Algorithm 1 under the conditions (C.1)–(C.2) and under the additional assumptions that the step sizes are diminishing and that the scaled stochastic error terms $\alpha_k \sigma_k^2$, $k \in \mathbb{N}$, are summable, which is a common requirement in the analysis of stochastic methods for nonsmooth, nonconvex optimization; see, e.g., [82, 30, 29].

Our basic idea is to show that both the proximal gradient and the semismooth Newton step yield approximate ψ-descent and that the error induced by gradient and Hessian sampling can be controlled in expectation. For a proximal gradient step this basically follows from Lemma 3.8. For a Newton step, we combine the growth conditions (2.8)–(2.9) and Lemma 3.9 to establish an estimate similar to (3.4). An analogous strategy was also used in [44, 45]. In our situation, however, a more careful discussion of the possible effects of the semismooth Newton steps is needed to cope with the stochastic situation. More specifically, since our convergence result is stated in expectation, all possible realizations of the random mini-batches S^k and T^k, $k \in \mathbb{N}_0$, and their influence on the conditions (2.8)–(2.9) have to be considered. In order to apply Lemma 3.9, we now set up some preparatory definitions.

Let $k \in \mathbb{N}_0$ be given and let $(\mathsf{R}_k)_k$ denote the stochastic process associated with the sequence of parameters $(\rho_k)_k$ used in the growth conditions (2.8)–(2.9). Furthermore, let us define the mapping $\mathsf{F}_k : \Omega \to \mathbb{R}^n$, $\mathsf{F}_k(\omega) := F_{\mathsf{S}^k(\omega)}^{\Lambda_k}(\mathsf{Z}_\mathsf{n}^{k-1}(\omega))$ and the set

$$P_k := \{\omega \in \Omega : \mathsf{Z}_\mathsf{n}^k(\omega) \in \operatorname{dom} r,\ \|\mathsf{F}_{k+1}(\omega)\| \leq (\eta + \nu_k)\mathsf{R}_k(\omega) + \varepsilon_k^1,$$
$$\psi(\mathsf{Z}_\mathsf{n}^k(\omega)) \leq \psi(\mathsf{X}^k(\omega)) + \beta \cdot \mathsf{R}_k(\omega)^{1-p}\|\mathsf{F}_{k+1}(\omega)\|^p + \varepsilon_k^2\}.$$

Then, setting $\mathsf{Y}_{k+1} : \Omega \to \{0,1\}$, $\mathsf{Y}_{k+1}(\omega) := \mathbb{1}_{P_k}(\omega)$, it holds that

$$\mathsf{Y}_{k+1}(\omega) = \begin{cases} 1 & \text{if } \mathsf{Z}_\mathsf{n}^k(\omega) \text{ is feasible and satisfies the conditions (2.8) and (2.9),} \\ 0 & \text{otherwise} \end{cases}$$

and consequently, each iterate x^{k+1} can be calculated as follows:

$$\begin{aligned} x^{k+1} &= (1 - y_{k+1}) z_\mathsf{p}^k + y_{k+1} z_\mathsf{n}^k \\ &= (1 - y_{k+1})[x^k - \alpha_k F_{s^k}^{\Lambda_k}(x^k)] + y_{k+1}[x^k - M_k^+ F_{s^k}^{\Lambda_k}(x^k)]. \end{aligned} \tag{3.7}$$

Here, y_{k+1} again represents a suitable realization of Y_{k+1} and the matrix M_k^+ denotes the *Moore–Penrose inverse* of the generalized derivative M_k. Let us note that this compact form of our iterative scheme turns out to be particularly useful in the proof

of Fact 3.4. We also introduce the functions $\mathsf{A}_k : \Omega \to \mathbb{R}_+$, $k \in \mathbb{N}_0$, which are defined recursively via

$$\mathsf{A}_0 := \rho_0 \in \mathbb{R}_+, \quad \mathsf{A}_{k+1} := (\eta + \nu_k)^{\mathsf{Y}_{k+1}} \mathsf{A}_k + \mathsf{Y}_{k+1}\varepsilon_k^1.$$

By construction of Algorithm 1 and by induction, we have $\mathsf{R}_k(\omega) \leq \mathsf{A}_k(\omega)$ and thus

$$\mathsf{Y}_{k+1}\|\mathsf{F}_{k+1}\| \leq \mathsf{Y}_{k+1}\mathsf{A}_{k+1} \quad \forall\, k \in \mathbb{N}_0 \quad (\forall\, \omega \in \Omega). \tag{3.8}$$

Since the estimates in Lemma 3.9 hold uniformly for all possible realizations of the random variables Y_k and A_k, we obtain the sample-independent bounds

$$\sum_{k=0}^{K-1} \mathsf{Y}_{k+1}\mathsf{A}_{k+1}^q \leq \frac{C_\nu^q}{1-\eta^q}\left[(\eta\rho_0)^q + \sum_{k=0}^{\infty}(\varepsilon_k^1)^q\right] =: \mathcal{C}_q < \infty \tag{3.9}$$

and $\mathsf{A}_{k+1} \leq C_\nu\left[\rho_0 + \sum_{k=0}^{\infty}\varepsilon_k^1\right] \leq \mathcal{C}_1/\eta$ for all $k \geq 0$, $K \in \mathbb{N}$, $q \in [p, 1]$, and $\omega \in \Omega$. We now state our main result of this section.

THEOREM 3.10. *Let the stochastic process $(\mathsf{X}^k)_k$ be generated by Algorithm 1. Suppose that assumptions (A.1)–(A.2), (B.1)–(B.2), and (C.1)–(C.2) are satisfied. Furthermore, suppose that the step sizes $\alpha_k \in [0, 1]$, $k \in \mathbb{N}$, are chosen such that the approximate descent condition (3.4) holds for some given γ and ρ. Then, we have the following:*

(i) *Under the additional assumptions*

$$(\alpha_k)_k \text{ is monotonically decreasing}, \quad \sum \alpha_k = \infty, \quad \sum \alpha_k\sigma_k^2 < \infty,$$

it follows $\liminf_{k\to\infty} \mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] = 0$ *and* $\liminf_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ *a.s. for any matrix* $\Lambda \in \mathbb{S}^n_{++}$.

(ii) *In the case* $\sum \sigma_k^2 < \infty$ *and if the step sizes* α_k *are bounded away from zero, that is, if there exists* $\underline{\alpha} > 0$ *with* $\alpha_k \geq \underline{\alpha}$ *for all* $k \in \mathbb{N}$, *it holds that* $\lim_{k\to\infty} \mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] = 0$ *and* $\lim_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ *a.s. for any* $\Lambda \in \mathbb{S}^n_{++}$.

Proof. We start with the proof of the first part. Assumption (A.1) implies that the gradient mapping $\nabla f(x)$ is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant L. Thus, for any matrix $\Gamma \in \mathbb{S}^n_{++}$ with $\lambda_M I \succeq \Gamma \succeq \lambda_m I$, we obtain the Lipschitz constant $1 + L\lambda_m^{-1}$ for $u^\Gamma(x)$. Since the proximity operator $\operatorname{prox}_r^\Gamma$ is Γ-nonexpansive, we now have

$$\begin{aligned}\|F^\Gamma(x) - F^\Gamma(y)\| &\leq \|x - y\| + \lambda_m^{-\frac{1}{2}}\|\operatorname{prox}_r^\Gamma(u^\Gamma(x)) - \operatorname{prox}_r^\Gamma(u^\Gamma(y))\|_\Gamma \\ &\leq \|x - y\| + (\lambda_m^{-1}\lambda_M)^{\frac{1}{2}}\|u^\Gamma(x) - u^\Gamma(y)\| \leq L_F\|x - y\|\end{aligned}$$

for all $x, y \in \mathbb{R}^n$ and $L_F := 1 + (\lambda_m^{-1}\lambda_M)^{\frac{1}{2}}(1 + L\lambda_m^{-1})$. Hence, by assumption (B.1), the functions $x \mapsto F^{\Lambda_k}(x)$, $k \in \mathbb{N}$, are all Lipschitz continuous on $\mathbb{R}^n$ with modulus L_F.

In the following, we consider an arbitrary but fixed realization of the involved stochastic processes, i.e., we choose $\omega \in \Omega$ and derive properties for the realizations $(\mathsf{X}^k(\omega))_k \equiv (x^k)_k$, $(\mathsf{Z}_\mathsf{n}^k(\omega))_k \equiv (z_\mathsf{n}^k)_k$, etc. We first analyze the case where $x^{k+1} = z_\mathsf{p}^k = x^k + \alpha_k v^k$ is generated by the proximal gradient method in step 6. Then, due to (B.1) and Remark 3.6, there exists a constant $\underline{\lambda} = \underline{\lambda}(\lambda_m, \lambda_M)$ such that

$$\begin{aligned}\|F^\Lambda(x^{k+1})\| &\leq \underline{\lambda}^{-1}(L_F\alpha_k\|v^k\| + \lambda_m^{-\frac{1}{2}}\|F^{\Lambda_k}(x^k) - F_{s^k}^{\Lambda_k}(x^k)\|_{\Lambda_k} + \lambda_m^{-\frac{1}{2}}\|F_{s^k}^{\Lambda_k}(x^k)\|_{\Lambda_k}) \\ &\leq \underline{\lambda}^{-1}\lambda_m^{-\frac{1}{2}}(L_F + 1)\|F_{s^k}^{\Lambda_k}(x^k)\|_{\Lambda_k} + (\underline{\lambda}\lambda_m)^{-1}\|\nabla f(x^k) - G_{s^k}(x^k)\|.\end{aligned}$$

Here, we utilized $\alpha_k \in [0,1]$ and the estimate $\|\mathrm{prox}_r^{\Lambda_k}(u^{\Lambda_k}(x^k)) - \mathrm{prox}_r^{\Lambda_k}(u_{s^k}^{\Lambda_k}(x^k))\|_{\Lambda_k} \leq \|\nabla f(x^k) - G_{s^k}(x^k)\|_{\Lambda_k^{-1}} \leq \lambda_m^{-1/2}\|\nabla f(x^k) - G_{s^k}(x^k)\|$, which again follows from the Λ_k-nonexpansiveness of the proximity operator. Thus, applying Lemma 3.8, using $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, for $a, b \in \mathbb{R}^n$, and setting $e_k := \|\nabla f(x^k) - G_{s^k}(x^k)\|$, we obtain

$$\begin{aligned}&\psi(x^k) - \psi(x^{k+1})\\ &\geq \underbrace{\frac{\gamma \underline{\lambda}^2 \lambda_m}{2(L_F+1)^2}}_{=:c_1} \cdot \alpha_k \|F^\Lambda(x^{k+1})\|^2 - \underbrace{\frac{1}{\lambda_m}\left(\frac{\gamma}{(L_F+1)^2} + \frac{1}{4\gamma(\rho-1)}\right)}_{=:c_2} \cdot \alpha_k e_k^2.\end{aligned} \tag{3.10}$$

Next, we derive analogous bounds for a Newton step $x^{k+1} = z_{\mathsf{n}}^k = x^k + d^k$. Similar to the last estimates and due to assumption (B.1) and Remark 3.6, we have

$$\|F^\Lambda(x^{k+1})\|^2 \leq 2\underline{\lambda}^{-2}\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^2 + 2(\underline{\lambda}\lambda_m)^{-2}\|\nabla f(x^{k+1}) - G_{s^{k+1}}(x^{k+1})\|^2. \tag{3.11}$$

Combining the growth condition (2.9), (3.11), and the bound $\alpha_{k+1} \leq 1$, it holds that

$$\begin{aligned}\psi(x^k) - \psi(x^{k+1}) &\geq -\beta\rho_k^{1-p}\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^p - \varepsilon_k^2\\ &\geq c_1 \cdot \alpha_{k+1}\|F^\Lambda(x^{k+1})\|^2 - \varepsilon_k^2 - 2c_1(\underline{\lambda}\lambda_m)^{-2} \cdot \alpha_{k+1}e_{k+1}^2\\ &\quad - \underbrace{\left(2c_1\underline{\lambda}^{-2}\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^{2-p} + \beta\rho_k^{1-p}\right)}_{=:C_k}\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^p.\end{aligned}$$

Furthermore, using $\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\| = \rho_{k+1} \leq a_{k+1} \leq \mathcal{C}_1/\eta$ and $\rho_k \leq a_k \leq \mathcal{C}_1/\,\eta$, it can be easily shown that the term C_k is bounded from above by a constant $\bar{\mathcal{C}}$ that does not depend on any of the random mini-batches S^j, T^j, $j \in \mathbb{N}_0$.

Now, let $K \in \mathbb{N}$ be arbitrary. Then, the monotonicity of $(\alpha_k)_k$, (3.8)–(3.9), and our last results imply

$$\begin{aligned}&\psi(x^0) - \psi(x^{K+1})\\ &\geq \sum_{k=0}^K c_1 \min\{\alpha_k, \alpha_{k+1}\}\,\|F^\Lambda(x^{k+1})\|^2 - \sum_{k=0}^K (1-y_{k+1})c_2 \cdot \alpha_k e_k^2\\ &\quad - \sum_{k=0}^K y_{k+1}\left[2c_1(\underline{\lambda}\lambda_m)^{-2} \cdot \alpha_{k+1}e_{k+1}^2 + \varepsilon_k^2 + \bar{\mathcal{C}} \cdot \|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^p\right]\\ &\geq \sum_{k=0}^K \left[c_1\alpha_{k+1}\|F^\Lambda(x^{k+1})\|^2 - c_2\alpha_k e_k^2 - 2c_1(\underline{\lambda}\lambda_m)^{-2}\alpha_{k+1}e_{k+1}^2\right] - \bar{\mathcal{C}}\mathcal{C}_p - \sum_{k=0}^K \varepsilon_k^2.\end{aligned}$$

Thus, taking expectation and setting $c_3 := c_2 + 2c_1(\underline{\lambda}\lambda_m)^{-2}$, we obtain

$$\sum_{k=0}^K c_1\alpha_{k+1}\mathbb{E}[\|F^\Lambda(\mathsf{X}^{k+1})\|^2] \leq \psi(x^0) - \mathbb{E}[\psi(\mathsf{X}^{K+1})] + \bar{\mathcal{C}}\mathcal{C}_p + \sum_{k=0}^K \varepsilon_k^2 + c_3\sum_{k=0}^{K+1}\alpha_k\sigma_k^2.$$

By assumption (A.2) the objective function ψ is bounded from below and hence, since the sequences $(\alpha_k\sigma_k^2)_k$ and $(\varepsilon_k^2)_k$ are summable, this yields $\sum \alpha_k\mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] < \infty$. Consequently, our first claim follows from $\sum \alpha_k = \infty$. On the other hand, Fatou's lemma implies

$$\mathbb{E}\left[\sum_{k=0}^{\infty}\alpha_k\|F^\Lambda(\mathsf{X}^k)\|^2\right] \le \liminf_{K\to\infty}\mathbb{E}\left[\sum_{k=0}^{K}\alpha_k\|F^\Lambda(\mathsf{X}^k)\|^2\right] < \infty$$

and hence, we have $\sum \alpha_k\|F^\Lambda(\mathsf{X}^k)\|^2 < \infty$ with probability 1. As before we can now infer $\liminf_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ a.s., which completes the proof of part (i). Next, using the additional lower boundedness of $(\alpha_k)_k$, we immediately obtain $\min\{\alpha_k, \alpha_{k+1}\} \ge \underline{\alpha}$. In this case, the latter results can be improved to $\sum \mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] < \infty$ and $\sum \|F^\Lambda(\mathsf{X}^k)\|^2 < \infty$ a.s., which proves part (ii). ∎

Remark 3.11. Let us assume that the samples S_i^k, T_j^k, $i \in [\mathbf{n}_k^{\mathsf{g}}]$, $j \in [\mathbf{n}_k^{\mathsf{h}}]$, $k \in \mathbb{N}_0$, are chosen independently of each other and that the conditions

$$\mathbb{E}[\mathcal{G}(x, \mathsf{S}_i^k)] = \nabla f(x), \qquad \mathbb{E}[\|\nabla f(x) - \mathcal{G}(x, \mathsf{S}_i^k)\|^2] \le \bar{\sigma}^2,$$

hold uniformly for all $i \in [\mathbf{n}_k^{\mathsf{g}}]$, $k \in \mathbb{N}_0$, and $x \in \mathbb{R}^n$ and for some $\bar{\sigma} > 0$. Then, as shown in [30] and setting $\mathsf{E}_k(x) := \|\nabla f(x) - \mathsf{G}_k(x)\|$, it follows that $\mathbb{E}[[\mathsf{E}_k(x)]^2] \le \bar{\sigma}^2[\mathbf{n}_k^{\mathsf{g}}]^{-1}$ for all $x \in \mathbb{R}^n$. Moreover, under the additional integrability conditions $\mathbb{E}[[\mathsf{E}_k(\mathsf{Z}_{\mathsf{n}}^{k-1})]^2] < \infty$, $\mathbb{E}[[\mathsf{E}_k(\mathsf{Z}_{\mathsf{p}}^{k-1})]^2] < \infty$, and using $\mathsf{Z}_{\mathsf{n}}^{k-1}, \mathsf{Z}_{\mathsf{p}}^{k-1} \in \mathcal{F}_{k-1}$, we obtain

$$\mathbb{E}[[\mathsf{E}_k(\mathsf{Z}_{\mathsf{n}}^{k-1})]^2] = \mathbb{E}[\mathbb{E}[[\mathsf{E}_k(\mathsf{Z}_{\mathsf{n}}^{k-1})]^2 \mid \mathcal{F}_{k-1}]] = \mathbb{E}[\mathbb{E}[[\mathsf{E}_k(\cdot)]^2](\mathsf{Z}_{\mathsf{n}}^{k-1})] \le \bar{\sigma}^2[\mathbf{n}_k^{\mathsf{g}}]^{-1}$$

and, similarly, $\mathbb{E}[[\mathsf{E}_k(\mathsf{Z}_{\mathsf{p}}^{k-1})]^2] \le \bar{\sigma}^2[\mathbf{n}_k^{\mathsf{g}}]^{-1}$; see, e.g., [10, Theorem 2.10]. Consequently, due to $\mathsf{Y}_k(\Omega) \subseteq \{0, 1\}$, we have

$$\mathbb{E}[\|\nabla f(\mathsf{X}^k) - \mathsf{G}_k(\mathsf{X}^k)\|^2] \le 2\mathbb{E}[(1-\mathsf{Y}_k)[\mathsf{E}_k(\mathsf{Z}_{\mathsf{p}}^{k-1})]^2] + 2\mathbb{E}[\mathsf{Y}_k[\mathsf{E}_k(\mathsf{Z}_{\mathsf{n}}^{k-1})]^2] \le \frac{4\bar{\sigma}^2}{\mathbf{n}_k^{\mathsf{g}}}.$$

Hence, one way to guarantee summability of the error terms σ_k^2 is to asymptotically increase the sample size $\mathbf{n}_k^{\mathsf{g}}$ and set $\mathbf{n}_k^{\mathsf{g}} = \mathcal{O}(k^{1+\varpi})$ for some $\varpi > 0$. This observation is similar to the results in [82, 30].

In the following, we present two situations where the approximate ψ-descent condition (2.9) is not needed in order to guarantee global convergence of the method. In many applications, such an adjustment of the algorithm can be significant, since calculating the full objective function ψ may be similarly expensive as evaluating the full gradient ∇f. Our first result shows that, in this case, it is still possible to establish global convergence of Algorithm 1 in the sense of

$$\liminf_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0 \quad \text{almost surely.}$$

Moreover, as a direct consequence of the proof of Theorem 3.10, we obtain convergence of the accepted Newton iterates in expectation.

THEOREM 3.12. *Let the random process $(\mathsf{X}^k)_k$ be generated by Algorithm* 1 *without checking the growth condition* (2.9) *and suppose that the setup and assumptions described in Theorem* 3.10 *are satisfied. It holds that*

(i) *under the assumptions*

$$\sum \alpha_k = \infty \quad \textit{and} \quad \sum \alpha_k\sigma_k^2 < \infty,$$

we have $\liminf_{k\to\infty}\mathbb{E}[\mathsf{Y}_k\|F^\Lambda(\mathsf{X}^k)\|^2] = 0$ *and* $\liminf_{k\to\infty}\mathsf{Y}_k F^\Lambda(\mathsf{X}^k) = 0$ *a.s. for any matrix* $\Lambda \in \mathbb{S}^n_{++}$;

(ii) *supposing that the conditions* $\sum \sigma_k^2 < \infty$ *and* $\alpha_k \in [\underline{\alpha}, \overline{\alpha}]$ *are fulfilled for* $\underline{\alpha} > 0$ *and all* $k \in \mathbb{N}$*, it follows that* $\lim_{k\to\infty} \mathbb{E}[\mathsf{Y}_k \|F^\Lambda(\mathsf{X}^k)\|^2] = 0$*,* $\lim_{k\to\infty} \mathsf{Y}_k F^\Lambda(\mathsf{X}^k) = 0$ *a.s., and* $\liminf_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ *a.s. for any matrix* $\Lambda \in \mathbb{S}^n_{++}$.

Proof. Reusing the notation $\mathsf{E}_k(x) = \|\nabla f(x) - \mathsf{G}_k(x)\|$ and arguing as in the proof of Theorem 3.10, the condition $\sum \mathbb{E}[\alpha_k[\mathsf{E}_k(\mathsf{X}^k)]^2] \leq \sum \alpha_k \sigma_k^2 < \infty$, which is satisfied under the assumptions stated in parts (i) and (ii), implies $\sum \alpha_k[\mathsf{E}_k(\mathsf{X}^k)]^2 < \infty$ almost surely. We now consider an arbitrary realization of the involved stochastic processes such that $\sum \alpha_k e_k^2 < \infty$. (As we have just shown such an event occurs with probability 1.) Following the proof of Theorem 3.10 and using (3.11), $\alpha_k \leq 1$, (3.8), $a_{k+1} \leq \mathcal{C}_1/\eta$, and (3.9) with $q = 1$, we have

$$\begin{aligned}\frac{\lambda^2}{2}\sum_{k=0}^{K} \alpha_{k+1} y_{k+1} \|F^\Lambda(x^{k+1})\|^2 &\leq \sum_{k=0}^{K} y_{k+1}[\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^2 + \lambda_m^{-2}\alpha_{k+1}e_{k+1}^2] \\ &\leq \sum_{k=0}^{K}[\eta^{-1}\mathcal{C}_1 \cdot y_{k+1}a_{k+1} + \lambda_m^{-2}\alpha_{k+1}e_{k+1}^2] \leq \eta^{-1}\mathcal{C}_1^2 + \lambda_m^{-2}\sum_{k=0}^{K}\alpha_{k+1}e_{k+1}^2 \end{aligned} \tag{3.12}$$

for all $K \in \mathbb{N}$ and hence, it holds that $\sum \alpha_k y_k \|F^\Lambda(x^k)\|^2 < \infty$. Thus, using the assumption $\sum \alpha_k = \infty$, we readily obtain $\liminf_{k\to\infty} y_k \cdot F^\Lambda(x^k) = 0$ and $\liminf_{k\to\infty} \mathsf{Y}_k$ $F^\Lambda(\mathsf{X}^k) = 0$ a.s. The first result stated in part (i) can be verified similarly by taking expectation in (3.12).

Let us now turn to the proof of part (ii). If infinitely many Newton steps are performed, the lower boundedness of $(\alpha_k)_k$ and (3.12) imply $\liminf_{k\to\infty} \|F^\Lambda(x^k)\|^2 = 0$. In the case $\sum y_k < \infty$, there exists $K_0 \in \mathbb{N}$ such that $y_k = 0$ for all $k > K_0$ and summing the estimate (3.10), it follows that

$$\sum_{k=K_0}^{K} c_1\alpha_k \|F^\Lambda(x^{k+1})\|^2 \leq \psi(x^{K_0}) - \psi(x^{K+1}) + c_2 \sum_{k=K_0}^{K} \alpha_k e_k^2$$

for all $K \geq K_0$. Assumption (A.2) and $\alpha_k \geq \underline{\alpha}$, $k \in \mathbb{N}$, now yield $\lim_{k\to\infty} \|F^\Lambda(x^k)\|^2 = 0$. Combining those two different cases, this establishes $\liminf_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ almost surely. The remaining results follow, as before, from (3.12) using the boundedness of $(\alpha_k)_k$. ∎

The following variant of Theorem 3.10 is mainly based on the strong convexity assumption (A.3) and on the boundedness condition (A.4).

THEOREM 3.13. *Let the sequence* $(\mathsf{X}^k)_k$ *be generated by Algorithm* 1 *without checking the growth condition* (2.9)*. Suppose that the assumptions* (A.1)*,* (A.3)–(A.4)*,* (B.1)–(B.2)*, and* (C.1)–(C.2) *are satisfied. Furthermore, suppose that the step sizes* $(\alpha_k)_k$ *are chosen via* $\alpha_k \in [\underline{\alpha}, \overline{\alpha}]$ *for some* $\underline{\alpha} > 0$ *and all* $k \in \mathbb{N}$*. Then, under the additional assumption*

$$\sum \sigma_k < \infty,$$

it holds that $\lim_{k\to\infty} \mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] = 0$ *and* $\lim_{k\to\infty} F^\Lambda(\mathsf{X}^k) = 0$ *a.s. for any* $\Lambda \in \mathbb{S}^n_{++}$.

Proof. As in the proof of Theorem 3.10, we want to derive suitable lower bounds for the ψ-descent $\psi(x^k) - \psi(x^{k+1})$ for an arbitrary realization of the occurring stochastic processes. We first consider the case where $x^{k+1} = z_{\mathsf{p}}^k$ is generated by the proximal gradient method. Then, using (3.10) and the bound on α_k, we have

$$\psi(x^k) - \psi(x^{k+1}) \geq c_1\underline{\alpha}\|F^\Lambda(x^{k+1})\|^2 - c_2 \cdot e_k^2.$$

Next, we discuss the second case $x^{k+1} = z_{\mathsf{n}}^k$. By Lemma 3.7 and reusing the estimate $\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\| = \rho_{k+1} \le a_{k+1} \le \mathcal{C}_1/\eta$ (see again (3.8)), it holds that

$$\|z_{\mathsf{n}}^k - x^*\|^2 \le B_1(\tau)\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\|^2 + B_2(\tau)e_{k+1}^2 \le \frac{B_1(\tau)\mathcal{C}_1}{\eta}\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\| + B_2(\tau)e_{k+1}^2$$

for some $\tau > 0$. By assumption (A.3), the functions $f - \frac{\mu_f}{2}\|\cdot\|^2$ and $r - \frac{\mu_r}{2}\|\cdot\|^2$ are convex (and directionally differentiable) and hence, we have

$$\psi(y) - \psi(x) \ge r'(x; y-x) + \langle \nabla f(x), y - x\rangle + \frac{\bar\mu}{2}\|y-x\|^2 \quad \forall\, x, y \in \operatorname{dom} r. \tag{3.13}$$

Now, applying the optimality of x^*, (3.13) with $x \equiv z_{\mathsf{n}}^k$ and $y \equiv x^*$, $z_{\mathsf{n}}^k \in \operatorname{dom} r$, the Lipschitz continuity of ∇f and F^Λ, the subadditivity of the square root, and defining

$$d_1 := \sqrt{B_1(\tau)}(\bar g_r + \|\nabla f(x^*)\|) + \eta^{-1}B_1(\tau)\mathcal{C}_1 L, \quad d_2 := \sqrt{B_2(\tau)}(\bar g_r + \|\nabla f(x^*)\|),$$

we obtain

$$\begin{aligned}
\psi(x^k) - \psi(x^{k+1}) &= \psi(x^k) - \psi(x^*) + \psi(x^*) - \psi(x^{k+1}) \\
&\ge r'(z_{\mathsf{n}}^k; x^* - z_{\mathsf{n}}^k) + \langle \nabla f(z_{\mathsf{n}}^k), x^* - z_{\mathsf{n}}^k\rangle + 0.5\bar\mu \cdot \|z_{\mathsf{n}}^k - x^*\|^2 \\
&\ge -(\bar g_r + \|\nabla f(x^*)\|) \cdot \|z_{\mathsf{n}}^k - x^*\| + (0.5\bar\mu - L)\,\|z_{\mathsf{n}}^k - x^*\|^2 \\
&\ge -d_1\|F_{s^{k+1}}^{\Lambda_{k+1}}(z_{\mathsf{n}}^k)\| - d_2 e_{k+1} - B_2(\tau)Le_{k+1}^2 + \frac{\bar\mu\underline\lambda^2}{2L_F^2}\|F^\Lambda(x^{k+1})\|^2.
\end{aligned}$$

Combining the last inequalities, setting $d_3 := \min\{c_1\underline\alpha, (2L_F^2)^{-1}\bar\mu\underline\lambda^2\}$ and using again (3.9) with $q = 1$, it holds that

$$\psi(x^0) - \psi(x^{K+1}) \ge \sum_{k=1}^{K+1}\left[d_3\|F^\Lambda(x^k)\|^2 - d_2 e_k\right] - (c_2 + B_2(\tau)L)\sum_{k=0}^{K+1} e_k^2 - d_1\mathcal{C}_1$$

for all $K \in \mathbb{N}$. Taking expectation, our first claim now follows from (C.2), Jensen's inequality, $\sum \sigma_k < \infty$, and the lower boundedness of $\psi(x^{K+1})$. The probabilistic convergence of the sequence $(F^\Lambda(\mathsf{X}^k))_k$ can then be inferred as in the proof of Theorem 3.10. ∎

Remark 3.14. Let us note that assumption (A.4) is required to derive a suitable lower bound for the difference terms $\psi(x^k) - \psi(z_{\mathsf{n}}^k)$ which allows us to apply Lemma 3.9. Furthermore, condition (A.4) is always satisfied in the following situation. Suppose that the mapping r has the special form $r = \iota_{\mathcal{S}} + \varphi$, where $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a real-valued, convex function and $\iota_{\mathcal{S}} : \mathbb{R}^n \to (-\infty, +\infty]$ is the indicator function of a nonempty, convex, and closed set $\mathcal{S} \subset \mathbb{R}^n$. Then, assumption (A.4) holds either if the set $\mathcal{S}$ is compact or if φ is positively homogeneous. In particular, condition (A.4) is satisfied if r is a norm.

Proof. For every feasible point $x \in \operatorname{dom} r = \mathcal{S}$, we have $0 \in \partial\iota_{\mathcal{S}}(x)$ and thus $\partial\varphi(x) \subset \partial r(x)$. By [7, Proposition 16.17], the set $\bigcup_{x\in\mathcal{S}}\partial\varphi(x)$ is bounded if $\mathcal{S}$ is bounded. On the other hand, if φ is positively homogeneous, then it follows $\partial\varphi(x) = \{\lambda \in \partial\varphi(0) : \langle\lambda, x\rangle = \varphi(x)\} \subset \partial\varphi(0)$; see, e.g., [7, Proposition 16.18] and [44, Example 2.5.17]. Since $\partial\varphi(0)$ is again a compact set, this proves our claim. ∎

The result in Theorem 3.13 can be further improved by additionally damping the semismooth Newton step and setting $z_{\mathsf{n}}^k := x^k + \alpha_k d^k$. Then, due to the convexity of ψ, we have $\psi(x^k) - \psi(z_{\mathsf{n}}^k) \geq \alpha_k(\psi(x^k) - \psi(x^k + d^k))$ and we can use the weaker conditions

$$(\alpha_k)_k \text{ is monotonically decreasing}, \quad \sum \alpha_k = \infty, \quad \sum \alpha_k \sigma_k < \infty,$$

to ensure $\liminf_{k\to\infty} \mathbb{E}[\|F^\Lambda(\mathsf{X}^k)\|^2] = 0$. We conclude with two important remarks.

Remark 3.15. We want to emphasize that the global results derived in this section do not explicitly depend on the sampling strategy or on any (uniform) invertibility properties of the stochastic oracle H_t or of the chosen generalized derivatives M_k. This is significantly different from other convergence results for stochastic second order–type methods that often rely on uniform invertibility bounds of the form

$$\inf_{\omega\in\Omega} \inf_{k\in\mathbb{N}} \lambda_{\min}(H_{\mathsf{T}^k(\omega)}(\mathsf{X}^k(\omega))) \geq \bar{\tau} \quad \text{or} \quad \inf_{x\in\mathbb{R}^n} \lambda_{\min}(\nabla^2 f(x)) \geq \bar{\tau}$$

for a fixed $\bar{\tau} > 0$; see, e.g., [12, section 2.1], [60, section 3.1], and [1, 17, 73] for comparison.

Remark 3.16. The developed convergence theory and globalization mechanism is still applicable if a different type of direction d^k is used instead of the semismooth Newton step $d^k = -M_k^+ F_{s^k}^{\Lambda_k}(x^k)$. In fact, we only require that the selected directions correspond to the trajectory of a stochastic process $(\mathsf{D}^k)_k$ of $\mathcal{F}_k$-measurable random mappings $\mathsf{D}^k : \Omega \to \mathbb{R}^n$, i.e., we have $\mathsf{D}^k \in \mathcal{F}_k$ for all k and for $\omega \in \Omega$ it holds that $(\mathsf{D}^k(\omega))_k \equiv (d^k)_k$. Consequently, the presented analysis also covers various extensions of our algorithmic framework. In particular, we can consider inexact variants of the semismooth Newton step (2.6) that, for instance, result from iterative solvers like the CG or Krylov subspace method or other directions such as stochastic quasi-Newton-type steps. Choosing a stochastic Newton direction as in (2.6), however, allows us to establish transition to fast local convergence with high probability. That is, for this choice, it is possible to show that the growth conditions (2.8)–(2.9) are always satisfied eventually under certain local assumptions. This theoretical property of the stochastic semismooth Newton method is discussed in detail in the companion paper [46].

4. Numerical results.

In this section, we demonstrate the efficiency of the proposed stochastic semismooth Newton framework and compare it with several state-of-the-art algorithms on a variety of test problems. All numerical experiments are performed using MATLAB R2018b on a MacBook Pro with Intel Core i7 2.6GHz and 16GB memory.

4.1. Logistic regression.

In our first experiment, we consider the well-known empirical ℓ_1-logistic regression problem

$$\min_{x\in\mathbb{R}^n} \ \psi(x) := f(x) + \mu\|x\|_1, \quad f(x) := \frac{1}{N}\sum_{i=1}^N f_i(x), \tag{4.1}$$

where $f_i(x) := \log(1 + \exp(-b_i \cdot \langle a_i, x\rangle))$ denotes the logistic loss function and the data pairs $(a_i, b_i) \in \mathbb{R}^n \times \{-1, 1\}$, $i \in [N]$, correspond to a given dataset or are drawn from a given distribution. The regularization parameter $\mu > 0$ controls the level of sparsity of a solution of problem (4.1). In the following, we always choose $\mu = 0.01$.

The datasets tested in our numerical comparison are summarized in Table 1. We linearly scale the entries of the data-matrix $(a_1, \ldots, a_N)$ to $[0, 1]$ for each dataset. The datasets for multiclass classification have been manually divided into two types or features. For instance, the `MNIST` dataset is used for classifying even and odd digits.

TABLE 1
A description of the datasets used in the experiments.

Dataset	Data points N	Variables n	Density	Reference
`CINA`	16 033	132	29.51%	[74]
`covtype`	581 012	54	22.12%	[11]
`gisette`	6 000	5 000	13.87%	[33]
`MNIST`	60 000	784	19.12%	[38]
`rcv1`	20 242	47 236	0.16%	[40]
`sido0`	12 678	4 932	9.84%	[75]

4.1.1. Algorithmic details. Next, we describe the implementational details of our approach and of the state-of-the-art algorithms used in the numerical comparison.

Stochastic oracles. In each iteration and similar to other stochastic second order methods, [15, 12, 17, 60], we generate stochastic approximations of the gradient and Hessian of f via first selecting two subsamples $\mathcal{S}_k, \mathcal{T}_k \subset [N]$ uniformly at random and without replacement from the index set $\{1, \dots, N\}$. We then define the mini-batch-type stochastic oracles

$$G_{s^k}(x) := \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \nabla f_i(x), \quad H_{t^k}(x) := \frac{1}{|\mathcal{T}_k|} \sum_{j \in \mathcal{T}_k} \nabla^2 f_j(x), \tag{4.2}$$

which correspond to the stochastic setup $\Xi := [N]$, $\mathcal{X} := \mathcal{P}(\Xi)$, $s_i^k :=$ ith element of $\mathcal{S}_k$, $\mathbf{n}_k^{\mathbf{g}} := |\mathcal{S}_k|$, and $\mathcal{G}(x, s_i^k) := \nabla f_{s_i^k}(x)$, etc. We refer to the variant of Algorithm 1 using the stochastic oracles (4.2) as S4N (**sub**sampled **s**emi**s**mooth **N**ewton method). Furthermore, motivated by the recent success of variance reduction techniques [36, 77, 56, 5, 73], we will also work with a variance reduced stochastic gradient that can be calculated as follows

$$\begin{cases} \mathbf{1} & \textbf{if} \quad k \bmod m = 0 \quad \textbf{then} \quad \text{set } \tilde{x} := x^k \text{ and calculate } \tilde{u} := \nabla f(\tilde{x}). \\ \mathbf{2} & \text{Compute } G_{s^k}(x^k) := \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} (\nabla f_i(x^k) - \nabla f_i(\tilde{x})) + \tilde{u}. \end{cases} \tag{4.3}$$

Here, $k \in \mathbb{N}$ is the current iteration and $m \in \mathbb{N}$ denotes the number of iterations after which the full gradient ∇f is evaluated at the auxiliary variable $\tilde{x}$. As in [56, 5, 73], this additional noise-free information is stored and utilized in the computation of the stochastic oracles for the next iterations.

Overview of the tested methods.

- Adagrad [23]. Adagrad is a stochastic proximal gradient method with a specific strategy for choosing the matrices Λ_k. We use the mini-batch gradient (4.2) as first order oracle in our implementation. This leads to the update rule

$$x^{k+1} = \operatorname{prox}_{\varphi}^{\Lambda_k}(x^k - \Lambda_k^{-1} G_{s^k}(x^k)), \quad \Lambda_k := \lambda^{-1} \operatorname{diag}(\delta \mathbb{1} + \sqrt{G_k}), \tag{4.4}$$

  where $\delta, \lambda > 0$, $G_k := G_{k-1} + G_{s^k}(x^k) \odot G_{s^k}(x^k)$, and the multiplication "$\odot$" and the square root "$\sqrt{\cdot}$" are performed componentwise.
- prox-SVRG [77]. Prox-SVRG is a variance reduced, stochastic proximal gradient method. Similar to [56, 57, 72], we substitute the basic variance reduction technique proposed in [36, 77] with the mini-batch version (4.3) to improve its performance.
- PG-BB. PG-BB is the basic proximal gradient method using Barzilai–Borwein step sizes and a monotone line search technique; see, e.g., [27, 76].

- PNOPT [39]. The proximal Newton method PNOPT is a deterministic higher order variant of the scheme (4.4). The matrix Λ_k is chosen as an SR1- or (L)BFGS-approximation of the true Hessian $\nabla^2 f$ to accelerate convergence.
- DAL [69]. The dual augmented Lagrangian algorithm DAL is designed for sparse estimation problems and can achieve fast q-superlinear convergence. It can be interpreted as a dual instance of the proximal point method applied to problem (4.1). In each iteration, a subproblem in the dual space $\mathbb{R}^N$ has to be solved approximately via special semismooth Newton steps to obtain the next iterate.
- S2N-D. S2N-D is the deterministic version of the stochastic semismooth Newton method using the full gradient and Hessian of f instead of stochastic oracles.
- S4N-HG. S4N with both subsampled gradient and Hessian (4.2). In the numerical experiments, the maximum sample size $|\mathcal{S}_k|$ of the stochastic oracle G_{s^k} is limited to 10%, 50%, and 100% of the training data size N, respectively.
- S4N-H. This version of S4N uses the full gradient ∇f and the subsampled Hessian H_{t^k} as defined in (4.2).
- S4N-VR. S4N-VR is a variant of S4N combining the variance reduced stochastic oracle (4.3) with the basic subsampling strategy (4.2) for the Hessian of f.

We compare Adagrad, prox-SVRG, PG-BB, PNOPT, and DAL with four different versions of our S4N method: S2N-D (deterministic), S4N-HG (subsampled gradient and Hessian), S4N-H (full gradient and subsampled Hessian), and S4N-VR (variance reduced stochastic gradient and subsampled Hessian).

Implementational details. For Adagrad, the sample size $|\mathcal{S}_k|$ of the stochastic gradient is fixed to 5% of the training data size N and we set $\delta = 10^{-7}$. The parameter λ varies for the different tested datasets and is chosen from the set $\{i \cdot 10^j : i \in [9], j \in \{-2, -1, 0, 1\}\}$ to guarantee optimal performance. The iterative scheme of prox-SVRG basically coincides with (4.4). Here, we also use a fixed sample size $|\mathcal{S}_k| = \lfloor 0.01N \rfloor$ and we set $m = 10$. The parameter matrix Λ_k is defined via $\Lambda_k := (1/\lambda_k)I$ and based on the full gradient values $\nabla f(\tilde{x})$, λ_k is chosen adaptively to approximate the Lipschitz constant of the gradient ∇f. The code for PNOPT and DAL is available online at http://web.stanford.edu/group/SOL/software/pnopt/ and https://github.com/ryotat/dal/blob/master/README.md. Instead of TFOCS, we use a specialized version of FISTA [8] to solve the inner subproblems in PNOPT and the algorithm is tested with different Hessian approximations: 0-memory SR1, 0-memory BFGS (LB-0), and LBFGS with memory 10 and 50 (LB-10, LB-50). In the comparison, we only report the performance of the quasi-Newton scheme corresponding to the best obtained result.

In S4N-HG, the initial sample size of the stochastic gradient is set to $|\mathcal{S}_0| = \lfloor 0.01N \rfloor$. The size of the mini-batch $\mathcal{S}_k$ is then increased by a factor of 3.375 every 30 iterations until $|\mathcal{S}_k|$ reaches the maximum sizes $\lfloor 0.1N \rfloor$, $\lfloor 0.5N \rfloor$, and N, respectively. In the following, we will use S4N-HG 10%, S4N-HG 50%, and S4N-HG 100% to denote the different variants of S4N-HG. In S4N-VR, we use the fixed sample size $|\mathcal{S}_k| = \lfloor 0.01N \rfloor$ for all k and $m = 10$. The mini-batch sizes of the stochastic Hessians are adjusted in a similar way. More specifically, in S4N-HG, S4N-H, and S4N-VR, we first set $|\mathcal{T}_0| = \lfloor 0.01N \rfloor$. As soon as the sample $\mathcal{S}_k$ reaches its maximum size, we repeatedly increase the size of the set $\mathcal{T}_k$ by a factor of 3.375 after 15 iterations. The upper limit of the Hessian sample size is set to 6% of the training data size, i.e., we have $|\mathcal{T}_k| \leq t_{\max}$ with $t_{\max} := \lfloor 0.06N \rfloor$ for all k.

In S4N-HG 10% and different from the other methods, the size of $\mathcal{T}_k$ is not changed, i.e., it holds that $t_{\max} = \lfloor 0.01N \rfloor$. As in prox-SRVG, we use $\Lambda_k := (1/\lambda_k)I$ and choose λ_k adaptively to estimate the (local) Lipschitz constant of the gradient. In particular, we compute

$$\lambda_k^1 = \frac{\|x^k - x^{k-1}\|}{\|G_{s^k}(x^k) - G_{s^{k-1}}(x^{k-1})\|}, \quad \lambda_k^2 = \max\{10^{-3}, \min\{10^4, \lambda_k^1\}\}.$$

In order to prevent outliers, we calculate a weighted mean of λ_k^2 and of the previous parameters λ_j, $j \in [k-1]$. This mean is then used as the new step size parameter λ_k. The initial step size is set to $\lambda_0 = 0.1$.

The proximity operator of the ℓ_1-norm has the explicit representation $\operatorname{prox}^{\Lambda_k}_{\mu\|\cdot\|_1}(u) = u - \mathcal{P}_{[-\mu\lambda_k, \mu\lambda_k]}(u)$ and is also known as the soft-thresholding function. Similar to [45, 52, 78], we will work with the following generalized Jacobian of $\operatorname{prox}^{\Lambda_k}_{\mu\|\cdot\|_1}$ at some $u \in \mathbb{R}^n$:

$$D(u) := \operatorname{diag}(d(u)), \quad d(u) \in \mathbb{R}^n, \quad d(u)_i := \begin{cases} 1, & |u_i| > \mu\lambda_k, \\ 0 & \text{otherwise.} \end{cases}$$

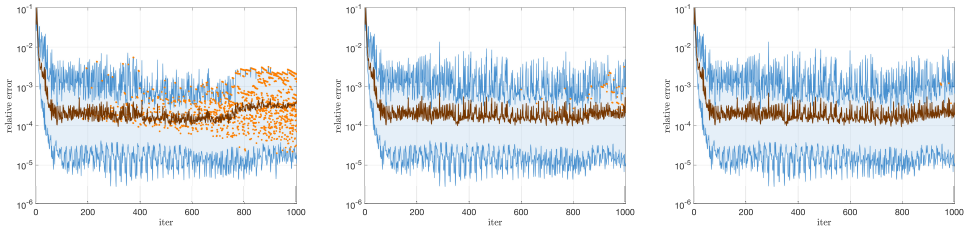
The generalized derivatives of $F^{\Lambda_k}_{s^k}$ are then built as in (2.7). As described in [45, 52, 78], we can exploit the structure of the resulting semismooth Newton system and reduce it to a smaller and symmetric linear system of equations. We utilize an early terminated CG method to solve this system approximately. The maximum number of iterations and the desired accuracy of the CG method are adjusted adaptively depending on the computed residual $\|F^{\Lambda_k}_{s^k}(x^k)\|$. When the residual is large, only a few iterations are performed to save time. The initial relative tolerance and the initial maximum number of iterations are set to 0.01 and 2, respectively. The total maximum number of CG iterations is restricted to 12. In order to numerically robustify the computation of the Newton step z^k_{n}, we also consider the following, regularized version of the Newton system:

$$(M_k + \theta_k I) \cdot d^k = -F^{\Lambda_k}_{s^k}(x^k), \quad M_k \in \mathcal{M}^{\Lambda_k}_{s^k, t^k}(x^k),$$

where $\theta_k > 0$ is a small positive number. We adjust θ_k according to the norm of the residual $F^{\Lambda_k}_{s^k}(x^k)$ so that $\theta_k \to 0$ as $\|F^{\Lambda_k}_{s^k}(x^k)\| \to 0$.

Finally, in our implementation of S4N, we only check the first growth condition (2.8) to measure the quality of the Newton step z^k_{n}. Although both growth conditions (2.8) and (2.9) are generally required to ensure strong global convergence, this adjustment does not affect the globalization process and convergence of S4N in the numerical experiments. In fact, as we have shown in Theorems 3.12 and 3.13, the condition (2.9) is not necessary for strongly convex problems or for almost sure liminf-convergence. Furthermore, as mentioned in Remark 3.16, it is possible to verify that the growth condition (2.9) is always satisfied locally close to a stationary point of problem (1.1) under certain assumptions. These different observations motivate us to restrict the acceptance test of the Newton steps to the cheaper condition (2.8). We use the following parameters: $\eta = 0.85$, $\nu_k = \varepsilon_k^1 = c_\nu k^{-1.1}$, $c_\nu = 250$, and $\alpha_k = 10^{-2}$.

In Figure 1 and Table 2, we exemplarily illustrate the acceptance of Newton steps and of the growth condition (2.8) for S4N-HG 10% depending on the choice of c_ν. The results in Table 2 indicate that a vast majority of semismooth Newton steps are accepted as a new iterate and only a comparably small number of proximal gradient steps



(a) $c_\nu = 50$ (b) $c_\nu = 150$ (c) $c_\nu = 250$

FIG. 1. *Performance of S4N-HG 10% on* `rcv1`*: range of the relative error with respect to the number of iterations and different choices of* c_ν *over* 25 *independent runs. The brown line depicts the averaged results. Occurring proximal gradient steps are marked in orange.*

TABLE 2

Acceptance rate of Newton steps for S4N-HG 10% on problem (4.1) *for different choices of* c_ν*. The algorithm is run* 25 *times for* 1000 *iterations.* $n_{\texttt{rej}}$*: maximum total number of rejected steps of a single run.* $p_{\texttt{acc}}$*: percentage acceptance rate averaged over the* 25 *runs.*

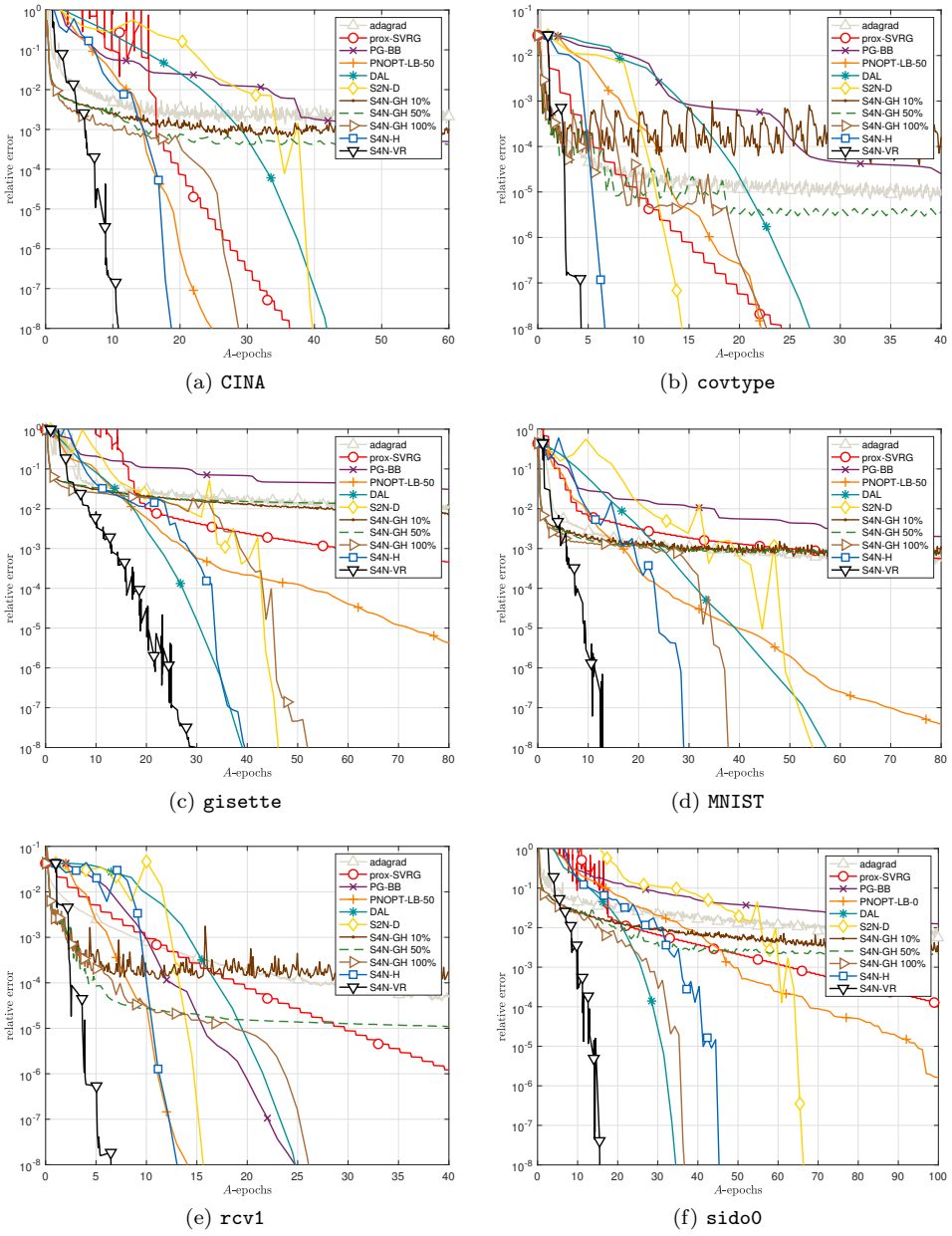
c_ν	CINA		covtype		gisette		MNIST		rcv1		sido0	
	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$	$n_{\texttt{rej}}$	$p_{\texttt{acc}}$
50	9	99.8%	64	99.3%	–	100%	–	100%	293	87.1%	1	99.9%
150	–	100%	6	99.9%	–	100%	–	100%	46	99.6%	–	100%
250	–	100%	–	100%	–	100%	–	100%	6	99.9%	–	100%

is performed. Figure 1 visualizes this behavior in more detail for the dataset `rcv1`. Here, the reported relative error is computed via $(\psi(x)-\psi(x^*))/\max\{1,|\psi(x^*)|\}$ and x^* is a reference solution generated by S2N-D. Figure 1 shows that most of the Newton steps are rejected in the case $c_\nu = 50$ when S4N-HG 10% stagnates and the Newton direction does not provide sufficient progress. We have also conducted similar experiments with other S4N variants and found that the corresponding acceptance rates are even higher and more robust than the rates of S4N-HG 10%.

4.1.2. Numerical comparison. In Figures 2 and 3, we show the performances of all methods for solving the logistic regression problem (4.1). The change of the *relative error* is reported with respect to *A-epochs* and *cpu-time*. Here, one A-epoch corresponds to *two* applications of the full matrix A in form of Ax or $A^\top y$. Specifically, if $\tilde{A} \in \mathbb{R}^{Q\times q}$ is a $Q \times q$ submatrix of A, the matrix-vector product $\tilde{A}z$, $z \in \mathbb{R}^q$, contributes $0.5 \cdot (Q/N)(q/n)$ to the number of A-epochs. Hence, the cost of a single evaluation of the stochastic oracle $G_{s^k}(x^k)$ is exactly $|\mathcal{S}_k|/N$. In contrast to the number of *epochs*, i.e., the number of full passes over a dataset, this also allows us to measure the performance of the CG method used in S4N and DAL. The results presented in Figures 2 and 3 are averaged over 25 independent runs. For all methods, we choose $x^0 = 0$ as initial point.

At first, we observe that S2N-D, S4N-HG 100%, S4N-H, and S4N-VR outperform the first order methods Adagrad and PG-BB with respect to both the required number of A-epochs and cpu-time. (PG-BB is only faster on the dataset `rcv1`). Furthermore, the different variants of S4N and S2N-D seem to be especially well suited for recovering high accuracy solutions.

The deterministic semismooth Newton method S2N-D decreases slowly in the early stage of the iteration process but converges rapidly when the iterates are close to an optimal solution. The behavior of the dual augmented Lagrangian method DAL



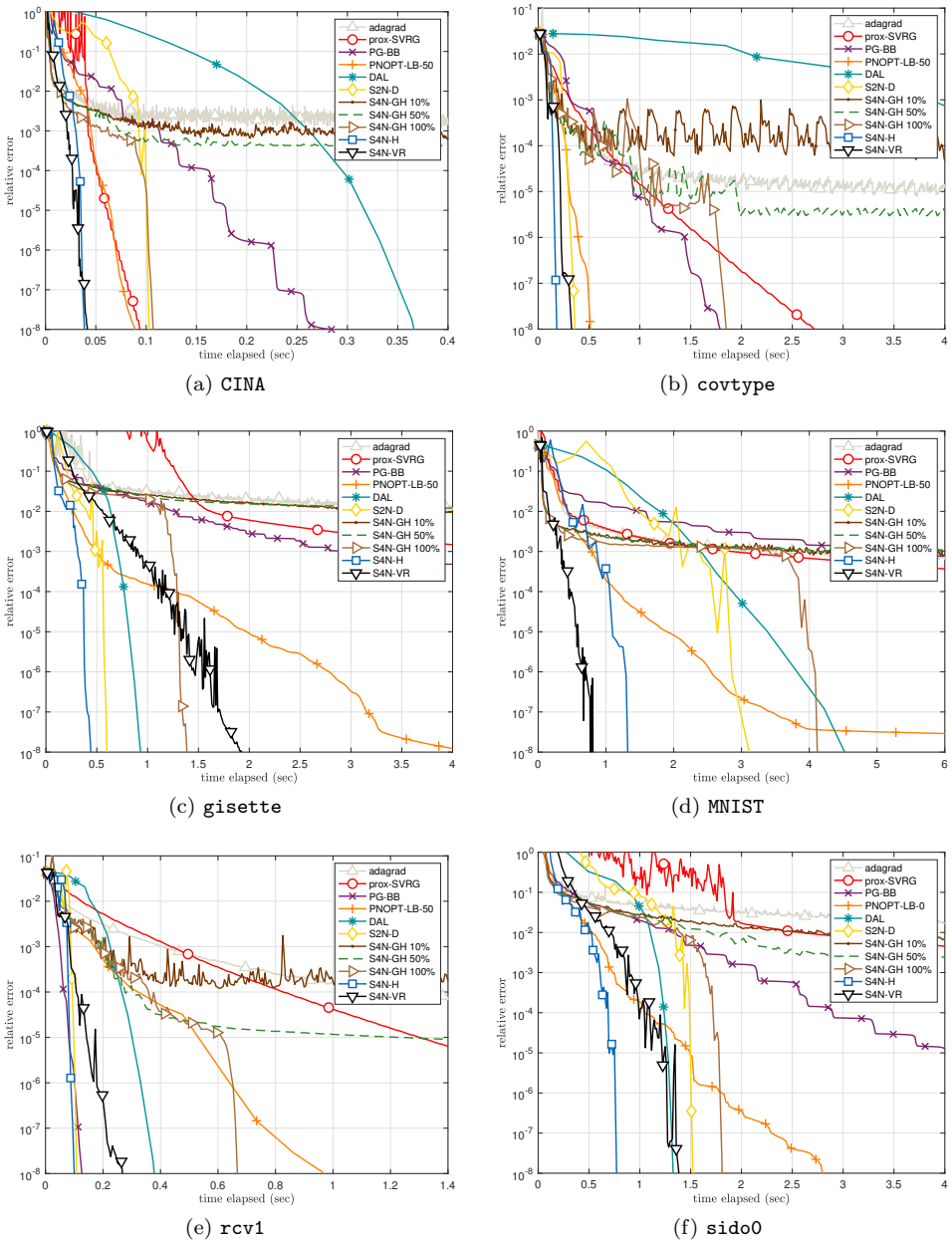
(a) `CINA` (b) `covtype`

(c) `gisette` (d) `MNIST`

(e) `rcv1` (f) `sido0`

FIG. 2. *Change of the relative error with respect to the required A-epochs for solving the* ℓ_1*-logistic regression problem* (4.1) *(averaged over* 25 *independent runs).*

is somewhat similar, although it typically requires a longer runtime than S2N-D. The results show that in the early stage the performances of both S2N-D and DAL are inferior to the performance of the other stochastic methods. If a higher precision is required, then both algorithms become more efficient. On the datasets `CINA` and `rcv1`, PNOPT recovers a high precision solution using a smaller number of A-epochs than DAL and S2N-D. However, similar to DAL, it requires more cpu-time for convergence. Overall (with few exceptions), the deterministic higher order methods DAL, PNOPT,



(a) `CINA` (b) `covtype`

(c) `gisette` (d) `MNIST`

(e) `rcv1` (f) `sido0`

FIG. 3. *Change of the relative error with respect to the cpu-time for solving the* ℓ_1*-logistic regression problem* (4.1) *(averaged over* 25 *independent runs).*

and S4N-D are not competitive with the stochastic variants S4N-H and S4N-VR and converge slower. These observations indicate the strength of stochastic algorithms in general.

Our numerical experiments show that the different performances of the stochastic methods can be roughly split into two categories. The first category includes Adagrad, S4N-HG 10%, and S4N-HG 50%, while the second category consists of S4N-H, S4N-HG 100%, and S4N-VR. The performance of prox-SVRG depends on the tested

datasets. While in `gisette`, `MNIST`, and `sido0`, it converges slowly and performs similarly to Adagrad, prox-SVRG shows much faster convergence on the datasets `CINA`, `covtype`, and `rcv1`. Considering the results in Figure 2, it appears that the performance of S4N-HG 10% and S4N-HG 50% is comparable to that of the first order method Adagrad. Since the maximum sample set size of the stochastic gradient in S4N-HG 10% and S4N-HG 50% is limited to $\lfloor 0.1N \rfloor$ and $\lfloor 0.5N \rfloor$, the associated gradient error terms still might be too large, preventing transition to fast local convergence and causing stagnation of the methods. Consequently and similar to the observations for stochastic quasi-Newton methods [17, 73], the performance of S4N is greatly affected by the sampling strategy and the accuracy of the gradient approximation. The results in Figures 2 and 3 demonstrate that the performance of S4N can be further improved by increasing the sample size of the gradient gradually to its full size, as in S4N-HG 100%, or by introducing an additional variance reduction technique as in S4N-VR. We also observe that S4N-VR outperforms all of the tested methods with respect to number of required A-epochs which indicates that the combination of second order information and variance reduction is advantageous and very promising.

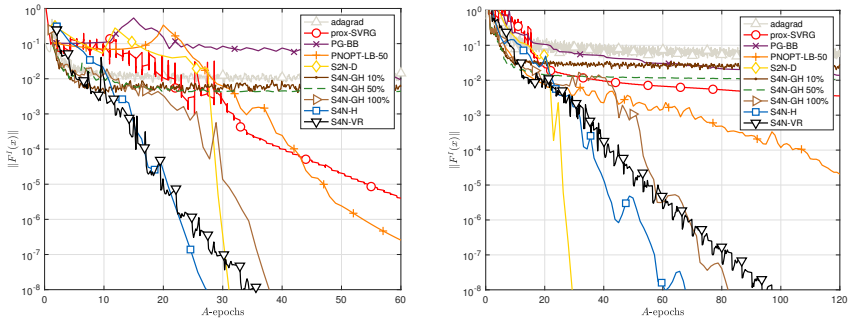
4.2. Nonconvex binary classification. In this subsection, we consider the following nonconvex, binary classification problem [43, 73]:

$$\min_{x \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^{N} [1 - \tanh(b_i \cdot \langle a_i, x \rangle))] + \mu \|x\|_1, \tag{4.5}$$

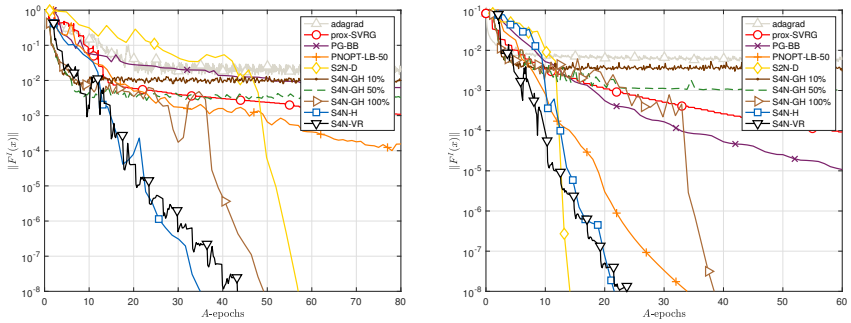
where $f_i(x) := 1 - \tanh(b_i \langle a_i, x \rangle))$ is the sigmoid loss function and $\mu = 0.01$ is a regularization parameter. We test the datasets `CINA`, `gisette`, `MNIST`, and `rcv1` to evaluate the performance of the different versions of S4N.

The sampling strategy and parameters are adjusted as follows. For all S4N methods, the initial mini-batch size of the stochastic gradient and the Hessian are increased to $|\mathcal{S}_0| = |\mathcal{T}_0| = \lfloor 0.03N \rfloor$. In S4N-HG 100% and S4N-H, we set $t_{\max} = \lfloor 0.24N \rfloor$ and in S4N-HG 50% and S4N-VR, we use $t_{\max} = \lfloor 0.18N \rfloor$. In S4N-HG 10% we do not adjust the size of the Hessian mini-batch and set $t_{\max} = |\mathcal{T}_0|$. We utilize the minimal residual method (MINRES) to solve the reduced Newton system and the maximum number of MINRES iterations is set to 32. Furthermore, we change the initial value for λ_0 to 100. All remaining parameters and strategies follow the setup discussed in the last subsection.

The numerical results are presented in Figures 4 and 5. We report the change of the residual $\|F^I(x)\|$ with respect to the required A-epochs and cpu-time. In general, the overall performance of the different methods is similar to the results shown in the last subsection. However, in contrast to the convex logistic regression problem, the more accurate approximation of the Hessian seems to be beneficial and can accelerate convergence. This observation is also supported by the improved performance of the deterministic semismooth Newton method S2N-D. Our results show that prox-SVRG now consistently outperforms S4N-HG 10% and S4N-HG 50% in recovering high precision solutions. Similar to the convex examples, S4N-HG 100% manages to significantly reduce the residual $\|F^I(x)\|$ in the first iterations. As soon as the stochastic error in the gradient approximation becomes negligible, the behavior of S4N-HG 100% changes and fast local convergence can be observed. S4N-H and S4N-VR still compare favorably with the other approaches and outperform the deterministic methods PG-BB and PNOPT. As in the last section and regarding the cpu-time, S4N-H achieves good results and converges quickly to highly accurate solutions.

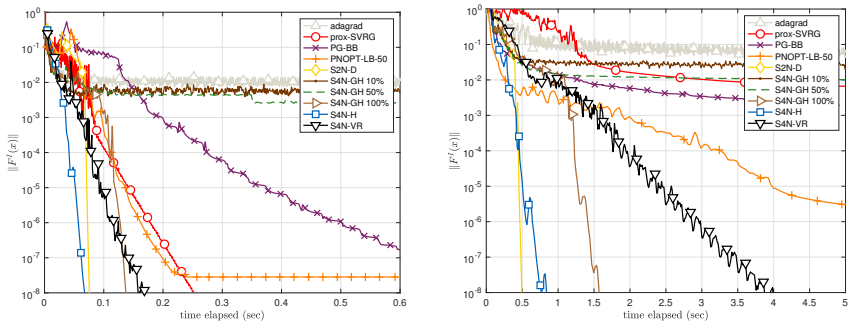


(a) CINA (b) gisette

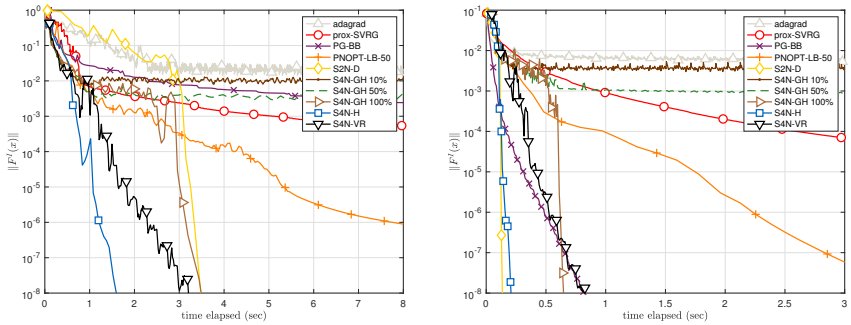


(c) MNIST (d) rcv1

FIG. 4. *Change of the residual* $\|F^I(x)\|$ *with respect to A-epochs for solving the nonconvex binary classification problem* (4.5) *(averaged over* 25 *independent runs).*



(a) CINA (b) gisette



(c) MNIST (d) rcv1

FIG. 5. *Change of the residual* $\|F^I(x)\|$ *with respect to cpu-time for solving the nonconvex binary classification problem* (4.5) *(averaged over* 25 *independent runs).*

5. Conclusion. In this paper, we investigate a stochastic semismooth Newton method for solving nonsmooth and nonconvex minimization problems. In the proposed framework, the gradient and Hessian of the smooth part of the objective function are approximated by general stochastic first and second order oracles. This allows the application of various subsampling and variance reduction techniques or other stochastic approximation schemes. The method is based on stochastic semismooth Newton steps, stochastic proximal gradient steps, and growth conditions and a detailed discussion of the global convergence properties is provided. The approach is tested on an ℓ_1-logistic regression and a nonconvex binary classification problem on a variety of datasets. The numerical comparisons indicate that our algorithmic framework and especially the combination of (generalized) second order information and variance reduction are promising and competitive.

Appendix A. Proofs of auxiliary results.

A.1. Proof of Fact 3.4. Let $(\Omega, \mathcal{F})$ and $(\Xi, \mathcal{X})$ be measurable spaces; then $\mathcal{F} \otimes \mathcal{X}$ denotes the usual product σ-algebra of the product space $\Omega \times \Xi$. We use $\mathcal{B}(\mathbb{R}^n)$ to denote the Borel σ-algebra of $\mathbb{R}^n$. We start with a preparatory result.

LEMMA A.1. *Let $(\Omega, \mathcal{F})$ be a measurable space and let $T : \Omega \to \mathbb{R}^{n\times n}$ and $y : \Omega \to \mathbb{R}^n$ be measurable functions. Then, the mapping $\omega \mapsto \zeta(\omega) := T(\omega)^+ y(\omega)$ is measurable.*

Proof. By [31], we have $\lim_{\lambda\to 0}(A^\top A+\lambda I)^{-1}A^\top = A^+$ for any matrix $A \in \mathbb{R}^{n\times n}$. Now, let $(\lambda_k)_k \subset \mathbb{R}$ be an arbitrary sequence converging to zero. Then, a continuity argument implies that the mapping $\omega \mapsto \zeta_k(\omega) := (T(\omega)^\top T(\omega) + \lambda_k I)^{-1} T(\omega)^\top y(\omega)$ is $\mathcal{F}$-measurable for all $k \in \mathbb{N}$. Since ζ is the pointwise limit of the sequence $(\zeta_k)_k$, this finishes the proof ∎

We now turn to the proof of Fact 3.4.

Proof. Since the stochastic oracles $\mathcal{G}$ and $\mathcal{H}$ are Carathéodory functions, it follows that $\mathcal{G}$ and $\mathcal{H}$ are jointly measurable, i.e., it holds that $\mathcal{G} \in \mathcal{B}(\mathbb{R}^n) \otimes \mathcal{X}$ and $\mathcal{H} \in \mathcal{B}(\mathbb{R}^n) \otimes \mathcal{X}$; see, e.g., [2, section 4.10]. We now prove the slightly extended claim

$$\mathsf{Z}_\mathsf{p}^k,\ \mathsf{Z}_\mathsf{n}^k \in \mathcal{F}_k, \quad \mathsf{Y}_{k+1},\ \mathsf{R}_{k+1} \in \hat{\mathcal{F}}_k, \quad \text{and} \quad \mathsf{X}^{k+1} \in \hat{\mathcal{F}}_k \tag{A.1}$$

inductively. Let us suppose that the statement (A.1) holds for $k-1$ with $k \in \mathbb{N}$. Then, due to $\mathsf{X}^k \in \hat{\mathcal{F}}_{k-1} \subset \mathcal{F}_k$ and using the $(\mathcal{F}_k, \mathcal{B}(\mathbb{R}^n) \otimes \mathcal{X})$-measurability of the functions $\omega \mapsto (\mathsf{X}^k(\omega), \mathsf{S}_i^k(\omega))$ and $\omega \mapsto (\mathsf{X}^k(\omega), \mathsf{T}_j^k(\omega))$ for $i \in [\mathbf{n}_k^\mathsf{g}]$ and $j \in [\mathbf{n}_k^\mathsf{h}]$, it follows that $\mathsf{G}_k(\mathsf{X}^k) \in \mathcal{F}_k$ and $\mathsf{H}_k(\mathsf{X}^k) \in \mathcal{F}_k$. Since the proximity operator $\operatorname{prox}_r^{\Lambda_k}$ is a continuous mapping, this also implies $u_{\mathsf{S}^k}^{\Lambda_k}(\mathsf{X}^k) \in \mathcal{F}_k$, $p_{\mathsf{S}^k}^{\Lambda_k}(\mathsf{X}^k) \in \mathcal{F}_k$, and $F_{\mathsf{S}^k}^{\Lambda_k}(\mathsf{X}^k) \in \mathcal{F}_k$ and thus we have $\mathsf{Z}_\mathsf{p}^k \in \mathcal{F}_k$. Moreover, by (C.1) and Lemma A.1, we can infer $\mathsf{Z}_\mathsf{n}^k \in \mathcal{F}_k$. Following the same line of reasoning and due to $\mathcal{F}_k \subset \hat{\mathcal{F}}_k$, it holds that $\mathsf{G}_{k+1}(\mathsf{Z}_\mathsf{n}^k) \in \hat{\mathcal{F}}_k$, $u_{\mathsf{S}^{k+1}}^{\Lambda_{k+1}}(\mathsf{Z}_\mathsf{n}^k) \in \hat{\mathcal{F}}_k$, $p_{\mathsf{S}^{k+1}}^{\Lambda_{k+1}}(\mathsf{Z}_\mathsf{n}^k) \in \hat{\mathcal{F}}_k$, and $\mathsf{F}_{k+1} = F_{\mathsf{S}^{k+1}}^{\Lambda_{k+1}}(\mathsf{Z}_\mathsf{n}^k) \in \hat{\mathcal{F}}_k$. Next, we study the measurability of the set P_k used in the definition of Y_{k+1}. Since lower semicontinuous functions are (Borel) measurable and we have $\mathsf{R}_k, \mathsf{X}^k, \mathsf{Z}_\mathsf{n}^k \in \hat{\mathcal{F}}_k$, the mappings $\mathsf{P}_1(\omega) := \|\mathsf{F}_{k+1}(\omega)\| - (\eta+\nu_k)\mathsf{R}_k(\omega)$ and $\mathsf{P}_2(\omega) := \psi(\mathsf{Z}_\mathsf{n}^k(\omega)) - \psi(\mathsf{X}^k(\omega)) - \beta \mathsf{R}_k(\omega)^{1-p}\|\mathsf{F}_{k+1}(\omega)\|^p$ are $\hat{\mathcal{F}}_k$-measurable. Hence, it follows that

$$P_k = \{\omega : r(\mathsf{Z}_\mathsf{n}^k(\omega)) < \infty\} \cap \{\omega : \mathsf{P}_1(\omega) \le \varepsilon_k^1\} \cap \{\omega : \mathsf{P}_2(\omega) \le \varepsilon_k^2\} \in \hat{\mathcal{F}}_k,$$

which shows that the binary variable Y_{k+1} is $\hat{\mathcal{F}}_k$-measurable. Using the representation (3.7), this implies $\mathsf{X}^{k+1} \in \hat{\mathcal{F}}_k$ and, due to $\mathsf{R}_{k+1} = \mathsf{Y}_{k+1}\|\mathsf{F}_{k+1}\| + (1 - \mathsf{Y}_{k+1})\mathsf{R}_k$, we

have $\mathsf{R}_{k+1} \in \hat{\mathcal{F}}_k$. Since the constant random variables X^0 and R_0 are trivially $\mathcal{F}_0$-measurable, we can use the same argumentation for the base case $k = 0$. This finishes the proof of Fact 3.4. ∎

We conclude with a remark on the existence of measurable selections of the multifunction $\mathcal{M}_k$. Due to [18], the generalized derivative $\partial \mathrm{prox}_r^{\Lambda_k} : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n\times n}$ is an upper semicontinuous, compact-valued function for all $k \in \mathbb{N}$. Hence, by [71, Lemma 4.4], $\partial \mathrm{prox}_r^{\Lambda_k}$ is (Borel) measurable for all k. As we have shown inductively, the function $\omega \mapsto u^{\Lambda_k}_{\mathsf{S}^k(\omega)}(\mathsf{X}^k(\omega))$ is $\mathcal{F}_k$-measurable and consequently, by [71, Lemma 4.5], the multifunction $\mathcal{D}_k : \Omega \rightrightarrows \mathbb{R}^{n\times n}$, $\mathcal{D}_k(\omega) := \partial \mathrm{prox}_r^{\Lambda_k}(u^{\Lambda_k}_{\mathsf{S}^k(\omega)}(\mathsf{X}^k(\omega)))$, is nonempty, closed-valued, and measurable with respect to $\mathcal{F}_k$ for all k. The Kuratowski–Ryll–Nardzewski selection theorem [59, 2] now implies that $\mathcal{D}_k$ admits an $\mathcal{F}_k$-measurable selection $\mathsf{D}_k : \Omega \to \mathbb{R}^{n\times n}$. Using $\mathsf{H}_k(\mathsf{X}^k) \in \mathcal{F}_k$, this implies that $\mathsf{M}_k := I - \mathsf{D}_k(I - \Lambda_k^{-1}\mathsf{H}_k(\mathsf{X}^k))$ is an $\mathcal{F}_k$-measurable selection of $\mathcal{M}_k$.

A.2. Proof of Lemma 3.7.

Proof. Since r is subdifferentiable at $p_s^\Lambda(x)$ with $\Lambda(u_s^\Lambda(x) - p_s^\Lambda(x)) \in \partial r(p_s^\Lambda(x))$ we have $r'(p_s^\Lambda(x); h) \geq \langle \Lambda F_s^\Lambda(x) - G_s(x), h\rangle$ for all $x, h \in \mathbb{R}^n$; see, e.g., [7, Proposition 17.17]. Now, using the convexity of $f - \frac{\mu_f}{2}\|\cdot\|^2$, the μ_r-strong convexity of r, (3.13) with $f \equiv 0$ and $\mu_f = 0$, and the descent lemma (3.3), it follows that

$$\begin{aligned}
\psi(y) &\geq f(x) + \langle \nabla f(x), y - x\rangle + 0.5\mu_f\|y - x\|^2 + r(p_s^\Lambda(x)) \\
&\quad + \langle \Lambda F_s^\Lambda(x) - G_s(x), y - p_s^\Lambda(x)\rangle + 0.5\mu_r\|y - p_s^\Lambda(x)\|^2 \\
&\geq \psi(p_s^\Lambda(x)) + \langle \nabla f(x) - G_s(x), F_s^\Lambda(x)\rangle - 0.5L\|F_s^\Lambda(x)\|^2 + 0.5\mu_f\|y - x\|^2 \\
&\quad + \|F_s^\Lambda(x)\|_\Lambda^2 + \langle \Lambda F_s^\Lambda(x) + \nabla f(x) - G_s(x), y - x\rangle + 0.5\mu_r\|y - p_s^\Lambda(x)\|^2 \\
&= \psi(p_s^\Lambda(x)) + \langle \nabla f(x) - G_s(x), F_s^\Lambda(x)\rangle + 0.5(\mu_r - L)\|F_s^\Lambda(x)\|^2 \\
&\quad + \|F_s^\Lambda(x)\|_\Lambda^2 + \langle (\Lambda + \mu_r I)F_s^\Lambda(x) + \nabla f(x) - G_s(x), y - x\rangle + 0.5\bar{\mu}\|y - x\|^2
\end{aligned}$$

for all $x, y \in \mathbb{R}^n$. Next, setting $e_s(x) := \|\nabla f(x) - G_s(x)\|$, $b_2 = (\lambda_M + \mu_r)^2\bar{\mu}^{-1}$, and applying Young's inequality twice for some $\tau, \alpha > 0$, we get

$$\begin{aligned}
&\left|\langle (\Lambda + \mu_r I)F_s^\Lambda(x) + \nabla f(x) - G_s(x), x^* - x\rangle\right| \\
&\qquad \leq \frac{1}{2}(1+\alpha)b_2\|F_s^\Lambda(x)\|^2 + \frac{(1+\tau(1+\alpha))\bar{\mu}}{2(1+\tau)(1+\alpha)}\|x - x^*\|^2 + \frac{(1+\tau)(1+\alpha)}{2\tau\alpha\bar{\mu}}\cdot e_s(x)^2.
\end{aligned}$$

Let us define $b_3(\alpha) := (2(1+\tau)(1+\alpha))^{-1}\alpha\bar{\mu}$. Setting $y = x^*$ in our first derived inequality, recalling $b_1 = L - 2\lambda_m - \mu_r$, and using the optimality of x^*, $\Lambda \succeq \lambda_m I$, Young's inequality, and the latter estimate, it holds that

$$\begin{aligned}
b_3(\alpha)\|x - x^*\|^2 &\leq 0.5(L - \mu_r + (1+\alpha)b_2)\|F_s^\Lambda(x)\|^2 - \|F_s^\Lambda(x)\|_\Lambda^2 \\
&\quad - \langle F_s^\Lambda(x), \nabla f(x) - G_s(x)\rangle + (4\tau b_3(\alpha))^{-1}e_s(x)^2 \\
&\leq 0.5(b_1 + \tau + (1+\alpha)b_2)\|F_s^\Lambda(x)\|^2 + (4\tau b_3(\alpha))^{-1}(2b_3(\alpha) + 1)e_s(x)^2.
\end{aligned}$$

Multiplying both sides with $b_3(\alpha)^{-1}$ and choosing $\alpha = (b_1 + b_2 + \tau)^{\frac{1}{2}}b_2^{-\frac{1}{2}}$ (this minimizes the factor in front of $\|F_s^\Lambda(x)\|^2$), we get (3.2) with $B_2(\tau) = (2b_3(\alpha) + 1)/(4\tau b_3(\alpha))$. If the full gradient is used, Young's inequality for $\tau > 0$ is not needed and we can set $B_1(\tau) \equiv B_1(0)$. ∎

Acknowledgments. The authors are grateful to Prof. Kim-Chuan Toh and two anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] N. Agarwal, B. Bullins, and E. Hazan, *Second-order stochastic optimization for machine learning in linear time*, J. Mach. Learn. Res., 18 (2017), pp. 1–40, http://jmlr.org/papers/v18/16-491.html.

[2] C. D. Aliprantis and K. C. Border, *Infinite Dimensional Analysis*, 3rd ed., Springer, Berlin, 2006.

[3] Z. Allen-Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, in Proceedings of the 49th Annual ACM SIGACT Symposium on the Theory of Computing, 2017, pp. 1200–1205.

[4] Z. Allen-Zhu, *Natasha: Faster non-convex stochastic optimization via strongly non-convex parameter*, in Proc. Mach. Learn. Res. 70, PMLR Press, 2017, pp. 89–97.

[5] Z. Allen-Zhu and E. Hazan, *Variance reduction for faster non-convex optimization*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 699–707.

[6] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization with sparsity-inducing penalties*, Found. Trends Mach. Learn., 4 (2011), pp. 1–106, https://doi.org/10.1561/2200000015.

[7] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math./Ouvrages Math. SMC, Springer, New York, 2011, https://doi.org/10.1007/978-1-4419-9467-7.

[8] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[9] A. S. Berahas, R. Bollapragada, and J. Nocedal, *An Investigation of Newton-Sketch and Subsampled Newton Methods*, https://arxiv.org/abs/1705.06211, 2017.

[10] R. Bhattacharya and E. C. Waymire, *A Basic Course in Probability Theory*, 2nd ed., Universitext, Springer, Cham, 2016, https://doi.org/10.1007/978-3-319-47974-3.

[11] J. A. Blackard and D. J. Dean, *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables*, Computers Electronics Agriculture, 24 (1999), pp. 131–151.

[12] R. Bollapragada, R. Byrd, and J. Nocedal, *Exact and inexact subsampled Newton methods for optimization*, IMA J. Numer. Anal., 39 (2019), pp. 545–578.

[13] A. Bordes, L. Bottou, and P. Gallinari, *SGD-QN: Careful quasi-Newton stochastic gradient descent*, J. Mach. Learn. Res., 10 (2009), pp. 1737–1754.

[14] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60 (2018), pp. 223–311, https://doi.org/10.1137/16M1080173.

[15] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, *On the use of stochastic Hessian information in optimization methods for machine learning*, SIAM J. Optim., 21 (2011), pp. 977–995, https://doi.org/10.1137/10079923X.

[16] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, *Sample size selection in optimization methods for machine learning*, Math. Program., 134 (2012), pp. 127–155, https://doi.org/10.1007/s10107-012-0572-5.

[17] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, *A stochastic quasi-Newton method for large-scale optimization*, SIAM J. Optim., 26 (2016), pp. 1008–1031, https://doi.org/10.1137/140954362.

[18] F. H. Clarke, *Optimization and Nonsmooth Analysis*, 2nd ed., Classics in Appl. Math. 5, SIAM, Philadelphia, 1990, https://doi.org/10.1137/1.9781611971309.

[19] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200, https://doi.org/10.1137/050626090.

[20] C. D. Dang and G. Lan, *Stochastic block mirror descent methods for nonsmooth and stochastic optimization*, SIAM J. Optim., 25 (2015), pp. 856–881, https://doi.org/10.1137/130936361.

[21] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 1646–1654.

[22] L. Deng and D. Yu, *Deep learning: Methods and applications*, Found. Trends Signal Process., 7 (2014), pp. 197–387, https://doi.org/10.1561/2000000039.

[23] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res., 12 (2011), pp. 2121–2159.

[24] M. A. Erdogdu and A. Montanari, *Convergence rates of sub-sampled Newton methods*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 3034–3042.

[25] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vol. II, Springer, New York, 2003.

[26] M. P. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM J. Sci. Comput., 34 (2012), pp. A1380–A1405, https://doi.org/10.1137/110830629.
[27] M. Fukushima and H. Mine, *A generalized proximal point algorithm for certain nonconvex minimization problems*, Internat. J. Systems Sci., 12 (1981), pp. 989–1000, https://doi.org/10.1080/00207728108963798.
[28] S. Ghadimi and G. Lan, *Stochastic first- and zeroth-order methods for nonconvex stochastic programming*, SIAM J. Optim., 23 (2013), pp. 2341–2368, https://doi.org/10.1137/120880811.
[29] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99, https://doi.org/10.1007/s10107-015-0871-8.
[30] S. Ghadimi, G. Lan, and H. Zhang, *Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization*, Math. Program., 155 (2016), pp. 267–305, https://doi.org/10.1007/s10107-014-0846-1.
[31] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 2013.
[32] R. Gower, D. Goldfarb, and P. Richtarik, *Stochastic block BFGS: Squeezing more curvature out of data*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 1869–1878.
[33] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, *Result analysis of the NIPS* 2003 *feature selection challenge*, in Advances in Neural Information Processing Systems, 17, MIT Press, MIT Press, Cambridge, MA, 2004, pp. 545–552, http://papers.nips.cc/paper/2728-result-analysis-of-the-nips-2003-feature-selection-challenge.pdf.
[34] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer Ser. Statist., Springer, New York, https://doi.org/10.1007/978-0-387-84858-7, 2009.
[35] K. Jiang, D. Sun, and K.-C. Toh, *A partial proximal point algorithm for nuclear norm regularized matrix least squares problems*, Math. Program. Comput., 6 (2014), pp. 281–325, https://doi.org/10.1007/s12532-014-0069-8.
[36] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2013, pp. 315–323.
[37] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, 521 (2015), pp. 436–444, https://doi.org/10.1038/nature14539.
[38] Y. LeCun, C. Cortes, and C. J. C. Burges, *The MNIST Database of Handwritten Digits*, http://yann.lecun.com/exdb/mnist (2010).
[39] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim., 24 (2014), pp. 1420–1443, https://doi.org/10.1137/130921428.
[40] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, *RCV1: A new benchmark collection for text categorization research*, J. Mach. Learn. Res., 5 (2004), pp. 361–397.
[41] H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 3384–3392.
[42] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual ICML, New York, 2009, pp. 689–696, https://doi.org/10.1145/1553374.1553463.
[43] L. Mason, J. Baxter, P. Bartlett, and M. Frean, *Boosting algorithms as gradient descent in function space*, in Proceedings of NIPS'99, 1999, pp. 512–518, http://dl.acm.org/citation.cfm?id=3009657.3009730.
[44] A. Milzarek, *Numerical Methods and Second Order Theory for Nonsmooth Problems*, Ph.D. dissertation, Technische Universität München, 2016.
[45] A. Milzarek and M. Ulbrich, *A semismooth Newton method with multidimensional filter globalization for l_1-optimization*, SIAM J. Optim., 24 (2014), pp. 298–333, https://doi.org/10.1137/120892167.
[46] A. Milzarek, X. Xiao, S. Cen, Z. Wen, and M. Ulbrich, *A Stochastic Semismooth Newton Method for Nonsmooth Nonconvex Optimization*, https://arxiv.org/abs/1803.03466, 2018.
[47] A. Mokhtari and A. Ribeiro, *RES: regularized stochastic BFGS algorithm*, IEEE Trans. Signal Process., 62 (2014), pp. 6089–6104, https://doi.org/10.1109/TSP.2014.2357775.
[48] J.-J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[49] P. Moritz, R. Nishihara, and M. Jordan, *A linearly-convergent stochastic L-BFGS algorithm*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, 2016, pp. 249–258.

[50] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2008), pp. 1574–1609, https://doi.org/10.1137/070704277.

[51] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim., 1 (2014), pp. 127–239, https://doi.org/10.1561/2400000003.

[52] P. Patrinos, L. Stella, and A. Bemporad, *Forward-Backward Truncated Newton Methods for Convex Composite Optimization*, https://arxiv.org/abs/1402.6655, 2014.

[53] M. Pilanci and M. J. Wainwright, *Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence*, SIAM J. Optim., 27 (2017), pp. 205–245, https://doi.org/10.1137/15M1021106.

[54] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244, https://doi.org/10.1287/moor.18.1.227.

[55] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Program., 58 (1993), pp. 353–367, https://doi.org/10.1007/BF01581275.

[56] S. J. Reddi, A. Hefny, S. Sra, B. Póczos, and A. J. Smola, *Stochastic variance reduction for nonconvex optimization*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 314–323.

[57] S. J. Reddi, S. Sra, B. Póczos, and A. J. Smola, *Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization*, in Advance in Neural Information Processing Systems 29, MIT Press, Cambridge, MA, 2016, pp. 1145–1153, http://papers.nips.cc/paper/pdf.

[58] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Stat., 22 (1951), pp. 400–407.

[59] R. T. Rockafellar, *Integral Functionals, Normal Integrands and Measurable Selections*, Lecture Notes in Math. 543, Springer, New York, 1976, pp. 157–207.

[60] F. Roosta-Khorasani and M. W. Mahoney, *Sub-sampled Newton methods*, Math. Program., 174 (2019), pp. 293–326.

[61] J. Schmidhuber, *Deep learning in neural networks: An overview*, Neural Netw., 61 (2015), pp. 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.

[62] M. Schmidt, N. Le Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Math. Program., 162 (2017), pp. 83–112, https://doi.org/10.1007/s10107-016-1030-6.

[63] N. N. Schraudolph, J. Yu, and S. Günter, *A stochastic quasi-Newton method for online convex optimization*, in Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, Vol. 2, 2007, pp. 436–443, http://proceedings.mlr.press/v2/schraudolph07a.html.

[64] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, New York, 2014.

[65] Z. Shi and R. Liu, *Large scale optimization with proximal stochastic Newton-type gradient descent*, in Mach. Learn. Knowl. Disc. Databases 9284, Springer, New York, 2015, pp. 691–704, https://doi.org/10.1007/978-3-319-23528-8_43.

[66] L. Stella, A. Themelis, and P. Patrinos, *Forward–backward quasi-Newton methods for nonsmooth optimization problems*, Comput. Optim. Appl., 67 (2017), pp. 443–487, https://doi.org/10.1007/s10589-017-9912-y.

[67] D. Sun and J. Sun, *Semismooth matrix-valued functions*, Math. Oper. Res., 27 (2002), pp. 150–169, https://doi.org/10.1287/moor.27.1.150.342.

[68] A. Themelis, M. Ahookhosh, and P. Patrinos, *On the Acceleration of Forward-Backward Splitting via an Inexact Newton Method*, https://arxiv.org/abs/1811.02935, 2018.

[69] R. Tomioka, T. Suzuki, and M. Sugiyama, *Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation*, J. Mach. Learn. Res., 12 (2011), pp. 1537–1586.

[70] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423, https://doi.org/10.1007/s10107-007-0170-0.

[71] M. Ulbrich, *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim., 13 (2002), pp. 805–842 (2003), https://doi.org/10.1137/S1052623400371569.

[72] J. Wang and T. Zhang, *Improved Optimization of Finite Sums with Minibatch Stochastic Variance Reduced Proximal Iterations*, https://arxiv.org/abs/1706.07001, 2017.

[73] X. Wang, S. Ma, D. Goldfarb, and W. Liu, *Stochastic Quasi-Newton methods for nonconvex stochastic optimization*, SIAM J. Optim., 27 (2017), pp. 927–956, https://doi.org/10.1137/15M1053141.

[74] *A Marketing Dataset*, Causality Workbench Team, http://www.causality.inf.ethz.ch/data/CINA.html (2008).
[75] *A Pharmacology Dataset*, Causality Workbench Team, http://www.causality.inf.ethz.ch/data/SIDO.html (2008).
[76] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493, https://doi.org/10.1109/TSP.2009.2016892.
[77] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM J. Optim., 24 (2014), pp. 2057–2075, https://doi.org/10.1137/140961791.
[78] X. Xiao, Y. Li, Z. Wen, and L. Zhang, *A regularized semi-smooth Newton method with projection steps for composite convex programs*, J. Sci. Comput., 76 (2018), pp. 364–389, https://doi.org/10.1007/s10915-017-0624-3.
[79] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney, *Newton-Type Methods for Non-Convex Optimization Under Inexact Hessian Information*, https://arxiv.org/abs/1708.07164, 2017.
[80] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney, *Second-Order Optimization for Non-Convex Machine Learning: An Empirical Study*, https://arxiv.org/abs/1708.07827, 2017.
[81] P. Xu, J. Yang, F. Roosta-Khorasani, C. Ré, and M. W. Mahoney, *Sub-sampled Newton methods with non-uniform sampling*, in Advances in Neural Information Processing Systems, 2016, pp. 3000–3008.
[82] Y. Xu and W. Yin, *Block stochastic gradient iteration for convex and nonconvex optimization*, SIAM J. Optim., 25 (2015), pp. 1686–1716, https://doi.org/10.1137/140983938.
[83] Z. Yao, P. Xu, F. Roosta-Khorasani, and M. W. Mahoney, *Inexact Non-Convex Newton-Type Methods*, https://arxiv.org/abs/1802.06925, 2017.
[84] H. Ye, L. Luo, and Z. Zhang, *Approximate Newton methods and their local convergence*, in Proc. Mach. Learn. Res. 70, PMLR Press, 2017, pp. 3931–3939, http://proceedings.mlr.press/v70/ye17a.html.