SIAM J. MATH. DATA SCI.
Vol. 6, No. 4, pp. 926–952

Stochastic Variance-Reduced Majorization-Minimization Algorithms*

Duy Nhat Phan†, Sedi Bartz‡, Nilabja Guha‡, and Hung M. Phan‡

Abstract. We study a class of nonconvex nonsmooth optimization problems in which the objective is a sum of two functions; one function is the average of a large number of differentiable functions, while the other function is proper, lower semicontinuous. Such problems arise in machine learning and regularized empirical risk minimization applications. However, nonconvexity and the large-sum structure are challenging for the design of new algorithms. Consequently, effective algorithms for such scenarios are scarce. We introduce and study three stochastic variance-reduced majorization-minimization (MM) algorithms, combining the general MM principle with new variance-reduced techniques. We provide almost surely subsequential convergence of the generated sequence to a stationary point. We further show that our algorithms possess the best-known complexity bounds in terms of gradient evaluations. We demonstrate the effectiveness of our algorithms on sparse binary classification problems, sparse multiclass logistic regressions, and neural networks by employing several widely used and publicly available data sets.

Key words. majorization-minimization, surrogate functions, variance reduction techniques

MSC codes. 90C26, 49M37, 65K05, 15A23, 15A83

DOI. 10.1137/23M1571836

1. Introduction.

We focus on a class of nonsmooth and nonconvex problems of the form

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

where $r : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper and lower semicontinuous function, d is a positive integer, and f has a large-sum structure, that is,

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1.2}$$

where n is a positive integer and $f_i : \mathbb{R}^d \to \mathbb{R}$ is differentiable (possibly nonconvex). The large-sum structure captures, in particular, *regularized empirical risk,* where f_i represents a loss function on a single data point and r is often a nonsmooth (possibly nonconvex) function that regularizes the promotion of sparse solutions, such as ℓ_1-norm, Geman [17], MCP [52],

*Received by the editors May 9, 2023; accepted for publication (in revised form) June 24, 2024; published electronically October 8, 2024.
https://doi.org/10.1137/23M1571836
Funding: The first, second, and fourth authors are partially supported by a seed grant from the Kennedy College of Sciences, University of Massachusetts Lowell. The second author is partially supported by a Simons Foundation Collaboration Grant for Mathematicians. The first and fourth authors are partially supported by a gift from Autodesk, Inc. The third author is partially supported by National Science Foundation grant NSF DMS 2015460.

†University of Dayton Research Institute, University of Dayton, Dayton, OH 45469 USA (dphan1@udayton.edu).
‡Department of Mathematics and Statistics, University of Massachusetts Lowell, Lowell, MA 01854 USA (sedi_bartz@uml.edu, nilabja_guha@uml.edu, hung_phan@uml.edu).

log-sum penalty [8], and exponential concave penalty [7]. Thus, problem (1.1) models a broad range of optimization problems from convex problems (i.e., f_i and r are convex functions), such as logistic regression, to fully nonconvex problems (i.e., both f_i and r are nonconvex) such as optimizing deep neural networks. Since nonconvex optimization became indispensable in recent advances in machine learning models, we focus our attention on the fully nonconvex scenario in problem (1.1). Specifically, we are interested in the case where the number of components n is extremely large since it is a key challenge in the era of big data applications.

1.1. Notation. We follow standard notation as in [4, 46]. For any real number a, $\lfloor a \rfloor$ denotes the largest integer less than or equal to a while $\lceil a \rceil$ denotes the smallest integer greater than or equal to a. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For $k \in \mathbb{N}$ and $b \in [n]$, we denote by I_k the index batch which is a *list* of (possibly repeated) indices $(i_1, i_2, \ldots, i_b)$ of fixed size b where each index i_j is independently and randomly chosen from $[n]$. We refer to I_k as a batch of size b. Let $x^0 \in \mathbb{R}^d$. With a sequence $x^k \in \mathbb{R}^d$ and a sequence of batches I_k, we associate $x_i^{-1} = x^0$ for all $i \in [n]$, and inductively, having defined x_i^{k-1}, we set

$$x_i^k = \begin{cases} x^k, & i \in I_k, \\ x_i^{k-1} & \text{otherwise.} \end{cases} \tag{1.3}$$

In other words, x_i^k is updated to x^k if and only if $i \in I_k$. Denote $\xi_k = I_k$ for `MM-SAGA` and $\xi_k = (I_k, d_k)$ for `MM-SVRG` and `MM-SARAH` (see descriptions of `MM-SAGA`, `MM-SVRG`, and `MM-SARAH` in section 3), where $d_k \in \{0, 1\}$.

Let Ω be the sample space of all sequences $\omega = \{\omega_k\}_{k=0}^{\infty}$, $\omega_k = \xi_k$. We define a sequence of σ-algebras $\mathcal{F}_k$ on Ω as follows. Fix $k \geq 1$. For each $(\xi_0, \xi_1, \ldots, \xi_{k-1})$, define the cylinder set

$$C(\xi_0, \xi_1, \ldots, \xi_{k-1}) := \{\omega \in \Omega : w_0 = \xi_0, \omega_1 = \xi_1, \ldots, \omega_{k-1} = \xi_{k-1}\}.$$

Denote by C^k the collection of all cylinder set $C(\xi_0, \xi_1, \ldots, \xi_{k-1})$. Now denote $\mathcal{F}_k := \sigma(C^k)$ the σ-algebra generated by C^k, and $\mathcal{F} := \sigma\left(\cup_{k=1}^{\infty} C^k\right)$. Clearly, the σ-algebra sequence $\{\mathcal{F}_k\}$ satisfies $\mathcal{F}_k \subset \mathcal{F}_{k+1} \subset \mathcal{F}$ for all $k \geq 1$. Thus, the σ-algebra $\mathcal{F}$ is associated with a probability measure p forming the probability space $(\Omega, \mathcal{F}, P)$. We use $\mathbb{E}_k$ as shorthand for the conditional expectation operator $\mathbb{E}[\cdot|\mathcal{F}_k]$ given $\mathcal{F}_k$.

Throughout, $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^d$ with induced norm $\|\cdot\|$ defined by $\|x\| = \sqrt{\langle x, x \rangle}, x \in \mathbb{R}^d$. We set $\mathbb{R}_+ = \{r \in \mathbb{R} : r \geq 0\}$. For a nonempty closed set $\mathcal{C} \subset \mathbb{R}^d$, the distance of x from $\mathcal{C}$ is defined by $\operatorname{dist}(x, \mathcal{C}) = \inf_{y \in \mathcal{C}} \|x - y\|$. An extended-real-valued function $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is said to be proper if its domain, the set $\operatorname{dom} g = \{x \in \mathbb{R}^d : g(x) < +\infty\}$, is nonempty. We say that g is lower semicontinuous if, at each $x^* \in \mathbb{R}^d$,

$$g(x^*) \leq \liminf_{x \to x^*} g(x).$$

Let $\alpha \in \mathbb{R}$. We say that g is α-convex if $g - \frac{\alpha}{2}\|\cdot\|^2$ is convex, equivalently, if

$$g((1-\lambda)x + \lambda y) \leq (1-\lambda)g(x) + \lambda g(y) - \frac{\alpha}{2}\lambda(1-\lambda)\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^d, \lambda \in [0, 1].$$

In particular, g is convex if and only if g is 0-convex. In the case where the function g is α-convex, we say that g is α-strongly convex if $\alpha > 0$ and we say that g is α-weakly convex if $\alpha < 0$. Finally, ∂f denotes the subdifferential of the function f (see Definition 2.1).

1.2. Motivation and related work. In the convex setting, a standard method for solving the noncomposite form ($r = 0$) of problem (1.1) is the gradient descent method (GD). Given an initial point $x^0 \in \mathbb{R}^d$, the iterative step of the GD method computes x^{k+1} by

$$x^{k+1} := x^k - \eta_k \nabla f(x^k),$$

where $\eta_k > 0$ is a stepsize and k is a nonnegative integer. In (1.2), if the number of components n is very large, each iteration of the GD method becomes extremely expensive since it requires the computation of the gradient for all of the components f_i. An effective alternative is the standard stochastic gradient method (SGD) [44]. In this case, in each iteration, the SGD draws randomly i_k from $[n]$ and updates x^{k+1} by

$$x^{k+1} := x^k - \eta_k \nabla f_{i_k}(x^k).$$

The advantage of the SGD method is that in each iteration, it only evaluates the gradient of a single component function. Consequently, the computational cost per iteration is only $1/n$ of that of the full step in the GD method. However, due to the *variance,* inadvertently generated by random sampling, the SGD method converges much slower than the full GD method. Fortunately, we can overcome this drawback by variance reduction techniques, utilizing information regarding the gradient from previous iterations to construct a better estimation of the gradient at the current step. To date, some of the most widely applied variance reduction methods in the literature are the *stochastic average gradient algorithm* (SAGA) [12], the *stochastic variance-reduced gradient* (SVRG) [22], and the *stochastic average gradient* (SAG) [49]. We note in passing that SAGA is an unbiased version of SAG. Variance reduction methods inherit the advantage of low iteration cost of the SGD method while providing similar convergence rates of the full GD method in convex settings.

Thus far, however, only several variance reduction methods have been developed in order to deal with nonconvex optimization problems possessing the large-sum structure. Furthermore, these methods mainly focus on special cases of (1.1), where $r = 0$, such as [1, 2, 38], or where r is convex, such as [21, 30]. For the fully nonconvex problem with an extremely large value of n (such as we study here), such developments become even more challenging. Consequently, research in this direction is sparse. Several recent studies, such as [27, 28, 31, 50], promote stochastic methods based on the difference-of-convex (DC) algorithm, developed in [29, 42], or majorization-minimization (MM), developed in [26]. In particular, if f_i is L-smooth and r has a DC structure, that is, $r = r_1 - r_2$ with r_1 being proper lower semicontinuous convex and r_2 being convex, problem (1.1) can be reformulated as a DC program

$$\min_{x \in \mathbb{R}^d} G(x) - H(x), \tag{1.4}$$

where $G(x) = \frac{\mu}{2}\|x\|^2 + r_1(x)$ and $H(x) = \frac{\mu}{2}\|x\|^2 - f(x) + r_2(x)$ are convex functions with $\mu \geq L$. The classic DC algorithm (DCA) linearizes the function H iteratively and updates x^{k+1} by

$$x^{k+1} = \operatorname*{argmin}_{x \in \mathbb{R}^d} \frac{\mu}{2}\|x\|^2 + r_1(x) - \left\langle \mu x^k - \nabla f(x^k) + y^k, x \right\rangle \tag{1.5}$$

for some $y^k \in \partial r_2(x^k)$.

In [27, 50], a stochastic version of DCA, named SDCA, was studied based on the idea of incrementally linearizing the components $\frac{\mu}{2}\|x\|^2 - f_i(x) + r_2(x)$ of H. More specifically, $\mu x^k - \nabla f(x^k) + y^k$ is replaced by the so-called SAG estimator

$$\widetilde{\nabla}_{\text{SAG}} H(x^k) := \frac{1}{n}\sum_{i=1}^{n}\left[\mu x_i^k - \nabla f_i(x_i^k) + y_i^k\right], \quad \text{where } y_i^k \in \partial r_2(x_i^k). \tag{1.6}$$

Consequently, subproblem (1.5) is replaced by

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^d} \frac{\mu}{2}\|x\|^2 + r_1(x) - \left\langle \widetilde{\nabla}_{\text{SAG}} H(x^k), x\right\rangle, \tag{1.7}$$

where $y_i^k \in \partial r_2(x_i^k)$ for all i.

Recently, Le Thi et al. [28] developed stochastic DC algorithms, named DCA-SAGA and DCA-SVRG, based on the so-called SAGA and SVRG estimators for the problem in which r is a DC function. Specifically, DCA-SAGA applied to (1.4) successively replaces $\mu x^k - \nabla f(x^k)$ in DCA's subproblem (1.5) by the SAGA stochastic gradient estimate,

$$\begin{aligned}\widetilde{\nabla}_{\text{SAGA}}\left(\frac{\mu}{2}\|\cdot\|^2 - f\right)(x^k) := \frac{1}{b}\sum_{i\in I_k}\left[\mu x^k - \nabla f_i(x^k) - \mu x_i^{k-1} + \nabla f_i(x_i^{k-1})\right]\\ + \frac{1}{n}\sum_{i=1}^{n}\left[\mu x_i^{k-1} - \nabla f_i(x_i^{k-1})\right],\end{aligned}$$

which, when combined with (1.5), implies that

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^d} \frac{\mu}{2}\|x\|^2 + r_1(x) - \left\langle \widetilde{\nabla}_{\text{SAGA}}\left(\frac{\mu}{2}\|\cdot\|^2 - f\right)(x^k) + y^k, x\right\rangle. \tag{1.8}$$

In comparison, DCA-SVRG replaces $\mu x^k - \nabla f(x^k)$ in DCA's subproblem by the SVRG stochastic gradient estimate:

$$\begin{aligned}\widetilde{\nabla}_{\text{SVRG}}\left(\frac{\mu}{2}\|\cdot\|^2 - f\right)(x^k) &:= \frac{1}{b}\sum_{i\in I_k}\left[\mu x^k - \nabla f_i(x^k) - \mu\tilde{x}^k + \nabla f_i(\tilde{x}^k)\right] + \mu\tilde{x}^k - \nabla f(\tilde{x}^k)\\ &= \mu x^k - \widetilde{\nabla}_{\text{SVRG}} f(x^k),\end{aligned} \tag{1.9}$$

where $\tilde{x}^k = x^k$ if $k \in m\mathbb{N}$, and $\tilde{x}^{k-1}$ otherwise, where m is a fixed positive integer. Consequently, the corresponding subproblem for DCA-SVRG is

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^d} \frac{\mu}{2}\|x - x^k\|^2 + r_1(x) - \left\langle -\widetilde{\nabla}_{\text{SVRG}} f(x^k) + y^k, x\right\rangle. \tag{1.10}$$

In general, problem (1.1) can be solved by an MM principle such that at each iteration, a complex objective function is approximated by an upper bound which is created around the current iteration and which can be minimized effectively. This step is called the *majorization* step. The minimum of this upper bound (the minimization step) is then used to sequentially

create another, hopefully tighter, upper bound (another majorization step) to be minimized. More specifically, the MM scheme applied to (1.1) computes x^{k+1} by

$$x^{k+1} \in \operatorname*{argmin}_{x\in\mathbb{R}^d} \frac{\mu}{2}\|x-x^k\|^2 + \left\langle \nabla f(x^k), x-x^k \right\rangle + u(x,x^k), \tag{1.11}$$

where $u(x,x^k)$ is an upper bound (or surrogate; see Definition 3.1) of r at x^k. Indeed, various deterministic approaches can be interpreted from the MM point of view such as proximal or gradient-based methods [3, 6, 11, 18, 23, 32, 35, 47], and expectation-maximization algorithms in statistics [13, 33]. To date, many extensions of MM have been developed, e.g., [9, 10, 19, 20, 24, 40, 43]. However, only a few algorithms have been applied in the large-sum structure settings. In particular, Mairal [31] introduced the *minimization of incremental surrogate* (MISO) that applies to problem (1.1) and successively updates x^{k+1} by

$$x^{k+1} \in \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\mu}{2}\|x-x_i^k\|^2 + \left\langle \nabla f_i(x_i^k), x-x_i^k \right\rangle + f_i(x_i^k) + u(x,x_i^k)\right], \tag{1.12}$$

where $u(\cdot,x_i^k)$ is a surrogate of r at x_i^k. However, in order to study asymptotic convergence, Mairal [31] employs a strong assumption, namely, that the approximation errors $h(x,x_i^k) := u(x,x_i^k) - r(x)$ are L-smooth in x. It is noteworthy that although MISO was inspired by SAG (see (1.6)), it does not recover SAG as a special case for smooth and composite convex optimization.

To the best of our knowledge, the incorporation of new stochastic gradient estimators SAGA, SVRG, and the *stochastic recursive gradient* (SARAH) [37] into MM algorithms for solving the nonconvex problem (1.1) was not previously studied.

1.3. Contribution and organization. For solving problem (1.1) in the case where it incorporates a large-sum structure and nonconvexity of the objective, we introduce three *stochastic variance-reduced majorization-minimization* (SVRMM) algorithms: MM-SAGA, MM-SVRG, and MM-SARAH. Unlike MISO, the SVRMM iterates on r and the large-sum f separately. In particular, at each iteration, MM-SAGA, MM-SVRG, and MM-SARAH replace the full gradient of f in the deterministic MM by stochastic gradient estimators employing SAGA, loopless SVRG, and loopless SARAH, respectively. It is important to note that MM-SAGA updates the proximal term $\frac{\mu}{2}\|x-x^k\|^2$ at the current iterate x^k in the same manner as in the MM scheme. This distinguishes MM-SAGA from DCA-SAGA when applied to the DC program (1.4), where the latter's update rule consists of the proximal $\frac{\mu}{2}\|x-\bar{x}^k\|^2$ at $\bar{x}^k = \frac{1}{b}\sum_{i\in I_k}[x^k - x_i^{k-1}] + \frac{1}{n}\sum_{i=1}^n x_i^{k-1}$. In addition, we point out that MM-SVRG employs the loopless SVRG estimator, which was shown to have superior performance [25] when compared to the classic estimator technique SVRG, employed in DCA-SVRG.

Under mild assumptions, we analyze the subsequential convergence for the generated sequence of the SVRMM algorithms. More concretely, we show that each limit point of the generated sequence is a stationary point of problem (1.1). Meanwhile, Le Thi et al. [28] showed that each limit point x^* of the generated sequence by their algorithms DCA-SAGA and DCA-SVRG is a DC critical point of $G-H$, i.e., $\partial G(x^*)\cap\partial H(x^*)\neq\emptyset$, which is weaker than the stationary point property, since $\partial F \subset \partial G - \partial H$. Furthermore, we show that our algorithms have $\mathcal{O}(k^{-1/2})$ convergence rate with respect to the proximity to a stationary point. In order to obtain an ϵ-stationary point, we show that MM-SAGA and MM-SVRG have

Table 1
SVRMM versus stochastic-based DCA.

Method	Requirement	Stepsize	Batch size	Complexity	Reference
DCA-SAGA [28]	$\frac{n\sqrt{n+b}}{b^2} \le \frac{1}{4}\frac{2\mu-L}{\mu+L}$	$\frac{1}{2L}$	$b = \lceil 2^{1/4} 2 n^{3/4} \rceil$	$\mathcal{O}(n^{3/4}/\epsilon^2)$	Theorem 5(5)
DCA-SVRG [28]	$\frac{m}{\sqrt{b}} \le \frac{1}{4\sqrt{e-1}}\frac{2\mu-L}{\mu+L}$	$\frac{1}{2L}$	$b = \lfloor n^{2/3} \rfloor$, $m = \lfloor \frac{\sqrt{b}}{4\sqrt{e-1}} \rfloor$	$\mathcal{O}(n^{2/3}/\epsilon^2)$	Theorem 2(4)
MM-SAGA (new)	$\frac{n}{b^{3/2}} \le \frac{1}{4}\frac{2\mu-L}{L}$	$\frac{1}{L}$	$b = \lceil 2^{5/3} n^{2/3} \rceil$	$\mathcal{O}(n^{2/3}/\epsilon^2)$	Corollary 4.10(a)
MM-SVRG (new)	$\frac{m}{\sqrt{b}} \le \frac{1}{4}\frac{2\mu-L}{L}$	$\frac{1}{L}$	$b = \lfloor n^{2/3} \rfloor$, $m = \frac{\sqrt{b}}{4\sqrt{2}}$	$\mathcal{O}(n^{2/3}/\epsilon^2)$	Corollary 4.10(b)
MM-SARAH (new)	$\frac{\sqrt{m}}{\sqrt{b}} < \frac{1}{2}\frac{2\mu-L}{L}$	$\frac{1}{L}$	$b = \lfloor n^{1/2} \rfloor$, $m = \frac{b}{8}$	$\mathcal{O}(n^{1/2}/\epsilon^2)$	Corollary 4.10(c)

complexity of $\mathcal{O}(n^{2/3}/\epsilon^2)$ while MM-SARAH has complexity of $\mathcal{O}(n^{1/2}/\epsilon^2)$ in terms of gradient evaluations. That is, our results are superior to those of DCA-SAGA and DCA-SVRG which have complexity of $\mathcal{O}(n^{3/4}/\epsilon^2)$ and $\mathcal{O}(n^{2/3}/\epsilon^2)$, respectively, for finding an ϵ-DC critical point. Another advantage of our methods is that we do not impose L-smoothness on the function r; it may be nonsmooth and nonconvex, but rather, in order to obtain our results, the stochastic DCA based algorithms require that the second DC component of r, namely, the component r_2 in the decomposition $r = r_1 - r_2$, is L-smooth. Table 1 contains a comparison between our new methods and DCA-SAGA and DCA-SVRG for solving nonsmooth nonconvex optimization problems, in terms of the requirement on our stepsize $1/\mu$ and our batchsize b, in order to achieve the corresponding complexity.

In Table 1, the stepsizes and batchsizes of DCA-SAGA and DCA-SVRG are taken from [28], and the stepsizes and batchsizes for MM-SAGA, MM-SVRG, and MM-SARAH are chosen in order to achieve the optimal order of complexity. We also note in passing that by setting the stepsize $\frac{1}{\mu} = \frac{1}{2L}$, we obtain the same complexity in all three SVRMM algorithms.

Finally, we apply our algorithms to solve three problems in order to illustrate their applicability and efficiency: sparse binary classification with nonconvex loss and regularizer, sparse multiclass logistic regression with nonconvex regularizer, and feedforward neural network training.

The paper is organized as follows. In section 2, we present basic concepts and properties in nonconvex optimization. In section 3, we present our algorithms. We analyze the convergence properties of our methods in section 4. In section 5, we provide a demonstration by numerical experiments, followed by conclusions in section 6.

2. Preliminaries.

Definition 2.1 (Fréchet and limiting subdifferential [46, Definition 8.3]). *Let $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function.*

(a) *For each $x \in \operatorname{dom} g$, we denote by $\hat{\partial} g(x)$ the Fréchet subdifferential of g at x. It contains all of the vectors $v \in \mathbb{R}^d$ which satisfy*

$$\liminf_{y \neq x, y \to x} \frac{1}{\|y - x\|} \left(g(y) - g(x) - \langle v, y - x \rangle \right) \ge 0.$$

If $x \notin \operatorname{dom} g$, we set $\hat{\partial} g(x) = \emptyset$.

(b) *The limiting subdifferential* $\partial g(x)$ *of* g *at* $x \in \operatorname{dom} g$ *is defined by*

$$\partial g(x) := \left\{ v \in \mathbb{R}^d : \exists x^k \to x,\, g(x^k) \to g(x),\, v^k \in \hat{\partial} g(x^k),\, v^k \to v \right\}.$$

If g is convex, then the Fréchet and limiting subdifferentials coincide with the convex subdifferential:

$$\partial g(x) = \left\{ v : g(y) \geq g(x) + \langle y - x, v \rangle \, \forall y \in \mathbb{R}^d \right\}.$$

Definition 2.2 (L-Lipschitz). *A mapping* $T : D \subset \mathbb{R}^d \to \mathbb{R}^k$ *is said to be* L*-Lipschitz,* $L \geq 0$*, if*

$$\forall x, y \in D, \quad \|Tx - Ty\| \leq L\|x - y\|.$$

Definition 2.3 (L-smooth). *Let* $g : \mathbb{R}^d \to \mathbb{R}$*. We say that* g *is* L*-smooth if it is everywhere differentiable and its gradient,* ∇g*, is* L*-Lipschitz.*

We now recall several useful basic facts.

Lemma 2.4. *Let* g *and* h *be proper and lower semicontinuous. Then,*

(a) $0 \in \partial g(\bar{x})$ *if* g *attains a local minimum at* $\bar{x} \in \operatorname{dom} g$*;*

(b) $\partial f(\bar{x}) = \partial g(\bar{x}) + \nabla h(\bar{x})$ *if* $f = g + h$ *and* h *is continuously differentiable in a neighborhood of* $\bar{x}$*;*

(c) $g(x) \geq g(y) + \rho\|x - y\|^2$ *if* g *is convex and* y *is defined by*

$$y = \operatorname*{argmin}_z \left\{ g(z) + \frac{\rho}{2}\|z - x\|^2 \right\};$$

(d) $|g(x) - g(y) - \langle \nabla g(y), x - y \rangle| \leq \frac{L}{2}\|x - y\|^2$ *for all* $x, y \in \mathbb{R}^d$ *if* g *is* L*-smooth.*

Proof. (a) See, e.g., [46, Theorem 8.15]. (b) See, e.g., [46, Exercise 8.8]. (c) See, e.g., [5, Theorem 6.39]. (d) See, e.g., [36, Lemma 1.2.3]. ∎

Definition 2.5 (ϵ-stationary point). *A point* x^* *is said to be an* ϵ*-stationary point of* g *if*

$$\operatorname{dist}(0, \partial g(x^*)) \leq \epsilon.$$

In particular, we say that x^* *is a stationary point of* g *if it is a* 0*-stationary point.*

The following lemma is a fundamental tool in our convergence analysis.

Lemma 2.6 (supermartingale convergence [45, Theorem 1]). *Let* $\{Y_k\}, \{Z_k\}$*, and* $\{W_k\}$ *be three sequences of random variables and let* $\{\mathcal{F}_k\}$ *be a sequence of sub-*σ*-algebras such that* $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ *for all* k*. Assume that, almost surely,*

(a) *for each* k*,* $\{Y_k\}, \{Z_k\}$*, and* W_k *are nonnegative* $\mathcal{F}_k$*-measurable random variables;*

(b) $\mathbb{E}[Y_{k+1}|\mathcal{F}_k] \leq Y_k - Z_k + W_k$ *for each* k*;*

(c) $\sum_{k=0}^{+\infty} W_k < +\infty$.

Then, $\sum_{k=0}^{+\infty} Z_k < +\infty$*, and* $\{Y_k\}$ *converges to a nonnegative random variable, almost surely.*

3. Stochastic variance-reduced majorization-minimization.

In this section, we introduce three `SVRMM` algorithms for solving problem (1.1). To this end, we define surrogate functions as follows.

Definition 3.1 (tangent majorant function). *A function $u:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$ is said to be a tangent majorant function of $r:\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$ if*

(a) $u(y,y)=r(y)$ *for all* $y\in\mathbb{R}^d$;

(b) $u(x,y)\geq r(x)$ *for all* $x,y\in\mathbb{R}^d$.

We introduce our first SVRMM algorithm, which we call MM-SAGA. It combines the deterministic MM and the SAGA-style of stochastic gradient update. In particular, we replace the full gradient $\nabla f(x^k)$ in the deterministic MM (1.11) with the stochastic gradient estimate $\widetilde{\nabla}_{\text{SAGA}}f(x^k)$ as follows:

$$\widetilde{\nabla}_{\text{SAGA}}f(x^k)=\frac{1}{b}\sum_{i\in I_k}\left(\nabla f_i(x^k)-\nabla f_i(x_i^{k-1})\right)+\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^{k-1}), \tag{3.1}$$

where x_i^k is determined by (1.3) with $x_i^{-1}=x^0$ and $\nabla f_i(x_i^{-1})=\nabla f(x^0)$ for $i=1,\dots,n$.

Our second SVRMM algorithm, named MM-SVRG, is inspired by a loopless SVRG estimator. Specifically, we replace the full gradient $\nabla f(x^k)$ in the deterministic MM (1.11) by the loopless SVRG stochastic gradient estimate $\widetilde{\nabla}_{\text{SVRG}}f(x^k)$ as follows:

$$\widetilde{\nabla}_{\text{SVRG}}f(x^k)=\frac{1}{b}\sum_{i\in I_k}\left(\nabla f_i(x^k)-\nabla f_i(\tilde{x}^k)\right)+\nabla f(\tilde{x}^k), \tag{3.2}$$

where $\tilde{x}^{-1}=x^0$ and $\tilde{x}^k=x^k$ if $d_k=1$ and $\tilde{x}^{k-1}$ otherwise, with d_k being randomly chosen from $\{0,1\}$ such that $d_k=1$ with probability (w.p.) $1/m$ and 0 otherwise with $m>1$. That is, $\tilde{x}^k=x^k$ w.p. $1/m$ and $\tilde{x}^{k-1}$ otherwise.

Our third SVRMM algorithm is named MM-SARAH, in which we replace the gradient $\nabla f(x^k)$ in the deterministic MM (1.11) with a loopless variant of SARAH as follows:

$$\widetilde{\nabla}_{\text{SARAH}}f(x^k)=\begin{cases}\nabla f(x^k) & \text{if } d_k=1,\\ \dfrac{1}{b}\displaystyle\sum_{i\in I_k}\left(\nabla f_i(x^k)-\nabla f_i(x^{k-1})\right)+\widetilde{\nabla}_{\text{SARAH}}f(x^{k-1}) & \text{otherwise,}\end{cases} \tag{3.3}$$

where $x^{-1}=x^0$, and $\widetilde{\nabla}_{\text{SARAH}}f(x^{-1})=\nabla f(x^0)$.

The general framework of our SVRMM algorithms is described in Algorithm 3.1, wherein MM-SAGA, MM-SVRG, and MM-SARAH employ their own gradient estimate $\widetilde{\nabla}_{\text{SAGA}}f(x^k)$, $\widetilde{\nabla}_{\text{SVRG}}f(x^k)$, and $\widetilde{\nabla}_{\text{SARAH}}f(x^k)$, accordingly. Additional details regarding our new algorithms are available in supplementary material section SM1.

Remark 3.2 (MM-SAGA vs. DCA-SAGA). The update rule (1.8) of DCA-SAGA [28] is

$$\min_{x\in\mathbb{R}^d}\frac{\mu}{2}\|x-\bar{x}^k\|^2+\left\langle\widetilde{\nabla}_{\text{SAGA}}f(x^k),x\right\rangle+r_1(x)-\left\langle y^k,x\right\rangle,$$

where $\bar{x}^k=\frac{1}{b}\sum_{i\in I_k}(x^k-x_i^{k-1})+\frac{1}{n}\sum_{i=1}^n x_i^{k-1}$ and $y^k\in\partial r_2(x^k)$. This update is different from our update rule (3.4) in MM-SAGA which employs the proximal term $\frac{\mu}{2}\|x-x^k\|^2$, in the same manner as in the deterministic MM.

It is worth mentioning an advantage of MM-SAGA when compared to DCA-SAGA in terms of memory storage, which can be described as follows. For MM-SAGA, one employs x_i^k solely

Algorithm 3.1. `SVRMM` framework.

Input: $x^0 \in \mathbb{R}^d$, a batch size $b \in [n]$, $\mu > \frac{L}{2}$, a gradient estimator $\widetilde{\nabla}$ (either $\widetilde{\nabla}_{\text{SAGA}}$, $\widetilde{\nabla}_{\text{SVRG}}$, or $\widetilde{\nabla}_{\text{SARAH}}$), $m > 1$ for $\widetilde{\nabla}_{\text{SVRG}}$ and $\widetilde{\nabla}_{\text{SARAH}}$, a tangent majorant function u of r, and $k = 0$.

repeat

Choose a random batch I_k of size b.

Compute the stochastic gradient estimate $\widetilde{\nabla} f(x^k)$: using (3.1) for `MM-SAGA`, using (3.2) for `MM-SVRG`, using (3.3) for `MM-SARAH`.

Compute

$$x^{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \frac{\mu}{2}\|x - x^k\|^2 + \left\langle \widetilde{\nabla} f(x^k), x \right\rangle + u(x, x^k). \tag{3.4}$$

until *Stopping criterion.*

Set $k \leftarrow k + 1$

for computing $\nabla f_i(x_i^k)$. In other words, (3.1) and (3.4) are defined and updated using only gradient values. Thus, it suffices to keep only the value $\nabla f_i(x_i^k)$, and then discard x_i^k. In many problems, such as binary classifications and multiclass logistic regressions, each gradient ∇f_i is a scalar multiple of the data point i, where the scalar can be updated in each iteration. In such cases, one may only store the weights, instead of the full gradient ∇f_i, e.g., [12]. In contrast, for `DCA-SAGA`, in addition to the values of gradients $\nabla f_i(x_i^k)$, one must store all vectors x_i^k in order to compute $\bar{x}^k$ in each iteration.

Remark 3.3. It is worth noting that `MM-SVRG` employs the loopless `SVRG` estimator which was demonstrated to have practical advantages in [25] when compared to the estimator (1.9) used in `DCA-SVRG` [28]. We also note that although the loopless `SVRG` estimator and `SVRG` may seem similar, they differ in the $\tilde{x}^k$-update. Specifically, the loopless `SVRG` estimator updates $\tilde{x}^k = x^k$ with a probability of $1/m$ ($m > 1$), while the `SVRG` estimator updates $\tilde{x}^k = x^k$ after every m iterations, i.e., if $k \in m\mathbb{N}$ (m is a fixed positive integer). In other words, the loopless `SVRG` estimator removes the explicit loop structure of `SVRG`.

4. Convergence analysis.

We now focus on convergence analysis of our `SVRMM` algorithms. To this end, throughout this paper, we assume the following basic assumptions regarding problem (1.1). Such assumptions are standard in optimization literature; see, e.g., [34, 41].

Assumption 1.

(a) F is bounded from below, that is, $F^* = \inf_{x \in \mathbb{R}^d} F(x) > -\infty$, and $\operatorname{dom} F \neq \emptyset$.

(b) Each f_i is L_i-smooth; equivalently, each f_i is continuously differentiable and there exists a positive constant L such that

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f_i(y)\|^2 \leq L^2\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^d. \tag{4.1}$$

Our following assumption is regarding the tangent majorant function u of r; see Definition 3.1.

Assumption 2.
(a) u is lower semicontinuous.
(b) For every x, $u(x,\cdot)$ is continuous.
(c) There exists a function $\bar{h}:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$ such that for every $y\in\mathbb{R}^d$, $\bar{h}(\cdot,y)$ is continuously differentiable at y with $\nabla\bar{h}(\cdot,y)(y)=0$, and the approximation error satisfies

$$u(\cdot,y)-r(\cdot)\le\bar{h}(\cdot,y). \tag{4.2}$$

Remark 4.1. We note that Assumption 2(c) encapsulates surrogate functions previously studied by Mairal [31, Definition 2.2], where it is assumed that the approximation error $h(\cdot,y):=u(\cdot,y)-r(\cdot)$ is L-smooth and $\nabla h(\cdot,y)(y)=0$. Indeed, in this case, we simply set $\bar{h}(\cdot,y)=h(\cdot,y)$.

We recall an important class of functions r with continuously differentiable approximation error $h(\cdot,y)$ such that $\nabla h(\cdot,y)(y)=0$. Consequently, this class satisfies Assumption 2. Additional examples can be found in [24, 31].

Example 4.2 (DC surrogates). If r is a DC function, that is, $r=r_1-r_2$ where r_1 and r_2 are convex, by assuming further r_2 is continuously differentiable, we consider the surrogate function

$$u(x,y)=r_1(x)-\Big[\langle\nabla r_2(y),x-y\rangle+r_2(y)\Big].$$

We now provide an example in which the approximation error function is nonsmooth; however, Assumption 2 is satisfied.

Example 4.3 (composite surrogates). Consider a class of functions of the form

$$r(x)=\sum_{i=1}^m\eta_i(g_i(x_i)),$$

where x is decomposed into m blocks $x=(x_1,\dots,x_m)$ with $x_i\in\mathbb{R}^{d_i}$, $\sum_{i=1}^m d_i=d$, and where $g_i:\mathbb{R}^{d_i}\to\mathbb{R}$ are convex and Lipschitz continuous with a common Lipschitz constant L_g, and $\eta_i:\mathbb{R}\to\mathbb{R}$ are concave and smooth with a common smoothness constant L_η on the image $g_i(\mathbb{R})$. This class includes composite functions, in particular, several existing sparsity-reduced regularizers, which are nonconvex and nonsmooth approximations of the ℓ_0-norm or $\ell_{q,0}$-norm; see, e.g., [7, 8]. Since η_i is concave, we can set up a surrogate function u for r as follows:

$$u(x,y)=r(y)+\sum_{i=1}^m\eta_i'(g_i(y_i))(g_i(x_i)-g_i(y_i)).$$

Since η_i is L_η-smooth on the image $g_i(\mathbb{R})$, it follows from Lemma 2.4(d) that

$$r(x)\ge r(y)+\sum_{i=1}^m\eta_i'(g_i(y_i))(g_i(x_i)-g_i(y_i))-\sum_{i=1}^m\frac{L_\eta}{2}|g_i(x_i)-g_i(y_i)|^2,$$

which, when combined with the L_g-Lipschitz continuity of g_i, implies that

$$u(x,y) - r(x) \le \sum_{i=1}^{m} \frac{L_\eta}{2} |g_i(x_i) - g_i(y_i)|^2 \le \frac{L_\eta L_g^2}{2} \|x - y\|^2.$$

Therefore, Assumption 2(c) is satisfied when we set $\bar{h}(x,y) = \frac{L_\eta L_g^2}{2}\|x-y\|^2$.

Before we proceed to our convergence analysis, for simplicity, we associate with each of our algorithms a sequence of random variables $\{\Upsilon^k\}$, which is defined by

(4.3)

Method	Υ^k
MM-SAGA	$\frac{1}{bn}\sum_{i=1}^{n} \left\|\nabla f_i(x^k) - \nabla f_i(x_i^{k-1})\right\|^2$
MM-SVRG	$\frac{1}{bn}\sum_{i=1}^{n} \left\|\nabla f_i(x^k) - \nabla f_i(\tilde{x}^{k-1})\right\|^2$
MM-SARAH	$\left\|\widetilde{\nabla}_{\text{SARAH}} f(x^{k-1}) - \nabla f(x^{k-1})\right\|^2$

In the following Lemmas 4.4 and 4.5, we estimate $\mathbb{E}_k\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\|^2$, where $\widetilde{\nabla} f(x^k)$ denotes either $\widetilde{\nabla}_{\text{SAGA}}$, $\widetilde{\nabla}_{\text{SVRG}}$, or $\widetilde{\nabla}_{\text{SARAH}}$. We also provide several properties of the sequence $\{\Upsilon^k\}$. Our arguments are similar to the arguments in the study [14] of stochastic proximal alternating linearized minimization algorithms using SAGA and SARAH for two-block optimization. However, we consider one block, which yields tighter bounds when compared to the bounds in [14, Proposition 2]. The proofs of Lemmas 4.4 and 4.5 are available in supplementary material section SM2.

Lemma 4.4. *Let $\{x^k\}$ be generated by MM-SAGA or by MM-SVRG with $m-1>0$ and let Υ^k be defined by (4.3). Then*

(a) $\mathbb{E}_k\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\|^2 \le \Upsilon^k$;

(b) $\mathbb{E}_k \Upsilon^{k+1} \le (1-\rho)\Upsilon^k + V_\Upsilon \mathbb{E}_k\|x^{k+1} - x^k\|^2$, *where ρ and V_Υ are defined by (4.4).*

(4.4)

Method	ρ	V_Υ
MM-SAGA	$\frac{b}{2n}$	$\frac{(2n-b)L^2}{b^2}$
MM-SVRG	$\frac{1}{2m}$	$\frac{(2m-1)L^2}{b}$

Lemma 4.5. *Let $\{x^k\}$ be generated by MM-SARAH with $m-1>0$ and let Υ^k be defined by (4.3). Then*

(a) $\mathbb{E}_k \left\|\widetilde{\nabla}_{\text{SARAH}} f(x^k) - \nabla f(x^k)\right\|^2 \le \Upsilon^k + \frac{L^2}{b}\left\|x^k - x^{k-1}\right\|^2$.

(b) $\mathbb{E}_k \Upsilon^{k+1} \le (1-\rho)\Upsilon^k + V_\Upsilon \left\|x^k - x^{k-1}\right\|^2$, *where*

$$\rho = \frac{1}{m} \quad \text{and} \quad V_\Upsilon = \frac{(m-1)L^2}{mb}. \tag{4.5}$$

We now present our main convergence results. The following theorem provides almost surely subsequential convergence of the iterations generated by our algorithms to a stationary point. For simplicity, we set the constant V in (4.6).

(4.6)

Method	V
MM-SAGA/MM-SVRG	0
MM-SARAH	$\frac{L^2}{b}$

Theorem 4.6 (almost surely subsequential convergence). *Let $\{x^k\}$ be a sequence generated by one of the* SVRMM *algorithms. Suppose that Assumptions 1 and 2 are satisfied, $m > 1$ for* MM-SVRG *and* MM-SARAH*, and that the condition*

$$2\mu - L - 2\sqrt{V + V_\Upsilon/\rho} > 0, \tag{4.7}$$

where V_Υ and ρ are determined by (4.4) for MM-SAGA *and* MM-SVRG *and by (4.5) for* MM-SARAH *and V is defined by (4.6), holds. Then*

(a) *the sequence $\{F(x^k)\}$ converges almost surely,*
(b) *the sequence $\{\|x^k - x^{k-1}\|^2\}$ has a finite sum (in particular, vanishes) almost surely,*
(c) *every limit point of $\{x^k\}$ is a stationary point of F, almost surely.*

Proof. First, we prove (a) and (b). Let us fix $k \in \mathbb{N}$. From the definition of a tangent majorant function (see Definition 3.1), it follows that

$$r(x^{k+1}) \le u(x^{k+1}, x^k). \tag{4.8}$$

By combining the L-smoothness of f with Lemma 2.4(d), we arrive at

$$f(x^{k+1}) \le f(x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k \right\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2. \tag{4.9}$$

Now, by combining the update rule for x^{k+1} with Lemma 2.4(c), we see that

$$\left\langle \widetilde{\nabla} f(x^k), x^{k+1} - x^k \right\rangle + u(x^{k+1}, x^k) + \mu\|x^{k+1} - x^k\|^2 \le u(x^k, x^k) = r(x^k). \tag{4.10}$$

Consequently, by summing up (4.8), (4.9), and (4.10), and by recalling that $F = f + r$, we obtain

$$\begin{aligned} F(x^{k+1}) + \frac{2\mu - L}{2}\left\|x^{k+1} - x^k\right\|^2 &\le F(x^k) + \left\langle \nabla f(x^k) - \widetilde{\nabla} f(x^k), x^{k+1} - x^k \right\rangle \\ &\le F(x^k) + \frac{\eta}{2}\left\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\right\|^2 + \frac{1}{2\eta}\left\|x^{k+1} - x^k\right\|^2, \end{aligned} \tag{4.11}$$

for any $\eta > 0$, where (4.11) holds due to the Cauchy–Schwarz inequality, more precisely, $\langle a, b\rangle \le \frac{\eta}{2}\|a\|^2 + \frac{1}{2\eta}\|b\|^2$ for all $\eta > 0$ and $a, b \in \mathbb{R}^d$. Later in our analysis, we will employ a particular choice of η. By taking the expectation in (4.11), conditioned on $\mathcal{F}_k$, we arrive at

$$\mathbb{E}_k\left[F(x^{k+1}) + \left(\frac{2\mu - L}{2} - \frac{1}{2\eta}\right)\left\|x^{k+1} - x^k\right\|^2\right] \le F(x^k) + \frac{\eta}{2}\mathbb{E}_k\left\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\right\|^2. \tag{4.12}$$

We now claim that

$$\mathbb{E}_k \Phi^{k+1} \le \Phi^k - \left(\frac{2\mu - L}{2} - \frac{1}{2\eta} - \frac{\eta V}{2} - \frac{\eta V_\Upsilon}{2\rho}\right)\left\|x^k - x^{k-1}\right\|^2, \tag{4.13}$$

where Φ^k is defined by (4.14).

Method	Φ^k
MM-SAGA/MM-SVRG	$F(x^k) + \left(\frac{2\mu-L}{2} - \frac{1}{2\eta} - \frac{\eta V}{2} - \frac{\eta V_\Upsilon}{2\rho}\right)\left\|x^k - x^{k-1}\right\|^2 + \frac{\eta}{2\rho}\Upsilon^k$
MM-SARAH	$F(x^k) + \left(\frac{2\mu-L}{2} - \frac{1}{2\eta}\right)\left\|x^k - x^{k-1}\right\|^2 + \frac{\eta}{2\rho}\Upsilon^k$

(4.14)

We will prove that the claim (4.13) holds by considering two cases. One case, in which the sequence $\{x^k\}$ is generated by either `MM-SAGA` or `MM-SVRG`, and the second case, in which the sequence $\{x^k\}$ is generated by `MM-SARAH`.

Case 1. Suppose that $\{x^k\}$ is generated by either `MM-SAGA` or `MM-SVRG`. Lemma 4.4(a) asserts that

$$\mathbb{E}_k \left\| \widetilde{\nabla} f(x^k) - \nabla f(x^k) \right\|^2 \leq \Upsilon^k + V \mathbb{E}_k \left\| x^{k+1} - x^k \right\|^2,$$

which, when combined with inequality (4.12), implies that

$$\mathbb{E}_k \left[F(x^{k+1}) + \left(\frac{2\mu - L}{2} - \frac{1}{2\eta} - \frac{\eta V}{2} \right) \left\| x^{k+1} - x^k \right\|^2 \right] \leq F(x^k) + \frac{\eta}{2} \Upsilon^k.$$

By recalling Lemma 4.4(b), which asserts that $\Upsilon^k \leq \frac{1}{\rho} \left(\Upsilon^k - \mathbb{E}_k \Upsilon^{k+1} \right) + \frac{V_\Upsilon}{\rho} \mathbb{E}_k \left\| x^{k+1} - x^k \right\|^2$, we obtain

$$\mathbb{E}_k \left[F(x^{k+1}) + \left(\frac{2\mu - L}{2} - \frac{1}{2\eta} - \frac{\eta V}{2} - \frac{\eta V_\Upsilon}{2\rho} \right) \left\| x^{k+1} - x^k \right\|^2 + \frac{\eta}{2\rho} \Upsilon^{k+1} \right] \leq F(x^k) + \frac{\eta}{2\rho} \Upsilon^k.$$

Consequently, (4.13) holds due to the definition of Φ^k in (4.14).

Case 2. Suppose that the sequence $\{x^k\}$ is generated by `MM-SARAH`. We now prove that (4.13) holds in this case as well.

First, Lemma 4.5(a) asserts that

$$\mathbb{E}_k \left\| \widetilde{\nabla} f(x^k) - \nabla f(x^k) \right\|^2 \leq \Upsilon^k + V \left\| x^k - x^{k-1} \right\|^2,$$

which, when combined with (4.12) and the relation from Lemma 4.5(b) where $\Upsilon^k \leq \frac{1}{\rho}(\Upsilon^k - \mathbb{E}_k \Upsilon^{k+1}) + \frac{V_\Upsilon}{\rho} \|x^k - x^{k-1}\|^2$, implies that

$$\begin{aligned} \mathbb{E}_k \left[F(x^{k+1}) + \left(\frac{2\mu - L}{2} - \frac{1}{2\eta} \right) \left\| x^{k+1} - x^k \right\|^2 + \frac{\eta}{2\rho} \Upsilon^{k+1} \right] \\ \leq F(x^k) + \frac{\eta}{2\rho} \Upsilon^k + \left(\frac{\eta V}{2} + \frac{\eta V_\Upsilon}{2\rho} \right) \left\| x^k - x^{k-1} \right\|^2. \end{aligned}$$

Thus, (4.13) holds due to the definition of Φ^k in (4.14).

Consequently, in both cases, (4.13) holds for the corresponding sequences $\{\Phi^k\}$. Now, due to (4.7), by setting $\eta = \frac{2\mu - L}{2(V + V_\Upsilon/\rho)}$, we see that

$$\frac{2\mu - L}{2} - \frac{1}{2\eta} - \frac{\eta V}{2} - \frac{\eta V_\Upsilon}{2\rho} = \frac{2\mu - L}{4} - \frac{V + V_\Upsilon/\rho}{2\mu - L} > 0.$$

We also have $2\mu - L > 0$ and $\frac{2\mu - L}{2} - \frac{1}{2\eta} > 0$. On the other hand, without the loss of generality, we may assume that $F^* \geq 0$. Thus, by supermartingale convergence (Lemma 2.6), the sequence $\{\|x^k - x^{k-1}\|^2\}$ almost surely has a finite sum (in particular, it vanishes), and $\{\Phi^k\}$ almost surely converges to a nonnegative random variable Φ^∞. Consequently, by combining Lemmas 4.4 and 4.5 and the supermartingale convergence in Lemma 2.6, Υ^k has a finite sum

(in particular, it vanishes) almost surely. It follows that the sequence $\{F(x^k)\}$ converges to Φ^∞ almost surely, which concludes the proof of (a) and (b).

Proof of (c). First, we claim that

$$\lim_{k\to+\infty}\left[\widetilde{\nabla} f(x^k)-\nabla f(x^k)\right]=0 \quad \text{almost surely.} \tag{4.15}$$

To prove this claim, by taking the total expectation in (4.13) with $\eta=\frac{2\mu-L}{2(V+V_\Upsilon/\rho)}$, we see that

$$\mathbb{E}\Phi^{k+1}\le \mathbb{E}\Phi^k-\kappa\mathbb{E}\left\|x^k-x^{k-1}\right\|^2, \tag{4.16}$$

where $\kappa=\frac{(2\mu-L)^2-4(V+V_\Upsilon/\rho)}{4(2\mu-L)}>0$. By telescoping (4.16) over k, we arrive at

$$\sum_{k=1}^{K}\kappa\mathbb{E}\left\|x^k-x^{k-1}\right\|^2\le \mathbb{E}\Phi^1-\mathbb{E}\Phi^{K+1}\le \mathbb{E}\Phi^1-F^*,$$

where we employed the fact that $\Phi^{K+1}\ge F(x^{K+1})\ge F^*$. Consequently, $\{\mathbb{E}\|x^k-x^{k-1}\|^2\}$ has a finite sum.

We now prove that the claim holds by considering two cases: Case 1, where $\{x^k\}$ is generated by either `MM-SAGA` or `MM-SVRG`, and Case 2, where the sequence $\{x^k\}$ is generated by `MM-SARAH`.

Case 1. Suppose that $\{x^k\}$ is generated by either `MM-SAGA` or `MM-SVRG`. Lemma 4.4 implies that

$$\mathbb{E}\left\|\widetilde{\nabla} f(x^k)-\nabla f(x^k)\right\|^2\le \mathbb{E}\Upsilon^k\le\left(\mathbb{E}\Upsilon^k-\mathbb{E}\Upsilon^{k+1}\right)/\rho+V_\Upsilon/\rho\mathbb{E}\left\|x^{k+1}-x^k\right\|^2. \tag{4.17}$$

By telescoping (4.17), we see that

$$\begin{aligned}\sum_{k=0}^{K}\mathbb{E}\left\|\widetilde{\nabla} f(x^k)-\nabla f(x^k)\right\|^2&\le\left(\mathbb{E}\Upsilon^0-\mathbb{E}\Upsilon^{K+1}\right)/\rho+V_\Upsilon/\rho\sum_{k=0}^{K}\mathbb{E}\left\|x^{k+1}-x^k\right\|^2\\&\le V_\Upsilon/\rho\sum_{k=0}^{K}\mathbb{E}\left\|x^{k+1}-x^k\right\|^2,\end{aligned} \tag{4.18}$$

where (4.18) is a consequence of $\Upsilon^k\ge 0$ and $\Upsilon^0=0$. The claim now follows from (4.18) and the fact that $\{\mathbb{E}\|x^k-x^{k-1}\|^2\}$ has a finite sum.

Case 2. Suppose that $\{x^k\}$ is generated by `MM-SARAH`. Lemma 4.5 implies that

$$\begin{aligned}\mathbb{E}\left\|\widetilde{\nabla} f(x^k)-\nabla f(x^k)\right\|^2&\le\mathbb{E}\Upsilon^k+V\mathbb{E}\left\|x^k-x^{k-1}\right\|^2\\&\le\left(\mathbb{E}\Upsilon^k-\mathbb{E}\Upsilon^{k+1}\right)/\rho+(V+V_\Upsilon/\rho)\,\mathbb{E}\left\|x^k-x^{k-1}\right\|^2.\end{aligned} \tag{4.19}$$

By telescoping (4.19), we see that

$$\begin{aligned}\sum_{k=0}^{K}\mathbb{E}\left\|\widetilde{\nabla} f(x^k)-\nabla f(x^k)\right\|^2&\le\left(\mathbb{E}\Upsilon^0-\mathbb{E}\Upsilon^{K+1}\right)/\rho+(V+V_\Upsilon/\rho)\sum_{k=0}^{K}\mathbb{E}\left\|x^k-x^{k-1}\right\|^2\\&\le(V+V_\Upsilon/\rho)\sum_{k=0}^{K}\mathbb{E}\left\|x^k-x^{k-1}\right\|^2,\end{aligned} \tag{4.20}$$

where (4.20) is a consequence of $\Upsilon^k \geq 0$ and $\Upsilon^0 = 0$. The claim now follows from (4.20) and the fact that $\{\mathbb{E}\|x^k - x^{k-1}\|^2\}$ has a finite sum.

This concludes the proof of claim (4.15).

By combining (4.15) and part (b), for any sequence x^k generated by one of our SVRMM algorithms,

$$\lim_{k\to\infty} \widetilde{\nabla} f(x^k) - \nabla f(x^k) = 0 \text{ and } \lim_{k\to\infty} x^k - x^{k-1} = 0 \tag{4.21}$$

almost surely. Let $\{x^k\}$ be generated by an SVRMM algorithm which satisfies (4.21). Let x^* be a limit point of $\{x^k\}$, that is, there is a subsequence $\{x^{k_j}\}$ of $\{x^k\}$ such that $x^{k_j} \to x^*$ as $j \to +\infty$. From the update rule of the SVRMM algorithms, it follows that

$$\nu^{k_j+1} := -\mu(x^{k_j+1} - x^{k_j}) - \widetilde{\nabla} f(x^{k_j}) \in \partial u(\cdot, x^{k_j})(x^{k_j+1}),$$

which implies that for any $x \in \mathbb{R}^d$,

$$u(x, x^{k_j}) \geq u(x^{k_j+1}, x^{k_j}) + \left\langle \nu^{k_j+1}, x - x^{k_j+1} \right\rangle. \tag{4.22}$$

By plugging $x = x^*$ into (4.22) and by letting $j \to +\infty$, we arrive at

$$r(x^*) \geq \limsup_{j\to+\infty} u(x^{k_j+1}, x^{k_j}), \tag{4.23}$$

where we employed the continuity of $u(x, y)$ in y (Assumption 2(b)), the fact that

$$\lim_{j\to+\infty} x^{k_j+1} = \lim_{j\to+\infty} x^{k_j} = x^*, \text{ and } \lim_{j\to+\infty} \nu^{k_j+1} = \lim_{j\to+\infty} -\widetilde{\nabla} f(x^{k_j}) = -\nabla f(x^*).$$

Here, $\lim_{j\to+\infty} -\widetilde{\nabla} f(x^{k_j}) = -\nabla f(x^*)$ follows from (4.15) and the continuity of ∇f. By combining (4.23) with the lower semicontinuity of $u(x, y)$ (Assumption 2(a)), we conclude that

$$\lim_{j\to+\infty} u(x^{k_j+1}, x^{k_j}) = r(x^*).$$

Consequently, by letting $j \to +\infty$ in (4.22), we see that for all $x \in \mathbb{R}^d$,

$$r(x^*) \leq u(x, x^*) + \langle \nabla f(x^*), x - x^* \rangle. \tag{4.24}$$

On the other hand, since f is L-smooth,

$$f(x^*) \leq f(x) - \langle \nabla f(x^*), x - x^* \rangle + \frac{L}{2}\|x - x^*\|^2. \tag{4.25}$$

By summing up (4.24) and (4.25), we arrive at

$$\begin{aligned} F(x^*) \leq u(x, x^*) + f(x) + \frac{L}{2}\|x - x^*\|^2 &= F(x) + u(x, x^*) - r(x) + \frac{L}{2}\|x - x^*\|^2 \\ &\leq F(x) + \bar{h}(x, x^*) + \frac{L}{2}\|x - x^*\|^2 \end{aligned}$$

for some function $\bar{h}$ which satisfies Assumption 2(c). Consequently, x^* is a minimizer of

$$\min_{x\in\mathbb{R}^d} F(x) + \bar{h}(x, x^*) + \frac{L}{2}\|x - x^*\|^2.$$

It follows that

$$0 \in \partial F(x^*) + \nabla \bar{h}(\cdot, x^*)(x^*) = \partial F(x^*), \tag{4.26}$$

which concludes part (c) and completes the proof. ∎

Remark 4.7 (feasibility of the batchsize b and the stepsize $\frac{1}{\mu}$). For `MM-SAGA`, since $V = 0$, $V_\Upsilon = \frac{(2n-b)L^2}{b^2}$, and $\rho = \frac{b}{2n}$, condition (4.7) is satisfied when

$$2\mu - L \geq \frac{4nL}{b^{3/2}}.$$

For `MM-SVRG`, since $V = 0$, $V_\Upsilon = \frac{(2m-1)L^2}{b}$, and $\rho = \frac{1}{2m}$, condition (4.7) is satisfied when

$$2\mu - L \geq \frac{4mL}{\sqrt{b}}.$$

For `MM-SARAH`, since $V = \frac{L^2}{b}$, $V_\Upsilon = \frac{(m-1)L^2}{mb}$, and $\rho = \frac{1}{m}$, condition (4.7) is satisfied when

$$2\mu - L > 2L\sqrt{\frac{m}{b}}.$$

We now provide iteration complexity in order to obtain an ϵ-stationary point. To this end, we incorporate the following additional assumption [19, Assumption 3(ii)] regarding the tangent majorant function u of r.

Assumption 3. There exists a (deterministic) constant L_u such that, almost surely, for any $k \in \mathbb{N}$ and for any $\nu \in \partial u(\cdot, x^k)(x^{k+1})$, there exists $\zeta \in \partial r(x^{k+1})$ such that $\|\nu - \zeta\| \leq L_u\|x^{k+1} - x^k\|$.

Remark 4.8. We consider the case where $h(x, y) := u(x, y) - r(x)$ is L_u-smooth in x and $\nabla h(\cdot, y)(y) = 0$, as assumed in [31]. We assert that Assumption 3 captures this case. Indeed, since $\partial u(\cdot, y)(x) = \partial r(x) + \nabla h(\cdot, y)(x)$, if $\nu \in \partial u(\cdot, y)(x)$, then there exists $\zeta \in \partial r(x)$ such that

$$\|\nu - \zeta\| = \|\nabla h(\cdot, y)(x)\| = \|\nabla h(\cdot, y)(x) - \nabla h(\cdot, y)(y)\| \leq L_u\|x - y\|.$$

It follows that the DC surrogates (Example 4.2), where r_2 is L-smooth, satisfy Assumption 3. Furthermore, Assumption 3 captures composite surrogates (Example 4.3) with nonsmooth approximation error functions. We assume further that for any x_j, $\partial g_j(x_j)$ are bounded sets with a common constant M, i.e., $\|\xi\| \leq M$ for any $\xi \in \partial g_j(x_j)$ and $x_j \in \mathbb{R}^{d_j}$. Let $\nu \in \partial u(\cdot, y)(x)$. Then

$$\nu = (\eta_1'(g_1(y_1))\xi_1, \ldots, \eta_m'(g_m(y_m))\xi_m), \quad \text{where} \quad \xi_j \in \partial g_j(x_j).$$

On the other hand, it follows from [48, Corollary 5Q] that

$$\partial(\eta_j \circ g_j)(x_j) = \eta_j'(g_j(x_j))\partial g_j(x_j).$$

Consequently, by letting $\zeta = (\eta_1'(g_1(x_1))\xi_1, \ldots, \eta_m'(g_m(x_m))\xi_m) \in \partial r(x)$, we arrive at

$$\|\nu - \zeta\| \le \sqrt{\sum_{j=1}^{m} L_\eta^2 |g_j(x_j) - g_j(y_j)|^2 \|\xi_j\|^2} \le L_\eta L_g M \|x - y\|,$$

which implies that Assumption 3 is satisfied by letting $L_u = L_\eta L_g M$.

Theorem 4.9 (iteration complexity). *Let $\{x^k\}$ be a sequence generated by one of the* SVRMM *algorithms. Suppose that Assumptions 1, 2, and 3, $m > 1$ for* MM-SVRG *and* MM-SARAH*, and condition (4.7) are satisfied. Then for any positive natural number K,*

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^k)\right) \le \frac{8(2\mu - L)[(L + \mu + L_u)^2 + V_\Upsilon/\rho](F(x^0) - F^*)}{K[(2\mu - L)^2 - 4(V + V_\Upsilon/\rho)]}. \tag{4.27}$$

Proof. From the update rule of the SVRMM algorithms, it follows that

$$\nu^{k+1} := -\mu(x^{k+1} - x^k) - \widetilde{\nabla} f(x^k) \in \partial u(\cdot, x^k)(x^{k+1}).$$

Thus, by Assumption 3, there exists $\zeta^{k+1} \in \partial r(x^{k+1})$ such that $\|\nu^{k+1} - \zeta^{k+1}\| \le L_u \|x^{k+1} - x^k\|$. Consequently, by invoking Lemma 2.4(b), it follows that

$$\nabla f(x^{k+1}) + \zeta^{k+1} \in \nabla f(x^{k+1}) + \partial r(x^{k+1}) = \partial F(x^{k+1}).$$

We see that

$$\begin{aligned}
\mathrm{dist}(0, \partial F(x^{k+1})) &\le \left\|\nabla f(x^{k+1}) + \zeta^{k+1}\right\| \\
&= \left\|\nabla f(x^{k+1}) - \nabla f(x^k) + \nabla f(x^k) - \widetilde{\nabla} f(x^k) - \mu(x^{k+1} - x^k) + \zeta^{k+1} - \nu^{k+1}\right\| \\
&\le (L + \mu + L_u)\left\|x^{k+1} - x^k\right\| + \left\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\right\|.
\end{aligned}$$

It now follows that

$$\mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^{k+1})\right) \le 2(L + \mu + L_u)^2\, \mathbb{E}\left\|x^{k+1} - x^k\right\|^2 + 2\mathbb{E}\left\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\right\|^2. \tag{4.28}$$

On the other hand, if the sequence is generated by either MM-SAGA or MM-SVRG, Lemma 4.4 implies that

$$\mathbb{E}\left\|\widetilde{\nabla} f(x^k) - \nabla f(x^k)\right\|^2 \le \left(\mathbb{E}\Upsilon^k - \mathbb{E}\Upsilon^{k+1}\right)/\rho + V_\Upsilon/\rho \mathbb{E}\left\|x^{k+1} - x^k\right\|^2. \tag{4.29}$$

By plugging (4.29) into (4.28) we arrive at

$$\mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^{k+1})\right) \le \left[2(L + \mu + L_u)^2 + 2V_\Upsilon/\rho\right]\mathbb{E}\left\|x^{k+1} - x^k\right\|^2 + 2\left(\mathbb{E}\Upsilon^k - \mathbb{E}\Upsilon^{k+1}\right)/\rho. \tag{4.30}$$

Moreover, it follows from (4.13) with $\eta = \frac{2\mu - L}{2(V + V_\Upsilon/\rho)}$ that

$$\mathbb{E}\left\|x^{k+1} - x^k\right\|^2 \le \kappa^{-1}\left(\mathbb{E}\Phi^{k+1} - \mathbb{E}\Phi^{k+2}\right), \tag{4.31}$$

where $\kappa = \frac{(2\mu - L)^2 - 4(V + V_\Upsilon/\rho)}{4(2\mu - L)}$. By combining (4.30) and (4.31), we see that

$$\mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^{k+1})\right) \le 2\kappa^{-1}\left[(L + \mu + L_u)^2 + V_\Upsilon/\rho\right]\left(\mathbb{E}\Phi^{k+1} - \mathbb{E}\Phi^{k+2}\right) + 2\left(\mathbb{E}\Upsilon^k - \mathbb{E}\Upsilon^{k+1}\right)/\rho. \tag{4.32}$$

Consequently, by telescoping (4.32) over $k = 0, \ldots, K-1$, we obtain

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^k)\right) &\le 2\kappa^{-1}\left[(L + \mu + L_u)^2 + V_\Upsilon/\rho\right]\left(\mathbb{E}\Phi^1 - \mathbb{E}\Phi^{K+1}\right) + 2\left(\mathbb{E}\Upsilon^0 - \mathbb{E}\Upsilon^K\right)/\rho \\ &\le 2\kappa^{-1}\left[(L + \mu + L_u)^2 + V_\Upsilon/\rho\right]\left(\mathbb{E}\Phi^1 - F^*\right), \qquad (4.33)\end{aligned}$$

where (4.33) follows from $\Upsilon^0 = 0$, $\Upsilon^k \ge 0$, and $\Phi^k \ge F(x^k) \ge F^*$. We conclude the proof in the case of `MM-SAGA` and `MM-SVRG` by combining (4.33) and the fact that

$$\begin{aligned}\Phi^1 &= F(x^1) + \left(\frac{2\mu - L}{4} - \frac{V + V_\Upsilon/\rho}{2\mu - L}\right)\left\|x^1 - x^0\right\|^2 + \frac{2\mu - L}{4(\rho V + V_\Upsilon)}\Upsilon^1 \qquad (4.34)\\ &\le F(x^1) + \frac{2\mu - L}{4}\left\|x^1 - x^0\right\|^2 + \frac{2\mu - L}{4(\rho V + V_\Upsilon)}\Upsilon^1 \\ &\le F(x^1) + \frac{2\mu - L}{2}\left\|x^1 - x^0\right\|^2 \qquad (4.35)\\ &\le F(x^0), \qquad (4.36)\end{aligned}$$

where (4.34) follows from (4.14) with $k = 1$ and $\eta = \frac{2\mu - L}{2(V + V_\Upsilon/\rho)}$, (4.36) follows from the first inequality in (4.11) with $k = 0$ and $\nabla f(x^0) = \tilde{\nabla} f(x^0)$, and (4.35) follows by recalling that

$$\frac{\Upsilon^1}{\rho V + V_\Upsilon} = \frac{1}{\rho V + V_\Upsilon}\frac{1}{bn}\sum_{i=1}^{n}\left\|\nabla f_i(x^1) - \nabla f_i(x^0)\right\|^2 \le \frac{1}{\rho V + V_\Upsilon}\frac{L^2}{b}\left\|x^1 - x^0\right\|^2 \le \left\|x^1 - x^0\right\|^2.$$

In the case where the sequence $\{x^k\}$ is generated by `MM-SARAH`, Lemma 4.5(b) implies that

$$\mathbb{E}\left\|\widetilde{\nabla}_{\text{SARAH}} f(x^k) - \nabla f(x^k)\right\|^2 = \mathbb{E}\Upsilon^{k+1} \le \left(\mathbb{E}\Upsilon^{k+1} - \mathbb{E}\Upsilon^{k+2}\right)/\rho + V_\Upsilon/\rho\mathbb{E}\left\|x^{k+1} - x^k\right\|^2. \tag{4.37}$$

By plugging (4.37) into (4.28), we see that

$$\mathbb{E}\,\mathrm{dist}^2(0, \partial F(x^{k+1})) \le \left[2\left(L + \mu + L_u\right)^2 + 2V_\Upsilon/\rho\right]\mathbb{E}\left\|x^{k+1} - x^k\right\|^2 + 2\left(\mathbb{E}\Upsilon^{k+1} - \mathbb{E}\Upsilon^{k+2}\right)/\rho. \tag{4.38}$$

By telescoping (4.38) over $k = 0, \ldots, K-1$, we obtain

$$\begin{aligned}\sum_{k=1}^{K} \mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(x^k)\right) &\le 2\left[(L+\mu+L_u)^2 + V_\Upsilon/\rho\right] \sum_{k=0}^{K-1} \mathbb{E}\left\|x^{k+1}-x^k\right\|^2 + 2\left(\mathbb{E}\Upsilon^1 - \mathbb{E}\Upsilon^{K+1}\right)/\rho \\ &\le 2\kappa^{-1}\left[(L+\mu+L_u)^2 + V_\Upsilon/\rho\right]\left(\mathbb{E}\Phi^1 - \mathbb{E}\Phi^{K+1}\right) \qquad (4.39)\\ &\le 2\kappa^{-1}\left[(L+\mu+L_u)^2 + V_\Upsilon/\rho\right]\left(\mathbb{E}\Phi^1 - F^*\right),\end{aligned}$$

where we employed the fact that $\Upsilon^1 = 0$, $\Upsilon^k \ge 0$, $\Phi^k \ge F(x^k) \ge F^*$, and (4.31). We conclude the proof in the case of `MM-SARAH` by combining (4.39) and the fact that

$$\begin{aligned}\Phi^1 &= F(x^1) + \left(\frac{2\mu - L}{2} - \frac{V + V_\Upsilon/\rho}{2\mu - L}\right)\left\|x^1 - x^0\right\|^2 + \frac{2\mu - L}{4(\rho V + V_\Upsilon)}\Upsilon^1 \qquad (4.40)\\ &= F(x^1) + \left(\frac{2\mu - L}{2} - \frac{V + V_\Upsilon/\rho}{2\mu - L}\right)\left\|x^1 - x^0\right\|^2 \qquad (4.41)\\ &\le F(x^1) + \frac{2\mu - L}{2}\left\|x^1 - x^0\right\|^2 \\ &\le F(x^0), \qquad (4.42)\end{aligned}$$

where (4.40) follows from (4.14) with $k = 1$ and $\eta = \frac{2\mu - L}{2(V + V_\Upsilon/\rho)}$, (4.41) holds due to $\Upsilon^1 = 0$, and (4.42) follows from the first inequality in (4.11) with $k = 0$ and $\nabla f(x^0) = \tilde{\nabla} f(x^0)$. ∎

The following corollary follows directly from Theorem 4.9.

Corollary 4.10. *Let $\{x^k\}$ be a sequence generated by one of the `SVRMM` algorithms. Suppose that Assumptions 1, 2, and 3, $m > 1$ for `MM-SVRG` and `MM-SARAH`, and condition (4.7) are satisfied. If $\hat{x}^K$ is chosen uniformly from $\{x^1, x^2, \ldots, x^K\}$, then*

$$\mathbb{E}\,\mathrm{dist}^2\left(0, \partial F(\hat{x}^K)\right) \le \frac{8(2\mu - L)[(L+\mu+L_u)^2 + V_\Upsilon/\rho](F(x^0) - F^*)}{K[(2\mu - L)^2 - 4(V + V_\Upsilon/\rho)]} = \mathcal{O}(1/K).$$

In other words, the number of iterations K needed to obtain an ϵ-stationary point $\hat{x}^K$ of F, in expectation, is at most $K = \frac{8(2\mu-L)[(L+\mu+L_u)^2+V_\Upsilon/\rho](F(x^0)-F^*)}{[(2\mu-L)^2-4(V+V_\Upsilon/\rho)]\epsilon^2} = \mathcal{O}(1/\epsilon^2)$.

The `SVRMM` algorithms incorporate three parameters: μ, the batchsize b, and m. Setting $\mu = L$ yields a larger stepsize $\frac{1}{L}$ when compared to other stochastic gradient methods such as `DCA-SAGA` and `DCA-SVRG` [28], with a stepsize of $\frac{1}{2L}$, `ProxSVRG` [21], with a stepsize of $\frac{1}{3L}$, and `ProxSVRG+` [30], with a stepsize of $\frac{1}{6L}$. After fixing μ, we select the batchsize that satisfies condition (4.7). The following corollary summarizes choices of the batchsize b and the associated complexity of each algorithm in terms of the number of individual stochastic gradient valuations ∇f_i. Its proof is available in supplementary material section SM3.

Corollary 4.11.

(a) *In the case of `MM-SAGA`, we set $\mu = L$ and $b = \lceil 2^{5/3} n^{2/3} \rceil$ and recall that $V = 0, V_\Upsilon = \frac{(2n-b)L^2}{b^2}$, and $\rho = \frac{b}{2n}$ (see (4.4) and (4.6)). Consequently, to obtain an ϵ-stationary*

point in expectation, the number of individual stochastic gradient evaluations ∇f_i *does not exceed*

$$Kb < \frac{16L[(2+L_u/L)^2+1/8](F(x^0)-F^*)}{\epsilon^2}\lceil 2^{5/3}n^{2/3}\rceil = \mathcal{O}(n^{2/3}/\epsilon^2).$$

In other words, the complexity is $\mathcal{O}(n^{2/3}/\epsilon^2)$.

(b) *In the case of* ***MM-SVRG****, we set* $\mu = L$, $b = \lfloor n^{2/3}\rfloor$ *and* $m = \frac{\sqrt{b}}{4\sqrt{2}}$ *and recall that* $V = 0, V_\Upsilon = \frac{(2m-1)L^2}{b}$, *and* $\rho = \frac{1}{2m}$ *(see* (4.4) *and* (4.6)*). Consequently, to obtain an* ϵ*-stationary point in expectation, the number of individual stochastic gradient evaluations* ∇f_i, *in expectation, does not exceed*

$$K\left(\left(1-\frac{1}{m}\right)2b+\frac{1}{m}n\right) < \frac{16L[(2+L_u/L)^2+1/8](F(x^0)-F^*)}{\epsilon^2}(10n^{2/3}+8) = \mathcal{O}(n^{2/3}/\epsilon^2).$$

In other words, the complexity is $\mathcal{O}(n^{2/3}/\epsilon^2)$.

(c) *In the case of* ***MM-SARAH****, we set* $\mu = L$, $b = \lfloor n^{1/2}\rfloor$ *and* $m = \frac{b}{8}$, *and recall that* $V = \frac{L^2}{b}, V_\Upsilon = \frac{(m-1)L^2}{mb}$, *and* $\rho = \frac{1}{m}$ *(see* (4.5) *and* (4.6)*). Consequently, to obtain an* ϵ*-stationary point in expectation, the number of individual stochastic gradient evaluations* ∇f_i, *in expectation, does not exceed*

$$K\left(\left(1-\frac{1}{m}\right)2b+\frac{1}{m}n\right) < \frac{288L[(2+L_u/L)^2+1/8](F(x^0)-F^*)}{\epsilon^2}n^{1/2} = \mathcal{O}(n^{1/2}/\epsilon^2).$$

In other words, the complexity is $\mathcal{O}(n^{1/2}/\epsilon^2)$.

Remark 4.12.

(a) Our results coincide with the best-known complexity bounds to obtain an ϵ-stationary point in expectation for `ProxSAGA` [21], `ProxSVRG` [21, 30], `SPIDER` [15], `SpiderBoost` [51], and `ProxSARAH` [41] methods in the particular case where $r = 0$ [15, 51] or the case where r is convex [21, 30, 41].

(b) To guarantee convergence, we need to choose the parameters μ, b, m to satisfy condition (4.7). Instead of fixing $\mu = L$, we may first fix a batchsize b, then choose a compatible parameter μ to comply with condition (4.7). In particular, we may pick a batchsize $b \in \{1, 2, \ldots, n-1\}$ and any $m > 1$, then set $\mu = (4nL/b^{3/2} + L)/2$ for `MM-SAGA`, $\mu = (4mL/b^{1/2}+L)/2$ for `MM-SVRG`, and $\mu = (2m^{1/2}L/b^{1/2}+L)/2+10^{-5}$ for `MM-SARAH`.

5. Numerical experiments. We now examine the applicability and efficiency of our `SVRMM` algorithms. To this end, we consider the following three problems: sparse binary classification with nonconvex loss and regularizer, sparse multiclass logistic regression with nonconvex regularizer, and feedforward neural network training.

We compare six algorithms:

- `MM-SAGA` with $\mu = L$ and $b = \lceil 2^{5/3}n^{2/3}\rceil$;
- `MM-SVRG` with $\mu = L$, $b = \lfloor n^{2/3}\rfloor$ and $m = \frac{\sqrt{b}}{4\sqrt{2}}$;
- `MM-SARAH` with $\mu = L$, $b = \lfloor n^{1/2}\rfloor$ and $m = \frac{b}{8}$;
- `SDCA` [27] with $\mu = 1.1L$ and $b = \lfloor n/10\rfloor$, which performed well in [27];

- `DCA-SAGA` [28] with $\mu = 2L$ and $b = \lceil 2\sqrt{n\sqrt{n+1}} \rceil$;
- `DCA-SVRG` [28] with $\mu = 2L$, $b = \lfloor n^{2/3} \rfloor$, and the inner loop length $M = \lfloor \frac{\sqrt{b}}{4\sqrt{e-1}} \rfloor$.

In our experiments, we run each algorithm 20 epochs repeated 20 times, where each epoch consists of n gradient evaluations. We are interested in the relative loss residuals $\frac{F(w^k)-F^*}{|F^*|}$, where F^* is the minimum loss values generated by all algorithms, and in classification accuracy on testing sets.

All tests are performed using Python on a Linux server with the following configuration: Intel Xeon Gold 5220R CPU 2.20 GHz of 64 GB RAM. The code is available at https://github.com/nhatpd/SVRMM.

5.1. Sparse binary classification with nonconvex loss and regularizer. Let $\{(a_i, b_i) : i = 1, \ldots, n\}$ be a training set with observation vectors $a_i \in \mathbb{R}^d$ and labels $b_i \in \{-1, 1\}$. We consider the sparse binary classification with nonconvex loss function and nonconvex regularizer:

$$\min_{w \in \mathbb{R}^d} \left\{ F(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(a_i^T w, b_i) + r(w) \right\}, \tag{5.1}$$

where ℓ is a nonconvex loss function and r is a regularization term. We revisit a nonconvex loss function from [53], $\ell(s,t) = (1 - \frac{1}{1+\exp(-ts)})^2$, and the exponential regularization from [7], $r(w) = \sum_{i=1}^{d} \eta \circ g(w_i)$, where η and g are the functions

$$\eta(t) = \lambda(1 - \exp(-\alpha t)) \quad \text{and} \quad g(w) = |w|, \tag{5.2}$$

where λ and α are nonnegative tuning parameters. The hessian matrix of $\ell(a_i^T \cdot, b_i)$ is evaluated as follows:

$$\nabla^2 \ell(a_i^T w, b_i) = \frac{4\exp(2b_i a_i^T w) - 2\exp(b_i a_i^T w)}{\left(\exp(2b_i a_i^T w) + 1\right)^4} a_i a_i^T.$$

We thus have

$$\|\nabla^2 \ell(a_i^T w, b_i)\| = \frac{\left|4\exp(2b_i a_i^T w) - 2\exp(b_i a_i^T w)\right|}{\left(\exp(2b_i a_i^T w) + 1\right)^4} \|a_i a_i^T\| \le \frac{39 + 55\sqrt{33}}{2304} \|a_i\|^2.$$

Therefore, $\ell(a_i^T \cdot, b_i)$ is L-smooth with $L = \frac{39+55\sqrt{33}}{2304} \max_{i=1,\ldots,n} \|a_i\|^2$ and, in this case, problem (5.1) is within the scope of problem (1.1) when we let $f_i(w) = \ell(a_i^T w, b_i)$. Moreover, since η is concave and $\lambda\alpha^2$-smooth on $\mathbb{R}_+$, and since g is convex and 1-Lipschitz continuous, we set a composite surrogate function u for r as follows:

$$u(w, w^k) = r(w^k) + \sum_{i=1}^{d} \lambda\alpha \exp(-\alpha |w_i^k|)(|w_i| - |w_i^k|).$$

Assumptions 2 and 3 are then satisfied; see Example 4.3 and Remark 4.8. The `SVRMM` algorithms update w^{k+1} to be the solution of the nonsmooth convex subproblem:

$$\min_{w \in \mathbb{R}^d} \frac{\mu}{2} \|w - w^k\|^2 + \left\langle \widetilde{\nabla} f(w^k), w \right\rangle + \sum_{i=1}^{d} \lambda\alpha \exp(-\alpha |w_i^k|)|w_i|,$$

for which a closed-form solution was provided in [39, section 6.5.2] by

$$w_i^{k+1} = \max\left\{|v_i^k| - \lambda\alpha\exp(-\alpha|w_i^k|)/\mu, 0\right\}\operatorname{sign}(v_i^k),$$

where $v^k = w^k - \widetilde{\nabla} f(x^k)/\mu$.

5.2. Sparse multiclass logistic regression with nonconvex regularizer. We revisit the multiclass logistic regression with a nonconvex regularizer:

$$\min_{W\in\mathbb{R}^{d\times q}}\left\{F(W) = \frac{1}{n}\sum_{i=1}^{n}\ell(b_i, a_i, W) + r(W)\right\}, \tag{5.3}$$

where q is the number of classes, $\{(a_i, b_i) : i = 1, 2, \ldots, q\}$ is a training set with the feature vectors $a_i \in \mathbb{R}^d$ and the labels $b_i \in \{1, 2, \ldots, q\}$, r is a regularizer, and $\ell(b_i, a_i, \cdot)$ is a loss function defined by

$$\ell(b_i, a_i, W) = \log\left(\sum_{k=1}^{q}\exp(a_i^T w_k)\right) - a_i^T w_{b_i},$$

where w_k is the kth column of W. We employ an exponential ℓ_2 regularizer, defined by $r(W) = \lambda\sum_{i=1}^{d}\eta(\|W_i\|)$, where η is defined as in (5.2), and W_i is the ith row of W. The gradient of $\ell(b_i, a_i, \cdot)$ is evaluated as follows:

$$\nabla\ell(b_i, a_i, W) = a_i\sigma(a_i, W) - a_i\delta_i,$$

where the softmax function $\sigma(a_i, \cdot)$ is defined by

$$\sigma(a_i, W) = \frac{1}{\sum_{k=1}^{q}\exp(a_i^T w_k)}\left[\exp(a_i^T w_1), \ldots, \exp(a_i^T w_q)\right],$$

and the indicator row-vector δ_i is defined by $\delta_{ik} = 1$ if $b_i = k$ and 0 otherwise. Since $\sigma(a_i, \cdot)$ is L-Lipschitz with $L = \|a_i\|$ due to [16, Proposition 4], it follows that $\ell(b_i, a_i, \cdot)$ is L-smooth with $L = \max_{i=1,..,n}\|a_i\|^2$. This implies that problem (5.3) is within the scope of problem (1.1) when we set $f_i(W) = \ell(b_i, a_i, W)$. The `SVRMM` algorithms applied to (5.3) iteratively determine a composite surrogate function of $r(W)$ at W^k by

$$u(W, W^k) = r(W^k) + \sum_{i=1}^{d}\lambda\alpha\exp(-\alpha\|W_i^k\|)(\|W_i\| - \|W_i^k\|)$$

and then update W^{k+1} by

$$W^{k+1} = \arg\min_{W\in\mathbb{R}^{d\times q}}\frac{\mu}{2}\|W - W^k\|^2 + \left\langle\widetilde{\nabla} f(W^k), W\right\rangle + \sum_{i=1}^{d}\lambda\alpha\exp(-\alpha\|W_i^k\|)\|W_i\|$$

for which a closed-form solution was provided in [39, section 6.5.1] by

$$W_i^{k+1} = \begin{cases}\left(1 - \dfrac{\lambda\alpha\exp(-\alpha\|W_i^k\|)/\mu}{\|V_i^k\|}\right)V_i^k & \text{if} \quad \|V_i^k\| \geq \lambda\alpha\exp(-\alpha\|W_i^k\|)/\mu,\\ 0 & \text{otherwise,}\end{cases}$$

where $V_i^k = W_i^k - \widetilde{\nabla} f(W^k)_i/\mu$.

5.3. Feedforward neural network training problem with nonconvex regularizer. We consider the nonconvex optimization model arising in a feedforward neural network configuration

$$\min_{w\in\mathbb{R}^D}\left\{F(w)=\frac{1}{n}\sum_{i=1}^{n}\ell(h(w,a_i),b_i)+r(w)\right\}, \tag{5.4}$$

where all of the weight matrices and bias vectors of the neural network are concatenated in one vector of variables w, $(a_i,b_i)_{i=1}^n$ is a training data set with the feature vectors $a_i\in\mathbb{R}^d$ and the labels $b_i\in\{1,2,\dots,q\}$, h is a composition of linear transforms and activation functions of the form $h(w,a)=\sigma_l(W_l\sigma_{l-1}(W_{l-1}\sigma_{l-2}(\cdots\sigma_0(W_0a+c_0)\cdots)+c_{l-1})+c_l)$, where W_i is a weight matrix, c_i is a bias vector, σ_i is an activation function, l is the number of layers, ℓ is the soft-max cross-entropy loss, and r is a regularizer. By considering the exponential regularization $r(w)=\sum_{i=1}^{D}\eta\circ g(w_i)$, where η and g are set in (5.2), problem (5.4) is within the scope of problem (1.1) when we let $f_i(w)=\ell(h(w,a_i),b_i)$. The SVRMM algorithms applied to (5.4) are different from the SVRMM algorithms for problem (5.1) only in computation of stochastic gradient estimates $\widetilde{\nabla}f$. In our experiment, we employ a one-hidden-layer fully connected neural network, $784\times100\times10$, as studied in [41]. The activation function σ_i of the hidden layer is ReLU.

5.4. Experiment setups and data sets. In our experiments, for the first two problems (5.1) and (5.3), all of the algorithms under study start at the zero point, while for the last problem (5.4), we use the global_variables_initializer function from Tensorflow. We set the regularization parameters $\alpha=5$ for the first two problems and $\alpha=0.05$ for the latter, and we fix $\lambda=1/n$. These regularization parameters are standard in the literature, e.g., [7, 41]. It is important to mention that in all of the experiments, we use the same problem settings for all of the algorithms.

We conducted experiments on five well-known data sets for sparse binary classification, including w8a, rcv1, real-sim, epsilon, and url. For sparse multiclass logistic regression, we tested all of the algorithms on four data sets: dna, shuttle, Sensorless, and connect-4. Finally, for the feedforward neural network training, we used two data sets, minist and fashion_mnist, to compare MM-SVRG and MM-SARAH with DCA-SVRG. It is worth noting that the last evaluation only considered the three algorithms that do not require storing the gradient of each component function f_i. For all experiments, we randomly pick 90% of the data for training and the rest for testing. The characteristics of the data sets are provided in supplementary material section SM4. The first nine data sets are obtained from the LIBSVM Data website[1] while the data sets minist and fashion_mnist are obtained from the library tensorflow.keras.datasets.

5.5. Results. We plotted the curves of the average value of relative loss residuals versus epochs in Figures 1 and 2. We also reported the average and the standard deviation of the relative loss residuals and the testing accuracy in supplementary material section SM5. We observe from Figures 1 and 2 that MM-SARAH has the fastest convergence on all of the data sets. This illustrates the theoretical results (see Corollary 4.11) where MM-SARAH has the best complexity among these algorithms. In addition, MM-SAGA performs better than DCA-SAGA, which is not stable on the first nine data sets. This illustrates the benefit of the proximal

[1] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/.

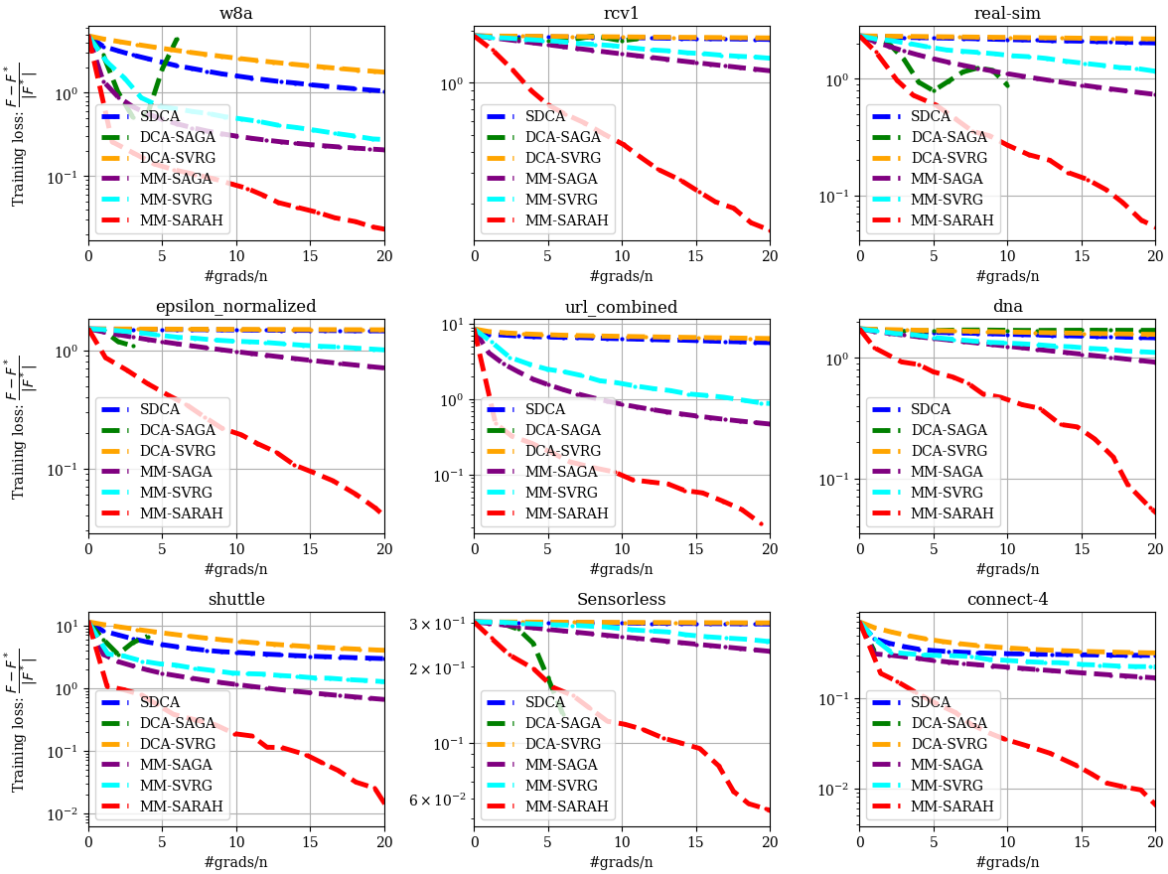

Figure 1. *Evolution of the average value of the relative loss residuals with respect to the epoch on w8a, rcv1, real-sim, epsilon_normalized, url_combined, dna, shuttle, Sensorless, and connect-4.*

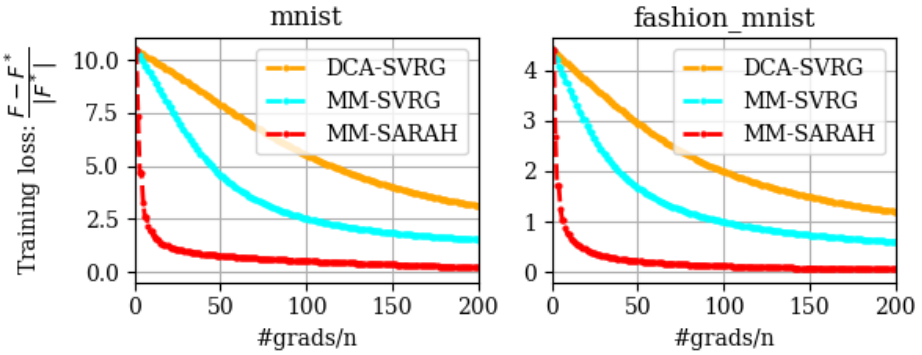

Figure 2. *Evolution of the average value of the relative loss residuals with respect to the epoch on mnist and fashion mnist.*

term in the iterate of MM-SAGA. Moreover, MM-SVRG performs better than DCA-SVRG on all of the data sets, which illustrates the benefit of the loopless variant of SVRG in MM-SVRG.

6. Conclusion. We introduced three stochastic variance-reduced MM algorithms, MM-SAGA, MM-SVRG, and MM-SARAH, combining the MM principle and the variance reduction techniques from SAGA, SVRG, and SARAH for solving a class of nonconvex nonsmooth optimization problems with the large-sum structure. The complex objective function is approximated by compatible surrogate functions, providing closed-form solutions in the updates of our algorithms. At the same time, we employ the benefits of the stochastic gradient estimators (SAGA, loopless SVRG, and loopless SARAH) to overcome the challenge of the large-sum structure. We provided almost surely subsequential convergence of MM-SAGA, MM-SVRG, and MM-SARAH to a stationary point under mild assumptions. In addition, we proved that our algorithms possess the state-of-the-art complexity bounds in terms of the number of gradient evaluations without assuming that the approximation errors of the regularizer r are L-smooth. We applied our new algorithms to three important problems in machine learning in order to demonstrate the advantages of combining the MM principle with SAGA, SVRG, and SARAH. Overall, MM-SARAH outperforms other stochastic algorithms under consideration. This is not surprising since the methods based on SAGA and SVRG have unvoidable limitations. In particular, SAGA requires storing the most recent gradient of each component function f_i while SVRG employs a pivot iterate $\tilde{x}^k$ that may be unchanged during many iterations and, thus, may no longer be highly correlated with the current iterate x^k.

Finally, nonconvexity and nonsmoothness are inherent in many problems in data science. The impact of our work stems from new accessible algorithms for such problems. Furthermore, we provided rigorous convergence guarantees and complexity analysis, which are important for data science practitioners who need reliable and efficient methods for solving complex optimization problems.

We employed the large-sum structure of the objective function, which is typical in regularized empirical risk minimization problems. By leveraging variance reduction techniques, we improved the convergence rate and reduced the computational cost. This is relevant for those who work with large-scale data sets and need scalable and fast algorithms.

We also employed surrogate functions to approximate the nonsmooth part of the objective function, prompting the application of the majorization-minimization principle. We provided general conditions on the surrogate functions and demonstrated how to verify them for various nonsmooth regularizers. This is useful for incorporating different types of regularization, such as sparsity, robustness, or low-rank matrices in various data science problems.

We demonstrated the effectiveness of the proposed algorithms on several real-world problems, such as sparse binary classification, sparse multiclass logistic regressions, and neural network training. This shows the practical applicability and potential impact of these algorithms in data science.

REFERENCES

[1] Z. Allen-Zhu, *Natasha* 2*: Faster non-convex optimization than SGD*, NIPS, 31 (2018).
[2] Z. Allen-Zhu and Y. Li, *Neon*2*: Finding local minima via first-order oracles*, NIPS, 31 (2018).

[3] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods*, Math. Program., 137 (2013), pp. 91–129.
[4] H. H. Bauschke and P. L. Combettes, eds., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math./Ouvrages Math. SMC 408, Springer, Berlin, 2017.
[5] A. Beck, *First-Order Methods in Optimization*, SIAM, Philadelphia, 2017.
[6] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.
[7] P. S. Bradley and O. L. Mangasarian, *Feature selection via concave minimization and support vector machines*, in Proceedings of the International Conference on Machine Learning, 1998.
[8] E. J. Candès, M. B. Wakin, and S. P. Boyd, *Enhancing sparsity by reweighted ℓ_1 minimization*, J. Fourier Anal. Appl., 14 (2008), pp. 877–905.
[9] E. Chouzenoux and J.-B. Fest, *Sabrina: A stochastic subspace majorization-minimization algorithm*, J. Optim. Theory Appl., 195 (2022), pp. 919–952.
[10] E. Chouzenoux and J.-C. Pesquet, *A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation*, IEEE Trans. Signal Process., 65 (2017), pp. 4770–4783.
[11] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, Berlin, 2011, pp. 185–212.
[12] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, NIPS, 27 (2014).
[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. R. Stat. Soc. Ser. B. Methodol., 39 (1977), pp. 1–22.
[14] D. Driggs, J. Tang, J. Liang, M. Davies, and C.-B. Schonlieb, *A stochastic proximal alternating minimization for nonsmooth and nonconvex optimization*, SIAM J. Imaging Sci., 14 (2021), pp. 1932–1970.
[15] C. Fang, C. J. Li, Z. Lin, and T. Zhang, *SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator*, NIPS, 31 (2018).
[16] B. Gao and L. Pavel, *On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning*, preprint, arXiv:1704.00805, 2017.
[17] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Trans. Image Process., 4 (1995), pp. 932–946.
[18] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: Methodology and convergence*, SIAM J. Optim., 19 (2008), pp. 1107–1130.
[19] L. Hien, D. Phan, and N. Gillis, *An inertial block majorization minimization framework for nonsmooth nonconvex optimization*, J. Mach. Learn. Res., 24 (2023), pp. 1–41.
[20] L. T. K. Hien, D. N. Phan, and N. Gillis, *Inertial alternating direction method of multipliers for non-convex non-smooth optimization*, Comput. Optim. Appl., 83 (2022), pp. 247–285.
[21] S. J. Reddi, S. Sra, B. Poczos, and A. J. Smola, *Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization*, NIPS, 29 (2016).
[22] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, NIPS, 26 (2013).
[23] A. Kaplan and R. Tichatschke, *Proximal point methods and nonconvex optimization*, J. Global Optim., 13 (1998), pp. 389–406.
[24] L. T. K. Hien, D. N. Phan, N. Gillis, M. Ahookhosh, and P. Patrinos, *Block Bregman majorization minimization with extrapolation*, SIAM J. Math. Data Sci., 4 (2022), pp. 1–25.
[25] D. Kovalev, S. Horváth, and P. Richtárik, *Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop*, in Algorithmic Learning Theory, PMLR, 2020.
[26] K. Lange, D. R. Hunter, and I. Yang, *Optimization transfer using surrogate objective functions*, J. Comput. Graph. Statist., 9 (2000), pp. 1–20.
[27] H. A. Le Thi, H. M. Le, D. N. Phan, and B. Tran, *Stochastic DCA for minimizing a large sum of DC functions with application to multi-class logistic regression*, Neural Networks, 132 (2020), pp. 220–231.
[28] H. A. Le Thi, H. P. H. Luu, H. M. Le, and T. P. Dinh, *Stochastic DCA with variance reduction and applications in machine learning*, J. Mach. Learn. Res., 23 (2022), pp. 1–44.

[29] H. A. Le Thi and T. Pham Dinh, *The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems*, Ann. Oper. Res., 133 (2005), pp. 23–46.
[30] Z. Li and J. Li, *A simple proximal stochastic gradient method for nonsmooth nonconvex optimization*, NIPS, 31 (2018).
[31] J. Mairal, *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM J. Optim., 25 (2015), pp. 829–855.
[32] B. Martinet, *Brève communication. régularisation d'inéquations variationnelles par approximations successives*, ESAIM Math. Model. Numer. Anal., 4 (1970), pp. 154–158.
[33] R. M. Neal and G. E. Hinton, *A view of the EM algorithm that justifies incremental, sparse, and other variants*, in Learning in Graphical Models, Springer, Berlin, 1998, pp. 355–368.
[34] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, Berlin, 2003.
[35] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.
[36] Y. Nesterov, *Lectures on Convex Optimization*, Springer Optim. Appl. 137, Springer, Berlin, 2018.
[37] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, *SARAH: A novel method for machine learning problems using stochastic recursive gradient*, in Proceedings of ICML, 2017, pp. 2613–2621.
[38] L. M. Nguyen, M. van Dijk, D. T. Phan, P. H. Nguyen, T.-W. Weng, and J. R. Kalagnanam, *Finite-sum smooth optimization with SARAH*, Comput. Optim. Appl., 82 (2022), pp. 561–593.
[39] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim., 1 (2014), pp. 127–239.
[40] S. N. Parizi, K. He, R. Aghajani, S. Sclaroff, and P. Felzenszwalb, *Generalized majorization-minimization*, in Proceedings of the International Conference on Machine Learning, PMLR, 2019, pp. 5022–5031.
[41] N. H. Pham, L. M. Nguyen, D. T. Phan, and Q. Tran-Dinh, *ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization*, J. Mach. Learn. Res., 21 (2020).
[42] T. Pham Dinh and H. A. Le Thi, *Convex analysis approach to D.C. programming: Theory, algorithms and applications*, Acta Math. Vietnam., 22 (1997), pp. 289–355.
[43] M. Razaviyayn, M. Hong, and Z.-Q. Luo, *A unified convergence analysis of block successive minimization methods for nonsmooth optimization*, SIAM J. Optim., 23 (2013), pp. 1126–1153.
[44] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statist., 22 (1951), pp. 400–407.
[45] H. Robbins and D. Siegmund, *A convergence theorem for non negative almost supermartingales and some applications*, in Optimizing Methods in Statistics, Elsevier, Amsterdam, 1971, pp. 233–257.
[46] R. Rockafellar and R. Wets, *Variational Analysis*, Springer, Berlin, 2009.
[47] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.
[48] R. T. Rockafellar, *The Theory of Subgradients and Its Applications to Problems of Optimization: Convex and Nonconvex Functions*, Heldermann Verlag, Berlin, 1981.
[49] M. Schmidt, N. Le Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Math. Program., 162 (2017), pp. 83–112.
[50] H. A. L. Thi, H. M. Le, P. D. Nhat, and B. Tran, *Stochastic DCA for the large-sum of non-convex functions problem and its application to group variable selection in classification*, in ICML, Vol. 70, 2017, pp. 3394–3403.
[51] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, *Spiderboost and momentum: Faster variance reduction algorithms*, NIPS, 32 (2019).
[52] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Statist., 38 (2010), pp. 894–942.
[53] L. Zhao, M. Mammadov, and J. Yearwood, *From convex to nonconvex: A loss function analysis for binary classification*, in Proceedings of the IEEE International Conference on Data Mining Workshops, 2010.