J Optim Theory Appl (2014) 160:832–853
DOI 10.1007/s10957-013-0409-2

Incrementally Updated Gradient Methods for Constrained and Regularized Optimization

Paul Tseng · Sangwoon Yun

Received: 26 July 2012 / Accepted: 19 August 2013 / Published online: 11 September 2013

Abstract We consider incrementally updated gradient methods for minimizing the sum of smooth functions and a convex function. This method can use a (sufficiently small) constant stepsize or, more practically, an adaptive stepsize that is decreased whenever sufficient progress is not made. We show that if the gradients of the smooth functions are Lipschitz continuous on the space of n-dimensional real column vectors or the gradients of the smooth functions are bounded and Lipschitz continuous over a certain level set and the convex function is Lipschitz continuous on its domain, then every cluster point of the iterates generated by the method is a stationary point. If in addition a local Lipschitz error bound assumption holds, then the method is linearly convergent.

Keywords Incrementally updated gradient method · Linear convergence · Error bound · Backpropagation · Neural network training · Regularization

1 Introduction

We consider a nonsmooth minimization problem whose objective is the sum of m smooth functions and a (possibly nonsmooth) convex function. Especially, m is large (exceeding 10^4). In this case, traditional gradient methods would be inefficient since they require evaluating all the gradients of smooth functions before updating the

Communicated by Masao Fukusima.

P. Tseng
Department of Mathematics, University of Washington, Seattle, WA 98195-4350, USA

S. Yun (✉)
Department of Mathematics Education, Sungkyunkwan University, Jongno-gu, Seoul 110-745, Republic of Korea
e-mail: yswmathedu@skku.edu

iterate. Incremental gradient methods, in contrast, update the iterate after evaluation of gradients for only one or a few smooth functions. In the unconstrained smooth minimization case (i.e., the convex function is the zero function), the smooth functions are selected for iteration according to a cyclic order for the basic form of the incremental gradient method (i.e., the gradient of only one smooth component function is used for updating the iterate). For global convergence of incremental gradient methods, the stepsize (also called "learning rate") needs to diminish to zero, which can lead to slow convergence; see [1–6]. If a constant stepsize is used, only convergence to an approximate solution can be shown [7, 8]. Methods to overcome this difficulty were proposed in [9, 10]. However, these methods need additional assumptions such that all the gradients of smooth component functions at a stationary point are zero, to achieve global convergence without the stepsize tending to zero. Moreover, its extension to the problem with a nonzero nonsmooth convex function is problematic. In the constrained case (i.e., the convex function is the indictor function for a nonempty closed convex set), [10] proposed projecting onto the feasible set after each cycle of m iterations, so feasibility is restored only at the end of each cycle.

Recently, Blatt, Hero, and Gauchman [11] proposed a method that computes the gradient of a single component function at each iteration, but instead of updating the iterate using this gradient, it uses the sum of m most recently computed gradients for the unconstrained smooth minimization case. Assuming the uniform boundedness and Lipschitz continuity of all the gradients of m smooth functions as well as the uniqueness of a stationary point and positive definiteness of Hessian of the sum of m smooth functions at the stationary point [11, Assumptions 1–4], the global convergence of this method with a sufficiently small stepsize is shown [11, Sect. 2.1]. If in addition each smooth component function is quadratic and the sum of m smooth functions is strictly convex, then the method achieves linear rate of convergence [11, Sect. 2.2]. Compared to the basic form of the incremental gradient method, this method requires more storage ($O(mn)$ instead of $O(n)$) and slightly more communication/computation per iteration but has the advantage that the global convergence can be achieved using a constant stepsize. The method in [11] may be viewed as belonging to a general class of gradient methods that update the gradients for only one or a few smooth functions at a time, which we call *incrementally updated gradient* (IUG) methods.

In this paper, we propose two IUG methods to solve the nonsmooth minimization problem whose objective is the sum of m smooth functions and a (possibly nonsmooth) convex function. We show the global convergence for the IUG method using a constant stepsize, assuming only the Lipschitz continuity of each gradient of m smooth functions; see Theorem 4.1. The linear convergence is shown assuming in addition that a local error bound holds; see Theorem 6.1. In particular, the linear convergence holds if each smooth component function is quadratic (not necessarily convex) and the convex function is polyhedral. Thus, compared to [11], we consider a more general method for solving a more general problem, and we show the global convergence and linear convergence under much weaker assumptions. The second IUG method uses adaptive stepsizes and hence is more practical, and it has a similar global convergence property as the first IUG method; see Sect. 5. Thus, our contributions are twofold: (i) we generalize the previous IUG method to handle constraints

and nonsmooth regularization, and (ii) we prove the global and linear convergence under much weaker assumptions.

2 Problem Setting and Framework for the IUG Method

In our notation, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors, ^{T} denotes transpose. For any $x \in \mathbb{R}^n$, x_j denotes the jth component of x, and $\|x\|_p = (\sum_{j=1}^n |x_j|^p)^{1/p}$ for $1 \le p < \infty$ and $\|x\|_\infty = \max_j |x_j|$. For simplicity, we write $\|x\| = \|x\|_2$. For any $x, y \in \mathbb{R}^n$, $\langle x, y\rangle = x^T y$ (so $\|x\| = \sqrt{\langle x, x\rangle}$. For $n \times n$ real symmetric matrices A, B, we write $A \succeq B$ (respectively, $A \succ B$) to mean that $A - B$ is positive semidefinite (respectively, positive definite). $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of A. We denote by I the identity matrix, and by 0_n the $n \times n$ matrix of zero entries. Unless otherwise specified, $\{x^k\}$ denotes the sequence $x^0, x^1, \ldots$. For any $\alpha \in \mathbb{R}$, $(\alpha)_+ = \max\{0, \alpha\}$.

In the applications of (supervised) learning, regression, pattern recognition, and data mining, for a given model $Y = S(x, X)$ of an input–output system parameterized by $x \in \mathbb{R}^n$, a training data set $(X_1, Y_1), \ldots, (X_m, Y_m) \in \mathbb{R}^p \times \mathbb{R}^q$, we seek an x that yields a small training error $S(x, X_i) \approx Y_i$. In a linear model case, $Y = \langle x, X\rangle$. In a feedforward neural network case with one hidden layer, $Y = \sum_{k=1}^N v_k \sigma(\langle X, u_k\rangle + \omega_k) + z$ and $x = (u_1, v_1, \omega_1, \ldots, u_N, v_N, \omega_N, z)$, with $\sigma : \mathbb{R} \to \mathbb{R}$ continuous and satisfying $\lim_{\theta\to\infty} \sigma(\theta) = 1$, $\lim_{\theta\to-\infty} \sigma(\theta) = 0$; see [2–6, 9, 12–15]. The training error is measured by a loss function $\ell : \mathbb{R}^q \times \mathbb{R}^q \to [0, \infty[$, so the error for the ith data point is

$$f_i(x) = \ell\big(S(x, X_i), Y_i\big).$$

Examples of a loss function include $\ell(Y', Y) = \frac{1}{2}\|Y' - Y\|^2$ in linear regression, $\ell(Y', Y) = \ln(1 + e^{Y'}) - YY'$ in logistic regression, and the piecewise-linear/quadratic Huber loss function. To avoid over-fitting and/or enable variable/feature selection and/or induce a sparse representation, a nonsmooth regularization term, such as the 1-norm or a weighted sum of 2-norms, may be added, as is done in compressed sensing, lasso, group lasso; see [16–21] and references therein. There may also be lower and upper bounds on the parameters. Then the resulting nonsmooth minimization problem has the form

$$\min_{x\in\mathbb{R}^n} F_c(x) := f(x) + cP(x), \tag{1}$$

where $c > 0$, $P : \mathbb{R}^n \to]-\infty, \infty]$ is a proper, convex, lower semicontinuous (lsc) function [22], and

$$f(x) := \sum_{i=1}^m f_i(x), \tag{2}$$

where each function f_i is real-valued and smooth (i.e., continuously differentiable) on an open subset of $\mathbb{R}^n$ containing $\operatorname{dom} P = \{x \mid P(x) < \infty\}$. A well-known special

case of (1) is smooth constrained convex optimization, for which P is the indicator function for a nonempty closed convex set $\mathcal{X} \subseteq \mathbb{R}^n$, i.e.,

$$P(x) = \begin{cases} 0 & \text{if } x \in \mathcal{X}, \\ \infty & \text{else.} \end{cases} \tag{3}$$

When P is the 1-norm, solving (1) would yield a model that is in some sense a minimal representation of the input/output system.

To efficiently solve (1), we propose two IUG methods that have the form

$$g^k = \sum_{i=1}^{m} \nabla f_i\big(x^{\tau_i^k}\big), \tag{4}$$

$$d^k = \arg\min_{d \in \mathbb{R}^n} \left\{ \langle g^k, d\rangle + \frac{1}{2}\langle d, H^k d\rangle + cP\big(x^k + d\big) \right\}, \tag{5}$$

$$x^{k+1} = x^k + \alpha_k d^k. \tag{6}$$

where $\tau_i^k \le k$ for $i = 1, \ldots, m$, $H^k \succ 0_n$, $\alpha_k \in]0, 1]$, and $x^0, x^{-1}, \ldots$ in $\operatorname{dom} P$ are given. The first IUG method uses a constant stepsize and is simpler to analyze. In particular, when $P \equiv 0$, (5)–(6) with $H^k = I$, $\alpha_k = \alpha$, and

$$\tau_i^k = \begin{cases} k & \text{if } i = (k \bmod m) + 1, \\ \tau_i^{k-1} & \text{otherwise,} \end{cases} \qquad 1 \le i \le m,\ k \ge m, \tag{7}$$

reduces to the method in [11]. A method of Fukushima and Mine [23] corresponds to $m = 1$, $H^k = \rho^k I$ ($\rho^k > 0$), and $\tau_i^k = k$ for all i, as does the iterative thresholding method for compressed sensing [17]; also see [20]. Thus the framework (4)–(6) is quite flexible and allows partially asynchronous updating of the component gradients [24, 25]. Moreover, we can accelerate convergence by using recent gradient approximations $g^k, g^{k-1}, \ldots$ to construct a Hessian approximation H^k as in, say, a limited-memory BFGS approach [26]. This is an additional advantage over the incremental gradient method, which has the following basic form:

$$x^{k+1} = x^k + \alpha_k \nabla f_{i_k}\big(x^k\big), \quad k = 0, 1, \ldots, \tag{8}$$

where i_k is chosen to cycle through $1, \ldots, m$ and $\alpha_k > 0$.

3 Properties of Search Directions

In this section, we collect the key properties of the search direction d^k given by (5) that will be used in our convergence analysis, i.e., proving Theorems 4.1, 5.1, and 6.1. In what follows, for any $x \in \operatorname{dom} P$, $g \in \mathbb{R}^n$, and $H \succ 0_n$, we denote

$$d_H^g(x) \ := \ \arg\min_d \left\{ \langle g, d\rangle + \frac{1}{2}\langle d, Hd\rangle + cP(x + d) \right\}. \tag{9}$$

Thus, $d^k = d_{H^k}^{g^k}(x^k)$. For H diagonal and P separable piecewise-linear/quadratic, $d_H^g(x)$ is computable in closed form. Some examples are given below.

- For P given by (3) with $\mathcal{X} = \prod_j [l_j, u_j]$, $d_H^g(x)_j = \text{mid}\{l_j - x_j, -\frac{g_j}{H_{jj}}, u_j - x_j\}$.
- For $P(x) = \|x\|_1$, $d_H^g(x)_j = -\text{mid}\{\frac{g_j - c}{H_{jj}}, x_j, \frac{g_j + c}{H_{jj}}\}$.
- For $P(x) = \|x\|_1 + \frac{\omega}{2}\|x\|^2$ $(\omega > 0)$ [18], $d_H^g(x)_j = -\text{mid}\{\frac{g_j - c + c\omega}{H_{jj} + c\omega}, x_j, \frac{g_j + c + c\omega}{H_{jj} + c\omega}\}$.

($\text{mid}\{a, b, c\}$ denotes the median (mid-point) of a, b, c.)

We have the following lemma, whose proof is identical to that of [20, Eq. (8)] and is thus omitted.

Lemma 3.1 *For any $x \in \text{dom}\, P$, $g \in \mathbb{R}^n$, and $H \succ 0_n$, let $d = d_H^g(x)$. Then*

$$\langle g, d\rangle + cP(x + d) - cP(x) \leq -\langle d, Hd\rangle.$$

We say that $x \in \mathbb{R}^n$ is *a stationary point* of F_c if $x \in \text{dom}\, P$ and $F_c'(x; d) \geq 0$ for all $d \in \mathbb{R}^n$. For any $x \in \text{dom}\, P$ and $H \succ 0_n$, let

$$d_H(x) := d_H^{\nabla f(x)}(x). \tag{10}$$

The following result from [20, Lemma 2] characterizes stationarity in terms of the zeros of d_H.

Lemma 3.2 *For any $H \succ 0_n$, an $x \in \text{dom}\, P$ is a stationary point of F_c if and only if $d_H(x) = 0$.*

For our linear convergence analysis (Theorem 6.1), we need in addition the following two lemmas. The first lemma shows that $\|d_H^g(x)\|$ changes not too fast with the quadratic coefficients H. Its proof is identical to that of [20, Lemma 3] and is thus omitted.

Lemma 3.3 *For any $x \in \text{dom}\, P$, $g \in \mathbb{R}^n$, $H \succ 0_n$, and $\tilde{H} \succ 0_n$, we have*

$$\left\|d_{\tilde{H}}^g(x)\right\| \leq \frac{1 + \lambda_{\max}(Q) + \sqrt{1 - 2\lambda_{\min}(Q) + \lambda_{\max}(Q)^2}}{2} \frac{\lambda_{\max}(H)}{\lambda_{\min}(\tilde{H})} \left\|d_H^g(x)\right\|, \tag{11}$$

where $Q = H^{-1/2}\tilde{H}H^{-1/2}$.

Next lemma gives an upper bound on $\langle g, x' - \bar{x}\rangle + cP(x') - cP(\bar{x})$, where $x' = x + \alpha d_H^g(x)$, in terms of quantities that diminish to zero as $g \approx \nabla f(x)$ and x nears a stationary point $\bar{x}$.

Lemma 3.4 *For any $x, \bar{x} \in \text{dom}\, P$, $\bar{\lambda} I \succeq H \succ 0_n$, and $\alpha \in]0, 1]$, we have*

$$\langle g, x' - \bar{x}\rangle + cP\left(x'\right) - cP(\bar{x}) \leq \bar{\lambda}\left\|d_H^g(x)\right\| \|x - \bar{x}\| - \Delta,$$

where $x' = x + \alpha d_H^g(x)$ and $\Delta = \langle g, d_H^g(x)\rangle + cP(x + d_H^g(x)) - cP(x)$.

Proof Let $d = d_H^g(x)$. By its definition (9) and Fermat's rule [31, Theorem 10.1],

$$d \in \arg\min_{d'} \{\langle g + Hd, d'\rangle + cP(x + d')\}.$$

Since $\bar{x} - x$ is a feasible solution of the above problem, we have

$$\langle g + Hd, d\rangle + cP(x + d) \leq \langle g + Hd, \bar{x} - x\rangle + cP(\bar{x}).$$

Using $\langle d, Hd\rangle \geq 0$, $H \preceq \bar{\lambda} I$, and the definition of Δ, this yields

$$\Delta \leq \langle g, \bar{x} - x\rangle + \bar{\lambda}\|d\|\|x - \bar{x}\| + cP(\bar{x}) - cP(x).$$

Since $\alpha \in]0, 1]$ and P is convex, this in turn yields

$$\begin{aligned}
\langle g, x' - \bar{x}\rangle + cP(x') - cP(x) &= \langle g, x - \bar{x}\rangle + \alpha\langle g, d\rangle + cP(x + \alpha d) - cP(x) \\
&\leq \langle g, x - \bar{x}\rangle + \alpha\big(\langle g, d\rangle + cP(x + d) - cP(x)\big) \\
&\leq \langle g, x - \bar{x}\rangle \\
&\leq \bar{\lambda}\|d\|\|x - \bar{x}\| + cP(\bar{x}) - cP(x) - \Delta,
\end{aligned}$$

where the second inequality uses Lemma 3.1 and $\alpha \geq 0$. This proves the desired result. □

4 IUG Method with Constant Stepsize

In this section, we study the global convergence of the IUG method (4)–(6) using a constant stepsize, which is easier to analyze. We describe this method formally below.

Algorithm 1

Step 1. Choose $x^0, x^{-1}, \cdots \in \mathrm{dom}\, P$ and $\alpha \in]0, 1]$. Initialize $k = 0$. Go to Step 2.
Step 2. Choose $H^k \succ 0_n$ and $0 \leq \tau_i^k \leq k$ for $i = 1, \ldots, m$, and compute g^k, d^k, and x^{k+1} by (4), (5), and (6) with $\alpha_k = \alpha$. Increment k by 1 and return to Step 2.

Note that $0 < \alpha \leq 1$ and the convexity of $\mathrm{dom}\, P$ ensure that $x^k \in \mathrm{dom}\, P$ for all k. We make the following assumptions about Algorithm 1:

Assumption 4.1

(a) $\tau_i^k \geq k - K$ *for all* i *and* k, *where* $K \geq 0$ *is an integer.*
(b) $\underline{\lambda} I \preceq H^k \preceq \bar{\lambda} I$ *for all* k, *where* $0 < \underline{\lambda} \leq \bar{\lambda}$.

Assumption 4.1(a) ensures that the gradient of f_i is updated at least once for every $K + 1$ consecutive iterations. This models a distributed computation setting where $\nabla f_1, \ldots, \nabla f_m$ are evaluated in a partially asynchronous manner by m processors

[24, Chap. 7]. As was remarked in Sect. 1, the method of Blatt et al. [11] corresponds to the special case of $P \equiv 0$, $H^k = I$, $K = m - 1$, and (7). The method of Fukushima and Mine [23] corresponds to the case of $H^k = \rho^k I$ ($\rho^k > 0$) and $K = 0$.

We make the following standard assumptions about $f_1, \ldots, f_m$:

Assumption 4.2

$$\left\|\nabla f_i(y) - \nabla f_i(z)\right\| \le L_i \|y - z\| \quad \forall y,\, z \in \operatorname{dom} P, \tag{12}$$

for some $L_i \ge 0$, $i = 1, \ldots, m$. *Let* $L = \sum_{i=1}^m L_i$.

We have the following global convergence result for the IUG method with sufficiently small constant stepsize.

Theorem 4.1 *Let* $\{x^k\}$, $\{d^k\}$, $\{H^k\}$ *be sequences generated by Algorithm 1 under Assumptions 4.1 and 4.2, and with* $\alpha < 2\underline{\lambda}/(L(2K+1))$, *and let* F_c *be bounded below. Then* $\{d^k\} \to 0$, *and every cluster point of* $\{x^k\}$ *is a stationary point of* (1).

Proof By (12), ∇f is Lipschitz continuous on dom P with constant L. Let

$$\Delta_k = \langle g^k, d^k \rangle + cP\left(x^k + d^k\right) - cP\left(x^k\right). \tag{13}$$

For each $k \in \{0, 1, \ldots\}$, we have

$$\begin{aligned} F_c\left(x^k + \alpha d^k\right) - F_c\left(x^k\right) &= f\left(x^k + \alpha d^k\right) - f\left(x^k\right) + cP\left(x^k + \alpha d^k\right) - cP\left(x^k\right) \\ &\le \alpha\left\langle \nabla f\left(x^k\right), d^k\right\rangle + \alpha^2 \frac{L}{2}\|d^k\|^2 \\ &\quad + \alpha\left(cP\left(x^k + d^k\right) - cP\left(x^k\right)\right) \\ &= \alpha\left\langle \nabla f\left(x^k\right) - g^k, d^k\right\rangle + \alpha^2 \frac{L}{2}\|d^k\|^2 + \alpha\Delta_k, \end{aligned}$$

where the first inequality uses the convexity of P, $\alpha \in]0, 1]$, and the Lipschitz continuity of ∇f on dom P [27, p. 667]. We also have

$$\begin{aligned} \left\|\nabla f\left(x^k\right) - g^k\right\| &= \left\|\sum_{i=1}^m \nabla f_i\left(x^k\right) - \nabla f_i\left(x^{\tau_i^k}\right)\right\| \le \sum_{i=1}^m \left\|\nabla f_i\left(x^k\right) - \nabla f_i\left(x^{\tau_i^k}\right)\right\| \\ &\le \sum_{i=1}^m L_i \left\|x^k - x^{\tau_i^k}\right\| \le \sum_{i=1}^m L_i \sum_{j=\tau_i^k}^{k-1} \alpha\|d^j\| \\ &\le \sum_{i=1}^m L_i \sum_{j=(k-K)_+}^{k-1} \alpha\|d^j\| = L \sum_{j=(k-K)_+}^{k-1} \alpha\|d^j\|. \end{aligned} \tag{14}$$

Combining the above two inequalities yields

$$F_c\big(x^k+\alpha d^k\big)-F_c\big(x^k\big)\le\alpha^2L\sum_{j=(k-K)_+}^{k-1}\|d^j\|\|d^k\|+\alpha^2\frac{L}{2}\|d^k\|^2+\alpha\Delta_k$$
$$\le\alpha^2\frac{L}{2}\sum_{j=(k-K)_+}^{k-1}\big(\|d^j\|^2+\|d^k\|^2\big)+\alpha^2\frac{L}{2}\|d^k\|^2-\alpha\underline{\lambda}\|d^k\|^2$$
$$\le\alpha\left(\alpha(K+1)\frac{L}{2}-\underline{\lambda}\right)\|d^k\|^2+\alpha^2\frac{L}{2}\sum_{j=(k-K)_+}^{k-1}\|d^j\|^2,\quad(15)$$

where the second inequality uses $\Delta_k\le-\langle d^k,H^kd^k\rangle\le-\underline{\lambda}\|d^k\|^2$ (see Lemma 3.1) and $ab\le a^2/2+b^2/2$.

Telescoping the above inequality using (6) yields

$$F_c\big(x^{k+1}\big)-F_c\big(x^0\big)\le\alpha\left(\alpha\frac{L}{2}(K+1)-\underline{\lambda}\right)\sum_{j=0}^{k}\|d^j\|^2+\alpha^2\frac{L}{2}K\sum_{j=0}^{k-1}\|d^j\|^2$$
$$\le\alpha\left(\alpha\frac{L}{2}(2K+1)-\underline{\lambda}\right)\sum_{j=0}^{k}\|d^j\|^2.$$

Since $\alpha<2\underline{\lambda}/(L(2K+1))$, this implies $F_c(x^k)\le F_c(x^0)$ for all k and either $F_c(x^k)\to-\infty$ or else $\sum_{j=0}^{\infty}\|d^j\|^2<\infty$, so that $\{d^k\}\to0$, and every limit point $\bar{x}$ of a convergent subsequence $\{x^k\}_{k\in\mathcal{K}}$ satisfies $\{x^{\tau_i^k}-x^k\}_{k\in\mathcal{K}}\to0$, and hence $\{g^k\}_{k\in\mathcal{K}}\to\nabla f(\bar{x})$. Then (5) implies that, for any $x\in\operatorname{dom}P$, we have

$$\langle g^k,d^k\rangle+\frac{1}{2}\langle d^k,H^kd^k\rangle+cP\big(x^k+d^k\big)$$
$$\le\langle g^k,x-x^k\rangle+\frac{1}{2}\big\langle x-x^k,H^k\big(x-x^k\big)\big\rangle+cP(x)$$

for all $k\in\mathcal{K}$, so, in the latter case, the lsc property of P yields in the limit that

$$cP(\bar{x})\le\big\langle\nabla f(\bar{x}),x-\bar{x}\big\rangle+\frac{1}{2}\big\langle x-\bar{x},\bar{H}(x-\bar{x})\big\rangle+cP(x)\quad\forall x\in\operatorname{dom}P,$$

where $\bar{H}$ is any cluster point of $\{H^k\}_{k\in\mathcal{K}}$. Since $H^k\succeq\underline{\lambda}I$ for all $k\in\mathcal{K}$, $\bar{H}\succ0_n$. Therefore, this shows that $d_{\bar{H}}(\bar{x})=0$, so that, by Lemma 3.2, $\bar{x}$ is a stationary point of (1). □

The only assumption Theorem 4.1 makes on the problem is the Lipschitz continuity of $\nabla f_1,\ldots,\nabla f_m$ on $\operatorname{dom}P$. This contrasts with the convergence result in [11, Sect. 2.1], which further assumes the boundedness of $\nabla f_1,\ldots,\nabla f_m$ as well as the uniqueness of a stationary point and global minimizer of f. Theorem 4.1 also covers more general problems and more general methods.

5 IUG Method with Adaptive Stepsize

The constant stepsize rule $\alpha_k = \alpha$ for all k has two practical drawbacks: (i) it requires knowledge of L, which may be difficult to estimate, (ii) the resulting stepsize is too conservative, leading to slow convergence. In the case of neural network training, various heuristic rules for choosing the stepsize have been proposed, the most popular of which entail keeping the stepsize fixed for as long as "progress" is made and decreasing the stepsize if otherwise. However, these heuristic rules are justified only by extensive experimentation (see [13] and references therein, [28, p. 124], and [29]). The following adaptive stepsize rule overcomes the above two drawbacks by making precise the notion of "progress" and backtracks to decrease the stepsize if progress is not made. The resulting IUG method is easy to implement and preserves the spirit of the heuristics.

Algorithm 2

Step 1. Choose $x^0, x^{-1}, \ldots \in \operatorname{dom} P$, $\underline{\alpha} \in]0, 1]$, $\beta \in]0, 1[$, and $\sigma > \frac{1}{2}$. Initialize $k = 0$. Go to Step 2.

Step 2. Choose $H^k \succ 0$ and $0 \le \tau_i^k \le k$ for $i = 1, \ldots, m$, compute g^k, d^k by (4), (5). Choose $\alpha_k^{\text{init}} \in [\underline{\alpha}, 1]$ and let α_k be the largest element of $\{\alpha_k^{\text{init}} \beta^j\}_{j=0,1,\ldots}$ satisfying

$$F_c\big(x^k + \alpha_k d^k\big) - F_c\big(x^k\big) \le -\sigma K L \|\alpha_k d^k\|^2 + \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \|\alpha_j d^j\|^2 \qquad (16)$$

and compute x^{k+1} by (6). Increment k by 1 and return to Step 2.

Algorithm 2 chooses the stepsize α_k adaptively by decreasing α_k whenever the nonmonotone descent condition (16) is violated. This is in the spirit of the Armijo rule [27, p. 29]. Condition (16) is motivated by the proof of Theorem 4.1 for the constant stepsize algorithm. In practice, L is not known a priori, but we can estimate L by starting with an arbitrary estimate of L and increasing L by, say, a factor of 2 whenever (16) fails to be satisfied when α_k is below $\bar{\alpha}$ defined in Theorem 5.1 below. This guarantees that (16) holds for some constant L for all k sufficiently large, which suffices for the global convergence proof in Theorem 5.1(b). Whether L is defined by Assumption 4.2 is actually not relevant.

Theorem 5.1 *Let* $\{x^k\}$, $\{d^k\}$, $\{H^k\}$, $\{\alpha_k\}$ *be sequences generated by Algorithm 2 under Assumptions 4.1 and 4.2. Then the following results hold.*

(a) *For each* $k \ge 0$, (16) *holds whenever* $\alpha_k \le \bar{\alpha}$, *where* $\bar{\alpha} = \frac{\lambda}{L(\sigma K + K/2 + 1/2)}$.

(b) *We have* $\alpha_k \ge \min\{\underline{\alpha}, \beta\bar{\alpha}\}$ *for all* k.

(c) *If* F_c *is bounded below, then* $\{d^k\} \to 0$, *and every cluster point of* $\{x^k\}$ *is a stationary point of* (1).

Proof (a) For each $k \in \{0, 1, \dots\}$, let Δ_k be given by (13). Suppose $\alpha_k \le \bar{\alpha}$. Then we have as in the proof of Theorem 4.1 that

$$\begin{aligned}
&F_c\big(x^k + \alpha_k d^k\big) - F_c\big(x^k\big) \\
&\quad \le L \sum_{j=(k-K)_+}^{k-1} \|\alpha_j d^j\| \|\alpha_k d^k\| + \frac{L}{2}\|\alpha_k d^k\|^2 + \alpha_k \Delta_k \\
&\quad \le \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \big(\|\alpha_j d^j\|^2 + \|\alpha_k d^k\|^2\big) + \frac{L}{2}\|\alpha_k d^k\|^2 - \alpha_k \underline{\lambda} \|d^k\|^2 \\
&\quad = \left((K+1)\frac{L}{2} - \frac{\underline{\lambda}}{\alpha_k}\right)\|\alpha_k d^k\|^2 + \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \|\alpha_j d^j\|^2 \\
&\quad \le -K\sigma L \|\alpha_k d^k\|^2 + \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \|\alpha_j d^j\|^2,
\end{aligned}$$

where the second inequality uses $\Delta_k \le -\langle d^k, H^k d^k\rangle \le -\underline{\lambda}\|d^k\|^2$ (see Lemma 3.1), and the last inequality uses $\alpha_k \le \bar{\alpha} = \frac{\underline{\lambda}}{L(\sigma K + K/2 + 1/2)}$. Thus, (16) holds.

(b) If $\alpha_k = \alpha_k^{\text{init}}$, then $\alpha_k \ge \underline{\alpha}$. Otherwise, $\alpha_k \le \beta \alpha_k^{\text{init}}$, in which case (a) implies $\alpha_k/\beta > \bar{\alpha}$.

(c) Let $\delta^k = \|\alpha_k d^k\|^2$. By (16),

$$F_c\big(x^{k+1}\big) - F_c\big(x^k\big) \le -\sigma K L \delta^k + \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \delta^j \quad \forall k.$$

Telescoping the above inequality yields

$$F_c\big(x^{k+1}\big) - F_c\big(x^0\big) \le -\sigma K L \sum_{j=0}^{k} \delta^j + \frac{KL}{2} \sum_{j=0}^{k-1} \delta^j \le -\left(\sigma - \frac{1}{2}\right) K L \sum_{j=0}^{k} \delta^j.$$

Since $\sigma > 1/2$, this implies $F_c(x^k) \le F_c(x^0)$ for all k and either $F_c(x^k) \to -\infty$ or else $\sum_{j=0}^{\infty} \delta^j < \infty$, so that $\{\delta^k\} \to 0$ or, equivalently, $\{\|\alpha_k d^k\|\} \to 0$. In the latter case, since $\inf_k \alpha_k > 0$ by (a), we have $\{d^k\} \to 0$. Then an argument identical to that used in the proof of Theorem 4.1 yields that every cluster point $\{x^k\}$ is a stationary point of (1). □

Assumption 4.2 can be replaced by the following assumption in which each ∇f_i is bounded and Lipschitz continuous on a certain level set and P is Lipschitz continuous on this level set intersected with dom P.

Assumption 5.1 *There exist scalars $\eta > F_c(x^0)$ and $\rho > 0$ such that, for $i = 1, \dots, m$, ∇f_i is bounded in norm by $\gamma_i \ge 0$ and Lipschitz continuous with constant*

$L_i \geq 0$ *on the convex set*

$$\mathcal{X}_\rho^\eta := \left\{x \in \mathbb{R}^n \mid F_c(x) \leq \eta\right\} + \rho\mathcal{B},$$

where $\mathcal{B} := \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$. *Moreover,* P *is Lipschitz continuous with constant* $L_p \geq 0$ *on* $\operatorname{dom} P \cap \mathcal{X}_\rho^\eta$.

Assumption 5.1 is satisfied when $f_1, \ldots, f_m$ are twice differentiable, P is Lipschitz continuous, and the level set $\{x \in \mathbb{R}^n \mid F_c(x) \leq \eta\}$ is bounded for some $\eta > F_c(x^0)$, as is the typical case with ℓ_1-regularized convex minimization such as ℓ_1-regularized linear least squares problem and ℓ_1-regularized logistic regression problem. The next lemma shows that, under Assumption 5.1 and reasonable assumptions on x, H, and g, $x + d_H^g(x)$ remains in $\operatorname{dom} P \cap \mathcal{X}_\rho^\eta$.

Lemma 5.1 *Under Assumption 5.1, for any* $x, x_1, \ldots, x_m \in \operatorname{dom} P \cap \mathcal{X}_\rho^\eta$ *with* $F_c(x) \leq \eta$, *any* $\underline{\lambda} \geq (\gamma + cL_p)/\rho$, *and any* $H \succ \underline{\lambda} I$, *we have*

$$x + d_H^g(x) \in \operatorname{dom} P \cap \mathcal{X}_\rho^\eta,$$

where $g = \sum_{i=1}^m \nabla f_i(x_i)$ *and* $\gamma = \gamma_1 + \cdots + \gamma_m$.

Proof Let $d = d_H^g(x)$. By definition (9), $x + d \in \operatorname{dom} P$. By Lemma 3.1,

$$\langle d, Hd\rangle \leq -\langle g, d\rangle - cP(x + d) + cP(x).$$

This, together with Assumption 5.1 and $H \succ \underline{\lambda} I$, yields

$$\underline{\lambda}\|d\|^2 \leq \langle d, Hd\rangle \leq \|g\|\|d\| + cL_p\|d\| \leq (\gamma + cL_p)\|d\|.$$

Hence, $0 \leq \|d\| \leq (\gamma + cL_p)/\underline{\lambda} \leq \rho$. Thus, provided that $F_c(x) \leq \eta$, this would imply $x + d \in \mathcal{X}_\rho^\eta$. □

By using Lemma 5.1, Theorems 4.1 and 5.1 still hold when Assumption 4.2 is replaced by Assumption 5.1.

Corollary 5.1 *Theorems 4.1 and 5.1 still hold under Assumption 5.1 instead of Assumption 4.2 and Assumption 4.1 with* $\underline{\lambda} \geq (\gamma + cL_p)/\rho$ *and* $\gamma = \gamma_1 + \cdots + \gamma_m$.

The stepsize rule for choosing α_k would depend on η, ρ, and $L_1, \ldots, L_m$ and, in the spirit of the Armijo–Goldstein stepsize rule for gradient descent methods, periodically checks if a certain descent condition is satisfied since the previous check was made and, if not, decreases the stepsize and restarts the method from when the previous check was made.

6 Convergence Rate Analysis

In this section we analyze the asymptotic convergence rate of Algorithm 1. In what follows, $\bar{\mathcal{X}}$ denotes the set of stationary points of F_c, and

$$\operatorname{dist}(x, \bar{\mathcal{X}}) := \min_{\bar{x} \in \bar{\mathcal{X}}} \|x - \bar{x}\| \quad \forall x \in \mathbb{R}^n.$$

Assumption 6.1

(a) $\bar{\mathcal{X}} \neq \emptyset$, *and, for any* $\zeta \geq \min_x F_c(x)$, *there exist scalars* $\tau > 0$ *and* $\epsilon > 0$ *such that*

$$\operatorname{dist}(x, \bar{\mathcal{X}}) \leq \tau \left\| d_I(x) \right\| \quad \text{whenever } F_c(x) \leq \zeta, \ \left\| d_I(x) \right\| \leq \epsilon.$$

(b) *There exists a scalar* $\delta > 0$ *such that*

$$\|x - y\| \geq \delta \quad \text{whenever } x \in \bar{X}, \ y \in \bar{X}, \ F_c(x) \neq F_c(y).$$

Assumption 6.1(a) is a local Lipschitzian error bound assumption, saying that the distance from x to $\bar{X}$ is locally in the order of the norm of the residual at x; see [20] and references therein. Assumption 6.1(b) says that the isocost surfaces of F_c restricted to the solution set $\bar{X}$ are "properly separated." Assumption 6.1(b) holds automatically if f is convex or f is quadratic and P is polyhedral; see [20, 30] for further discussions. We have from [20, Theorem 6.1] the following sufficient conditions for Assumption 6.1(a) to hold.

Proposition 6.1 *Suppose that* $\bar{\mathcal{X}} \neq \emptyset$ *and any of the following conditions hold.*

(C1) f *is strongly convex and satisfies* (12) *for some* $L_i \geq 0$.
(C2) f *is quadratic.* P *is polyhedral.*
(C3) $f(x) = g(Ex) + \langle q, x \rangle$ *for all* $x \in \mathbb{R}^n$, *where* $E \in \mathbb{R}^{m \times n}$, $q \in \mathbb{R}^n$, *and* g *is a strongly convex differentiable function on* $\mathbb{R}^m$ *with* ∇g *Lipschitz continuous on* $\mathbb{R}^m$. P *is polyhedral.*
(C4) $f(x) = \max_{y \in Y}\{\langle Ex, y \rangle - g(y)\} + \langle q, x \rangle$ *for all* $x \in \mathbb{R}^n$, *where* Y *is a polyhedral set in* $\mathbb{R}^m$, $E \in \mathbb{R}^{m \times n}$, $q \in \mathbb{R}^n$, *and* g *is a strongly convex differentiable function on* $\mathbb{R}^m$ *with* ∇g *Lipschitz continuous on* $\mathbb{R}^m$. P *is polyhedral.*

Then Assumption 6.1(a) *holds.*

Next theorem establishes, under Assumptions 4.1, 4.2, and 6.1, the linear rate of convergence of the IUG method with sufficiently small constant stepsize. Its proof, based on ideas in [25] for a partially asynchronous block-coordinate gradient-projection method, uses Lemmas 3.3 and 3.4.

Theorem 6.1 *Let* $\{x^k\}$, $\{d^k\}$, $\{H^k\}$ *be sequences generated by Algorithm* 1 *under Assumptions* 4.1, 4.2, *and* 6.1. *Suppose that there exists a scalar* $\bar{\alpha} > 0$, *depending on* K, $\underline{\lambda}$, $\bar{\lambda}$, $L = \sum_{i=1}^m L_i$, *and* τ (*see* (19)) *only, such that if* $\lim_{k \to \infty} F_c(x^k) > -\infty$ *and* $0 < \alpha < \bar{\alpha}$. *Then the sequence* $\{x^k\}$ *converges at least linearly to an element of* $\bar{\mathcal{X}}$ *with a K-step convergence ratio of* $\sqrt{1 - C\alpha}$, *where* $C > 0$ *is some scalar constant.*

Proof Let $\hat{d}^k = d_{H^k}(x^k)$. By Fermat's rule [31, Theorem 10.1],

$$d^k \in \arg\min_d \langle g^k + H^k d^k, d\rangle + cP\big(x^k + d\big),$$
$$\hat{d}^k \in \arg\min_d \big\langle \nabla f\big(x^k\big) + H^k \hat{d}^k, d\big\rangle + cP\big(x^k + d\big).$$

Hence,

$$\langle g^k + H^k d^k, d^k\rangle + cP\big(x^k + d^k\big) \le \langle g^k + H^k d^k, \hat{d}^k\rangle + cP\big(x^k + \hat{d}^k\big),$$
$$\big\langle \nabla f\big(x^k\big) + H^k \hat{d}^k, \hat{d}^k\big\rangle + cP\big(x^k + \hat{d}^k\big) \le \big\langle \nabla f\big(x^k\big) + H^k \hat{d}^k, d^k\big\rangle + cP\big(x^k + d^k\big).$$

Summing the above two inequalities and rearranging terms, we have

$$\big\langle \nabla f\big(x^k\big) - g^k, d^k - \hat{d}^k\big\rangle \ge \langle H^k d^k - H^k \hat{d}^k, d^k - \hat{d}^k\rangle \ \ge\ \underline{\lambda}\|d^k - \hat{d}^k\|^2.$$

Since the left-hand side is at most $\|\nabla f(x^k) - g^k\|\|d^k - \hat{d}^k\|$, this yields

$$\underline{\lambda}\|d^k - \hat{d}^k\| \le \big\|\nabla f\big(x^k\big) - g^k\big\| \le L \sum_{j=(k-K)_+}^{k-1} \alpha\|d^j\|,$$

where the second inequality uses (14). Hence,

$$\|\hat{d}^k\| \le \frac{L\alpha}{\underline{\lambda}} \sum_{j=(k-K)_+}^{k-1} \|d^j\| + \|d^k\|. \tag{17}$$

By Theorem 4.1 and $\lim_{k\to\infty} F_c(x^k) > -\infty$, if α is sufficiently small (in fact, it suffices to take $\alpha < 2\underline{\lambda}/(L(2K+1))$, this implies $F_c(x^k) \le F_c(x^0)$ for all k and $\sum_{j=0}^{\infty}\|d^j\|^2 < \infty$, so that $\sum_{j=(k-K)_+}^{k-1}\|d^j\| \to 0$ and $\{d^k\} \to 0$. By (17), $\{\hat{d}^k\} \to 0$. By Lemma 3.3 with $H = H^k$ and $\tilde{H} = I$, we have

$$\big\|d_I\big(x^k\big)\big\| \le \tilde{\lambda}\,\|\hat{d}^k\| \quad \forall k, \tag{18}$$

where $\tilde{\lambda} = \bar{\lambda}(1 + 1/\underline{\lambda} + \sqrt{1 - 2/\bar{\lambda} + (1/\underline{\lambda})^2})/2$. Hence, $\{d_I(x^k)\} \to 0$. Then, by Assumption 6.1(a), there exist $\bar{k}$ and $\tau > 0$ such that

$$\|x^k - \bar{x}^k\| \le \tau\big\|d_I\big(x^k\big)\big\| \quad \forall k \ge \bar{k}, \tag{19}$$

where $\bar{x}^k \in \bar{\mathcal{X}}$ satisfies $\|x^k - \bar{x}^k\| = \mathrm{dist}(x^k, \bar{\mathcal{X}})$. Since $\{d_I(x^k)\} \to 0$, this implies $\{x^k - \bar{x}^k\} \to 0$. Since $\{x^{k+1} - x^k\} = \{\alpha d^k\} \to 0$, this and Assumption 6.1(b) imply that $\{\bar{x}^k\}$ eventually settles down at some isocost surface of F_c, i.e., there exist an index $\hat{k} \ge \bar{k}$ and a scalar $\bar{\upsilon}$ such that

$$F_c\big(\bar{x}^k\big) = \bar{\upsilon} \quad \forall k \ge \hat{k}. \tag{20}$$

Fix any index $k \in \mathcal{K},\ k \geq \hat{k}$. Since $\bar{x}^k$ is a stationary point of F_c, we have

$$\langle \nabla f(\bar{x}^k), x^k - \bar{x}^k \rangle + cP(x^k) - cP(\bar{x}^k) \geq 0.$$

We also have from the mean value theorem that

$$f(x^k) - f(\bar{x}^k) = \langle \nabla f(\psi^k), x^k - \bar{x}^k \rangle$$

for some ψ^k lying on the line segment joining x^k with $\bar{x}^k$. Since x^k and $\bar{x}^k$ lie in the convex set $\mathrm{dom}P$, so does ψ^k. Combining these two relations and using (20), we obtain

$$\begin{aligned}
\bar{\upsilon} - F_c(x^k) &\leq \langle \nabla f(\bar{x}^k) - \nabla f(\psi^k), x^k - \bar{x}^k \rangle \\
&\leq \|\nabla f(\bar{x}^k) - \nabla f(\psi^k)\| \|x^k - \bar{x}^k\| \\
&\leq \sum_{i=1}^m \|\nabla f_i(\bar{x}^k) - \nabla f_i(\psi^k)\| \|x^k - \bar{x}^k\| \\
&\leq \sum_{i=1}^m L_i \|\bar{x}^k - \psi^k\| \|x^k - \bar{x}^k\| \leq L\|x^k - \bar{x}^k\|^2,
\end{aligned}$$

where the fourth inequality uses (12), and the fifth inequality uses the inequality $\|\psi^k - \bar{x}^k\| \leq \|x^k - \bar{x}^k\|$. This, together with $\{x^k - \bar{x}^k\} \to 0$, proves

$$\liminf_{k\to\infty} F_c(x^k) \geq \bar{\upsilon}. \tag{21}$$

From the first inequality in (15), for each index $k \geq \tilde{k}$, where $\tilde{k} = \max\{\hat{k}, K\}$, we have

$$\begin{aligned}
F_c(x^k + \alpha d^k) - F_c(x^k) &\leq \alpha^2 L \sum_{j=k-K}^{k-1} \|d^j\| \|d^k\| + \alpha^2 \frac{L}{2} \|d^k\|^2 + \alpha \Delta_k \\
&\leq \alpha^2 \frac{L}{2} \sum_{j=k-K}^{k-1} \left(\|d^j\|^2 + \|d^k\|^2\right) + \alpha^2 \frac{L}{2} \|d^k\|^2 + \alpha \Delta_k \\
&\leq \alpha \left(1 - \alpha(K+1)\frac{L}{2\underline{\lambda}}\right) \Delta_k - \alpha^2 \frac{L}{2\underline{\lambda}} \sum_{j=k-K}^{k-1} \Delta_j,
\end{aligned}$$

where the second inequality uses $ab \leq a^2/2 + b^2/2$, and the last inequality uses $\underline{\lambda}\|d^j\|^2 \leq \langle d^j, H^j d^j \rangle \leq -\Delta_j$ (see Lemma 3.1). Applying the above argument successively to $k, k+1, \ldots, k+K-1$, we obtain

$$F_c(x^{k+K}) - F_c(x^k) \leq \alpha \left(1 - \alpha(K+1)\frac{L}{2\underline{\lambda}}\right) \sum_{j=k}^{k+K-1} \Delta_j - \alpha^2 \frac{LK}{2\underline{\lambda}} \sum_{j=k-K}^{k+K-1} \Delta_j$$

$$= \alpha\left(1-\alpha(2K+1)\frac{L}{2\underline{\lambda}}\right)\sum_{j=k}^{k+K-1}\Delta_j - \alpha^2\frac{LK}{2\underline{\lambda}}\sum_{j=k-K}^{k-1}\Delta_j$$

$$\le C_1\sum_{j=k}^{k+K-1}\alpha\Delta_j - C_2\sum_{j=k-K}^{k-1}\alpha^2\Delta_j, \tag{22}$$

where $C_1 = 1-\alpha C_2$ and $C_2 = (K+.5)L/\underline{\lambda}$.

Also, we have from (21) that

$$\begin{aligned}
&F_c\big(x^{k+1}\big) - \bar{\upsilon}\\
&= f\big(x^{k+1}\big) + cP\big(x^{k+1}\big) - f\big(\bar{x}^k\big) - cP\big(\bar{x}^k\big)\\
&= \big\langle \nabla f\big(\tilde{x}^k\big), x^{k+1}-\bar{x}^k\big\rangle + cP\big(x^{k+1}\big) - cP\big(\bar{x}^k\big)\\
&= \big\langle \nabla f\big(\tilde{x}^k\big) - g^k, x^{k+1}-\bar{x}^k\big\rangle + \langle g^k, x^{k+1}-\bar{x}^k\rangle + cP\big(x^{k+1}\big) - cP\big(\bar{x}^k\big)\\
&\le \sum_{i=1}^m \big\|\nabla f_i\big(\tilde{x}^k\big) - \nabla f_i\big(x^{\tau_i^k}\big)\big\|\|x^{k+1}-\bar{x}^k\| + \bar{\lambda}\|x^k-\bar{x}^k\|\|d^k\| - \Delta_k\\
&\le \sum_{i=1}^m L_i\|\tilde{x}^k - x^{\tau_i^k}\|\|x^{k+1}-\bar{x}^k\| + \bar{\lambda}\|x^k-\bar{x}^k\|\|d^k\| - \Delta_k\\
&\le \sum_{i=1}^m L_i\big(\|x^{k+1}-x^k\| + \|x^k-\bar{x}^k\| + \|x^k - x^{\tau_i^k}\|\big)\|x^{k+1}-\bar{x}^k\|\\
&\quad + \bar{\lambda}\|x^k-\bar{x}^k\|\|d^k\| - \Delta_k,
\end{aligned} \tag{23}$$

where the second step uses the mean value theorem with $\tilde{x}^k$ a point lying on the segment joining x^{k+1} with $\bar{x}^k$, the fourth step uses Lemma 3.4, the fifth step uses (12), and the last step uses the inequalities $\|\tilde{x}^k - x^{\tau_i^k}\| \le \|\tilde{x}^k - x^k\| + \|x^k - x^{\tau_i^k}\|$ and $\|\tilde{x}^k - x^k\| \le \|x^{k+1}-x^k\| + \|x^k-\bar{x}^k\|$.

Using $\|x^{k+1}-\bar{x}^k\| \le \|x^{k+1}-x^k\| + \|x^k-\bar{x}^k\|$ and $\|x^{k+1}-x^k\| = \alpha\|d^k\|$, we see from (17)–(19) that the right-hand side of (23) is bounded above by

$$C_0\left(\sum_{j=k-K}^{k-1}\alpha\|d^j\|\right)^2 + C_0\sum_{j=k-K}^{k-1}\alpha\|d^j\|\|d^k\| + C_0\|d^k\|^2 - \Delta_k \tag{24}$$

for all $k \ge \tilde{k}$, where C_0 is depending on $L, \underline{\lambda}, \bar{\lambda}, \tau$ only.

By using $ab \le (a^2+b^2)/2$ and $\underline{\lambda}\|d^j\|^2 \le \langle d^j, H^j d^j\rangle \le -\Delta_j = |\Delta_j|$, we have from (23) and (24) that

$$\begin{aligned}
&F_c\big(x^{k+1}\big) - \bar{\upsilon}\\
&\le \frac{C_0}{2}\sum_{j=k-K}^{k-1}\left(\sum_{i=k-K}^{k-1}\alpha\big(\|d^i\|^2 + \|d^j\|^2\big) + \alpha\big(\|d^j\|^2 + \|d^k\|^2\big)\right) + C_0\|d^k\|^2 - \Delta_k
\end{aligned}$$

$$= \frac{C_0}{2}\left(\sum_{j=k-K}^{k-1}\alpha\big(\|d^j\|^2 + K\|d^j\|^2\big) + K\sum_{i=k-K}^{k-1}\alpha\|d^i\|^2 + (\alpha K+2)\|d^k\|^2\right) - \Delta_k$$

$$\le C_3\sum_{j=k-K}^{k-1}\alpha|\Delta_j| + C_3|\Delta_k| \tag{25}$$

for all $k \ge \tilde{k}$, where $C_3 = (K+1)C_0/\underline{\lambda} + 1$.

Since (25) holds for all $k \ge \tilde{k}$, it holds in particular when k is replaced by $k+K-1$. This, together with $\alpha \in (0,1]$, implies that

$$\begin{aligned} F_c\big(x^{k+K}\big) - \bar{\upsilon} &\le C_3\sum_{j=k-1}^{k+K-2}\alpha|\Delta_j| + C_3|\Delta_{k+K-1}| \\ &\le C_3\sum_{j=k}^{k+K-1}|\Delta_j| + C_3\sum_{j=k-K}^{k-1}\alpha|\Delta_j|. \end{aligned} \tag{26}$$

By using the bounds (21), (22), and (26), we can now prove the linear convergence of $\{x^k\}$. To simplify the notation, let

$$e_k = F_c\big(x^k\big) - \bar{\upsilon}, \qquad \Gamma_k = \sum_{j=k-K}^{k-1}\alpha^2|\Delta_j|,$$

for all $k \ge \tilde{k}$. Then, we have from (21), (22), and (26), respectively, that, for any $k \ge \tilde{k}$,

$$\begin{aligned} 0 &\le \liminf_{k\to\infty} e_k, \\ e_{k+K} &\le e_k - \alpha^{-1}C_1\Gamma_{k+K} + C_2\Gamma_k, \\ e_{k+K} &\le \alpha^{-2}C_3\Gamma_{k+K} + \alpha^{-1}C_3\Gamma_k. \end{aligned}$$

By proceeding as in the statement of [25, pp. 614–616], there exists a scalar $\bar{\alpha} > 0$ (depending on $L, K, \underline{\lambda}, \bar{\lambda}, \tau$ only) such that if $0 < \alpha < \bar{\alpha}$, then, for all $r = 0, 1, 2, \ldots,$

$$e_{\tilde{k}+rK} \le a\rho^{r-1}, \quad \Gamma_{\tilde{k}+rK} \le b\rho^{r-1}, \tag{27}$$

where $\rho = 1 - \alpha C$, $C = \frac{C_1}{2C_3+2C_1}$, a and b are taken sufficiently large so that

$$e_{\tilde{k}} \le a, \quad e_{\tilde{k}+K} \le a, \quad \Gamma_{\tilde{k}} \le b, \quad \Gamma_{\tilde{k}+K} \le b.$$

(27) implies that $\{\Gamma_k\}$ converges at least linearly with a K-step convergence ratio of $1-\alpha C$. Since $\|x^k - x^{k-K}\|^2 \le \alpha^2\sum_{j=k-K}^{k-1}\|d^j\|^2 \le \frac{1}{\underline{\lambda}}\Gamma_k$ for all k (see Lemma 3.1 and the definition of Γ_k), this shows that $\{x^k\}$ converges at least linearly with a K-step convergence ratio of $\sqrt{1-\alpha C}$, which is at most $1 - \alpha C/2$. □

7 Numerical Illustration

In this section, we briefly describe our implementation of Algorithms 1 (Alg1) and 2 (Alg2) and report the performance of them with ℓ_1-regularized logistic regression problems with randomly generated data.

The ℓ_1-regularized logistic regression problem has the following form:

$$\min_{x\in\mathbb{R}^{n-1},\, y\in\mathbb{R}} \ \frac{1}{m}\sum_{i=1}^{m} \log\bigl(1+\exp\bigl(-\bigl(x^T a_i + y b_i\bigr)\bigr)\bigr) + c\|x\|_1, \tag{28}$$

where $c>0$, $a_i = b_i z_i$, and $(z_i, b_i) \in \mathbb{R}^{n-1}\times\{-1,1\}$, $i=1,\ldots,m$, are a given set of (observed or training) examples. Here, we assume that $m>n$ and there are $j,k\in\{1,\ldots,m\}$ such that $b_j=1$ and $b_k=-1$. Since the Hessian of the function $\log(1+\exp(-(x^T a_i + y b_i)))$ is

$$\frac{\exp(-(x^T a_i + y b_i))}{(\exp(-(x^T a_i + y b_i))+1)^2}\begin{pmatrix} a_i \\ b_i\end{pmatrix}\bigl(a_i^T \ b_i\bigr),$$

we can set $L_i = \frac{a_i^T a_i + b_i^2}{4m}$, and so we can take $L = \sum_{i=1}^m \frac{a_i^T a_i + b_i^2}{4m}$.

For Algorithm 1, we choose the constant stepsize $\alpha = 1/L(K+0.5+\varepsilon)$ with $\varepsilon = 10^{-6}$ and $H^k = I$. For Algorithm 1, we choose $\bar{\alpha} = 1/L(1.1K+0.5)$, $H^k = I$, and $\sigma = 0.6$. The stepsize α_k is chosen by the rule (16) with

$$\beta = 0.5, \quad \underline{\alpha} = 10^{-7}, \quad \alpha_0^{\text{init}} = 1, \quad \alpha_k^{\text{init}} = \max\left\{\underline{\alpha}, \min\left\{1, \frac{\alpha_{k-1}}{\beta}\right\}\right\} \quad \forall k \geq 1.$$

We experimented with other values of the above parameters, but the above choice works well. And, for τ_i^k in (4), the natural scheme would be cyclic as follows:

$$\tau_i^k = \begin{cases} k & \text{if } i = (k \bmod (K+1))\cdot m_K + j, \\ \tau_i^{k-1} & \text{otherwise,}\end{cases} \qquad 1\leq i\leq m,\ 1\leq j\leq m_K,\ k\geq m, \tag{29}$$

with $m_K = \frac{m}{K+1}$. But, the cyclic scheme with random permutation (i.e., we randomly reorder $f_1,\ldots,f_m$ after each cycle of $K+1$ iterations) overall works better than the ordinary cyclic scheme (29). Hence, we use the cyclic scheme with random permutation for all experiments.

We terminate Algorithms 1 and 2 when

$$\|d^k\| \leq \text{Tol}, \tag{30}$$

where Tol is a moderately small tolerance.

All runs are performed on a Desktop with Intel Core i5-2400 CPU (3.10 GHz) and 4 GB Memory, running 32-bit Windows 7 and MATLAB (Version 8.1). Throughout the experiments, we choose the initial iterate to be $(x^0, y^0) = (0,0)$ for Algorithms 1 and 2.

Table 1 reports the number of iterations, the number of times that $\nabla f_1,\ldots,\nabla f_m$ have been evaluated, the number of times that $\sum_{i=1}^m f_i$ has been evaluated, the final

Table 1 Test results with random data sets with $m = 1000$ and $n = 100$ with Tol $= 5 \times 10^{-4}$

$K+1$		$c = 0.047262$		$c = 0.0473915$	
		Alg1	Alg2	Alg1	Alg2
1	iters	1163	69	1199	71
	ngrad[1]	1164000	70000	1200000	72000
	nfunc[2]		71		73
	obj	0.242861	0.242861	0.248551	0.248551
	CPU	75.8	5.73	77.9	5.87
2	iters	3475	67	3590	71
	ngrad[1]	1738500	34500	180500	36500
	nfunc[2]		72		76
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	114	3.40	117	4.02
5	iters	10433	82	10773	82
	ngrad[1]	2087600	17400	2155600	17400
	nfunc[2]		113		111
	obj	0.242861	0.242860	0.248551	0.248550
	CPU	136	2.54	140	2.46
10	iters	22026	156	22743	156
	ngrad[1]	2203600	16600	2275300	16600
	nfunc[2]		313		313
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	143	3.73	149	3.74
20	iters	45214	420	46684	448
	ngrad[1]	2261700	22000	2335200	23400
	nfunc[2]		843		899
	obj	0.242861	0.242861	0.248551	0.248551
	CPU	150	8.69	154	9.25

[1] Denotes the number of times that $\nabla f_1, \ldots, \nabla f_m$ have been evaluated

[2] Denotes the number of times that $\sum_{i=1}^{m} f_i$ has been evaluated

objective value, and the CPU time (in seconds) for solving randomly generated problems described in [32] to see the effect of the choice of the number (i.e., K) of updated gradients at each iteration. Each randomly generated problem has an equal number of positive and negative examples. Features of positive (negative) examples are independent and identically distributed, drawn from a normal distribution $\mathcal{N}(\nu, 1)$, where ν is in turn drawn from a uniform distribution on $[0, 1]([-1, 0])$. For each instance, we chose $c = 0.1c_{\max}$, where $c_{\max} = \frac{1}{m}\|\frac{m_-}{m}\sum_{b_i=1} a_i + \frac{m_+}{m}\sum_{b_i=-1} a_i\|_\infty$, m_- is the number of negative examples, and m_+ is the number of positive examples. If $c \geq c_{\max}$, we get a maximally sparse weight vector, i.e., $x = 0$; see [32] for details. We note that, to see the effect of the choice of K at each iteration, we do not use multiplications of a matrix and a vector when more than one gradient is updated

Table 1 (*Continued*)

$K+1$		$c = 0.047262$		$c = 0.0473915$	
		Alg1	Alg2	Alg1	Alg2
50	iters	114780	1087	118510	1141
	ngrad[1]	2296600	22740	2371200	23820
	nfunc[2]		2178		2286
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	157	20.4	160	21.2
100	iters	230708	2176	238211	2287
	ngrad[1]	2308080	22760	2383110	23870
	nfunc[2]		4357		4579
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	162	40.0	168	41.2
200	iters	462584	4358	477637	4567
	ngrad[1]	2313920	22790	2389185	23835
	nfunc[2]		8722		9140
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	176	77.8	179	80.6
500	iters	1158253	15689	1195868	9428
	ngrad[1]	2317506	32378	2392736	19856
	nfunc[2]		31386		18863
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	210	276	217	164
1000	iterations	2317690	18304	2392912	18927
	ngrad[1]	2318690	19304	2393912	19927
	nfunc[2]		36616		37862
	obj	0.242861	0.242861	0.248551	0.248550
	CPU	266	317	278	334

[1] Denotes the number of times that $\nabla f_1, \ldots, \nabla f_m$ have been evaluated

[2] Denotes the number of times that $\sum_{i=1}^{m} f_i$ has been evaluated

and a function evaluation is required. Also note that, when $K = 0$, i.e., all gradients $\nabla f_1, \ldots, \nabla f_m$ are updated at each iteration, Algorithm 1 becomes a proximal gradient method [33]. From Table 1, Alg2 with $K = 4$ performs better than all others. Since the Lipschitz constant is large, the constant stepsize of Alg1 is small even for $K = 0$, and hence Alg1 converges slowly. However, Alg2 is slower than Alg1 when K is large. Note that the incremental scheme, i.e., updating some of gradients, works only for Alg2 in this numerical example. Alg1 with large K may be a good choice in the case where only partial data can be accessed to use at each iteration.

In the following, we compare Algorithm 2 to an incrementally updated gradient method with a heuristic rule for choosing the stepsize [13, 28]. We call the heuristic incrementally updated gradient method IUG-h and describe the rule for choosing the

Table 2 Comparing Algorithm 2 to IUG-h with $m = 1000$ and $n = 100$

$K+1$		$c = 0.047262$		$c = 0.0473915$	
		IUG-h	Alg2	IUG-h	Alg2
1	iters	70	69	71	71
	ngrad[1]	71000	70000	72000	72000
	nfunc[2]	71	71	72	73
	obj	0.242861	0.242861	0.248551	0.248551
	CPU	5.82	5.73	5.86	5.87
5	iters	143	82	124	82
	ngrad[1]	29600	17400	25800	17400
	nfunc[2]	144	113	125	111
	obj	0.242860	0.242860	0.248550	0.248550
	CPU	4.35	2.54	3.76	2.46
20	iters	628	420	560	448
	ngrad[1]	32400	22000	29000	23400
	nfunc[2]	629	843	561	899
	obj	0.242861	0.242861	0.248550	0.248551
	CPU	13.0	8.69	11.4	9.25
100	iters	6284	2176	6810	2287
	ngrad[1]	63840	22760	69100	23870
	nfunc[2]	6285	4357	6811	4579
	obj	0.242861	0.242861	0.248550	0.248550
	CPU	113	40.0	121	41.2
200	iters	10969	4358	12834	4567
	ngrad[1]	55845	22790	65170	23835
	nfunc[2]	10970	8722	12835	9140
	obj	0.242861	0.242861	0.248550	0.248550
	CPU	208	77.8	234	80.6
500	iters	63796	15689	64626	9428
	ngrad[1]	128592	32378	130252	19856
	nfunc[2]	63797	31386	64627	18863
	obj	0.242861	0.242861	0.248550	0.248550
	CPU	1290	276	1230	164

[1] Denotes the number of times that $\nabla f_1, \ldots, \nabla f_m$ have been evaluated

[2] Denotes the number of times that $\sum_{i=1}^{m} f_i$ has been evaluated

stepsize below

$$\alpha_{k+1} = \begin{cases} \alpha_k & \text{if } F_c(x^{k+1}) < F_c(x^k), \\ \max\{0.99\alpha_k, \hat{\alpha}\} & \text{otherwise,} \end{cases}$$

where $\hat{\alpha} = \frac{1}{L(K+0.5+10^{-6})}$ and $\alpha_0 = 1$. The scaling constant 0.99 was found after some experimentation. To perform the comparison, we first ran Algorithm 2 with a

stopping tolerance 5×10^{-4} and then IUG-h until it reached the same value of the objective function reached by Algorithm 2 since there is no known theoretic convergence property but the objective value eventually decreases. From Tables 1 and 2 we see that IUG-h converges slower than Algorithm 2 except when $K = 0$ but faster than Algorithm 1 when $K < 199$.

8 Conclusions

In this paper, we have proposed two incrementally updated gradient methods for minimizing the sum of smooth functions and a convex function. Algorithms 1 and 2 can be considered as the generalization of previous incremental gradient methods [11] for minimizing only the sum of smooth functions or minimizing the sum of smooth functions with constraints. A drawback of Algorithms 1 and 2 is the $O(mn)$ storage, which is expensive when m is large. Instead of storing a past gradient of f_i for each i, we use a running average of *all* past gradients. Then this has the advantage of using only $O(n)$ storage. This gradient update is also used in a subgradient averaging method of Nesterov [34, Sect. 6] and its extension by Xiao [21] (also see [35]) for convex stochastic optimization of the form (1) with f being the expectation of convex functions parameterized by a random variable. This new method is described in detail in [36].

In the future, we will give comprehensive numerical study of the two proposed methods and the new $O(n)$ storage method mentioned above.

Acknowledgements This research was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A1006406).

We thank anonymous referees for their detailed comments to improve the paper.

References

1. Bertsekas, D.P.: A new class of incremental gradient methods for least squares problems. SIAM J. Optim. **7**, 913–926 (1997)
2. Gaivoronski, A.A.: Convergence properties of back-propagation for neural nets via theory of stochastic gradient methods. Part I. Optim. Methods Softw. **4**, 117–134 (1994)
3. Luo, Z.-Q., Tseng, P.: Analysis of an approximate gradient projection method with applications to the backpropagation algorithm. Optim. Methods Softw. **4**, 85–101 (1994)
4. Mangasarian, O.L., Solodov, M.V.: Serial and parallel backpropagation convergence via nonmonotone perturbed minimization. Optim. Methods Softw. **4**, 103–116 (1994)
5. White, H.: Learning in artificial neural networks: a statistical perspective. Neural Comput. **1**, 425–464 (1989)
6. White, H.: Some asymptotic results for learning in single hidden-layer feedforward network models. J. Am. Stat. Assoc. **84**, 1003–1013 (1989)
7. Luo, Z.-Q.: On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks. Neural Comput. **3**, 226–245 (1991)
8. Solodov, M.V.: Incremental gradient algorithms with stepsizes bounded away from zero. Comput. Optim. Appl. **11**, 23–35 (1998)
9. Grippo, L.: A class of unconstrained minimization methods for neural network training. Optim. Methods Softw. **4**, 135–150 (1994)
10. Tseng, P.: An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. SIAM J. Optim. **8**, 506–531 (1998)

11. Blatt, D., Hero, A.O., Gauchman, H.: A convergent incremental gradient method with a constant step size. SIAM J. Optim. **18**, 29–51 (2007)
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing—Explorations in the Microstructure of Cognition, pp. 318–362. MIT press, Cambridge (1986)
13. Tesauro, G., He, Y., Ahmad, S.: Asymptotic convergence of back propagation. Neural Comput. **1**, 382–391 (1989)
14. Werbos, P.J.: Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Committee on Applied Mathematics, Harvard University, Cambridge (1974)
15. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**, 1550–1560 (1990)
16. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**, 33–61 (1999)
17. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**, 1413–1457 (2004)
18. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**, 1–22 (2010)
19. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B **58**, 267–288 (1996)
20. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**, 387–423 (2009)
21. Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. J. Mach. Learn. Res. **11**, 2543–2596 (2010)
22. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
23. Fukushima, M., Mine, H.: A generalized proximal point algorithm for certain non-convex minimization problems. Int. J. Syst. Sci. **12**, 989–1000 (1981)
24. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Englewood Cliffs (1989)
25. Tseng, P.: On the rate of convergence of a partially asynchronous gradient projection algorithm. SIAM J. Optim. **1**, 603–619 (1991)
26. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
27. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
28. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City (1991)
29. Denoeux, T., Lengellé, R.: Initializing back propagation networks with prototypes. Neural Netw. **6**, 351–363 (1993)
30. Luo, Z.-Q., Tseng, P.: Error bounds and convergence analysis of feasible descent methods: a general approach. Ann. Oper. Res. **46**, 157–178 (1993)
31. Rockafellar, R.T., Wets, R.J.-B.: Variational Analysis. Springer, New York (1998)
32. Koh, K., Kim, S.-J., Boyd, S.: An interior-point method for large-scale ℓ_1-regularized logistic regression. J. Mach. Learn. Res. **8**, 1519–1555 (2007)
33. Beck, A., Teboulle, M.: Gradient-Based Algorithms with Applications in Signal Recovery Problems. In: Palomar, D., Eldar, Y. (eds.) Convex Optimization in Signal Processing and Communications, pp. 33–88. Cambribge University Press, Cambribge (2010)
34. Nesterov, Y.: Primal-dual subgradient methods for convex problems. Math. Program. **120**, 221–259 (2009)
35. Juditsky, A., Lan, G., Nemirovski, A., Shapiro, A.: Stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**, 1574–1609 (2009)
36. Tseng, P., Yun, S.: Incrementally updated gradient methods for constrained and regularized optimization. Report, Department of Mathematics Education, Sungkyunkwan University, Seoul (2012)