A MEMORY EFFICIENT INCREMENTAL GRADIENT METHOD FOR REGULARIZED MINIMIZATION

Sangwoon Yun

Abstract. In this paper, we propose a new incremental gradient method for solving a regularized minimization problem whose objective is the sum of \( m \) smooth functions and a (possibly nonsmooth) convex function. This method uses an adaptive stepsize. Recently proposed incremental gradient methods for a regularized minimization problem need \( O(mn) \) storage, where \( n \) is the number of variables. This is the drawback of them. But, the proposed new incremental gradient method requires only \( O(n) \) storage.

1. Introduction

In this paper, we consider the regularized minimization problem whose form is

\[
\min_{x \in \mathbb{R}^n} F_\lambda(x) := f(x) + \lambda P(x),
\]

where \( \lambda > 0, \ P: \mathbb{R}^n \to (-\infty, \infty] \) is a proper, convex, lower semicontinuous (lsc) function \([20]\), and

\[
f(x) := \sum_{i=1}^m f_i(x),
\]

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

The minimization problem (1) we consider arises in many applications such as (supervised) learning \([7, 12, 27]\), regression \([17, 23]\), neural network training \([11, 21, 29]\), and data mining/classification \([5, 15, 22, 28]\). For the \( \ell_1 \)-regularized linear least squares problem \([6, 23]\),

\[
f_i(x) = \frac{1}{2}(a_i^T x - b_i)^2, \quad P(x) = \lambda \|x\|_1,
\]

Received April 9, 2015; Revised October 8, 2015.

2010 Mathematics Subject Classification. Primary 49M27, 49M37, 65K05, 90C25, 90C30.

Key words and phrases. incremental gradient method, nonsmooth, regularization, running average.

This work was financially supported by Basic Science Research Program through NRF funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A2056038).

©2016 Korean Mathematical Society

589
where $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, and $\lambda > 0$. In this problem, $f$ can be interpreted as a linear model under Gaussian errors on $b$. For the $\ell_1$-regularized logistic regression problem [15]:

$$f_i(x) = \frac{1}{m} \log(1 + \exp(-(a_i^T x_{1:n-1} + b_i x_n))), \quad P(x) = \lambda \|x_{1:n-1}\|_1,$$

where $x_{1:n-1} = (x_1, \ldots, x_{n-1})^T$, $a_i = b_i z_i$ with $(z_i, b_i) \in \mathbb{R}^{n-1} \times \{-1, 1\}$, and $\lambda > 0$. As is done in compressed sensing, lasso, and group lasso, a nonsmooth regularization term $P(x)$, such as the 1-norm, is added to avoid over-fitting and/or induces a sparse representation; see [6, 8, 9, 23, 25, 31] and references therein. Another important problem of the form (1) is L2-loss support vector regression [12]:

$$f_i(x) = \max\left(|a_i^T x - b_i| - \epsilon, 0\right), \quad P(x) = \frac{\lambda}{2} \|x\|_2^2,$$

where $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, and $\epsilon, \lambda > 0$. Note that $a_i$ and $b_i$ are a given set of (observed or training) data.

In many applications, the number of functions $m$ is large, say, more than $10^4$. In this case, traditional gradient based algorithms would be inefficient since they require evaluating $\nabla f_i(x)$ for all $i$ before $x$ is updated. In contrast, incremental gradient methods update $x$ after $\nabla f_i(x)$ is evaluated for only one or a few $i$. In the unconstrained case, i.e., $P \equiv 0$, the classical incremental gradient method has the following basic form

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

where $i_k$ is chosen to cycle through $1, \ldots, m$ (i.e., $i_0 = 1, i_1 = 2, \ldots, i_{m-1} = m, i_m = 1, \ldots$) and $\alpha_k > 0$. To guarantee global convergence of them, the stepsize requires to diminish to zero. This can lead to slow convergence; see [1, 10, 16, 18, 29, 30]. Moreover, its extension to the nonsmooth regularized minimization problem is hard.

Recently, Tseng and Yun [26] proposed incremental gradient methods to solve the problem (1), i.e., the (nonsmooth) regularized minimization problem. The method proposed in [26] has the following form

$$g^k = \sum_{i=1}^{m} \nabla f_i(x_{\tau_i^k}^k),$$

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

$$x^{k+1} = x^k + \alpha_k d^k,$$

where $\tau_i^k \leq k$ for $i = 1, \ldots, m$, $H^k \succ 0$, $\alpha_k \in (0, 1]$, and $x^0, x^1, \ldots$ in dom$P$ are given. In particular, when there is no regularization term, i.e., $P \equiv 0,$
(6)–(7) with $H^k = I$ and $\alpha_k = \alpha$, and
\[
(8) \quad \ \tau^k_i = \begin{cases} 
  k & \text{if } i = (k \mod m) + 1; \\
  \tau^{k-1}_i & \text{otherwise,}
\end{cases} \quad 1 \leq i \leq m, \ k \geq m,
\]
reduces to the incremental gradient method proposed by Blatt et al. [4] which has the form
\[
g^k = g^{k-1} + \nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^{k-m}),
\]
\[
x^{k+1} = x^k - \alpha_k g^k.
\]
The algorithm (5)–(7) is a more general method and the gradient components can be partially asynchronously updated [3, 24]. Hence the incremental gradient method in [26] has several advantages over the classical incremental gradient method (4). But, this incremental gradient method requires $O(mn)$ storage, which is expensive when $m$ is large. This is the main drawback of it.

In this paper, we propose a new incremental gradient (IG) method that, instead of storing a past gradient of $f_i$ for each $i$, uses a running average of all past gradients. This has the advantage of using only $O(n)$ storage. Specifically, the proposed IG method replaces (5) by
\[
(9) \quad g^k = \frac{k}{k+1} g^{k-1} + \frac{m}{k+1} \nabla f_{i_k}(x^k) \quad \text{with} \quad i_k = (k \mod m) + 1,
\]
with $g^{-1} = 0$. This gradient update is also used in a subgradient averaging method of Nesterov [19, Section 6] and its extension by Xiao [31] (also see [13]) for convex stochastic optimization of the form (1) with $f$ being the expectation of convex functions parametrized by a random variable. In contrast to the IG method proposed in [26], the proposed IG method can choose $i_k$ randomly.

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 $j$th component of $x$, and $\|x\|_p = \left(\sum_{j=1}^{n} |x_j|^p\right)^{1/p}$ for $1 \leq p < \infty$ 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). 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 $x^0, x^1, \ldots$.

2. Memory efficient incremental gradient method

In this section we describe the proposed IG method in which $g^k$ is updated by a weighted average of past component gradients (9) and give some lemmas that will be used in our convergence analysis, i.e., Theorem 3.1.

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

**Assumption 1.**
\[
(10) \quad \|\nabla f_i(y) - \nabla f_i(z)\| \leq L_i \|y - z\| \quad \forall y, z \in \text{dom} P,
\]
for some $L_i \geq 0$, $i = 1, \ldots, m$. Let $L = \sum_{i=1}^{m} L_i$.

The following assumptions are required for global convergence analysis of the proposed method.

**Assumption 2.** $\sigma I \preceq H^k$ for all $k$, where $0 < \sigma$.

Convergence of the proposed method requires $g^k - \nabla f(x^k) \to 0$. To ensure this, the stepsize $\alpha_k$ needs to be chosen carefully.

**Assumption 3.**

(a) $\sum_{k=0}^{\infty} \alpha_k = \infty$.

(b) $\lim_{\ell \to \infty} \sum_{j=0}^{\ell} \frac{j+1}{\ell+1} = 0$, where $\delta_j := \max_{i=0,1,\ldots,m} \|x^{k+i} - x^{k+m}\|_{k=jm-1}$.

Assumption 3(b) is satisfied by, for example, the adaptive stepsize rule

$$\alpha_{k+i} = \min\left\{1, \frac{\phi(j+1)}{(j+1)!d^{k+i}}\right\}, \quad 0 \leq i < m, j = 0, 1, \ldots,$$

where $\phi : [1, \infty) \to (0, \infty)$ is continuous, decreasing, and $\lim_{t \to \infty} \phi(t) = 0$. This is because, by (11), $\delta_j \leq m \frac{\phi(j+1)}{j+1}$. Also, for any $\epsilon > 0$, there exists $\bar{j}$ such that $\phi(j) \leq \epsilon$ for all $j > \bar{j}$. Thus

$$\sum_{j=0}^{\ell} \frac{j+1}{\ell+1} \delta_j \leq m \sum_{j=0}^{\ell} \frac{\phi(j+1)}{\ell+1} \leq m \left( \sum_{j=0}^{\ell} \frac{\phi(j+1)}{\ell+1} + \frac{\ell - \bar{j}}{\ell+1} \right) \to m \epsilon \text{ as } \ell \to \infty.$$

If in addition $\int_{1}^{\infty} \frac{\phi(t)}{t} dt = \infty$ and $\{d^k\}$ is bounded (such as when $\text{dom}P$ is bounded), then Assumption 3(a) holds as well. Examples of such $\phi$ include $\phi(t) = \frac{1}{\ln t}$ and $\phi(t) = \frac{1}{\ln \ln \ln \ln t}$.

Now, we formally describe our proposed method below.

**Algorithm 1** Memory Efficient Incremental Gradient Method

Choose $x^0 \in \text{dom}P$ and set $g^{-1} = 0$. For $k = 0, 1, 2, \ldots$, generate $x^{k+1}$ from $x^k$ according to the following iteration:

**Step 1:** Choose $H^k \succ 0_n$.

**Step 2:** Solve

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

with $g^k = \frac{1}{k+1} g^{k-1} + \frac{m}{k+1} \nabla f_i(x^k)$ with $i_k = (k \mod m) + 1$.

**Step 3:** Set $x^{k+1} = x^k + \alpha_k d^k$ with $\alpha_k \in (0, 1]$. 

In what follows, for any \( x \in \text{dom} P, \, g \in \mathbb{R}^n, \) and \( H \succ 0_n, \) we denote

\[
d^0_H(x) := \arg \min_d \left\{ \langle g, d \rangle + \frac{1}{2} \|d, Hd\| + \lambda P(x + d) \right\}.
\]

Thus \( d^k = d^k_H(x^k). \) For \( H \) diagonal and \( P \) separable piecewise-linear/quadratic, \( d^k_H(x) \) is computable in closed form. Some examples are given below.

1. For \( P(x) = \|x\|_1, \) \( d^k_H(x) = -\text{mid}\left\{ \frac{g_j - \lambda + \omega}{H_{jj}}, x_j, \frac{g_j + \lambda + \omega}{H_{jj}} \right\}. \)

2. For \( P(x) = \|x\|_1 + \frac{\omega}{2} \|x\|^2 \) \((\omega > 0) \) [9],

\[
d^k_H(x) = -\text{mid} \left\{ \ell_j - x_j, -\text{mid}\left\{ \frac{g_j - \lambda}{H_{jj}}, x_j, \frac{g_j + \lambda}{H_{jj}} \right\}, u_j - x_j \right\},
\]

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

We have the following lemma, whose proof is identical to that of [25, Eq. (8)] and is thus omitted.

**Lemma 2.1.** For any \( x \in \text{dom} P, \, g \in \mathbb{R}^n, \) and \( H \succ 0_n, \) let \( d = d^k_H(x). \) Then

\[
\langle g, d \rangle + \lambda P(x + d) - \lambda P(x) \leq -\langle d, Hd \rangle.
\]

We say that \( x \in \mathbb{R}^n \) is a stationary point of \( F_\lambda \) if \( x \in \text{dom} P \) and \( F_\lambda'(x; d) \geq 0 \) for all \( d \in \mathbb{R}^n. \) The following result from [25, Lemma 2] characterizes stationarity in terms of \( d^k_H(x). \)

**Lemma 2.2.** For any \( H \succ 0_n, \) an \( x \in \text{dom} P \) is a stationary point of \( F_\lambda \) if and only if \( d^k_H(x) = 0. \)

### 3. Convergence analysis

In this section, we analyze convergence properties of our proposed IG method under Assumptions 1–3. The proof uses Lemmas 2.1 and 2.2.

**Theorem 3.1.** Let \( \{x^k\}, \{d^k\}, \{H^k\}, \{\alpha_k\} \) be sequences generated by Algorithm 1 under Assumptions 1, 2 and 3. Then the following results hold.

(a) \( \{\|x^{k+1} - x^k\|\} \to 0 \) and \( \{\|\nabla f(x^k) - g^k\|\} \to 0. \)

(b) \( \lim \inf_{k \to \infty} \|d^k\| = 0. \)

(c) If \( \{x^k\} \) is bounded, then there exists a cluster point of \( \{x^k\} \) that is a stationary point of (1).

**Proof.** (a) Assumption 3(b) implies \( \delta_j \to 0, \) so \( \|x^k - x^{k+1}\| \to 0. \) For each \( j \in \{0, 1, \ldots\}, \) letting \( k = jm - 1, \) we have \( t_{k+1} = 1, t_{k+2} = 2, \ldots, t_{k+m} = m, \) and hence

\[
g^{k+m} = \frac{k + m}{k + m + 1} g^{k+m-1} + \frac{m}{k + m + 1} \nabla f_m(x^{k+m})
\]
\[
\begin{align*}
&= \frac{k + m}{k + m + 1} \left( \frac{k + m - 1}{k + m} g^{k+m-2} + \frac{m}{k + m} \nabla f_{m-1}(x^{k+m-1}) \right) \\
&\quad + \frac{m}{k + m + 1} \nabla f_m(x^{k+m}) \\
&= \frac{k + m}{k + m + 1} g^{k+m-2} + \frac{m}{k + m + 1} \nabla f_{m-1}(x^{k+m-1}) \\
&\quad + \frac{m}{k + m + 1} \nabla f_m(x^{k+m}) \\
&= \frac{k + 1}{k + m + 1} g^k + \frac{m}{k + m + 1} \sum_{i=1}^{m} \nabla f_i(x^{k+i}) \\
&= \frac{j}{j + 1} g^k + \frac{1}{j + 1} \sum_{i=1}^{m} \nabla f_i(x^{k+i}).
\end{align*}
\]

Thus, upon letting
\[
e_j := \|g^k - \nabla f(x^k)\|_{k=jm-1}
\]
and using \(k = jm - 1\), we have
\[
e_{j+1} = \|g^{k+m} - \nabla f(x^{k+m})\| \\
= \left\| \frac{j}{j + 1} (g^k - \nabla f(x^{k+m})) + \frac{1}{j + 1} \sum_{i=1}^{m} \left( \nabla f_i(x^{k+i}) - \nabla f_i(x^{k+m}) \right) \right\| \\
\leq \frac{j}{j + 1} \|g^k - \nabla f(x^k)\| + \frac{1}{j + 1} \sum_{i=1}^{m} \|\nabla f_i(x^{k+i}) - \nabla f_i(x^{k+m})\| \\
\leq \frac{j}{j + 1} \|g^k - \nabla f(x^k)\| + L \|x^k - x^{k+m}\| + \frac{1}{j + 1} \sum_{i=1}^{m} L_i \|x^{k+i} - x^{k+m}\| \\
\leq \frac{j}{j + 1} (e_j + L\delta_j) + \frac{1}{j + 1} L\delta_j \\
= \frac{j}{j + 1} e_j + L\delta_j,
\]
where the second inequality uses (10) and Assumption 1. Propagating this recursion backwards yields
\[
e_{j+1} \leq L \left( \frac{1}{j + 1} \delta_0 + \frac{2}{j + 1} \delta_1 + \cdots + \frac{j}{j + 1} \delta_{j-1} + \delta_j \right).
\]

Under Assumption 3(b), the right-hand side tends to zero as \(j \to \infty\). This shows that \(\{e_j\} \to 0\).

By replacing “\(k = jm - 1\)” in the above argument with “\(k = jm - 1 + \nu\)”, where \(\nu \in \{1, \ldots, m - 1\}\), we obtain
\[
\|x^k - x^{k+m}\| \leq \|x^k - x^{k+m-\nu}\| + \|x^{k+m-\nu} - x^{k+m}\| \leq \delta_j + 2\delta_{j+1}
\]
and similarly \( \|x^{k+1} - x^{k+m}\| \leq \delta_j + 2\delta_{j+1} \), so a similar argument yields
\[
e_{j+1} = \frac{j + \frac{1}{2\nu}}{j + 1 + \frac{1}{2m}} e_j + L(\delta_j + 2\delta_{j+1}) \leq \frac{j + 1}{j + 2} e_j + L(\delta_j + 2\delta_{j+1})
\]
and then \( \{e_j\} \to 0 \). Since the choice of \( \nu \) was arbitrary, this proves that \( \{\|\nabla f(x^k) - g^k\|\} \to 0 \).

(b) Let
\[
\Delta_k = \langle g^k, d^k \rangle + \lambda P(x^k + d^k) - \lambda P(x^k).
\]
Then, for each \( k \in \{0, 1, \ldots\} \),
\[
F_\lambda(x^{k+1}) - F_\lambda(x^k) = F_\lambda(x^k + \alpha_k d^k) - F_\lambda(x^k)
\]
\[
= f(x^k + \alpha d^k) - f(x^k) + \lambda P(x^k + \alpha d^k) - \lambda P(x^k)
\]
\[
\leq \alpha \|\nabla f(x^k), d^k\| + \alpha^2 \frac{L}{2} \|d^k\|^2 + \alpha (\lambda P(x^k + d^k) - \lambda P(x^k))
\]
\[
\leq \alpha_k \|\nabla f(x^k) - g^k, d^k\| + \alpha^2 \frac{L}{2} \|d^k\|^2 + \alpha_k \Delta_k
\]
\[
\leq \alpha_k \|\nabla f(x^k) - g^k\| \|d^k\| + \alpha^2 \frac{L}{2} \|d^k\|^2 - \alpha_k \|g^k\| \|d^k\|^2
\]
(12)
\[
= -\alpha_k \left( \frac{\alpha}{\|d^k\|} - \|\nabla f(x^k) - g^k\| - \frac{L}{2} \|x^{k+1} - x^k\| \right) \|d^k\|,
\]
where the first inequality uses the convexity of \( P \), \( \alpha \in (0, 1] \), and the Lipschitz continuity of \( \nabla f \) on \( \text{dom}P \) [2, page 667] and the third inequality uses \( \Delta_k \leq -\langle d^k, H^k x^k \rangle \leq -\frac{\alpha}{2} \|d^k\|^2 \) (see Lemma 2.1 and Assumption 2).

We argue \( \lim \inf_{k \to \infty} \|d^k\| = 0 \) by contradiction. Suppose the contrary, so that there exists an \( \epsilon > 0 \) such that \( \|d^k\| \geq \epsilon \) for all \( k \). By (a), we have \( \{\|\nabla f(x^k) - g^k\|\} \to 0 \) and \( \{\|x^{k+1} - x^k\|\} \to 0 \) and hence there exists an integer \( k \) such that
\[
\|\nabla f(x^k) - g^k\| \leq \frac{\alpha}{4} \|d^k\|, \quad \frac{L}{2} \|x^{k+1} - x^k\| \leq \frac{\alpha}{4} \|d^k\| \quad \forall k \geq \bar{k}.
\]
Then (12) yields
\[
F_\lambda(x^{k+1}) - F_\lambda(x^k) \leq -\alpha_k \frac{\|d^k\|^2}{2} \leq -\alpha_k \frac{\epsilon^2}{2} \quad \forall k \geq \bar{k},
\]
so that
\[
\lim_{k \to \infty} F_\lambda(x^{k+1}) \leq F_\lambda(x^k) - \sum_{k=\bar{k}}^{\infty} \alpha_k \frac{\epsilon^2}{2} = -\infty,
\]
where the equality is due to Assumption 3(a). This contradicts inf \( F_\lambda \geq -\infty \).

(c) Suppose \( \{x^k\} \) is bounded. By (b), \( \{d^k\}_{k \in K} \to 0 \) for some \( K \subseteq \{0, 1, \ldots\} \). Since \( \{\|\nabla f(x^k) - g^k\|\} \to 0 \), for every limit point \( \bar{x} \) of a convergent subsequence \( \{x^k\}_{k \in K \subseteq K'} \), \( \{g^k\}_{k \in K} \to \nabla f(\bar{x}) \).

Then (6) implies that, for any \( x \in \text{dom}P \), we have
\[
\langle g^k, d^k \rangle + \frac{1}{2} \langle d^k, H^k d^k \rangle + \lambda P(x^k + d^k) \leq \langle g^k, x - x^k \rangle + \frac{1}{2} \langle x - x^k, H^k (x - x^k) \rangle + \lambda P(x) \]
for all \( k \in \tilde{K} \), so the lsc property of \( P \) yields in the limit that
\[
\lambda P(\bar{x}) \leq \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{1}{2} \| x - \bar{x} \| H(x - \bar{x}) + \lambda P(x) \quad \forall x \in \text{dom} P,
\]
where \( \bar{x} \) is any cluster point of \( \{ H^k \}_{k \in \tilde{K}} \). Since \( H^k \succeq \alpha I \) for all \( k \in \tilde{K} \), \( \bar{x} \) is a stationary point of (1). Hence every cluster point of \( \{ x^k \}_{k \in \tilde{K}} \) is a stationary point of (1).

\[\Box\]

4. Numerical experiments

In this section we support Theorem 3.1 by numerical experiments. We apply our proposed method to solve the \( \ell_1 \)-regularized logistic regression problem (3) on randomly generated data. Note that we test our proposed method only on small size problems to show it works well.

Here, we assume that there are \( j, k \in \{1, \ldots, m\} \) such that \( b_j = 1 \) and \( b_k = -1 \). If we take \( x^0 = 0, \lambda \| x_{1:n-1} \|_1 \leq \log 2 \) for all \( x \) with \( x \in \mathcal{X}_0 := \{ x \mid F_\lambda(x) \leq F_\lambda(x^0) \} \) since \( \log(1 + \exp(-((a_j^T x_{1:n-1} + b_j x_n))) > 0 \) for all \( x \in \mathbb{R}^n \) and \( i = 1, \ldots, m \). Hence \( |x_i| \leq \frac{\log 2}{\lambda} \) for \( i = 1, \ldots, n-1 \). Also, since \( \lambda \| x_{1:n-1} \|_1 \geq 0, \log(1 + \exp(-((a_j^T x_{1:n-1} + b_j x_n))) \leq m \log 2 \) for all \( x \in \mathcal{X}_0 \) and \( i = 1, \ldots, m \). Hence, for each \( i \), \( \exp(-(a_j^T x_{1:n-1} + b_j x_n)) \leq 2^m - 1 \). This together with the assumption on the choice of \( b \) and the boundedness of \( x_{1:n-1} \) with \( x \in \mathcal{X}_0 \) implies that
\[
\| x^0 \|_1 \leq m \log 2 - \frac{\log 2}{\lambda} \min_{b_j > 0} \| a_i \|_1 \leq m \log 2 + \frac{\log 2}{\lambda} \min_{b_j < 0} \| a_i \|_1.
\]
Therefore \( \mathcal{X}_0 \) with \( x^0 = 0 \) is bounded. Optimal solutions are contained in \( \mathcal{X}_0 \). Hence we apply our proposed method to the following bounded \( \ell_1 \)-regularized logistic regression problem:
\[
\min_x \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-(a_i^T x_{1:n-1} + b_i x_n))) + \lambda \| x_{1:n-1} \|_1 + \epsilon_B(x),
\]
where \( B = \{ x \mid |x_i| \leq \frac{\log 2}{\lambda} \) for \( i = 1, \ldots, n-1, x_n \) satisfies (13) \}. Since \( B \) is bounded, this implies that Assumption 3 with (11) and \( \phi(t) = \frac{1}{\max_t} \) is satisfied.

Our proposed method is implemented as follows. We choose \( H^k = I \). The stepsize \( \alpha_k \) is chosen by the rule (11) with \( \phi(t) = \frac{1}{\max_t} \).

We stop the algorithm when the relative error of the iterates satisfies the following condition:
\[
\| x^k - x^{k-1} \| \leq \text{Tol},
\]
where Tol is a moderately small tolerance.

All runs are performed on a Desktop with Intel Core i7-3770 CPU (3.40GHz) and 8GB Memory, running 64-bit windows 8.1 and MATLAB (Version 8.3). Throughout the experiments, we choose the initial iterate to be \( x^0 = 0 \).
Table 1. Test results with ten random data sets with \( m = 100 \) and \( n = 101 \) with Tol = \( 10^{-4} \).

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>iters</th>
<th>obj</th>
<th>( \lambda )</th>
<th>iters</th>
<th>obj</th>
<th>( \lambda )</th>
<th>iters</th>
<th>obj</th>
<th>( \lambda )</th>
<th>iters</th>
<th>obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0473272</td>
<td>35508</td>
<td>0.2393</td>
<td>0.0474351</td>
<td>44336</td>
<td>0.2308</td>
<td>0.0489608</td>
<td>33911</td>
<td>0.2365</td>
<td>0.0543297</td>
<td>53836</td>
<td>0.2453</td>
</tr>
<tr>
<td>0.0503979</td>
<td>36393</td>
<td>0.2402</td>
<td>0.0503678</td>
<td>35306</td>
<td>0.2341</td>
<td>0.0503678</td>
<td>52595</td>
<td>0.2369</td>
<td>0.0527553</td>
<td>28196</td>
<td>0.2368</td>
</tr>
<tr>
<td>0.0503678</td>
<td></td>
<td></td>
<td>0.0527553</td>
<td></td>
<td></td>
<td>0.0529956</td>
<td></td>
<td></td>
<td>0.0527553</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. (a) Objective values versus epoch (one epoch is \( m \) iterations). It shows that the objective values eventually decrease. (b) Relative errors versus epoch. It shows that the relative errors eventually converge to zero. (c) The differences between the gradient of \( f \) at \( x^k \) and the approximated gradient \( g^k \) versus epoch. It shows that the differences eventually converge to zero. (d) The norm of directions versus epoch. It shows that the norm of directions also eventually converge to zero. Note that these are numerical results when \( \lambda = 0.0473272 \).
Table 1 reports the number of iterations and the final objective value for the bounded $\ell_1$-regularized logistic regression problem (14) on ten randomly generated data with size $m = 100$ and $n = 101$. As suggested in [14], each randomly generated problem has an equal number of positive and negative data points. Features of positive (negative) points are independent and identically distributed, drawn from a normal distribution $\mathcal{N}(\xi, 1)$, where $\xi$ is in turn drawn from a uniform distribution on $[0, 1]([-1, 0])$. For each instance, we chose $\lambda = 0.1\lambda_{\text{max}}$ where $\lambda_{\text{max}} = \frac{1}{m} \| \sum_{b=1}^{m} a_i \|_\infty + \frac{m-1}{m} \| \sum_{b=-1}^{-1} a_i \|_\infty$, $m_-$ is the number of negative points, and $m_+$ is the number of positive points.

Figure 1(a) shows that the objective values eventually decrease and Figure 1(b)-(d) show that the relative errors, the differences between the gradient of $f$ at $x^k$ and the approximated gradient $g^k$, the norm of directions converge to zero, respectively. Note that similar performance is observed in other randomly generated data. Hence Figure 1 supports that Theorem 3.1 works well for the bounded $\ell_1$-regularized logistic regression problem (14).

5. Conclusions and extensions

In this paper we have proposed the new incremental gradient method for minimizing the sum of smooth functions and a (possibly nonsmooth) convex function. The proposed method uses much less storage and so is a memory efficient method.

Our adaptive stepsize rule (11) is somewhat complicated. Can a diminishing stepsize which is used for classical incremental gradient methods or a constant stepsize be used? These stepsize rules are much simple to use. Can Theorem 3.1(c) be strengthened to show that every cluster point of $\{x^k\}$ is stationary? In this paper, we have tested our proposed method on small size problems only to show Theorem 3.1 works well. Hence, it will be interested in the comprehensive numerical study of the proposed method. These are some issues that need further investigation.

References


