Accelerated Difference of Convex functions Algorithm and its Application to Sparse Binary Logistic Regression

Duy Nhat Phan, Hoai Minh Le, Hoai An Le Thi
Department Computer Science and Application, LGIPM, University of Lorraine
{duy-nhat.phan, minh.le, hoai-an.le-thi}@univ-lorraine.fr

Abstract
In this work, we present a variant of DCA (Difference of Convex function Algorithm) with the aim of improving its performance. The proposed algorithm, named Accelerated DCA (ADCA), consists in incorporating the Nesterov’s acceleration technique into DCA. We first investigate ADCA for solving the standard DC program and rigorously study its convergence properties and the convergence rate. Secondly, we develop ADCA for a special case of the standard DC program whose the objective function is the sum of a differentiable function with Lipchitz continuous gradient (possibly nonconvex) and a DC function. We exploit the special structure of the problem to propose an efficient DC decomposition for which the corresponding ADCA scheme is inexpensive. As an application, we consider the sparse binary logistic regression problem. Numerical experiments on several benchmark datasets illustrate the efficiency of our algorithm and its superiority over well-known methods.

1 Introduction
A standard DC (Difference of Convex functions) program is of the form

$$\min_{x \in \mathbb{R}^n} \{ F(x) := G(x) - H(x) \}, \quad (1)$$

where $G$ and $H$ are lower semi-continuous proper convex functions on $\mathbb{R}^n$. Such a function $F$ is called a DC function while $G$ and $H$ are the DC components of $F$. Note that a convex constraint $x \in C$ can be incorporated into the objective function $F$ by using the indicator function on $C$, defined by $\chi_C = 0$ if $x \in C$ and $\chi_C = +\infty$ if $x \notin C$. The DC program (1) plays a key role in optimization since almost nonconvex programs encountered in practice can be formulated/reformulated as a DC program.

DCA (DC Algorithm) for solving the DC program (1) was introduced in 1985 by Pham Dinh Tao and extensively developed by Le Thi Hoai An and Pham Dinh Tao since 1994 to become now classic and increasingly popular ([Le Thi and Pham Dinh, 2005; 2018; Pham Dinh and Le Thi, 1997; 2014] and references therein). The original key idea of DCA relies on the structure DC of the objective function. DCA works with the DC components $G$ and $H$ but not directly with the function $F$. The main idea of DCA is simple: each iteration $l$ of DCA approximates the concave part $-H$ by its affine majorization (that corresponds to taking $y^k \in \partial H(x^k)$) and computes $x^{k+1}$ by solving the resulting convex problem.

$$\min \{ G(x) - \langle y^k, x \rangle : x \in \mathbb{R}^n \} \quad (P_k).$$

DCA has been successfully applied to various nonconvex/nonsmooth programs thanks to its versatility, flexibility, robustness, inexpensiveness and their adaptation to specific structure of considered problems. It has been proved that with an appropriate DC decompositions and a suitably equivalent DC reformulations, DCA permits to recover most of standard methods in convex and nonconvex programming. Several well-known methods in Machine Learning such as Expectation-Maximization (EM) [Dempster et al., 1977], Successive Linear Approximation (SLA) [Bradley and Mangasarian, 1998], Iterative Shrinkage-Thresholding Algorithms (ISTA) [Chambolle et al., 1998], and Convex-Concave Procedure (CCCP) [Yuille and Rangarajan, 2003] are special cases of DCA. The readers are referred to the recent paper [Le Thi and Pham Dinh, 2018] for an extensive overview of thirty years of development of DCA.

Nowadays, especially with the Big Data explosion, it has been becoming more and more important to develop advanced optimization methods able to handle very large-scale problems. Motivated by the success of DCA on several nonconvex/nonsmooth programs to which it was proved to be more robust and more efficient than related standard methods, we aim to investigate a variant of DCA in order to improve its performance.

Paper’s contribution: the contributions of this paper are multiples.

Firstly, we introduce Accelerated DCA (ADCA) for solving the standard DC program (1). In ADCA, we incorporate the Nesterov’s acceleration technique into standard DCA in order to improve its performance. The idea of ADCA is different to the usual line search acceleration using Armijo type rule which can be computationally costly. The acceleration step in ADCA which consists in using an extrapolated point from the current iterate and the previous one, aims to find a point $z^k$ which is better than $x^k$ for the computation of $x^{k+1}$.
We then provide a rigorous argument to prove the interesting convergence properties of ADCA as well as the convergence rate under the Lojasiewicz assumption.

Secondly, we investigate ADCA for a special case of the standard DC program, namely the sum of two nonconvex function minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) + r(x),$$

(2)

where $f$ is a differentiable function with $L$-Lipschitz continuous gradient (possibly nonconvex) and $r$ is a DC function. The problem (2) covers several nonconvex and nonsmooth problems arising from various fields such as machine learning, computational biology, signal processing, etc.

The problem (2) has been attracting attention of many researchers. Proximal gradient (PG) methods, which are also known as different names ISTA, fixed point iteration, forward-backward splitting (see e.g. [Daubechies et al., 2004; Beck and Teboulle, 2009b; Combettes and Wajs, 2005; Hale et al., 2008]), have been extensively developed for the convex case of (2), i.e., both $f$ and $r$ are convex. In [Nesterov, 1983], the author introduced the first accelerated proximal gradient (APG) method for solving (2) with $f$ convex and $r = 0$. Later, Beck and Teboulle [Beck and Teboulle, 2009a; 2009b] extended it for the case where both $f$ and $r$ are convex. These algorithms exhibit a global convergence rate $O(1/k^2)$, where $k$ is the iteration counter. Recently, several extensions of APG method for the convex case of (2) have been proposed for the nonconvex cases. For instance, [Li and Lin, 2015] have extended the method of [Beck and Teboulle, 2009a] for $f$ differentiable with $L$-Lipschitz gradient and $r$ nonconvex. In [Gu et al., 2018], the authors presented inexact versions of PG and APG for solving (2) with $f$ smooth and $r$ nonsmooth (possibly nonconvex). Another inexact APG for the nonconvex case of (2) which only requires one proximal step at each iteration, were proposed in [Yao et al., 2017]. However, the aforementioned algorithms have to compute the proximal map of nonconvex functions $r$ which do not has closed form in many cases. Usually, this computation can be very expensive or impossible.

In this work, by exploiting the properties of $f$ and $r$, we propose an efficient DC decomposition for which the corresponding DCA and ADCA are inexpensive.

Finally, as an application, we consider the sparse binary logistic regression and carefully perform numerical experiments of all proposed algorithms.

The remainder of the paper is organized as follows. In Section 2 we introduce ADCA for the standard DC program and study its convergence properties. ADCA for solving the problem (2) is presented in Section 3. The numerical experiments on the sparse binary logistic regression problem are reported in Section 4 and Section 5 concludes the paper.

2 Accelerated DCA for Standard DC Program

Before presenting the Accelerated DCA, let us recall some basis notations that will be used in the sequel.

The modulus of strong convexity of $\theta$ on $\Omega$, denoted by $\mu(\theta, \Omega)$ or $\mu(\theta)$ if $\Omega = \mathbb{R}^n$, is given by

$$\mu(\theta, \Omega) = \sup\{|\mu \geq 0 : \theta - (\mu/2)||.||^2 \text{ is convex on } \Omega\}.$$  

One says that $\theta$ is strongly convex on $\Omega$ if $\mu(\theta, \Omega) > 0$.

For a convex function $\theta$, the subdifferential of $\theta$ at $x_0 \in \text{dom} \theta := \{x \in \mathbb{R}^n : \theta(x_0) < +\infty\}$, denoted by $\partial \theta(x_0)$, is defined by

$$\partial \theta(x_0) := \{y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + \langle x-x_0, y \rangle, \forall x \in \mathbb{R}^n\}.$$  

The subdifferential $\partial \theta(x_0)$ generalizes the derivative in the sense that $\theta$ is differentiable at $x_0$ if and only if $\partial \theta(x_0) = \{\nabla \theta(x_0)\}$. A point $x^*$ is called a critical point of $G - H$, or a generalized Karush-Kuhn-Tucker point (KKT) of $(P_{dc})$ if

$$\partial H(x^*) \cap \partial G(x^*) \neq \emptyset.$$

(3)

We now introduce the Accelerated DCA (ADCA) for solving the standard DC program (1). According to the DCA scheme, at each iteration, one computes $y^k \in \partial H(x^k)$ then solves the convex sub-problem (1) to get $x^{k+1}$. The idea of ADCA, in order to accelerate DCA, is to find a point $z^k$ which is better than $x^k$ for the computation of $x^{k+1}$. In this work, we consider $z^k$ as an extrapolated point of the current iterate $x^k$ and the previous iterate $x^{k-1}$:

$$z^k = x^k + \frac{t_k - 1}{t_{k+1}} (x^k - x^{k-1}),$$

where $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$. If $z^k$ is better than one of last $q$ iterates $\{x^{k-q}, \ldots, x^{k-1}, x^k\}$ in term of objective function, i.e., $F(z^k) \leq \max_{t=max(0,k-q),\ldots,k} F(x^t)$ then $z^k$ will be used instead of $x^k$ to compute $y^k$. This condition allows the objective function $F$ to increase and consequently to escape from a potential bad local minimum [Grippo and Sciandrone, 2002; Wright et al., 2009]. Theoretically, a large value of $q$ increases the chance of using the extrapolated points $z^k$ in ADCA and consequently increases its chance to accelerate. Note that if $q = 0$ then $F(z^k) \leq F(x^k)$ and ADCA is a monotone algorithm like DCA. ADCA is described in Algorithm 1.

Algorithm 1 ADCA for solving the standard DC program (1)

Initialization: Choose an initial point $x^0, z^0 = x^0, q \in \mathbb{N}$, $t_0 = 1$, and $k \leftarrow 0$.

repeat

1: Compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$.
2: Compute $z^k = x^k + \frac{t_k - 1}{t_{k+1}} (x^k - x^{k-1})$ if $k \geq 1$.
3: If $F(z^k) \leq \max_{t=max(0,k-q),\ldots,k} F(x^t)$ then set $y^k = z^k$, otherwise set $y^k = x^k$.
4: Compute $y^k \in \partial H(y^k)$.
5: Compute $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \{G(x) - \langle y^k, x \rangle\}$.
6: $k \leftarrow k + 1$.

until Stopping criterion.
Remark 1. ADCA does not require any particular property of the sequence \{t_k\} for extrapolation. The above sequence \(t\) was chosen thanks to its interesting convergence rate proved in [Beck and Teboulle, 2009b]. More precisely, the well-known FISTA algorithm [Beck and Teboulle, 2009b] for solving the convex case of (2), i.e., both \(f(x)\) and \(r(x)\) are convex, is nothing else but a special case of ADCA in which we always use the extrapolated point \(z^k\) instead of the last iterate \(x^k\). In FISTA, the extrapolated point \(v^k\) is updated with \(t_{k+1} = \frac{1 + \sqrt{1+4t_k^2}}{2}\) and the authors have proved that FISTA has a interesting convergence rate \(O(\frac{1}{r^k})\).

2.1 Convergence Analysis of ADCA

In this subsection, we study the convergence of ADCA. Our first result provides the behavior of the limit points of the sequence \(\{x^{(k)}\}\) generated by ADCA, where \(\phi(k) = \arg\min_{t = k+1, \ldots, k+q} \|x^t - v^t - 1\|^2\).

Theorem 1. Let \(\mu(G)\) and \(\mu(H)\) be the convex modulus of \(G\) and \(H\), respectively. If \(\alpha = \inf_{x \in \mathbb{R}^n} F(x) > -\infty\) and \(\min \{\mu(G), \mu(H)\} > 0\), then for any subsequence \(\{x^{(k)}\}\), converging to \(x^*\) such that \(\{y^{(k+1)}\} = 0\) bounded, the limit point \(x^*\) is a critical point of (1).

To prove Theorem 1, we will use the following lemma. Denote by \(\{I^k\}\) the sequence defined as

\[
\Gamma^k = \max_{t = k+1, \ldots, k+q} F(x^t).
\]

Lemma 1. Let \(\{x^k\}\) and \(\{v^k\}\) be sequences generated by ADCA. The following statements hold.

(i) For any \(k \in 0, 1, \ldots, \Gamma^k - \Gamma^{k+1} > \frac{\mu(G) + \mu(H)}{2} \|x^{(k)} - v^{(k)} - 1\|^2\). (4)

As a result, by choosing \(q = 0\), we get the monotone property of \(\{F(x^k)\}\), i.e., \(F(x^k) - F(x^{k+1}) \geq \frac{\mu(G) + \mu(H)}{2} \|x^k - x^{k+1} - 1\|^2\).

(ii) If \(\alpha = \inf_{x \in \mathbb{R}^n} F(x) > -\infty\) and \(\min \{\mu(G), \mu(H)\} > 0\), then

\[
\sum_{k = 0}^{\infty} \|x^{(k)} - v^{(k)} - 1\|^2 < +\infty, \text{ and therefore } \lim_{k \to +\infty} \|x^{(k)} - v^{(k)} - 1\| = 0.
\]

Proof. First let us justify (i) by noting from the \(\mu\)-convexity of \(G\) and \(y^k \in \partial G(x^{k+1})\) that

\[
G(y^k) \geq G(x^{k+1}) + \langle y^k, v^k - x^{k+1} \rangle + \frac{\mu(G)}{2} \|v^k - x^{k+1}\|^2.
\]

It follows from the \(\mu\)-convexity of \(H\) and \(y^k \in \partial H(v^k)\) that

\[
H(x^{k+1}) \geq H(v^k) + \langle y^k, x^{k+1} - v^k \rangle + \frac{\mu(H)}{2} \|x^{k+1} - v^k\|^2.
\]

Summing the above two inequalities gives us

\[
F(v^k) - F(x^{k+1}) \geq \frac{\mu(G) + \mu(H)}{2} \|x^{k+1} - v^k\|^2.
\]

Observe that \(F(v^k) \leq \max_{t = k+1, \ldots, k+q} F(x^t) = \Gamma^k\). It follows from this and (5) that

\[
F(x^{k+1}) \leq \Gamma^k - \frac{\mu(G) + \mu(H)}{2} \|x^{k+1} - v^k\|^2.
\]

This implies that \(F(x^{k+1}) \leq \Gamma^k\). We prove by induction that for all \(t = 0, \ldots, q\)

\[
F(x^{k+1+t}) \leq \Gamma^k - \frac{\mu(G) + \mu(H)}{2} \|x^{k+1+t} - v^{k+t}\|^2.
\]

Indeed, it follows from (6) that the claim holds for \(t = 0\). We suppose that it also holds for \(t = 0, \ldots, p - 1\) with \(1 \leq p \leq q\). Thus, we have

\[
F(x^{k+1+p}) \leq \Gamma^k - \frac{\mu(G) + \mu(H)}{2} \|x^{k+1+p} - v^{k+p}\|^2
\leq \max \{\Gamma^k, F(x^{k+1}), \ldots, F(x^{k+p})\} - \frac{\mu(G) + \mu(H)}{2} \|x^{k+1+p} - v^{k+p}\|^2
\leq \Gamma^k - \frac{\mu(G) + \mu(H)}{2} \|x^{k+1+p} - v^{k+p}\|^2,
\]

where last inequality follows from \(F(x^{k+1+t}) \leq \Gamma^k\) for \(t = 0, \ldots, p - 1\). Therefore, we obtain

\[
\Gamma^k + q^k = \max_{t = k+1, \ldots, k+q} F(x^t)
\leq \Gamma^k - \min_{t = k+1, \ldots, k+q} \frac{\mu(G) + \mu(H)}{2} \|x^t - v^t - 1\|^2
\leq \Gamma^k - \frac{\mu(G) + \mu(H)}{2} \|x^k - v^k - 1\|^2.
\]

Next let us prove (ii) by noting that \(\Gamma^k \geq \alpha\) for all \(k\). Summing (4) over \(k = 0, \ldots, N\) gives us

\[
\sum_{k=0}^{N} \|x^{(k)} - v^{(k)} - 1\|^2 \leq \frac{\sum_{k=0}^{N} \|F(x^k) - \alpha\|^2}{\mu(G) + \mu(H)}
\leq \frac{(q + 1)\|F(x^q) - \alpha\|^2}{\mu(G) + \mu(H)}.
\]

Since \(\min \{\mu(G), \mu(H)\} > 0\), we have

\[
\sum_{k=0}^{N} \|x^{(k)} - v^{(k)} - 1\|^2 \leq \frac{(q + 1)\|F(x^q) - \alpha\|^2}{\mu(G) + \mu(H)}.
\]

Passing to the limit over the sequence \(\{N\}_{N \in \mathbb{N}}\) tells us that

\[
\sum_{k=0}^{+\infty} \|x^{(k)} - v^{(k)} - 1\|^2 \leq +\infty,
\]

and therefore \(\lim_{k \to +\infty} \|x^{(k)} - v^{(k)} - 1\| = 0\). □

Proof of Theorem 1. Let \(\{x^{(k)}\}\) be a subsequence of \(\{x^{(k)}\}\) that converges to \(x^*\). It follows from (ii) of Lemma 1 that \(\lim_{k \to +\infty} v^{(k)} = x^*\). Without loss of generality, we can suppose that the sequence \(\{y^{(k)}\}\) converges to \(y^*\). By the closedness of the subdifferential mapping \(\partial H\), we have \(y^* \in \partial H(x^*)\). We note that \(x^{(k)} \in \arg\min \{G(x) - \langle y^{(k)} - 1, x \rangle\}\). This implies that \(y^{(k)} = \partial G(x^{(k)})\). By the closedness of \(G\), we obtain \(y^* \in \partial G(x^*)\). Therefore, \(y^* \in \partial G(x^*) \cap \partial H(x^*)\). It follows from this that \(x^*\) is a critical point of the DC program (1). □

Recall that a lower semicontinuous function \(F\) has the Lojasiewicz property [Attouch and Bolte, 2009] if for any limiting-critical point \(x^*\), that \(0 \in \partial^2 F(x^*)\), there exist \(C, c > 0\) and \(\theta \in [0, 1]\) such that

\[
|F(x) - F(x^*)|^\theta \leq C|x - x^*|, \quad \forall x \in B(x^*, \varepsilon), \forall \varepsilon \in \partial^2 F(x).
\]

Here \(\partial^2 F(x)\) denotes the limiting-subdifferential of \(F\) at \(x\). The class of functions having the Lojasiewicz property is very ample, for example, semi-algebraic, subanalytic, and log-exp functions. For studying the convergence rate of ADCA, we state the results concerning the following lemma.
Lemma 2. Consider the settings of Theorem 1. Let \( \{x^k\} \) be sequence generated by ADCA with \( q = 0 \). Denote by \( \Omega \) the set of limit points of \( \{x^k\} \). Suppose further that \( \{x^k\} \) and \( \{y^k\} \) are bounded, and \( F \) has lower semicontinuity. The following statements hold.

(i) \( \Omega \) is a compact set and \( \lim_{k \to \infty} F(x^k) = F(x^*) \) for some \( x^* \in \Omega \). Thus, \( F \) has the same value on \( \Omega \), which is denoted by \( F^* \).

(ii) If \( F \) has the Lojasiewicz property and \( H \) is differentiable, then there exist \( C, \epsilon > 0 \) and \( \theta \in (0, 1) \) such that for all \( x \in \mathbb{R}^n \),
\[
\|x - x^+\| \leq \epsilon, \quad \forall \epsilon > 0, \theta \in (0, 1)
\]
and
\[
F(x) - F^* \leq C\|x - x^+\|.
\]
Furthermore, we have
\[
\|x - x^+\| \leq \epsilon, \quad \forall \epsilon > 0, \theta \in (0, 1)
\]
implies
\[
F(x) - F^* \leq C\|x - x^+\|.
\]

(iii) If \( F \) has the Lojasiewicz property and \( G \) is differentiable, then there exist \( C, \epsilon > 0 \) and \( \theta \in (0, 1) \) such that for all \( x \in \mathbb{R}^n \),
\[
\|x - x^+\| \leq \epsilon, \quad \forall \epsilon > 0, \theta \in (0, 1)
\]
and
\[
F(x) - F^* \leq C\|x - x^+\|.
\]

We now provide the asymptotic convergence rate of ADCA under the Lojasiewicz assumption.

Theorem 2. Consider the settings of Theorem 1. Suppose further that either \( G \) or \( H \) is differentiable with locally Lipschitz derivative. Let \( \{x^k\} \) be sequence generated by ADCA with \( q = 0 \). Assume that \( F \) is lower semicontinuous and has the Lojasiewicz property, and \( \{x^k\}, \{y^k\} \) are bounded. Denote by \( \theta \) the parameter, which is defined as in Lemma 2. The following estimations hold

(i) If \( \theta = 0 \), then the sequence \( \{F(x^k)\} \) converges to \( F^* \) in a finite number of steps.

(ii) If \( \theta \in (0, 1/2) \), then the sequence \( \{F(x^k)\} \) converges linearly to \( F^* \).

(iii) If \( \theta \in (1/2, 1) \), then there exist positive constants \( \eta \) and \( N_0 \) such that
\[
F(x^k) - F^* \leq \eta k - \frac{\|x^k - x^{k+1}\|}{\mu(G) + \mu(H)}.
\]

Proof. Let us consider the following cases.

Case 1. \( G \) is differentiable and its derivative is locally Lipschitz. For each \( x \in \Omega \), there exist \( L_x, \epsilon_x > 0 \) such that
\[
\|\nabla G(u) - \nabla G(v)\| \leq L_x \|u - v\| \quad \forall u, v \in \Omega, \quad \|x - x^+\| \leq \epsilon_x.
\]

Therefore, we have
\[
\|x - x^+\| \leq \epsilon_x, \quad \forall \epsilon_x > 0, \theta \in (0, 1)
\]
implies
\[
F(x) - F^* \leq C\|x - x^+\|.
\]

Therefore, we have
\[
\lim_{k \to \infty} \sup \{G(x^k) - H(x^k)\} \leq C\|x^k - x^{k+1}\|.
\]

On the other hand, since \( G \) is differentiable and the definition of \( x^{k+1} \), we have
\[
\nabla G(x^{k+1}) - \nabla G(x^k) = \nabla H(x^k) - \nabla G(x^k)
\]
\[
\nabla H(x^k) - \nabla G(x^k) = \nabla (F^* - F(x^k))
\]
where the second inequality follows from (11) and the last inequality holds by the lower semicontinuity of \( H \). On the other hand, from the lower semicontinuity of \( F \), we obtain
\[
\lim_{k \to \infty} \inf F(x^k) \geq F(x^*)
\]
Hence, by the uniqueness of limit, we have \( F^* = F(x^*) \).

We use the following critical points of \( F \). According to Lemma 1 in [Attouch and Bolte, 2009], applied to the function \( F \), there exist \( C, \epsilon > 0 \) and \( \theta \in (0, 1) \) such that \( \forall x \in \mathbb{R}^n \), \( \text{dist}(x, \Omega) \leq \epsilon \), \( \forall \epsilon > 0, \theta \in (0, 1) \), one has
\[
\|x - x^+\| \leq \epsilon, \quad \forall \epsilon > 0, \theta \in (0, 1)
\]
implies
\[
F(x) - F^* \leq C\|x - x^+\|.
\]
By using a similar argument, we have the result (iii).
Since $H$ is differentiable and by the definition of $x^{k+1}$, we have
\[
\nabla H(y^k) - \nabla H(x^{k+1}) = y^k - \nabla H(x^{k+1}) \\
\in \partial G(x^{k+1}) - \nabla H(x^{k+1}) \\
= \partial L(F(x^{k+1}) - y^k).
\]
Therefore, from the (ii) of Lemma 2 and $\text{dist}(x^{k+1}, \Omega) \rightarrow 0$, by increasing $N_2$ if necessary, we have for all $k \geq N_2$,
\[
|F(x^{k+1}) - F^*| \leq C(\nabla H(y^k) - \nabla H(x^{k+1}))^2 \\
\leq (CL)^2\|v^k - x^{k+1}\|^2 \\
\leq \frac{2(CL)^2}{\mu(G) + \mu(H)}|F(x^k) - F(x^{k+1})|,
\]
where the second inequality follows from (15) and the last inequality follows from the (i) of Lemma 1. Hence, we obtain
\[
r^2_{k+1} \leq \frac{2(CL)^2}{\mu(G) + \mu(H)}|r_k - r_{k+1}|.
\]
Thus, from (14) and (17), we have shown that, in both cases, there exists $\tau > 0$ such that
\[
r^2_{k+1} \leq \tau|r_k - r_{k+1}|.
\]
By using similar arguments in [Frankel et al., 2015], it is easy to show that sequence $\{r_k\}$ satisfying the above inductive property converges to zero at different rates according to $\theta$ as stated in the theorem.\\

3 ADCA for the Sum of Two Nonconvex Functions Minimization Problem

In this section, we consider the sum of two nonconvex functions minimization problem (2), i.e.,
\[
\min_{x \in \mathbb{R}^n} f(x) + r(x),
\]
where $f$ is a differentiable function with $L$-Lipschitz continuous gradient (possibly nonconvex) and $r$ is a DC function. Clearly, $f$ can be expressed as a DC function: $f(x) = \frac{\rho}{2}\|x\|^2 - \frac{\rho}{2}\|x\|^2 - f(x)$, where $\rho \geq L$. Let $r(x) = g(x) - h(x)$, then a DC decomposition of $F(x) = f(x) + r(x)$ is given by
\[
F(x) = G(x) - H(x),
\]
where $G(x) = \frac{\rho}{2}\|x\|^2 + g(x)$ and $H(x) = \frac{\rho}{2}\|x\|^2 - f(x) + h(x).$ According to Algorithm 1, ADCA for solving the problem (2) is described in Algorithm 2. We note that $\mu(G) > \rho > 0$ and the convergence results of Algorithm 2 are guaranteed by Theorem 1 and 2. Furthermore, with most of existing nonconvex regularizers $r$ in the literature, the resolution of the subproblem (20) is inexpensive or can be explicitly computed.

4 Numerical Experiment

To study the performance of the ADCA, we consider the sparse binary logistic regression problem. The problem can be described as follows. Let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set with observation vectors $x_i \in \mathbb{R}^d$ and labels $y_i \in \{-1, 1\}$. We aim to find a hyperplane $f = \langle w, x \rangle + b$ that separates the two classes. To find $w$ and $b$, we maximize the log-likelihood function $\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(x_i^Tw + b)))$. On the other hand, to deal with irrelevant and redundant features in high-dimensional data, we use features selection method which consists in minimizing the zero-norm of $w$.

Hence the sparse binary logistic regression is formulated by
\[
\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} -\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(x_i^Tw + b))) + \lambda\|w\|_0.
\]
Note that with this DC decomposition, the corresponding $\lambda > 0$ is the trade-off parameter between the two terms. The minimization of zero-norm is known to be NP-hard for which several methods have been developed in the literature. The readers are referred to [Le Thi et al., 2015] for an extensive overview of existing methods for the minimization of zero-norm. In this work, we replace $\|w\|_0$ by a nonconvex approximation, namely the exponential concave function defined by $r_{exp}(w) = \sum_{i=1}^{d} (1 - \exp(-\alpha|w_i|))$. Thus, the problem (21) becomes
\[
\min_{(w,b)} -\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(x_i^Tw + b))) + \lambda r_{exp}(w).
\]
Thus, the problem (22) takes the form of (2) and can be solved by DCA and ADCA. Note that with this DC decomposition, the corresponding DCA and ADCA are very inexpensive: it only required the soft thresholding operator (proximal operator of $\ell_0$-norm), which is explicitly computed.

The experiments are performed on 4 algorithms: DCA, ADCA, inexact APG (inAPG) [Yao et al., 2017] and non-monotone APG (nmAPG) [Li and Lin, 2015]. For DCA and ADCA, we estimated a Lipschitz constant $L$ by computing a bound of Hessian matrix of logistic loss. Note that inAPG and nmAPG require to compute the proximal mapping of the DC function $r$. However, this proximal mapping do not have a closed form. We therefore use DCA to compute the proximal mapping of $r$ in inAPG and nmAPG. All the algorithms are terminated when the change of two consecutive objective function values is less than $10^{-6}$. We also stop algorithms after $5h$ (18,000 seconds) of CPU time.

The detailed information of used datasets is summarized in the first column of Table 1. $n_{train}$ (resp. $n_{test}$) represents

\begin{algorithm}
\caption{ADCA for solving (2)}
\begin{algorithmic}
\State \textbf{Initialization:} Choose an initial point $x^0, z^0 = x^0, q \in \mathbb{N}$, $t_0 = 1, \rho > L$, and $k \leftarrow 0$.
\Repeat
\State 1: Compute $t_{k+1} = 1 + \sqrt{1 + 4t_k^2}$.
\State 2: Compute $z_k = x^k + \frac{t_k}{t_{k+1}}(x^k - x^{k-1})$ if $k \geq 1$.
\State 3: If $F(z^k) \leq \max_{i=\max(0,k-q),\ldots,k} F(x^t)$ then set $v_k = z^k$, otherwise set $v_k = x^k$.
\State 4: Compute $y_k = \rho v_k - \nabla f(v_k) + \xi_k$, where $\xi_k \in \partial h(v_k)$.
\State 5: Compute $x^{k+1}$ by solving strongly convex problem
\[
\min_{x \in \mathbb{R}^n} \left\{ \frac{\rho}{2} \|x\|^2 + g(x) - \langle y^k, x \rangle \right\}.
\]
\Until {Stopping criterion.}
\end{algorithmic}
\end{algorithm}
the number of points in training set (resp. test set) while $d$ is the number of features. All data sets can be downloaded from well-known data repertory LibSVM.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Time (s)</th>
<th>Acc (§)</th>
<th>Sparsity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>madelon</td>
<td>DCA</td>
<td>1.14</td>
<td>62.17</td>
<td>0.4</td>
</tr>
<tr>
<td>w8a</td>
<td>ADCA</td>
<td>0.54</td>
<td>62.17</td>
<td>0.4</td>
</tr>
<tr>
<td>rcv1</td>
<td>inAPG</td>
<td>0.86</td>
<td>62.17</td>
<td>0.4</td>
</tr>
<tr>
<td>a9a</td>
<td>nmAPG</td>
<td>1.23</td>
<td>62.17</td>
<td>0.4</td>
</tr>
<tr>
<td>epsilon</td>
<td>DCA</td>
<td>32.8</td>
<td>98.43</td>
<td>16</td>
</tr>
<tr>
<td>epsilon</td>
<td>ADCA</td>
<td>5.58</td>
<td>98.51</td>
<td>15.67</td>
</tr>
<tr>
<td>epsilon</td>
<td>inAPG</td>
<td>36.42</td>
<td>98.45</td>
<td>17</td>
</tr>
<tr>
<td>epsilon</td>
<td>nmAPG</td>
<td>54.81</td>
<td>98.4</td>
<td>19.33</td>
</tr>
<tr>
<td>rcv1</td>
<td>DCA</td>
<td>113.03</td>
<td>91.8</td>
<td>0.87</td>
</tr>
<tr>
<td>a9a</td>
<td>ADCA</td>
<td>33.74</td>
<td>94.23</td>
<td>0.79</td>
</tr>
<tr>
<td>a9a</td>
<td>inAPG</td>
<td>39.65</td>
<td>91.1</td>
<td>0.85</td>
</tr>
<tr>
<td>a9a</td>
<td>nmAPG</td>
<td>112.37</td>
<td>93.9</td>
<td>0.72</td>
</tr>
<tr>
<td>real-sim</td>
<td>DCA</td>
<td>7.11</td>
<td>84.95</td>
<td>32.52</td>
</tr>
<tr>
<td>real-sim</td>
<td>ADCA</td>
<td>1.11</td>
<td>84.98</td>
<td>33.33</td>
</tr>
<tr>
<td>real-sim</td>
<td>inAPG</td>
<td>6.38</td>
<td>84.97</td>
<td>32.52</td>
</tr>
<tr>
<td>real-sim</td>
<td>nmAPG</td>
<td>14.01</td>
<td>84.97</td>
<td>32.52</td>
</tr>
<tr>
<td>epsilon</td>
<td>DCA</td>
<td>33.25</td>
<td>94.49</td>
<td>2.71</td>
</tr>
<tr>
<td>epsilon</td>
<td>ADCA</td>
<td>7.74</td>
<td>94.5</td>
<td>2.59</td>
</tr>
<tr>
<td>epsilon</td>
<td>inAPG</td>
<td>33.49</td>
<td>94.42</td>
<td>2.62</td>
</tr>
<tr>
<td>epsilon</td>
<td>nmAPG</td>
<td>53.11</td>
<td>94.39</td>
<td>2.82</td>
</tr>
</tbody>
</table>

Table 1: Comparative results. Bold values correspond to the best results for each dataset.

All experiments are performed on a PC Intel i7 CPU3770, 3.40 GHz of 8GB RAM and the codes were written in MATLAB. We fix $\alpha = 5$ as proposed in [Bradley and Mangasarian, 1998]. The parameter $q$ is set to 5. The trade-off parameter $\lambda$ is fixed to $10^{-3}$ on rcv1, epsilon and $10^{-3}$ on the other data sets. For the epsilon, the nmAPG algorithm did not furnish any result due to a out of memory problem.

We reported in Table 1 the running time in second, the classification accuracy on test set and the sparsity (percentage of selected features) of solution. We also plot the curves of objective function values versus training time in Figure 1.

Concerning the classification accuracy, ADCA gives the best results for all 5 datasets (all four algorithms give the same result on madelon). As for the sparsity of solution, the four algorithms are comparable. All four algorithms give the same results on madelon and the best sparsity of solution on a9a is obtained by DCA, inAPG, nmAPG. nmAPG suppresses more features than the others on rcv1 while ADCA gives the best sparsity of solution on w8a, epsilon. In term of running time, ADCA is clearly the fastest one among the four compared algorithms. ADCA improve considerably the running time comparing to DCA: ADCA is up to 12.00 times faster than DCA (gisette). ADCA is faster than inAPG and nmAPG which also use acceleration technique. We can observe from Figure 1 that the objective functions of ADCA decrease drastically in few first iterations comparing to the others three algorithms.

Overall, among the four compared algorithms, ADCA is the fastest one while giving better classification accuracy and sparsity of solution.

5 Conclusion

We have rigorously studied ADCA, a variant of DCA with the aim of improving its performance. ADCA consists in incorporating the Nesterov’s acceleration technique into DCA. We have proved that ADCA converges to a critical point of the standard DC program. Furthermore, we proved the convergence rate of ADCA under the Lojasiewics assumption. ADCA is then developed for the minimization of sum of two nonconvex functions problem, a special case of the standard DC program. Exploiting the fact that $f(x)$ is differentiable with L-Lipschitz gradient, we propose, an efficient DC decomposition for which the corresponding ADCA scheme is inexpensive. To evaluate the performance of proposed algorithm, we consider the sparse binary logistic regression problem. ADCA for solving the latter is very inexpensive: it only required the soft thresholding operator which is explicitly computed. Numerical results showed that ADCA improves considerably the running time of DCA (up to 12.09 times faster than DCA) while giving similar or better classification accuracy and sparsity of solution. Furthermore, ADCA outperformed related accelerated proximal gradient methods such as non-monotone APG and inexact APG.
References


