Convergence guarantees for a class of non-convex and non-smooth optimization problems

Koulik Khamaru [1] **Martin J. Wainwright** [1 2]

Abstract

Non-convex optimization problems arise frequently in machine learning, including feature selection, structured matrix learning, mixture modeling, and neural network training. We consider the problem of finding critical points of a broad class of non-convex problems with non-smooth components. We analyze the behavior of two gradient-based methods—namely a sub-gradient method, and a proximal method. Our main results are to establish rates of convergence for general problems, and also exhibit faster rates for sub-analytic functions. As an application of our theory, we obtain a simplification of the popular CCCP algorithm, which retains all the desirable convergence properties of the original method, along with a significantly lower cost per iteration. We illustrate our methods and theory via application to the problems of best subset selection, robust estimation and shape from shading reconstruction.

1. Introduction

Non-convex optimization problems arise frequently in statistical machine learning; examples include the use of non-convex penalties for enforcing sparsity, non-convexity in likelihoods in mixture modeling, and non-convexity in neural network training. Of course, minimizing a non-convex problem is NP-hard in general, but in machine learning applications, it is often sufficient to find critical points that are first-order (and possibly second-order) stationary. There have been a number of recent papers demonstrating that all first (and/or second) order critical points have desirable properties for certain statistical problems; for instance, see the papers (Loh & Wainwright, 2013; Ge et al., 2017) as well as references therein. Accordingly, recent years have witnessed an explosion of research on different algorithms for non-convex problems, with the goal of trying to characterize the nature of their fixed points, and their convergence properties.

The most straightforward approach to obtaining a first-order critical point is via gradient descent which can be shown, under suitable regularity conditions and step size choices, to compute (approximate) first-order critical points. Recently, Lee et al. (2016) showed that for twice continuously differentiable smooth functions, gradient descent with random initializations converges to a second order stationary solution almost surely. Despite the empirical success of (sub)gradient-based methods in many non-convex problems, an issue is that all the available theory applies to smooth non-convex functions. In practice, many machine learning problems have non-smooth components; examples include the hinge loss in support vector machines, the rectified linear unit in neural networks, and various types of matrix regularizers in collaborative filtering and recommender systems. Accordingly, a natural goal is to develop subgradient-based techniques that apply to a broader class of non-convex functions, allowing for non-smoothness.

The main contribution of this paper is to provide precisely such a set of techniques, along with non-asymptotic guarantees on their convergence rates. In particular, we study algorithms that can be used to obtain first-order (and in some cases, also second-order) optimal solutions to a relatively broad class of non-convex functions, allowing for non-smoothness in certain portions of the problem. For each sequence $\{x^k\}_{k\geq 0}$ generated by one of our algorithms, we provide non-asymptotic bounds on the convergence rate of the gradient sequence $\{\|\nabla f(x^k)\|_2\}_{k\geq 0}$. Moreover, for functions that satisfy a form of the Kurdaya-Łojasiewicz inequality, we show that our methods achieve faster rates.

Our work also makes interesting points of contact with a second line of work on non-convex optimization, namely to functions that can be represented as a difference of two convex functions, popularly known as DC functions. One of the most popular DC optimization algorithms is Convex Concave Procedure (CCCP); see the papers (Yuille & Rangarajan, 2003; Lipp & Boyd, 2016) for further details.

[1]Department of Statistics, UC Berkeley, Berkeley, USA [2]Department of EECS, UC Berkeley, Berkeley, USA. Correspondence to: Koulik Khamaru <koulik@berkeley.edu>, Martin J. Wainwright <wainwrig@eecs.berkeley.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

This is a double loop algorithm that minimizes a convex relaxation of the non-convex function at each iteration. The CCCP algorithm has some attractive convergence properties; see (Lanckriet & Sriperumbudur, 2009) for details. But one of the drawbacks is that it can be slow in many situations due to its double loop structure. In this paper, we develop a single-loop prox-method that retains the convergence guarantees of CCCP, but is much faster to run, as demonstrated in our simulation performance.

Related Work: There is a lengthy literature on non-convex optimization, dating back more than six decades. A particular class that has been extensively studied are those functions that can be represented as the difference of convex functions, or DC functions for short; see the papers (Tuy, 1995; Hartman, 1959; Lanckriet & Sriperumbudur, 2009; Yuille & Rangarajan, 2003) for more details. Recent years have witnessed a re-surgence of interest in guarantees for non-convex and non-smooth problems, in part driven by the non-convexity of many objectives that arise in machine learning and statistics. Bolte et al. (2014) developed a proximal type algorithm when the objective function is a sum of smooth (possibly non-convex) and a convex (possibly non-differentiable) function. Some recent work (Xu & Yin, 2017) extended these ideas and provided analysis for block co-ordinate descent methods for non-convex functions. In other recent work, Hong et al. (2016) provided a analysis of the ADMM method for non-convex problems. An and Nam (2017) proposed a proximal type method for non-convex functions which can be written as a sum of a smooth function, a concave continuous function and a convex lower semi-continuous function; we also analyze this class in one of our results (Theorem 2).

Overview of our results

- Our first main result, stated in Theorem 1, gives a gradient type algorithm for minimizing problem 2 over a closed convex set $\mathcal{C}$. In Theorem 1, we provide a rate of convergence of the gradient (or a gradient type object in case of non-differentiable functions); we show that the algorithm is optimal among all first order algorithms. In Corollary 1 we prove that Algorithm 1 escapes strict saddle points with random initializing, without smoothness assumption on the entire function $f = g - h$. We next illustrate some consequences of Theorem 1 by obtaining a new rate of convergence of gradients in CCCP.

- In the second result, stated in Theorem 2, we provide a proximal type algorithm for problem (1) and provide rate of convergence of the proximal type algorithm. In Section 5.2, we demonstrate how this proximal type algorithm can be used to solve the best subset selection problem in linear regression. We demonstrate the performance of Algorithm 2 and CCCP for best subset selection problem.

- Finally, in Theorems 3 and 4 we show that under a mild assumption, the rate of convergence of gradient can be improved to at least $1/k$. We also provide a large class of function, namely subanalytic function (which include semi-algebraic, sub-analytic, analytic functions), that satisfy our assumption. We also provide examples of functions where the rate of convergence of gradient is $1/k^r$ for $r = 1, 2, \ldots$.

Notation: Given a set $\mathcal{C} \subset \mathbb{R}^d$, we use $\mathrm{int}(\mathcal{C})$ to denote its interior. We use $\|x\|_2$ and $\|x\|_1$ to denote the Euclidean and ℓ_1-norms, respectively, of a vector $x \in \mathbb{R}^d$. In many examples considered in this paper, the objective function f is linear combination of a differentiable function g and one or more convex functions h and φ. If $f = g + \varphi - h$, then we use $\nabla f(x)$ to denote the Minkowski sum of $\{\nabla g(x)\}, \partial\varphi(x)$ and $-\partial h(x)$, i.e. $\nabla f(x) := \nabla g(x) + \partial\varphi(x) - \partial h(x)$. For any sequence $\{a^k\}_{k\geq 0}$ we use Avg $\left(a^k\right) := \frac{1}{k+1}\sum_{\ell=0}^{\ell=k} a^\ell$, the running arithmetic mean of the sequence $\{a^k\}_{k\geq 0}$. Similarly, we use GAvg $\left(a^k\right)$ to denote the running geometric mean.

2. Problem setup

In this paper, we study the problem of minimizing a non-convex and possibly non-smooth function over a closed convex set. Consider the optimization problem

$$\min_{x\in\mathcal{C}} f(x) \;=\; \min_{x\in\mathcal{C}} \Big\{ g(x) - h(x) + \varphi(x) \Big\}, \qquad (1)$$

where the domain $\mathcal{C}$ is a closed convex set. In all cases, we assume the function f is bounded below over $\mathcal{C}$, and that the function h is continuous and convex. Our aim is to derive algorithms for problem (1) for various types of g and φ.

Structural assumption on functions g and φ

(a) Theorems 1 and 3 assume that the function g is continuously differentiable and smooth, and that $\varphi \equiv 0$.

(b) In Theorems 2 and 4, we assume that the function g is continuously differentiable and smooth, and that the function φ is proper convex and lower semi-continuous.[1]

The class of non-convex functions covered in part (a) includes the class of difference of convex (DC) functions, for

[1] Taking $\varphi \equiv 0$ yields part (a) as a special case, but it is worthwhile to point out that the assumptions in Theorem 1 are weaker than Theorem 2. Furthermore, we can prove some interesting results about saddle points when $\varphi \equiv 0$; see Corollary 1.

which the first convex function is smooth and the second convex function is continuous, as a special case. Note that we only put a mild assumption of continuity on the convex function h, meaning that the difference function $g - h$ can be non-smooth and non-differentiable in general. It can be shown that when h is continuously differentiable but non-smooth, and g is smooth, then the difference function $g - h$ is non-smooth. Furthermore, if we take $h \equiv 0$, then we recover the class of smooth functions as a special case.

3. Main results

Our main results are to analyze two algorithms for this class of non-convex non-smooth problems; in particular, deriving non-asymptotic bounds on their rate of convergence. The first algorithm is a (sub)-gradient type method, and is mainly suited for unconstrained optimization; the second algorithm is based on a proximal operator, and can be applied to constrained optimization problems.

3.1. Gradient type method

In this section, we analyze a (sub)-gradient-based method for solving a certain class of non-convex optimization problems. In particular, consider a pair of functions (g, h) such that:

Assumption GR:

(a) The function g is continuously differentiable and M_g-smooth.

(b) The function h is continuous and convex.

(c) There is a closed convex set $\mathcal{C}$ such that the difference function $f := g - h$ is bounded below on $\mathcal{C}$.

Under these conditions, we then analyze the behavior of a (sub)-gradient method in application to the problem

$$f^* = \min_{x \in \mathcal{C}} f(x) \;=\; \min_{x \in \mathcal{C}} \big\{ g(x) - h(x) \big\}. \tag{2}$$

With a slight abuse of notation, we refer to a vector of the form $\nabla g(x) - u(x)$, where $u(x) \in \partial h(x)$—the convex subgradient set of the function h at the point x— as a gradient of the function f.

Algorithm 1 Subgradient type method

1: Given an initial point $x^0 \in \operatorname{int}(\mathcal{C})$ and step size $\alpha \in (0, \frac{1}{M_g}]$:
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Choose subgradient $u^k \in \partial h(x^k)$.
4: Update $x^{k+1} = x^k - \alpha\big(\nabla g(x^k) - u^k\big)$.
5: **end for**

In our analysis, we assume that the initial vector $x^0 \in \operatorname{int}(\mathcal{C})$ is chosen such that the associated level set

$$\mathcal{L}(f(x^0)) := \big\{ x \in \mathbb{R}^d \mid f(x) \leq f(x^0) \big\}$$

is contained within $\operatorname{int}(\mathcal{C})$. This condition is standard in the analysis of non-convex optimization (e.g., see Nesterov and Polyak (2006)). When $\mathcal{C} = \mathbb{R}^d$, it holds trivially.

With this set-up, we have the following guarantee on the convergence rate of Algorithm 1

Theorem 1. Under Assumption GR, any sequence $\{x^k\}_{k \geq 0}$ produced by Algorithm 1 has the following properties:

(a) Any limit point is a critical point of f, and the sequence of function values $\{f(x^k)\}_{k \geq 0}$ is strictly decreasing and convergent.

(b) For all $k = 0, 1, 2, \ldots$, we have

$$\text{Avg}\left(\|\nabla f(x^k)\|_2^2 \right) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}. \tag{3}$$

We provide a proof sketch here, providing the full proof in Appendix C.1. At a high level, the main part of the analysis is devoted to establishing that Algorithm 1 with step size $\alpha \leq \frac{1}{M_g}$ enjoys the following descent property:

$$f(x^{k+1}) \leq f(x^k) - \frac{\alpha}{2} \|\nabla f(x^k)\|_2^2. \tag{4}$$

Combining this descent property with the fact that f is bounded below, some further calculations then allow us to establish the claimed convergence bounds on the function value sequence $\{f(x^k)\}_{k \geq 0}$ as well as the gradient sequence $\{\|\nabla f(x^k)\|_2\}_{k \geq 0}$.

3.1.1. Comments on convergence rates

Note that the bound (3) guarantees that the gradient norm sequence $\min_{j \leq k} \|\nabla f(x^j)\|_2$ converges to zero at the rate $\mathcal{O}(1/\sqrt{k})$. It is natural to wonder whether this convergence rate can be improved. Interestingly, the answer is no, at least for the general class of functions covered by Theorem 1. Indeed, note that the class of M-smooth functions is contained within the class of functions covered by Theorem 1. It follows from past works (Cartis et al., 2010; Carmon et al., 2017) that for the class of smooth functions, the rate of convergence of any algorithm, given access to only the function gradients and function values, cannot be better than $\Omega(1/\sqrt{k})$. Finally, observe that in the special case $h \equiv 0$, Algorithm 1 reduces to the ordinary gradient descent with fixed step size α. Putting together the pieces, we conclude that for the class of functions which can be written as a difference of smooth and a continuous convex function, Algorithm 1 is *optimal* among all algorithms which has access to the function gradients (and/or the sub-gradients) and the function values.

3.1.2. ESCAPING STRICT SADDLE POINTS

One of obstacles in gradient-based optimization is that these methods can in principle converge to a saddle point. In this section, we analyze the behavior of the gradient type Algorithm 1 under random initialization. Our main result is to show that under random initialization, Algorithm 1 escapes strict saddle points almost surely.

Corollary 1. In additions to the conditions on $(g, h, \mathcal{C})$ from Theorem 1, suppose that (g, h) are twice continuously differentiable. Then the set of initial points for which Algorithm 1 converges to a strict saddle point has measure zero.

As a proof sketch, the main step is to show that the gradient map $x \mapsto x - \alpha \nabla g(x)$ is a diffeomorphism when the step size $\alpha \leq \frac{1}{M_g}$. The rest of the proof follows from a simple application of Stable Manifold Theorem; see Theorem 4.4 of Lee et al. (2016). See Appendix C.2 for a detailed proof. We note that Lee et al. (2016) provided the same guarantee when the function $f = g - h$ is twice continuously differentiable and M-smooth. The novelty of Corollary 1 is that the same guarantee holds without imposing a smoothness condition on the entire function f.

3.2. Connections to the convex-concave procedure

We show that one can obtain a rate of convergence result for CCCP(convex-concave procedure) as a corollary of Algorithm 1; which is heavily used algorithm in Difference of Convex (DC) optimization problems. Before we do so, let us provide a brief description of DC functions and CCCP algorithm.

DC functions: Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a convex set. We say a function $f : \mathcal{C} \mapsto \mathbb{R}$ is DC on $\mathcal{C}$ if there exist convex functions g and h such that $f = g - h$ on $\mathcal{C}$. Note that the DC representation $f = g - h$ mentioned in definition is not unique; in fact, for any convex function p, we can write $f = (g + p) - (h + p)$. The class of DC functions includes a very large number of non-convex problems encountered in practice. Obviously, both convex and concave functions are (trivially) DC. Moreover, the class of DC functions remains closed under addition and subtraction. More interestingly, under mild restrictions on the domain, the class of non-zero DC functions are also closed under multiplication, division, and composition (Hartman, 1959; Tuy, 1995). The maximum and minimum of a finite collection of DC functions are also DC functions.

Convex-concave procedure: One interesting class of DC optimization problem is minimizing a DC function over a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, i.e.

$$f^* := \min_{x \in \mathcal{C}} f(x) \;=\; \min_{x \in \mathcal{C}} \big\{ g(x) - h(x) \big\}, \tag{5}$$

where g and h are proper convex functions. The above problem has been studied intensively, and there are various methods for solving it; for instance, see the papers (Tuy, 1995; Pham Dinh et al., 2013; Lipp & Boyd, 2016) and references therein for details. One of the popular algorithm to solve problem (5) is Convex Concave Procedure (CCCP), which was introduced by Yuille and Rangarajan (2003). This iterative algorithm is a special case of a Majorization-Minimization algorithm, which uses the DC structure of the objective function in problem (5) to construct a convex majorant at each step. We start with a feasible point $x^0 \in \mathrm{int}(\mathcal{C})$. Let x^k denote the iterate at k^{th} iteration; at $(k+1)^{\text{th}}$ iteration we construct a convex majorant $q(\cdot,\, x^k)$ of f via

$$f(x) \leq \underbrace{g(x) - h(x^k) - \langle u^k,\, x - x^k \rangle}_{=:\, q(x, x^k)}, \tag{6}$$

where $u^k \in \partial h(x^k)$. The next iterate x^{k+1} is obtained by solving the convex program

$$x^{k+1} \in \arg\min_{x \in \mathcal{C}} q(x, x^k). \tag{7}$$

The CCCP algorithm has some attractive convergence properties. For example, it is a descent algorithm, and when g is strongly convex differentiable and h is continuously differentiable, then it can be shown that any limit point of the sequence $\{x^k\}_{k \geq 0}$ obtained from CCCP is stationary. Under the same assumptions, one can also show that $\lim_{k \to \infty} \|x^k - x^{k+1}\|_2 = 0$.

We now turn to an analysis of CCCP using the techniques that underlie Theorem 1. Earlier analyses of CCCP, including the papers (Lanckriet & Sriperumbudur, 2009; Yuille & Rangarajan, 2003), are mainly based on the assumption of strong convexity of g. Here we derive a rate of convergence of the gradient sequence, and show that all limit points of $\{x^k\}_{k \geq 0}$ are stationary. Unlike previous works, we only assume g is M_g-smooth, and h is a convex continuous function. When the function g is strongly convex, our analysis recovers the well-known convergence result in past work (Lanckriet & Sriperumbudur, 2009). In particular we show that CCCP enjoys the same rate of convergence as that of Algorithm 1.

Proposition 1. Under Assumption GR and with g being convex, the CCCP sequence (7) has the following properties:

(a) Any limit point of the sequence $\{x^k\}_{k \geq 0}$ is a critical point, and the sequence of function values $\{f(x^k)\}_{k \geq 0}$ is strictly decreasing and convergent.

(b) Furthermore, for all $k = 1, 2, \ldots$, we have

$$\text{Avg}\left(\|\nabla f(x^k)\|_2^2\right) \leq \frac{2M_g\big(f(x^0) - f^*\big)}{(k+1)}, \tag{8a}$$

and assuming moreover that g is μ-strongly convex,

$$\text{Avg}\left(\|x^k - x^{k+1}\|_2^2\right) \leq \frac{2\big(f(x^0) - f^*\big)}{\mu(k+1)}. \tag{8b}$$

As a high level proof sketch, the main observation here is that the update step 7 is stronger than update equation of Algorithm 1; hence CCCP update step 7 also enjoys the descent property (4), and the rate of convergence in terms of $\|\nabla f(x^k)\|_2$ follows from proof of Algorithm 1. When function h is strongly convex, we note that the convex majorant $q(x, x^k)$ is also strongly convex. We leverage this fact to obtain a rate of convergence of the successive difference $\|x^k - x^{k+1}\|_2$. See Appendix C.3 for detailed proof.

3.2.1. Simplifying CCCP

Algorithm 1 provides us an alternative algorithm for minimizing difference of convex functions when the first convex function is smooth. The benefit of Algorithm 1 over standard CCCP is that Algorithm 1 is a single loop algorithm and is expected to be faster than standard double-loop CCCP algorithm in many situations. Furthermore, Algorithm 1 shares convergence guarantees similar to a standard CCCP algorithm.

3.3. Proximal type method

We now turn a more general class of optimization problems of the form

$$f^* := \min_{x\in\mathbb{R}^d} f(x) \;=\; \min_{x\in\mathbb{R}^d} \Big\{\big(g(x) - h(x)\big) + \varphi(x)\Big\}. \tag{9}$$

We assume that the functions g, h and φ satisfy the following conditions:

Assumption PR

(a) The function $f = g - h + \varphi$ is bounded below on $\mathbb{R}^d$.

(b) The function g is continuously differentiable and M_g-smooth; the function h is continuous and convex; and the function φ is proper, convex and lower semi-continuous.

Typical example of φ could be $\varphi(x) = \|x\|_1$ or indicator of a closed convex convex set $\mathcal{X}$. For general lower semi-continuous function φ, the gradient type algorithm does not work, because the function $g + \varphi$ is neither differentiable nor a smooth function. One way to minimize such functions is to apply a Proximal type method instead of gradient type method.

Algorithm 2 Proximal type algorithm

1: Given an initial vector $x^0 \in \mathcal{C}$ and step size $\alpha \in \big(0, \frac{1}{M_g}\big)$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Update $x^{k+1} = \text{prox}_{1/\alpha}^{\varphi}\Big(x^k - \alpha\big(\nabla g(x^k) - u^k\big)\Big)$ for some $u^k \in \partial h(x^k)$.
4: **end for**

The proximal update in Line 3 of Algorithm 2 is very easy to compute and often has a closed form solution (e.g., see Parikh et al. (2014)). Let us now derive the rate of convergence result of Algorithm 2.

Theorem 2. Under Assumption PR, any sequence $\big\{x^k\big\}_{k\geq 0}$ obtained from Algorithm 2 has the following properties:

(a) The sequence $\big\{f(x^k)\big\}_{k\geq 0}$ is strictly decreasing and convergent, and any limit point of the sequence $\big\{x^k\big\}_{k\geq 0}$ is stationary.

(b) For all $k = 0, 1, 2, \ldots$, we have

$$\text{Avg}\left(\|x^k - x^{k-1}\|_2^2\right) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}. \tag{10a}$$

If moreover h is M_h-smooth, then

$$\text{Avg}\left(\|\nabla f(x^k)\|_2^2\right) \leq \frac{2C_{M,\alpha}\big(f(x^0) - f^*\big)}{\alpha(k+1)}, \tag{10b}$$

where $C_{M,\alpha} = \big(M_g + M_h + \frac{1}{\alpha}\big)^2$.

The proof of Theorem 2 reveals that the smoothness condition on function h in Theorem 2 can be replaced by smoothness of function h in a set which contains all the iterates $\big\{x^k\big\}_{k\geq 0}$. Such a condition can be easily verified when the sequence $\big\{x^k\big\}_{k\geq 0}$ is bounded and the norm of the Hessian of the function f is upper bounded in that bounded set. The boundedness assumption on the iterates $\big\{x^k\big\}_{k\geq 0}$ holds in many situations. For instance, if f is coercive, meaning that $f(x) \to \infty$ as $\|x\|_2 \to \infty$, then it follows that the iterates remain bounded.

A special case of the Algorithm 2 is when $\varphi = \mathbb{1}_{\mathcal{X}}$ is the indicator function of a closed convex set $\mathcal{X}$. Consider the following constrained optimization problem

$$\min_{x\in\mathcal{X}} \big\{g(x) - h(x)\big\}, \tag{11}$$

where $\mathcal{X}$ is a closed convex set, the function g is M_g-smooth, and h is a convex continuous function. Following Algorithm 2, the update equation in this case is given by

$$x^{k+1} = \Pi_{\mathcal{X}}\big(x^k - \alpha(\nabla g(x^k) - u^k)\big). \tag{12}$$

In projected gradient, we should not expect a rate in terms of gradient, because the projected gradient step may not be aligned with the gradient direction, and even when the projected gradient step is along the gradient direction, the step size may be arbitrarily small due to projection. In this case, an appropriate analogue of gradient is follows:

$$\nabla f_{\mathcal{X}}\left(x^k\right) = \frac{1}{\alpha}\big(x^k - \Pi_{\mathcal{X}}(x^k - \alpha(\nabla g(x^k) - u^k))\big).$$

The analysis of Projected Gradient method using $\nabla f_{\mathcal{X}}\left(x^k\right)$ is standard in optimization literature (e.g., see Bubeck (2015)). It is worth pointing out that the quantity $\nabla f_{\mathcal{X}}\left(x^k\right)$ is the analogue of gradient in constrained optimization setup, and coincides with the gradient in unconstrained setup. Concretely, we have $\nabla f_{\mathcal{X}}\left(x^k\right) = \nabla f(x^k)$ where $f = g - h$ and $\mathcal{X} = \mathbb{R}^d$, and hence $\text{Avg}\left(\|\nabla f_{\mathcal{X}}\left(x^k\right)\|_2^2\right) \leq \frac{2\left(f(x^0)-f^*\right)}{\alpha(k+1)}$.

4. Faster rates under KL conditions

In the preceding sections, we have derived rates of convergence for the gradient norms for various classes of problems. It is natural to wonder if faster convergence rates are possible when the objective function is equipped with some additional structure. Based on Theorems 1 and 2, we see that both Algorithms 1 and 2 ensure that $\|x^k - x^{k+1}\|_2 \to 0$, meaning that the successive differences between the iterates converge to zero. Although we proved that any limit point of the sequence $\{x^k\}_{k\geq 0}$ has desirable properties, the condition $\|x^k - x^{k+1}\|_2 \to 0$ is not sufficient—at least in general—to prove convergence of the sequence $\{x^k\}_{k\geq 0}$, when no further information is available about the sequence $\{x^k\}_{k\geq 0}$. In this section, we provide a sufficient conditions under which Algorithms 1 and 2 yield convergent sequences of iterates $\{x^k\}_{k\geq 0}$, and we establish that gradient sequence $\{\|\nabla f(x)\|_2\}_{k\geq 0}$ converge at a faster rate.

4.1. Kurdaya-Łojasiewicz inequality

We prove a faster local rate of convergence of Algorithms 1 and 2 for a large class of functions which satisfy a form of the Kurdaya-Łojasiewicz (KL) inequality, as formalized in Assumption KL. We assume that there exists $\theta \in [0, 1)$ such that the ratio $\frac{|f(x)-f(\bar{x})|^\theta}{\|\nabla f(x)\|_2}$ is bounded above in a neighborhood of every point $\bar{x} \in \text{dom}_f$. This type of inequality is known as a Kurdaya-Łojasiewicz inequality, and the exponent θ is known as the Kurdaya-Łojasiewicz exponent (*KL-exponent*) of the function f at a point $\bar{x}$. These type of inequalities were first proved by Łojasiewicz (1963) for real analytic functions. Kurdaya (1998) and Bolte et al. (2007) proved similar inequalities for non-smooth functions, and also provided a large class of functions which satisfy a form KL inequality. For sake of completeness we provide a discussion on functions satisfying KL inequalities in Appendix B.2.

Assumption KL: For any point [2] $\bar{x} \in \text{dom}_f$, there exists a scalar $\theta \in [0, 1)$ such that the ratio $\frac{|f(x)-f(\bar{x})|^\theta}{\|\nabla f(x)\|_2}$ is bounded above in a neighborhood of $\bar{x}$.

Note that the neighborhood mentioned in the Assumption KL above may depend on the the function f and the point $\bar{x}$.

4.2. Convergence guarantees

Theorem 3. Under Assumptions GR & KL any bounded sequence $\{x^k\}_{k\geq 0}$ obtained from Algorithm 1 has the following properties:

(a) The sequence $\{x^k\}_{k\geq 0}$ converges to a critical point $\bar{x}$, and

$$\text{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k} \quad \text{for all } k = 1, 2, \ldots.$$

(b) Suppose $\bar{\theta}$ be the KL-exponent of f at $\bar{x}$ with $\frac{1}{2} \leq \bar{\theta} < \frac{r}{2r-1}$ for some $r \in \{2, 3, \ldots\}$, then

$$\text{GAvg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_2}{k^r} \qquad \text{for all} \quad k = 1, 2, \ldots.$$

We provide a proof sketch here, providing the full proof in Appendix E.2. The main step of the proof is to show that under Assumption KL, Algorithm 1 enjoys following stronger version of the descent property (4):

$$\left(f(x^{k+1})\right)^{1-\gamma\theta} \leq \left(f(x^k)\right)^{1-\gamma\theta} - c\|\nabla f(x^k)\|_2^{2-\gamma}, \quad (13)$$

where $\frac{1}{\theta} > \gamma \geq 1$ and $c > 0$ universal constants independent of k. The rate of convergence in terms of $\text{Avg}\left(\|\nabla f(x^k)\|_2\right)$ follows by leveraging the last descent inequality, and substituting $\gamma = 1$. When $\frac{1}{2} \leq \bar{\theta} < \frac{r}{2r-1}$, we show that one is allowed to take $\gamma = \frac{2r-1}{r}$, thereby ensuring $2 - \gamma = \frac{1}{r}$. We then utilize descent property 13 with $2 - \gamma = \frac{1}{r}$, and the Arithmetic-Geometric mean inequality to obtain a rate of convergence in terms of $\text{GAvg}\left(\|\nabla f(x^k)\|_2\right)$.

Theorem 4. Suppose the function h in Algorithm 2 is smooth, then under Assumptions PR & KL any bounded sequence $\{x^k\}_{k\geq 0}$ obtained from Algorithm 2 has the following properties:

(a) The sequence $\{x^k\}_{k\geq 0}$ converges to a critical point $\bar{x}$, and for all $k = 0, 1, 2 \ldots$

$$\text{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k}.$$

[2] It can be shown that such an inequality would hold at non-critical point of a continuous function f; see Remark 3.2 of Bolte et al. (2007).

(b) Given some $r \in \{2, 3, \ldots\}$, suppose that f at $\bar{x}$ has a KL exponent $\bar{\theta} \in \left[\frac{1}{2}, \frac{r}{2r-1}\right)$. Then

$$\text{GAvg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_2}{k^r} \qquad \text{for all} \quad k = 1, 2, \ldots.$$

Here $c_1, c_2 > 0$ are constants independent of k.

See Appendix E.2 for the proof.

Comments: Note that $\min_{1 \leq i \leq k} \|\nabla f(x^k)\|_2$ is upper bounded by the quantities $\text{Avg}\left(\|\nabla f(x^k)\|_2\right)$ and $\text{GAvg}\left(\|\nabla f(x^k)\|_2\right)$. It thus follows that the sequence $\{\|\nabla f(x^k)\|_2\}_{k \geq 0}$ converges to zero at a rate of at least $1/k$, thereby improving the rate of convergence of $\|\nabla f(x)\|_2$ obtained in Theorems 1 and 2. A simple modification of the proof by substituting $\gamma = 2$ shows that when $\theta < \frac{1}{2}$, Algorithms 1 and 2 converge in a finite number of steps.

5. Some illustrative applications

In this section, we discuss various consequences of Theorems 1 – 4 as well as Corollary 1 when applied to the problem of Robust regression, Best subset selection. We also add an application to the problem of *shape from shading* reconstruction in the Appendix A.

5.1. Robust regression using Tuky's bi-weight

We begin by considering robust regression problem with Tuky's bi-weight penalty function. Recall that in the framework of robust mean estimation, we assume the following model

$$y_i = \langle z_i, \mu^* \rangle + w_i \qquad \text{for } i = 1, \ldots, n.$$

Here the parameter $\mu^* \in \mathbb{R}^d$ is the unknown parameter of interest, where as $\{z_i, y_i\}_{i=1}^n$ are the observations. In robust regression, we obtain an estimate of μ^* by minimizing the following optimization problem,

$$\min_{\mu \in \mathbb{R}^d} \underbrace{\left\{ \frac{1}{n} \sum_{i=1}^{n} \Psi\big(y_i - \langle z_i, \mu \rangle\big) \right\}}_{=: f(\mu)}, \tag{14}$$

where Ψ is a known function with some robustness properties. One popular example of loss function Ψ is Tukey's bi-weight function, which is given by

$$\Psi(t) = \begin{cases} 1 - (1 - (t/\lambda)^2)^3 & \text{if } |t| \leq \lambda \\ 1 & \text{otherwise} \end{cases}, \tag{15}$$

where $\lambda > 0$ is a tuning parameter. Note that p is a smooth function, whence the function f in the objective (14) is also smooth, meaning that Algorithm 1 is suitable for the problem. With this set-up, applying Theorem 1, Theorem 3 and Corollary 1 yields the following corollary:

Corollary 2. Given a random initialization, the sequence $\{\mu^k\}_{k \geq 0}$ obtained by applying Algorithm 1 to the objective 14 has the following properties:

(a) Almost surely with respect to the random initialization, the sequence $\{\mu^k\}_{k \geq 0}$ converges to a point $\bar{\mu}$ such that

$$\nabla f(\bar{\mu}) = 0, \quad \text{and} \qquad \nabla^2 f(\bar{\mu}) \succeq 0.$$

(b) There is a universal constant c_1 such that

$$\text{Avg}\left(\|\nabla f(\mu^k)\|_2\right) \leq \frac{c_1}{k} \qquad \text{for all } k = 1, 2, \ldots.$$

We provide a detailed proof in Appendix F.1

5.2. Best subset selection

Moving beyond robust regression problem, we consider the problem of best subset selection in linear regression. Suppose that we observe the vector $y \in \mathbb{R}^n$ and a matrix $B \in \mathbb{R}^{n \times d}$ that are linked via the standard linear model $y = Bx^* + w$. Here $w \in \mathbb{R}^n$ is a noise vector, while $x^* \in \mathbb{R}^d$ is the unknown regression vector. We wish to estimate the unknown parameter x^* subject to a sparsity constraint, and we do so by solving the following optimization problem:

$$\min_{\substack{x \in \mathbb{R}^d \\ \|x\|_0 \leq s}} \|y - Bx\|_2^2. \tag{16}$$

Here the positive integer s is a tuning parameter that controls maximum number of allowable non-zero entries in x. It is known (Gotoh et al., 2017) that the set of s-sparse vectors can be expressed as the level set of a certain DC function. In particular, let $|x|_{(d)} \geq |x|_{(d-1)} \geq \cdots \geq |x|_{(1)}$ denote the values of $x \in \mathbb{R}^d$ re-ordered in terms of their absolute magnitudes. In terms of this notation, we have $\|x\|_1 \geq \sum_{i=d-s+1}^{d} |x|_{(i)}$ for all $x \in \mathbb{R}^d$, with equality holding if and only if x is s–sparse. This fact ensures that

$$\Big\{x \in \mathbb{R}^d : \|x\|_0 \leq s\Big\} = \Big\{x \in \mathbb{R}^d : \|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \leq 0\Big\}. \tag{17}$$

Since both the maps $x \mapsto \|x\|_1$ and $x \mapsto \sum_{i=d-s+1}^{d} |x|_{(i)}$ are convex (Boyd & Vandenberghe, 2004), this level set formulation is a DC constraint. Now using the representation (17), we can rewrite problem (16) as $\min_{x \in \mathbb{R}^d} \|y - Bx\|_2^2$ such that $\|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \leq 0$. For our experiments, it is more convenient to solve the penalized analogue of this problem, given by

$$\min_{x \in \mathbb{R}^d} \left\{ \|y - Bx\|^2 + \lambda\Big(\|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)}\Big) \right\}, \tag{18}$$

where $\lambda > 0$ is a tuning parameter. The optimization problem 18 can be solved using Algorithm 2 with $g = g$, $\varphi(x) = \lambda\|x\|_1$ and $h(x) = \lambda\sum_{i=d-s+1}^{d}|x|_{(i)}$. For the non-smooth component $\varphi(s) = \lambda\|x\|_1$, there is a closed form expression of the proximal update in Algorithm 2, so that the method is especially efficient in this case. It is interesting to note that problem (18) is a DC optimization problem, and is hence amenable to standard DC optimization techniques like CCCP. In Section 6, we show through simulations that solving the sparse linear regression problem (18) using Algorithm 2 is significantly faster than using CCCP.

In order to obtain a satisfactory convergence result for an algorithm applied to the problem (18), we need some further assumption. Interestingly, it turns out that the convergence of Algorithm 2 is dependent on the uniqueness of the solution of a convex relaxation of the original problem (18). For any point $\bar{x} \in \mathbb{R}^d$ with $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$, consider the following convex relaxation of problem (18)

$$\mathcal{P}(\bar{x}) := \min_{x\in\mathbb{R}^d}\left\{\|y - Bx\|_2^2 + \lambda\|x\|_1 - \lambda\langle\nabla h(\bar{x}),\, x - \bar{x}\rangle\right\}. \tag{19}$$

Note that $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$ implies the differentiability of the function h, ensuring that the last problem is well-defined.

Corollary 3. Let $\left\{x^k\right\}_{k\geq 0}$ be the sequence obtained by applying Algorithm 2 on problem (18). Suppose there exists a limit point $\bar{x}$ of the sequence $\left\{x^k\right\}_{k\geq 0}$ satisfying $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$, and the convex problem (19) has unique solution. Then the sequence $\left\{x^k\right\}_{k\geq 0}$ converges to the point $\bar{x}$, and for all $k = 0, 1, 2, \ldots$, we have

$$\text{Avg}\left(\|\nabla f(x^k)\|\right) \leq \frac{c_1}{k}, \quad \text{and} \quad \|x^k - \bar{x}\|_2 \leq cq^k, \tag{20}$$

where $q \in (0, 1)$ and $c_1, c > 0$ are constants independent of k.

The proof of this corollary is along the lines of the proof of Theorem 4, but it requires additional care, since the function $h(x) := \lambda\sum_{i=d-s+1}^{d}|x|_{(i)}$ is not smooth. See Appendix F.2 for the full proof.

6. Simulation

In this section, we compare the performance of Algorithm 2 (prox-type method for short) with the convex-concave Procedure (CCCP). on best-subset selection problem described in problem 18. Recall that, problem 18 can be decomposed as a difference of two convex functions and hence amenable to DC optimization algorithms like CCCP. In this simulation experiment, we use CCCP update (7). The inner convex optimization problem in update (7) is solved by proximal methods for minimizing sum of smooth convex function and ℓ_1 regularizer. The results for Algorithm 2 was obtained by applying Algorithm 2 on problem (18) with $g(x) = \|y - Bx\|_2^2$, $h(x) = \lambda\sum_{i=d-s+1}^{d}|x|_{(i)}$, and $\varphi(x) = \lambda\|x\|_1$.

Synthetic data generation: We generated the rows of the $n \times d$ matrix B from a d-dimensional Gaussian distribution with zero mean and an equicovariance matrix Σ satisfying $\Sigma_{ii} = 1$ for all i, and $\Sigma_{ij} = 0.7$ for all $i \neq j$. The regression vector $x^* \in \mathbb{R}^d$ (true value) was chosen to be a binary vector with sparsity s ($s \ll d$). The location of the nonzero entries of the true regression vector x^* was chosen uniformly form $\{1, \ldots, d\}$.

Performance comparison: We compare the two methods in terms of the total runtime of the algorithms and in terms of the estimator error. Recall that if $\bar{x}$ is the estimated value of the unknown regression vector $x^* \in \mathbb{R}^d$, then the average estimation error is defined as $\frac{\|\bar{x}-x^*\|_2}{\sqrt{p}\|x^*\|_2}$. Note that the average estimation error used here is scale invariant. For both algorithms, the tolerance level η was set to $\eta = 10^{-8}$, maximum number of iterations was to be 1000, and both had same initializations. The performance comparison between the two algorithms is documented in the figure 1.

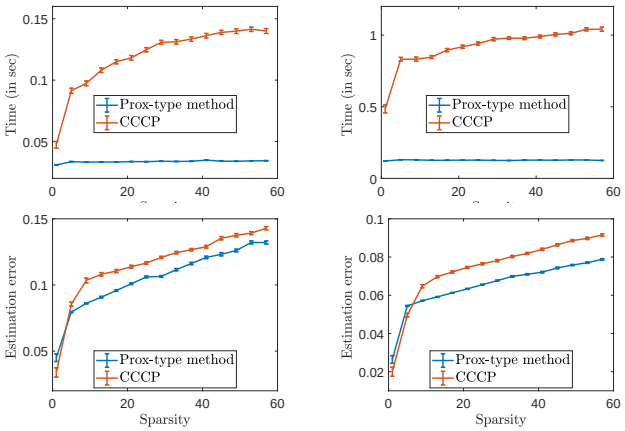

Figure 1. Figure showing performance of CCCP and Algorithm 2 on best subset selection problem for synthetic data for different values of (n, p). The left and right columns correspond to the settings $(n, p) = (190, 300)$ and $(n, p) = (380, 600)$, respectively. Plots in the first and second rows provide comparisons of running times and estimation error, respectively. The prox-type method (Algorithm 2) outperforms CCCP in terms of both the criteria. Results shown above are averaged over 100 replications and the bands represent the point-wise error bars.

Acknowledgments

This work was partially supported by the Office of Naval Research Grant DOD ONR-N00014 and National Science

Foundation Grant NSF-DMS-1612948.

References

An, N. T. and Nam, N. M. Convergence analysis of a proximal point algorithm for minimizing differences of functions. *Optimization*, 66(1):129–147, 2017.

Bolte, J., Daniilidis, A., and Lewis, A. The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

Cartis, C., Gould, N. I., and Toint, P. L. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *Siam journal on optimization*, 20(6):2833–2852, 2010.

Ecker, A. and Jepson, A. D. Polynomial shape from shading. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 145–152. IEEE, 2010.

Facchinei, F. and Pang, J.-S. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017.

Gotoh, J.-Y., Takeda, A., and Tono, K. DC formulations and algorithms for sparse optimization problems. *Mathematical Programming*, pp. 1–36, 2017.

Hartman, P. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3): 707–713, 1959.

Hong, M., Luo, Z.-Q., and Razaviyayn, M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

Kurdyka, K. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pp. 769–784. Chartres: L'Institut, 1950-, 1998.

Lanckriet, G. R. and Sriperumbudur, B. K. On the convergence of the concave-convex procedure. In *Advances in neural information processing systems*, pp. 1759–1767, 2009.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pp. 1246–1257, 2016.

Li, G. and Pong, T. K. Calculus of the exponent of Kurdya-Lojasiewicz inequality and its applications to linear convergence of first-order methods. *arXiv preprint arXiv:1602.02915*, 2016.

Lipp, T. and Boyd, S. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, 2016.

Loh, P.-L. and Wainwright, M. J. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pp. 476–484, 2013.

Lojasiewicz, S. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 117:87–89, 1963.

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Ostrowski, A. M. *Solution of equations and systems of equations: Pure and applied mathematics: A Series of monographs and textbooks*, volume 9. Elsevier, 2016.

Parikh, N., Boyd, S., et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Pham Dinh, T., Ngai, H., and Le Thi, H. Convergence analysis of the DC algorithm for DC programming with subanalytic data. *preprint*, 2013.

Rockafellar, R. T. and Wets, R. J.-B. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Tuy, H. DC optimization: theory, methods and algorithms. In *Handbook of global optimization*, pp. 149–216. Springer, 1995.

Wang, S., Schwing, A., and Urtasun, R. Efficient inference of continuous markov random fields with polynomial potentials. In *Advances in neural information processing systems*, pp. 936–944, 2014.

Xu, Y. and Yin, W. A globally convergent algorithm for non-convex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.

Yuille, A. L. and Rangarajan, A. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.

Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.