Successive Convex Approximation Algorithms for Sparse Signal Estimation with Nonconvex Regularizations

Yang Yang[1], Marius Pesavento[2], Symeon Chatzinotas[1], Björn Ottersten[2]
1. University of Luxembourg, L-1855 Luxembourg. Email: {yang.yang, symeon.chatzinotas, bjorn.ottersten}@uni.lu
2. Technische Universität Darmstadt, Darmstadt 64283, Germany. Email: pesavento@nt.tu-darmstadt.de

***Abstract*—In this paper, we propose a successive convex approximation framework for sparse optimization where the nondifferentiable regularization in the objective function is nonconvex and it can be written as the difference of two convex functions. The proposed framework is based on a nontrivial combination of the majorization-minimization method and successive convex approximation for nonconvex optimization where the regularization function is convex. The proposed framework is flexible and it leads to algorithms that exploit the problem structure and have a low complexity. We demonstrate these advantages by an example application where the nonconvex regularization is the capped ℓ_1-norm function. Customizing the proposed framework, we obtain a best-response type algorithm for which all elements of the unknown parameter are updated in parallel according to closed-form expressions. Finally, the proposed algorithms are numerically tested.**

***Index Terms*—Big Data, Line Search, Nonconvex Regularization, Successive Convex Approximation**

I. Problem Formulation

In this paper, we consider the following optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & h(\mathbf{x}) \triangleq f(\mathbf{x}) + g^+(\mathbf{x}) - g^-(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is a proper and differentiable function (with a continuous gradient) that is not necessarily convex, $g^+(\mathbf{x})$ and $g^-(\mathbf{x})$ are convex but not necessarily differentiable, and $\mathcal{X}$ is a closed and convex set.

Such a formulation plays a fundamental role in parameter estimation, where $f(\mathbf{x})$ models the estimate error and $g^{+(-)}(\mathbf{x})$ are regularization (penalty) functions promoting in the solution a certain structure known a priori such as sparsity [1]. Among others, the linear regression problem has received extensive attention in the past ten years and it is a special case of (1) by setting $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2$ with known $\mathbf{A} \in \mathbb{R}^{N\times K}$ and $\mathbf{b} \in \mathbb{R}^{K\times 1}$, $g^+(\mathbf{x}) = \lambda\left\|\mathbf{x}\right\|_1$, and $g^-(\mathbf{x}) = 0$ so that

$$g(\mathbf{x}) \triangleq g^+(\mathbf{x}) - g^-(\mathbf{x}) = \lambda\left\|\mathbf{x}\right\|_1.$$

Many algorithms have been proposed for the linear regression problem, for example, fast iterative soft-thresholding algorithm (FISTA) [2], block coordinate descent (BCD) method [3], alternating direction method of multiplier (ADMM) [4], proximal algorithm [5] and parallel best-response algorithms [6].

In linear regression, the function $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2$ is convex in $\mathbf{x}$. This is generally desirable in the design of numerical algorithms solving problem (1) iteratively. However, this desirable property is not available in nonlinear regression problems [7], where $f(\mathbf{x})$ is for example specified by $f(\mathbf{x}) = \frac{1}{2}\left\|\boldsymbol{\sigma}(\mathbf{A}\mathbf{x})-\mathbf{b}\right\|_2^2$ and $\boldsymbol{\sigma}(\mathbf{x})$ is a given function specifying the nonlinear regression model, e.g., the cosine or sigmoid function.

The ℓ_1-norm has been used as a standard regularization function to make the solution of problem (1) sparse. However, it was pointed out in [8, 9] that the ℓ_1-norm is a loose approximation of the ℓ_0-norm and it tends to produce biased estimates for large coefficients. A more desirable regularization function to achieve sparsity should be singular at the origin while flat elsewhere. Along this direction, several nonconvex regularization functions have been proposed, for example, the smoothly clipped absolute deviation [8], the capped ℓ_1-norm [10], and the logarithm function [11]; we refer the interested reader to [12, Fig. 1] for their shapes.

The nonconvexity of the regularization function $g(\mathbf{x})$ renders many well studied algorithms inapplicable, including the SCA framework [6], because the nondifferentiable function $g(\mathbf{x})$ is assumed to be convex. It is shown in [12] that if the differentiable function $f(\mathbf{x})$ is convex and the nonconvex regularization functions can be written as the difference of two convex functions, the standard majorization-minimization (MM) method can be applied to find a stationary point of (1). Nevertheless, this algorithm has a high complexity, because it is a two-layer algorithm that involves iterating within iterations: a new instance of regression problems must be solved at each iteration while solving regression problems repeatedly is not a trivial task, even with a warm start that sets the optimal point of the previous instance as the initial point of the new instance.

To reduce the complexity of the MM method, a generalized iterative soft-thresholding algorithm (GIST) is proposed in [13]. The GIST algorithm consists of solving a sequence of approximate problems, and in each approximate problem, the function $f(\mathbf{x})$ is replaced by its linear approximation while $g^{+(-)}(\mathbf{x})$ are not changed. Although the GIST algorithm converges to a stationary point of (1), it suffers from two limitations. Firstly, the convergence speed with the linear approximation is usually slower than some other approximations, for example, the best-response approximation [6]. Secondly, the approximate problem solved in each iteration is nonconvex, and it may not be easy to solve except in the few cases discussed in [13].

In this paper, we propose a SCA framework for problem (1). This SCA framework is based on a nontrivial combination of the SCA framework developed in [6] for convex regularization functions and standard MM framework. In particular, in each iteration, we first construct a (possibly nonconvex) upper bound of the original objective function $h(\mathbf{x})$ by the MM method, and then minimize an approximate function of the upper bound which can be constructed by the SCA framework developed in [6]. To guarantee the convergence, the approximate function only needs to satisfy some mild assumptions on convexity, gradient consistency and continuity. To further speed up the convergence, we design a new line search scheme to calculate the stepsize. On the one hand, the proposed algorithm exhibits a fast convergence behavior because i) it is a one-layer algorithm, ii) the problem structure can be better exploited by a proper choice of the approximate function, and iii) the use of the line search. On the other hand, the proposed algorithm enjoys a low complexity because i) the approximate function is convex and easy to optimize, and ii) the proposed line search scheme is over a properly constructed differentiable function while in traditional schemes line search is

The work of Yang, Chatzinotas and Ottersten is supported by the ERC project AGNOSTIC, and the work of Pesavento is supported by the EXPRESS Project within the DFG Priority Program CoSIP (DFG-SPP 1798).

directly applied to the original nonconvex nondifferentiable objective function. We then illustrate the above attractive features by customizing the proposed framework for an example application of the capped ℓ_1-norm minimization problem, where both the optimal point of the (best-response type) approximate functions and the stepsize obtained from the exact line search have closed-form expressions.

II. The Proposed Successive Convex Approximation Algorithms

Since $f(\mathbf{x})$ is not necessarily convex and $g^-(\mathbf{x})$ is convex, $h(\mathbf{x})$ is a nonconvex function. Since both $g^+(\mathbf{x})$ and $g^-(\mathbf{x})$ are assumed to be nondifferentiable, $h(\mathbf{x})$ is in general a nondifferentiable function. In this section, we develop an iterative algorithm that converges to a stationary point $\mathbf{x}^\star$ of problem (1) that satisfies the first order optimality condition:

$$(\mathbf{x}-\mathbf{x}^\star)^T(\nabla f(\mathbf{x}^\star)+\boldsymbol{\xi}^+(\mathbf{x}^\star)-\boldsymbol{\xi}^-(\mathbf{x}^\star))\geq 0, \forall \mathbf{x}\in\mathcal{X},$$

where $\boldsymbol{\xi}^+(\mathbf{x})$ (and $\boldsymbol{\xi}^-(\mathbf{x})$) is a subgradient of $g^+(\mathbf{x})$ (and $g^-(\mathbf{x})$).

At any arbitrary but given point $\mathbf{x}^t$, assume the subgradient of $g^-(\mathbf{x})$ is $\boldsymbol{\xi}^-(\mathbf{x}^t)$. Since $g^-(\mathbf{x})$ is convex, it follows from Jensen's inequality that

$$g^-(\mathbf{x})\geq g^-(\mathbf{x}^t)+(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t), \forall \mathbf{x}\in\mathcal{X}. \tag{2}$$

Define $\overline{h}(\mathbf{x};\mathbf{x}^t)$ as

$$\overline{h}(\mathbf{x};\mathbf{x}^t)\triangleq f(\mathbf{x})-g^-(\mathbf{x}^t)-(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)+g^+(\mathbf{x}). \tag{3}$$

We can readily infer from (2) that $\overline{h}(\mathbf{x};\mathbf{x}^t)$ is a global upper bound of $h(\mathbf{x})$ which is tight at $\mathbf{x}=\mathbf{x}^t$:

$$\overline{h}(\mathbf{x};\mathbf{x}^t)\geq h(\mathbf{x}),\ \text{and}\ \overline{h}(\mathbf{x}^t;\mathbf{x}^t)=h(\mathbf{x}^t). \tag{4}$$

In the standard MM method for problem (1) proposed in [12], a sequence of points $\{\mathbf{x}^t\}_t$ is generated by minimizing the upper bound function $\overline{h}(\mathbf{x};\mathbf{x}^t)$:

$$\mathbf{x}^{t+1}=\arg\min_{\mathbf{x}\in\mathcal{X}}\ \overline{h}(\mathbf{x};\mathbf{x}^t). \tag{5}$$

This and (4) imply that $\{h(\mathbf{x}^t)\}_t$ is a decreasing sequence as

$$h(\mathbf{x}^{t+1})\leq\overline{h}(\mathbf{x}^{t+1};\mathbf{x}^t)\leq\overline{h}(\mathbf{x}^t;\mathbf{x}^t)=h(\mathbf{x}^t).$$

However, the optimization problem (5) is not necessarily easy to solve due to two possible reasons: $\overline{h}(\mathbf{x};\mathbf{x}^t)$ may be nonconvex, and $\mathbf{x}^{t+1}$ may not have a closed-form expression and must be found iteratively.

The proposed algorithm consists of minimizing a sequence of successively refined approximate functions. Given $\mathbf{x}^t$ at iteration t, we propose to minimize a properly designed *approximate function* of the *upper bound function* $\overline{h}(\mathbf{x};\mathbf{x}^t)$, denoted as $\tilde{h}(\mathbf{x};\mathbf{x}^t)$:

$$\tilde{h}(\mathbf{x};\mathbf{x}^t)=\tilde{f}(\mathbf{x};\mathbf{x}^t)-(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)+g^+(\mathbf{x}), \tag{6}$$

where $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ is an approximate function of $f(\mathbf{x})$ at $\mathbf{x}^t$ that satisfies several technical conditions that are in the same essence as those specified in [6], namely,

(A1) The approximate function $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ is convex in $\mathbf{x}$ for any given $\mathbf{x}^t\in\mathcal{X}$;

(A2) The approximate function $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ is continuously differentiable in $\mathbf{x}$ for any given $\mathbf{x}^t\in\mathcal{X}$ and continuous in $\mathbf{x}^t$ for any $\mathbf{x}\in\mathcal{X}$;

(A3) The gradient of $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ and the gradient of $f(\mathbf{x})$ are identical at $\mathbf{x}=\mathbf{x}^t$ for any $\mathbf{x}^t\in\mathcal{X}$, i.e., $\nabla_{\mathbf{x}}\tilde{f}(\mathbf{x}^t;\mathbf{x}^t)=\nabla_{\mathbf{x}}f(\mathbf{x}^t)$.

Comparing $\overline{h}(\mathbf{x};\mathbf{x}^t)$ in (3) with $\tilde{h}(\mathbf{x};\mathbf{x}^t)$ in (6), we see that replacing $f(\mathbf{x})$ in $\overline{h}(\mathbf{x};\mathbf{x}^t)$ by its approximate function $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ leads to the proposed approximate function $\tilde{h}(\mathbf{x};\mathbf{x}^t)$. Note that $\tilde{h}(\mathbf{x};\mathbf{x}^t)$ is not necessarily a global upper bound of $\overline{h}(\mathbf{x};\mathbf{x}^t)$ (or the original function $h(\mathbf{x})$), because according to Assumptions (A1)-(A3), $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ does not have to be a global upper bound of $f(\mathbf{x})$.

At iteration t, the approximate problem consists of minimizing the approximate function $\tilde{h}(\mathbf{x};\mathbf{x}^t)$ over the same constraint set $\mathcal{X}$:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}}\ \tilde{h}(\mathbf{x};\mathbf{x}^t)=\tilde{f}(\mathbf{x};\mathbf{x}^t)-(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)+g^+(\mathbf{x}). \tag{7}$$

Since $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ is convex by assumption (A1), (7) is a convex optimization problem. We denote as $\mathbb{B}\mathbf{x}^t$ an (globally) optimal solution of (7) and as $\mathcal{S}(\mathbf{x}^t)$ the set of (globally) optimal solutions:

$$\mathbb{B}\mathbf{x}^t\in\mathcal{S}(\mathbf{x}^t)=\left\{\mathbf{x}^\star:\mathbf{x}^\star\in\arg\min_{\mathbf{x}\in\mathcal{X}}\tilde{h}(\mathbf{x};\mathbf{x}^t)\right\}. \tag{8}$$

Based on (8), we define the mapping $\mathbb{B}\mathbf{x}$ that is used to generate the sequence of points in the proposed algorithm:

$$\mathcal{X}\ni\mathbf{x}\longmapsto\mathbb{B}\mathbf{x}\in\mathcal{X}. \tag{9}$$

Given the mapping $\mathbb{B}\mathbf{x}$, the following properties hold.

Proposition 1 (Stationary point and descent direction)**.** *Provided that Assumptions (A1)-(A3) are satisfied: (i) A point $\mathbf{x}^t$ is a stationary point of (1) if and only if $\mathbf{x}^t\in\mathcal{S}(\mathbf{x}^t)$ defined in (8); (ii) If $\mathbf{x}^t$ is not a stationary point of (8), then $\mathbb{B}\mathbf{x}^t-\mathbf{x}^t$ is a descent direction of $\overline{h}(\mathbf{x};\mathbf{x}^t)$ at $\mathbf{x}=\mathbf{x}^t$ in the sense that*

$$(\mathbb{B}\mathbf{x}^t-\mathbf{x}^t)^T(\nabla f(\mathbf{x}^t)-\boldsymbol{\xi}^-(\mathbf{x}^t))+g^+(\mathbb{B}\mathbf{x}^t)-g^+(\mathbf{x}^t)<0. \tag{10}$$

Proof: From (8), there are two possibilities: $\tilde{h}(\mathbb{B}\mathbf{x}^t;\mathbf{x}^t)=\tilde{h}(\mathbf{x}^t;\mathbf{x}^t)$ and $\tilde{h}(\mathbb{B}\mathbf{x}^t;\mathbf{x}^t)<\tilde{h}(\mathbf{x}^t;\mathbf{x}^t)$.

We first show that the first possibility is equivalent to $\mathbf{x}^t$ being a stationary point of (1).

If $\tilde{h}(\mathbb{B}\mathbf{x}^t;\mathbf{x}^t)=\tilde{h}(\mathbf{x}^t;\mathbf{x}^t)$, then $\mathbf{x}^t$ is already an optimal point of the following convex problem:

$$\min_{\mathbf{x}}\ \tilde{h}(\mathbf{x};\mathbf{x}^t),$$

and it satisfies the first-order optimality condition:

$$(\mathbf{x}-\mathbf{x}^t)(\nabla\tilde{f}(\mathbf{x}^t;\mathbf{x}^t)+\boldsymbol{\xi}^+(\mathbf{x}^t)-\boldsymbol{\xi}^-(\mathbf{x}^t))\geq 0, \forall\mathbf{x}.$$

This is exactly the first-order optimality condition of problem (1) after replacing $\nabla\tilde{f}(\mathbf{x}^t;\mathbf{x}^t)$ by $\nabla f(\mathbf{x}^t)$ in view of Assumption (A3). Therefore, $\mathbf{x}^t$ is a stationary point of (1).

If $\mathbf{x}^t$ is a stationary point of (1), then it satisfies the first-order optimality condition:

$$(\mathbf{x}-\mathbf{x}^t)(\nabla f(\mathbf{x}^t)+\boldsymbol{\xi}^+(\mathbf{x}^t)-\boldsymbol{\xi}^-(\mathbf{x}^t))\geq 0, \forall\mathbf{x}.$$

By assumption (A3), the above condition is equivalent to

$$(\mathbf{x}-\mathbf{x}^t)(\nabla\tilde{f}(\mathbf{x}^t;\mathbf{x}^t)+\boldsymbol{\xi}^+(\mathbf{x}^t)-\boldsymbol{\xi}^-(\mathbf{x}^t))\geq 0, \forall\mathbf{x}.$$

In other words, $\mathbf{x}^t$ is an optimal solution of (8) and $\tilde{h}(\mathbf{x}^t;\mathbf{x}^t)=\min_{\mathbf{x}\in\mathcal{X}}\tilde{h}(\mathbf{x};\mathbf{x}^t)$.

To prove the second part of the proposition, we first remark that $\tilde{h}(\mathbf{x};\mathbf{x}^t)$ is convex and it is the sum of a differentiable function $f(\mathbf{x})-g^-(\mathbf{x}^t)-(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)$ and a convex function $g^+(\mathbf{x})$. Furthermore, problem (7) is equivalent to the following problem:

$$\begin{aligned}&\underset{\mathbf{x},y}{\text{minimize}}\tilde{f}(\mathbf{x};\mathbf{x}^t)-(\mathbf{x}-\mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)+y\\&\text{subject to}\mathbf{x}\in\mathcal{X}, g^+(\mathbf{x})\leq y,\end{aligned}$$

where the objective function is convex and differentiable, and thus also pseudoconvex [6, Figure 1]. It follows from the definition of pseudoconvex functions that $\tilde{h}(\mathbb{B}\mathbf{x}^t;\mathbf{x}^t)<\tilde{h}(\mathbf{x}^t;\mathbf{x}^t)$ implies

$$(\mathbb{B}\mathbf{x}^t-\mathbf{x}^t)^T(\nabla\tilde{f}(\mathbf{x}^t;\mathbf{x}^t)-\boldsymbol{\xi}^-(\mathbf{x}^t))+y^\star(\mathbb{B}\mathbf{x}^t)-y^t<0,$$

where $y^\star(\mathbb{B}\mathbf{x}^t) = g^+(\mathbb{B}\mathbf{x}^t)$, and y^t can be set to $g^+(\mathbf{x}^t)$ without loss of optimality [6, Sec. III-A]. Therefore, we readily obtain that

$$(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T(\nabla f(\mathbf{x}^t) - \boldsymbol{\xi}^-(\mathbf{x}^t)) + g^+(\mathbb{B}\mathbf{x}^t) - g^+(\mathbf{x}^t) < 0.$$

The proof is thus completed. ■

Theorem 2 (Convergence to a stationary point)**.** *Consider the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 1. Provided that Assumptions (A1)-(A3) as well as the following assumptions are satisfied:*

(A4) The solution set $\mathcal{S}(\mathbf{x}^t)$ is nonempty for $t = 1, 2, \ldots$;

(A5) Given any convergent subsequence $\{\mathbf{x}^t\}_{t\in\mathcal{T}}$ where $\mathcal{T} \subseteq \{1, 2, \ldots\}$, the sequence $\{\mathbb{B}\mathbf{x}^t\}_{t\in\mathcal{T}}$ is bounded.

Then any limit point of $\{\mathbf{x}^t\}$ is a stationary point of (1).

Proof: The proof follows the same line of analysis of [6, Theorem 2]. ■

If $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$ is a descent direction of $\overline{h}(\mathbf{x}; \mathbf{x}^t)$ at $\mathbf{x} = \mathbf{x}^t$, there always exists a scalar $\gamma^t \in (0, 1]$ such that [14, 8.2.1]

$$\overline{h}(\mathbf{x}^t + \gamma^t(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) < \overline{h}(\mathbf{x}^t).$$

This motivates us to update the variable as follows

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \gamma^t(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t), \tag{11}$$

so that we have in view of (4)

$$h(\mathbf{x}^{t+1}) \le \overline{h}(\mathbf{x}^{t+1}; \mathbf{x}^t) < \overline{h}(\mathbf{x}^t; \mathbf{x}^t) = h(\mathbf{x}^t). \tag{12}$$

In other words, the function value $h(\mathbf{x}^t)$ is monotonically decreasing.

There are several commonly used stepsize rules, for example, the constant/decreasing stepsize rules and the line search. In this paper, we restrict the discussion to the (exact) line search because it leads to a fast convergence speed as shown in [6]. On the one hand, the traditional exact line search aims at finding the stepsize that yields the largest decrease of $h(\mathbf{x})$ along the direction $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$:

$$\begin{aligned}\gamma^t &= \arg\min_{0\le\gamma\le 1} h(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ &= \arg\min_{0\le\gamma\le 1} \left\{ \begin{array}{l} f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ +g^+(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ -g^-(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \end{array} \right\}. \end{aligned} \tag{13}$$

Although it is a scalar problem, it is not necessarily easy to solve because it is nonconvex (even when $f(\mathbf{x})$ is convex) and nondifferentiable. On the other hand, as $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$ is a descent direction of $\overline{h}(\mathbf{x}; \mathbf{x}^t)$ according to Proposition 1, it is possible to perform the exact line search over $\overline{h}(\mathbf{x}; \mathbf{x}^t)$ along the direction $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$:

$$\begin{aligned}\gamma^t &= \arg\min_{0\le\gamma\le 1} \overline{h}(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t); \mathbf{x}^t) \\ &= \arg\min_{0\le\gamma\le 1} \left\{ \begin{array}{l} f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ -(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t) - \mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t) \\ g^+(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \end{array} \right\}. \end{aligned} \tag{14}$$

However, this is not favorable in practice either because the above minimization problem involves the nondifferentiable function g^+.

To reduce the complexity, we propose to perform the line search by solving the following one dimensional optimization problem:

$$\begin{aligned}\gamma^t &= \arg\min_{0\le\gamma\le 1} \left\{ \begin{array}{l} f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ -(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t) - \mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t) \\ +g(\mathbf{x}^t) + \gamma(g^+(\mathbb{B}\mathbf{x}^t) - g^+(\mathbf{x}^t)). \end{array} \right\} \\ &= \arg\min_{0\le\gamma\le 1} \left\{ \begin{array}{l} f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ +\gamma(g^+(\mathbb{B}\mathbf{x}^t) - g^+(\mathbf{x}^t) - (\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)) \end{array} \right\}. \end{aligned} \tag{15}$$

Algorithm 1 The parallel best-response algorithm with exact line search for problem (1)

Data: $t = 0$, $\mathbf{x}^0$ (arbitrary but fixed, e.g., $\mathbf{x}^0 = \mathbf{0}$), stop criterion δ.
S1: Compute $\mathbb{B}\mathbf{x}^t$ according to (8).
S2: Determine the stepsize γ^t by the exact line search (15).
S3: Update $\mathbf{x}^{t+1}$ according to (11).
S4: If $h(\mathbf{x}^t) - h(\mathbf{x}^{t+1}) \le \delta$, STOP; otherwise $t \leftarrow t + 1$ and go to **S1**.

Note that the objective function in (15) is differentiable, which is furthermore convex if $f(\mathbf{x})$ is convex. As a matter of fact, it is an upper bound of the objective function in (14) after applying Jensen's inequality to the convex but nondifferentiable function $g^+(\mathbf{x})$:

$$g(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \le g^+(\mathbf{x}^t) + \gamma(g^+(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t) - g^+(\mathbf{x}^t)).$$

We remark that the same line of analysis can also be used to design a low complexity successive line search (the Armijo rule) which is carried out over a differentiable function. It is an useful alternative for the exact line search (15) if the optimization problem in (15) is difficult to solve. The details are omitted here due to the page limit.

The proposed single-layer algorithm is summarized in Algorithm 1 and its convergence properties are given in the following theorem.

Sufficient conditions for Assumptions (A4)-(A5) are that either the feasible set $\mathcal{X}$ in (7) is bounded or the approximate function in (7) is strongly convex [15]. We will show that these assumptions are satisfied by the example application in the next section.

In what follows, we draw some comments on the proposed algorithm's features and connections to existing algorithms.

On the complexity of the proposed algorithm. The Algorithm 1 has a low complexity due to the use of an approximate function and the line search scheme over a differentiable function. The benefits of employing the approximate function $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ are twofold. On the one hand, it is a convex function by Assumption (A1), so the approximate problem (7) is a convex problem, which is presumably easier to solve than (5) which is nonconvex if $f(\mathbf{x})$ is nonconvex. On the other hand, it can be tailored according to the structure of the problem at hand so that the approximate problem (7) even easier to solve. For example, if $g^+(\mathbf{x})$ is separable among the scalar elements of $\mathbf{x}$ (as in, e.g., ℓ_1-norm $\|\mathbf{x}\|_1 = \sum_{k=1}^K |x_k|$), we could choose $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ to be separable as well, so that the problem (7) can be decomposed into independent subproblems which are then solved in parallel. Furthermore, the proposed line search scheme (15) is carried out over a differentiable function, which is presumably easier to implement than traditional schemes (13)-(14) over nonconvex nondifferentiable functions and leads to faster convergence than constant and decreasing stepsizes.

On the choice of approximate function. Note that different choices of $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ lead to different algorithms. For example, when $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ is based on linear approximation, the resulting algorithm would be a proximal-like algorithm. When $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ is based on the best-response approximation, the resulting algorithm would be a Jacobi-like algorithm. We refer interested readers to [6, Sec. III-B] for a comprehensive discussion.

On the connection to the MM method [12]. The function $f(\mathbf{x})$ is assumed to be convex in [12]. The proposed algorithm includes as a special case the MM method proposed in [12] by setting $\tilde{f}(\mathbf{x}; \mathbf{x}^t) = f(\mathbf{x})$, i.e., no approximation is employed. For this particular choice of approximate function, it can be verified that the assumptions (A1)-(A3) are satisfied, and additionally the approximate function is a global upper bound of the original function $h(\mathbf{x})$. It is possible to show that in this case the proposed Algorithm 1 converges (in the

sense specified by Theorem 2) under a constant unit stepsize $\gamma^t = 1$. We omit the detailed steps due to the page limit.

On the comparison with GIST [13]. In the GIST algorithm proposed in [13], the variable is updated by

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \begin{array}{l} f(\mathbf{x}^t) + \nabla f(\mathbf{x}^t)^T(\mathbf{x}-\mathbf{x}^t) + \frac{c^t}{2}\left\|\mathbf{x}-\mathbf{x}^t\right\|^2 \\ +g^+(\mathbf{x}) - g^-(\mathbf{x}) \end{array} \right\},$$

where $c^t > L_{\nabla f}$ and $L_{\nabla f}$ is the Lipschitz constant of ∇f. When the value of $L_{\nabla f}$ is not known, c^t should be estimated iteratively: for some constant $0 < \alpha < 1$ and $\beta > 1$, $\mathbf{x}^{t+1} = \mathbf{x}^\star(\beta^{m_t})$, where $\mathbf{x}^\star(\beta^m)$ is defined as

$$\mathbf{x}^\star(\beta^m) \triangleq \arg\min_{\mathbf{x}} \left\{ \begin{array}{l} f(\mathbf{x}^t) + \nabla f(\mathbf{x}^t)^T(\mathbf{x}-\mathbf{x}^t) + \frac{\beta^m}{2}\left\|\mathbf{x}-\mathbf{x}^t\right\|^2 \\ +g^+(\mathbf{x}) - g^-(\mathbf{x}) \end{array} \right\} \tag{16}$$

and m_t is the first nonnegative integer such that $h(\mathbf{x}^\star(\beta^m)) - h(\mathbf{x}^t) \le -\alpha/2\beta^m \left\|\mathbf{x}^\star(\beta^m) - \mathbf{x}^t\right\|^2$. As a result, $\mathbf{x}^\star(\eta^m)$ must be evaluated repeatedly for m_t times, namely, $m = 0, 1, \ldots, m_t$ and it incurs additional complexity. This is however not necessary in the proposed algorithm, because computing the descent direction and the stepsize according to (8) and (15) does not depend on any unknown parameters. Furthermore, (16) may not be easy to solve for a general $g^-(\mathbf{x})$ except for some specific choices studied in [13].

III. Sparse Parameter Estimation with Capped ℓ_1 Regularization

In this section, we consider as an application the sparse signal estimation problem with a capped ℓ_1 regularization [10, 12, 13]:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2 + \mu\left\|\min(\mathbf{x}, \theta\mathbf{1})\right\|_1, \tag{17}$$

where $\mathbf{1}$ is a vector with all elements equal to 1, and $\min(\mathbf{x},\mathbf{y}) \triangleq (\min(x_k, y_k))_{k=1}^K$. Problem (17) is a special case of (1) by setting $f(\mathbf{x}) \triangleq \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2$,

$$g^+(\mathbf{x}) \triangleq \mu\left\|\mathbf{x}\right\|_1, \text{ and } g^-(\mathbf{x}) \triangleq \mu\left\|\mathbf{x}\right\|_1 - \mu\left\|\min(\mathbf{x}, \theta\mathbf{1})\right\|_1.$$

Since $f(\mathbf{x})$ is convex, we adopt the approximate function that is of a best-response type: the approximate function consists of K component functions, and in the k-th component function, only the k-th element, x_k, of $\mathbf{x}$ is treated as a variable while other elements $\mathbf{x}_{-k} \triangleq (x_j)_{j\neq k}$ are fixed,

$$\tilde{f}(\mathbf{x};\mathbf{x}^t) = \frac{1}{2}\sum_{k=1}^K f(x_k, \mathbf{x}_{-k}^t) = \frac{1}{2}\sum_{k=1}^K \left\|\mathbf{a}_k x_k + \sum_{j\neq k}\mathbf{a}_j x_j^t - \mathbf{b}\right\|_2^2. \tag{18}$$

To obtain the update direction, we solve the approximate problem

$$\begin{aligned} \mathbb{B}\mathbf{x}^t &= \arg\min_{\mathbf{x}} \{\tilde{f}(\mathbf{x};\mathbf{x}^t) - (\mathbf{x}-\mathbf{x}^t)\boldsymbol{\xi}^-(\mathbf{x}^t) + g^+(\mathbf{x})\} \\ &= \mathbf{d}(\mathbf{A}^T\mathbf{A})^{-1} \circ \mathcal{S}_{\mu\mathbf{1}}(\mathbf{r}(\mathbf{x}^t, \boldsymbol{\xi}^-(\mathbf{x}^t))), \end{aligned} \tag{19}$$

where $\mathbf{r}(\mathbf{x}^t, \boldsymbol{\xi}^-(\mathbf{x}^t)) \triangleq \mathbf{d}(\mathbf{A}^T\mathbf{A}) \circ \mathbf{x}^t + \boldsymbol{\xi}^-(\mathbf{x}^t) - \mathbf{A}^T(\mathbf{A}\mathbf{x}^t - \mathbf{b})$, $\mathbf{d}(\mathbf{A}^T\mathbf{A})$ is the diagonal vector of $\mathbf{A}^T\mathbf{A}$, $\mathbf{a}\circ\mathbf{b}$ denotes the Hadamard product between $\mathbf{a}$ and $\mathbf{b}$, $\mathcal{S}_{\mathbf{a}}(\mathbf{b}) \triangleq [\mathbf{b}-\mathbf{a}]^+ - [-\mathbf{b}-\mathbf{a}]^+$ is the soft-thresholding operator, and the subgradient of $g^-(\mathbf{x})$ is $\boldsymbol{\xi}^-(\mathbf{x}) = (\xi_k^-(x_k))_{k=1}^K$ with $\xi_k^-(x_k) = \mu(\text{sign}(x_k-\theta) - \text{sign}(-x_k-\theta))/2$.

Given the update direction $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$, we calculate the stepsize γ^t according to the proposed exact line search (15):

$$\begin{aligned} \gamma^t &= \arg\min_{0\le\gamma\le 1} \left\{ \begin{array}{l} f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \\ +\gamma(g^+(\mathbb{B}\mathbf{x}^t) - g^+(\mathbf{x}^t) - (\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T\boldsymbol{\xi}^-(\mathbf{x}^t)) \end{array} \right\} \\ &= \left[\frac{(\boldsymbol{\xi}^-(\mathbf{x}^t) - \mathbf{A}^T(\mathbf{A}\mathbf{x}^t - \mathbf{b}))^T(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t) - \mu(\left\|\mathbb{B}\mathbf{x}^t\right\|_1 - \left\|\mathbf{x}^t\right\|_1)}{(\mathbf{A}(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t))^T(\mathbf{A}(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t))} \right]_0^1. \end{aligned} \tag{20}$$

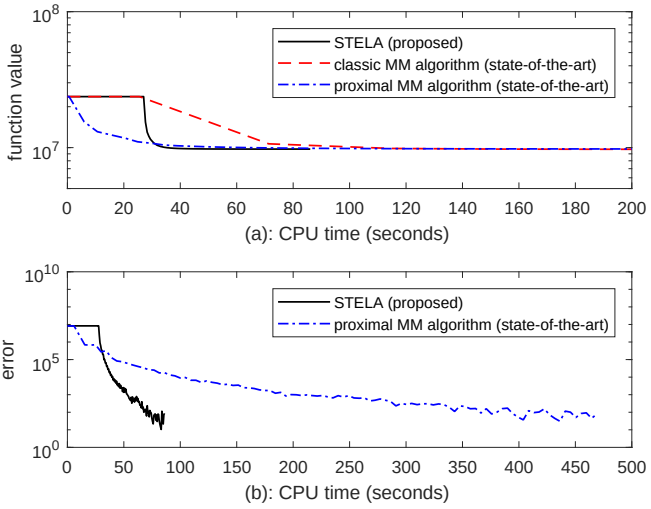

Figure 1. Capped ℓ_1-norm minimization: Achieved function value $h(\mathbf{x}^t)$ and error $h(\mathbf{x}^{t-1}) - h(\mathbf{x}^t)$ versus the CPU time (in seconds).

We name the proposed update (19)-(20) as Soft-Thresholding with Exact Line search Algorithm (STELA).

For the capped ℓ_1-norm minimization problem (17), the proposed algorithm (19)-(20) exhibits a fast convergence to a stationary point and has a low complexity due to several attractive features: i) it is a single-layer algorithm; ii) all elements are updated in parallel with a closed-form expression; iii) the approximate function is of a best-response type; iv) the stepsize is based on the exact line search which has a closed-form expression; v) the proposed algorithm converges to a stationary point of the nonconvex problem (17) by Theorem 2 (note that $\tilde{f}(\mathbf{x};\mathbf{x}^t)$ in (18) is strongly convex so Assumptions (A4)-(A5) are satisfied). We remark that feature i) is an advantage over the traditional MM method [12]; feature ii) is an advantage over the BCD algorithm [3] and traditional MM method [12]; feature iii) is an advantage over the proximal type algorithm [13] with a linear approximation; feature iv) is an advantage over commonly used constant and decreasing stepsizes [16, 17]; feature v) is an advantage over the BCD algorithm [3], FISTA [2], and the standard SCA framework [6, 17, 18].

Simulations. The dimension of $\mathbf{A}$ is 10000×50000 and all of its elements are generated randomly by the normal distribution. The density (the proportion of nonzero elements) of the sparse vector $\mathbf{x}_{\text{true}}$ is 0.1. The vector $\mathbf{b}$ is generated as $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \mathbf{e}$ where $\mathbf{e}$ is drawn from an i.i.d. Gaussian distribution with variance 10^{-4}. The regularization gain μ is set to $\mu = 0.1\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty$, which allows $\mathbf{x}_{\text{true}}$ to be recovered to a high accuracy [19], and the parameter θ in the capped ℓ_1-norm is set to 1.

We compare the proposed algorithm STELA with the MM method [12] and the GIST algorithm [13]. The comparison is made in terms of CPU time that is required until the maximum number of iterations (100 for STELA and the GIST algorithm and 10 for the MM method) is reached. The running time consists of both the initialization stage required for preprocessing (represented by a flat curve) and the formal stage in which the iterations are carried out. For example, in STELA, $\mathbf{d}(\mathbf{A}^T\mathbf{A})$ is computed in the initialization stage since it is required in the iterative variable update in the formal stage, cf. (19). The upper bound function in the MM method, cf. (5), is minimized by STELA for ℓ_1-norm (with a warm start), which was presented in [6, Sec. IV-III]. All algorithms have the same initial point, $\mathbf{x}^0 = \mathbf{0}$. The simulation results are averaged over 20 instances.

The achieved function value $h(\mathbf{x}^t)$ and error $h(\mathbf{x}^{t-1}) - h(\mathbf{x}^t)$ versus the CPU time (in seconds) is plotted in Figure 1 (a) and 1 (b), respectively. We see from Figure 1 (a) that all algorithms converge to the same value. Furthermore, the initialization stage of STELA is much longer than that of the GIST algorithm, because computing $\mathbf{d}(\mathbf{A}^T\mathbf{A})$, the diagonal vector of $\mathbf{A}^T\mathbf{A}$, is computationally expensive, especially when the dimension of $\mathbf{A}$ is large. Nevertheless, in the formal stage, the convergence speed of STELA is much faster than the GIST algorithm, and this is mainly due to the use of the best-response type approximate function (18), and more specifically, the use of $\mathbf{d}(\mathbf{A}^T\mathbf{A})$, cf. (19), which represents partial second order information of the function $f(\mathbf{x})$ in (17) (note that $\nabla^2 f(\mathbf{x}) = \mathbf{A}^T\mathbf{A}$). We see from Figure 1 (b) that the long initialization stage is compensated by the fast convergence speed in the formal stage. We mention for the paper's completeness that $\mathbf{d}(\mathbf{A}^T\mathbf{A})$ can be calculated analytically in some applications, e.g., when $\mathbf{A}$ is a Vandermonde matrix.

We see from Figure 1 (a) that the major complexity of the MM method lies in the beginning iterations, as the initial point in the beginning iterations is usually far away from the optimal point that minimizes the upper bound function and more iterations are needed. The most notable difference between the MM method and the STELA is that the upper bound function is only approximately minimized in the STELA, and this leads to a significant reduction in the computational complexity. Using the approximate function is also beneficial when the upper bound function $\overline{h}(\mathbf{x}; \mathbf{x}^t)$ is not easy to minimize, e.g., $f(\mathbf{x})$ is nonconvex.

IV. Concluding Remarks

In this paper, we have proposed a successive convex approximation framework for sparse optimization where the nondifferentiable nonconvex regularization function can be written as the difference of two convex functions. The proposed procedure is to apply the standard successive convex approximation for nonconvex optimization where the regularization function is convex to an upper bound of the original objective function that can be obtained following the standard convex-concave procedure. This procedure also facilitates the design of the line search which is carried out over a differentiable function. The proposed framework is flexible and it leads to algorithms that exploit the problem structure and have a low complexity. Customizing the proposed framework for the example application where the nonconvex regularization is the capped ℓ_1-norm function, we obtain a best-response type algorithm for which all elements are updated in parallel according to closed-form expressions. The advantages of the proposed algorithms are finally numerically illustrated.

References

[1] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 1st ed. Academic Press, 2015.

[2] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm," *Society for Industrial and Applied Mathematics Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[3] P. Tseng, "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, 2010.

[5] N. Parikh and S. Boyd, "Proximal Algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[6] Y. Yang and M. Pesavento, "A Unified Successive Pseudoconvex Approximation Framework," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.

[7] Z. Yang, Z. Wang, H. Liu, Y. C. Eldar, and T. Zhang, "Sparse Nonlinear Regression: Parameter Estimation and Asymptotic Inference," in *International Conference on Machine Learning (ICML)*, 2016.

[8] J. Fan and R. Li, "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[9] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing Sparsity by Reweighted L1 Minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.

[10] T. Zhang, "Analysis of Multi-stage Convex Relaxation for Sparse Regularization," *Journal of Machine Learning Research*, vol. 11, pp. 1081–1107, 2010.

[11] J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping, "The use of zero-norm with linear models and kernel methods," *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, 2003.

[12] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and DC programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, Dec. 2009.

[13] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye, "A General Iterative Shrinkage and Thresholding Algorithm for Non-convex Regularized Optimization Problems," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 37–45.

[14] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. Academic, New York, 1970.

[15] S. M. Robinson and R. H. Day, "A sufficient condition for continuity of optimal sets in mathematical programming," *Journal of Mathematical Analysis and Applications*, vol. 45, no. 2, pp. 506–511, Feb. 1974.

[16] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by Partial Linearization: Parallel Optimization of Multi-Agent Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[17] M. Razaviyayn, M. Hong, Z.-Q. Luo, and J.-S. Pang, "Parallel Successive Convex Approximation for Nonsmooth Nonconvex Optimization," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, pp. 1440–1448.

[18] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel Selective Algorithms for Nonconvex Big Data Optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, Nov. 2015.

[19] S. Wright, R. Nowak, and M. Figueiredo, "Sparse Reconstruction by Separable Approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.