A majorized proximal point dual Newton algorithm for nonconvex statistical optimization problems

Defeng Sun

Department of Applied Mathematics

THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

The Sixth International Conference on Continuous Optimization
Technical University (TU) of Berlin, August 3-8, 2019.

Based on joint works with Kim-Chuan Toh (NUS), Xudong Li (Fudan)
Peipei Tang (ZUCC), Chengjing Wang (SWJTU), Ning Zhang (PolyU)

The Purpose

Second Order Methods

with

First Order Costs

Newton's methods

Our approach — Newton's methods (second order methods) with low costs [can be very low]

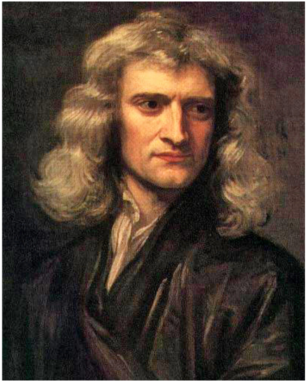

Figure: Sir Isaac Newton (Niu Dun) (4 January 1643 - 31 March 1727)

Solving a sparse + low rank linear equation problem: a basic idea

1. One core mathematical problem is to solve the following linear equation

$$Bx = b,$$

where $B \in \Re^{n\times n}$ and $b \in \Re^n$

2. Assume that B is non-singular. One can use the Gaussian elimination method [the ancient Chinese (Jiu Zhang Suan Shu) and Indians discovered this method thousand years ago] to get

$$x = B^{-1}b$$

with the cost of $O(n^3)$ flops – way too high for a big n.

3. If B is "sparse", e.g., $B = I + uv^T$, where $u, v \in \Re^n$, one can reduce the cost from $O(n^3)$ to $O(n)$ via the Sherman-Morrison-Woodbury formula

$$x = B^{-1}b = (I + uv^T)^{-1}b = b - v\underbrace{(1 + v^Tu)^{-1}}_{\text{number}}(u^Tb).$$

4. Solving structured linear equations can be cheap $\Longrightarrow$ Second order methods with first order computational costs possible!!!

The Setting

Let $X \in \Re^{m \times n}$ be the input data and b be the response variables with a noise vector $\varepsilon = b - X\ddot{\beta}$. Let $\lambda > 0$. One of the most commonly used models to control the overfitting and/or variable selection is the Lasso model

$$\min_{\beta \in \Re^n} \left\{ \frac{1}{2}\|X\beta - b\|^2 + \lambda\|\beta\|_1 \right\}$$

which relies on knowing the standard deviation of the noise. Here in the convex case we are interested in the more general model

$$\min_{\beta \in \Re^n} \left\{ \underbrace{h(X\beta)}_{f(\beta)} + p(\beta) \right\}$$

where both $h(\cdot)$ and $p(\cdot)$ are proper and closed convex functions, which can be nonsmooth or non-Lipschitzian. Here, h is not assumed to be differentiable!!!

Simple convex examples

One interesting example is the square-root Lasso (srLasso) model (Alex Belloni et al. (2011))

$$\min_{\beta \in \Re^n} \{\|X\beta - b\| + \lambda \|\beta\|_1\}$$

which is equivalent to the robust least regression (Huan Xu et al. 2010)

$$\min_{\beta \in \Re^n} \left\{ \max_{\Delta X \in \mathcal{U}} \|b - (X + \Delta X)\beta\| \right\}$$

with the uncertainty set

$$\mathcal{U} := \{(\Delta_1, \ldots, \Delta_n) \mid \|\Delta_i\| \leq \lambda, \ i = 1, \ldots, n\}$$

Another commonly used example is the constrained Lasso model

$$\min_{\beta \in \Re^n} \{\|\beta\|_1 \ \mid \ \|X\beta - b\| \leq \tau\}$$

Here $h(\cdot)$ is the indicator function

$$h(y) := \delta_{B^\tau}(y) \quad \forall\, y \in \Re^m$$

over the ball $B^\tau := \{y \in \Re^m \,|\, ||y|| \leq \tau\}$ centered at 0 with radius $\tau > 0$

More Lasso-type problems

There are many more convex Lasso-type models:

(LASSO)

$$\min\left\{\frac{1}{2}\|X\beta - b\|^2 + \lambda\|\beta\|_1 \mid \beta \in \Re^n\right\}$$

where $\lambda > 0$.

(Fused LASSO)

$$\min\left\{\frac{1}{2}\|X\beta - b\|^2 + \lambda\|\beta\|_1 + \lambda_2\|B\beta\|_1\right\}$$

$$B = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix}$$

More Lasso-type problems (continued)

(Clustered LASSO)

$$\min\left\{\frac{1}{2}\|X\beta - b\|^2 + \lambda\|\beta\|_1 + \lambda_2 \sum_{i=1}^{n}\sum_{j=i+1}^{n} |\beta_i - \beta_j|\right\}$$

Note that the above problem is not numerically solvable if n is large as the objective function value computation itself would cost $O(n^2)$ flops.

Fortunately, [Lin-Liu-S.-Toh 2018] showed

$$\sum_{1\le i<j\le n} |\beta_i - \beta_j| \;=\; \langle w, \beta^{\downarrow}\rangle,$$

where $\beta^{\downarrow}$ is the vector whose components are those of β sorted in a non-increasing order, i.e. $\beta_1^{\downarrow} \ge \beta_2^{\downarrow} \ge \cdots \ge \beta_n^{\downarrow}$ [costs $O(n\log n)$] and the weight vector $w \in \Re^n$ is defined by

$$w_k = n - 2k + 1,\; k = 1, \cdots, n.$$

More Lasso-type problems (continued)

(SLOPE, Ordered Lasso)

$$\min\left\{\frac{1}{2}\|X\beta - b\|^2 + \sum_{i=1}^{n} \lambda_i |\beta|_i^{\downarrow}\right\}$$

with parameters $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\lambda_1 > 0$.

We are interested in n (number of features) large and/or m (number of samples) large. Note that the regularization term in SLOPE (ordered Lasso) is not separable.

More on the loss functions

In the loss function part, f can be the logistic regression function, defined as below: given $b \in \Re^m$ and $X \in \Re^{m \times n}$,

$$f(\beta) = \sum_{i=1}^{m} \log(1 + \exp(-b_i (X\beta)_i)) \tag{1}$$

- Define $h : \Re^m \to \Re$ as follows

$$h(z) = \sum_{i=1}^{m} \log(1 + \exp(-b_i z_i)) \quad \forall\, z \in \Re^m$$

The function f defined by (1) can be written as

$$f(\beta) = h(X\beta)$$

More on the loss functions (continued)

In general, the loss function $f : \Re^{pK} \to \Re$ can take as the multinomial logistic regression function: given $A := (A_1, \ldots, A_N)^T \in \Re^{N \times p}$,

$$f(\beta) = -\sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik} A_i^T \beta_k - \log \sum_{k=1}^{K} \exp(A_i^T \beta_k) \right) \tag{2}$$

- Define $h : \Re^{NK} \to \Re$ as follows:

$$h(z) := -\sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik} Z_{ik} - \log \sum_{k=1}^{K} \exp(Z_{ik}) \right), \text{ with } Z := \mathrm{mat}(z)$$

Then the function f defined by (2) can be written as

$$f(\beta) := h(X\beta), \text{ with } X := I \otimes A \in \Re^{NK \times pK}$$

The nonconvex case

A proper nonconvex regularization can achieve a sparse estimation with fewer measurements, faster convergence and more robust against noises.

In this talk, we aim to develop an efficient and robust algorithm for solving the following nonconvex problem **(P)**:

$$\min_{\beta \in \Re^n} \left\{ g(\beta) := \underbrace{h(X\beta)}_{f(\beta)} + \underbrace{p(\beta) - q(\beta)}_{r(\beta)} \right\} \tag{3}$$

Here $p : \Re^n \to (-\infty, +\infty]$ is a proper closed convex function and $q : \Re^n \to \Re$ is a finite-valued (smooth, not essential) convex function. Moreover, we require the proximal functions of h and p to be (strongly) semismooth.

Nonconvex regularizers: SCAD

For $\lambda > 0$, the SCAD regularization is defined by $r(\beta) = p(\beta) - q(\beta)$ with

$$p(\beta) = \lambda \|\beta\|_1$$

$$q(\beta) = \sum_{i=1}^{n} \begin{cases} 0, & \text{if} \quad |\beta_i| < \lambda \\ \frac{(|\beta_i| - \lambda)^2}{2(a_s - 1)}, & \text{if} \quad \lambda \le |\beta_i| \le a_s \lambda \\ \lambda |\beta_i| - \frac{a_s + 1}{2} \lambda^2, & \text{if} \quad |\beta_i| > a_s \lambda \end{cases}$$

Note that $q(\cdot)$ is continuously differentiable. In our numerical experiments, we take $a_s = 3.7$.

Nonconvex regularizers: MCP

For two positive parameters $a_m > 2$ and λ, the MCP regularization can be defined as $r(\beta) = p(\beta) - q(\beta)$ with

$$p(\beta) = \lambda \|\beta\|_1$$

$$q(\beta) = \sum_{i=1}^{n} \begin{cases} \frac{\beta_i^2}{a_m}, & \text{if } |\beta_i| \leq a_m\lambda, \\ 2\lambda|\beta_i| - a_m\lambda^2, & \text{if } |\beta_i| > a_m\lambda \end{cases}$$

The function $q(\cdot)$ is continuously differentiable with its derivative given by

$$\frac{\partial q(\beta)}{\partial \beta_i} = \begin{cases} \frac{2\beta_i}{a_m}, & \text{if } |\beta_i| \leq a_m\lambda, \\ 2\lambda \,\mathsf{sign}(\beta_i), & \text{if } |\beta_i| > a_m\lambda \end{cases}$$

In our numerical experiments, we take $a_m = 3.7$.

Nonconvex regularizers: Difference of Ky Fan norms

For any positive integer k, let $\|\cdot\|_{(k)}$ denote Ky Fan's k-norm function, i.e., for any $\beta \in \Re^n$, $\|\beta\|_{(k)}$ is the sum of the first k largest absolute values of β.

By noting that the cardinality constraint

$$\|\beta\|_0 \leq k$$

can be written equivalently as

$$\|\beta\|_{(k)} = \|\beta\|_1,$$

in the majorized penalty method (Gao & S., 2010) we define the regularization term $r(\beta) = p(\beta) - q(\beta)$ with

$$p(\beta) = ||\beta\|_1 \quad \& \quad q(\beta) = \|\beta\|_{(k)} \quad \forall\, \beta \in \Re^n$$

or

$$p(\beta) = ||\beta\|_{(k+1)} \quad \& \quad q(\beta) = \|\beta\|_{(k)} \quad \forall\, \beta \in \Re^n$$

Note that $q(\cdot)$ is continuously differentiable near any β with $\|\beta\|_0 = k$.

mPPDNA

Next, we shall consider the promised majorized proximal point dual Newton algorithm (mPPDNA) to solve the following problem [e.g., the srLasso problem for example]

$$\min_{\beta \in \Re^n} \Big\{ h(X\beta) + p(\beta) - q(\beta) \Big\}$$

- In Stage 1, replace q by its linear approximation at the origin [when $q(0) = 0$ and $0 \in \partial q(0)$, which hold true for many interesting cases of q, we just delete q from the original problem] and add "proper" proximal terms to obtain an initial point for the second stage.
- In Stage 2, a series of majorized proximal subproblems are solved to obtain an approximate solution point.

The subproblem

Given $\sigma > 0$, $\tau > 0$, $\tilde{\beta} \in \Re^n$, $\tilde{v} \in \Re^n$, $\tilde{b} \in \Re^m$, in our main algorithm (mPPDNA) we need to solve the following minimization subproblem

$$\min_{\beta \in \Re^n} \Big\{ g(\beta; \sigma, \tau, \tilde{\beta}, \tilde{v}, \tilde{b}) \quad := \quad \underbrace{h(X\beta) + p(\beta)}_{\text{convex}} - \underbrace{(q(\tilde{\beta}) + \langle \tilde{v}, \beta - \tilde{\beta} \rangle)}_{\text{linear}} + \frac{\sigma}{2}\|\beta - \tilde{\beta}\|^2 + \frac{\tau}{2}\|X\beta - \tilde{b}\|^2 \Big\} \qquad (4)$$

Here, $\tilde{v} \in \partial q(\tilde{\beta})$. Obviously, $g(\cdot; \sigma, \tau, \tilde{\beta}, \tilde{v}, \tilde{b})$ is a strongly convex function albeit nonsmooth or non-Lipschitian.

- The big question is how one can solve (4) in a fast and robust way!!!
- For the convex case: $q \equiv 0$
- If h is strongly convex on $\mathcal{E}$, we can take $\tau = 0$ though not necessary

The dual of the subproblem

The dual of (4), after converting it into the minimization form and ignoring the constant term, is

$$\min_{u\in\Re^m} \Big\{\phi(u;\sigma,\tau) := \frac{\tau}{2}\|\tilde{b}+\tau^{-1}u\|^2 - e_\tau h(\tilde{b}+\tau^{-1}u) \\ +\frac{\sigma}{2}\|\tilde{\beta}+\sigma^{-1}(\tilde{v}-X^*u)\|^2 - e_\sigma p(\tilde{\beta}+\sigma^{-1}(\tilde{v}-X^*u))\Big\}.$$

Recall that for any $t>0$, $e_t f(\cdot)$ is the Moreau envelope of a closed proper convex function f, associated with t, given by

$$e_t f(x) := \min_{z\in\Re^n} \left\{ f(z) + \frac{t}{2}\|z-x\|^2 \right\}, \quad \forall\, x\in\Re^n. \tag{5}$$

Here $e_t f(\cdot)$ is continuously differentiable with

$$\nabla e_t f(x) = t[x - \mathsf{P}_t f(x)], \quad \forall\, x\in\Re^n,$$

where $\mathsf{P}_t f(x)$ is the unique optimal solution to problem (5). $\mathsf{P}_t f(\cdot)$, called the proximal mapping of f, is globally Lipschitz continuous with modulus 1.

A semismooth Newton method for the subproblem

We shall apply the superlinearly (quadratically) convergent sparse semismooth Newton method to find the solution $\bar{u}$ of the nonsmooth equations

$$\nabla\phi(u;\sigma,\tau) \quad = \quad \mathsf{P}_\tau h(\tilde{b}+\tau^{-1}u) - X\mathsf{P}_\sigma p(\tilde{\beta}+\sigma^{-1}(\tilde{v}-X^*u)) = 0.$$

Then the unique optimal solution $\bar{\beta}$ to problem (4) is

$$\bar{\beta} = \mathsf{P}_\sigma p(\tilde{\beta}+\sigma^{-1}(\tilde{v}-X^*\bar{u})).$$

Proposition

Suppose that problem (4) is nondegenerate, which holds true if $f(\cdot) \equiv h(X\cdot)$ is continuously differentiable near $\bar{\beta}$ (this is the no-overfitting assumption for the squared root Lasso problem). Then all the elements in Clarke's generalized Jacobian $\partial^2\phi(\bar{u})$ are self-adjoint and positive definite.

A globalized SSN algorithm for the subproblem (4)

Algorithm SSN (SSN(σ, τ)): Given $\mu \in (0, \frac{1}{2})$, $\overline{\eta} \in (0,1)$, $\bar{\tau} \in (0,1]$, ν_1, $\nu_2 \in (0,1)$, and $\delta \in (0,1)$, choose $u^0 \in \Re^m$. Set $j = 0$ and iterate the following steps.

1. Choose $V^j \in \partial \mathsf{P}_\tau h(\tilde{b} + \tau^{-1} u^j)$ and $U^j \in \partial \mathsf{P}_\sigma p(\tilde{\beta} + \sigma^{-1}(\tilde{v} - X^* u^j))$. Let $H^j = \tau^{-1} V^j + \sigma^{-1} X U^j X^*$ and find the exact solution Δu^j or apply the PCG method to find an approximate solution Δu^j to

   $$(H^j + \varepsilon_j I)\Delta u = -\nabla \phi(u^j; \sigma, \tau)$$

   such that

   $$\|H^j \Delta u^j + \nabla \phi(u^j; \sigma, \tau)\| \leq \eta_j := \min(\overline{\eta}, \|\nabla \phi(u^j; \sigma, \tau)\|^{1+\bar{\tau}})$$

   where $\varepsilon_j := \nu_1 \min\left\{\nu_2, \|\nabla \phi(u^j; \sigma, \tau)\|\right\}$
2. Set $\alpha_j = \delta^{l_j}$, where l_j is the first nonnegative integer l for which

   $$\phi(u^j + \delta^l \Delta u^j; \sigma, \tau) \leq \phi(u^j; \sigma, \tau) + \mu \delta^l \langle \nabla \phi(u^j; \sigma, \tau), (\Delta u^j) \rangle$$
3. Set $u^{j+1} = u^j + \alpha_j \Delta u^j$

Nonsmooth Newton's method for inner problems

Theorem

Assume that $P_\tau h(\cdot)$ and $P_\sigma p(\cdot)$ are strongly semismooth. If problem (4) is nondegenerate, in particular if $f(\cdot) \equiv h(X\cdot)$ is continuously differentiable near $\bar{\beta}$, then $\{u^j\}$ converges to the unique optimal solution $\bar{u}$ and

$$\|u^{j+1} - \bar{u}\| = O(\|u^j - \bar{u}\|^{1+\bar{\tau}}).$$

Note that if $\bar{\tau} = 1$, we get the quadratic convergence.

mPPDNA

Algorithm. Let $\sigma^0, \sigma^1 > 0, \tau^0, \tau^1 > 0$ be given parameters

1. Compute

$$\beta^1 \approx \operatorname*{argmin}_{\beta \in \Re^n} \left\{ g(\beta; \sigma^0, \tau^0, 0, 0, b) \right\}$$

   via solving its dual problem such that a prescribed stopping criterion is satisfied. Let $k = 1$ and go to Step 2.1.

2.1 Choose $v^k \in \partial q(\beta^k)$ and compute

$$\beta^{k+1} = \operatorname*{argmin}_{\beta \in \Re^n} \left\{ g(\beta; \sigma^k, \tau^k, \beta^k, v^k, X\beta^k) + \langle \delta^k, \beta - \beta^k \rangle \right\}$$

   via solving its dual problem such that the vector δ^k satisfies a prescribed accuracy condition.

2.2. If β^{k+1} satisfies a prescribed stopping condition, terminate; otherwise update $\sigma^{k+1} = \rho_k \sigma^k$, $\tau^{k+1} = \rho_k \tau^k$ with $\rho_k \in (0, 1)$ and return to Step 2.1 with $k = k + 1$.

PPDNA: for the convex case of $q(\cdot) \equiv 0$

Algorithm. Let $\sigma^0, \sigma^1 > 0, \tau^0, \tau^1 > 0$ be given parameters. $\beta^0 \in \mathrm{dom}(p)$.

1. Compute

$$\beta^1 \approx \operatorname*{argmin}_{\beta \in \Re^n} \left\{ f(\beta) + p(\beta) + \frac{\sigma^0}{2}\|\beta - \beta^0\|^2 + \frac{\tau^0}{2}\|X\beta - b\|^2 \right\}$$

via solving its dual problem such that a prescribed stopping criterion is satisfied. Let $k = 1$ and go to Step 2.1.

2.1 Compute

$$\beta^{k+1} \approx \operatorname*{argmin}_{\beta \in \Re^n} \left\{ f(\beta) + p(\beta) + \frac{1}{2}\|\beta - \beta^k\|^2_{\sigma^k I + \tau^k X^* X} \right\}$$

via solving its dual problem satisfying a prescribed accuracy condition.

2.2. If β^{k+1} satisfies a prescribed stopping condition, terminate; otherwise update $\sigma^{k+1} = \rho_k \sigma^k$, $\tau^{k+1} = \rho_k \tau^k$ with $\rho_k \in (0, 1)$ and return to Step 2.1 with $k = k + 1$.

PPDNA: Rates of Convergence

For simplicity, assume that we take for some constant $c > 0$ that

$$\tau_k \equiv c\sigma_k \quad \forall k.$$

Then the k-th subproblem of PPDNA can be written as

$$\beta^{k+1} \approx \underset{\beta \in \Re^n}{\text{argmin}} \left\{ g_k(\beta) := f(\beta) + p(\beta) + \frac{\sigma_k}{2} \|\beta - \beta^k\|_M^2 \right\},$$

where

$$M := I + cX^*X \succ 0.$$

PPDNA: Stopping Criterion

The stopping criterion for inner subproblems

$$\text{(A)}\quad g_k(\beta_{k+1}) - \inf g_k \le \sigma_k \varepsilon_k^2/2, \quad \sum \varepsilon_k < \infty.$$

Theorem (Global convergence)

*Suppose that the solution set to (**P**) is nonempty. Then, $\{\beta^k\}$ is bounded and converges to an optimal solution β^* of (**P**).*

PPDNA: Fast linear local convergence

Assumption (Error bound)

For a maximal monotone operator $\mathcal{T}(\cdot)$ with $\mathcal{T}^{-1}(0) \neq \emptyset$, $\exists\, \varepsilon > 0$ and $a > 0$ s.t.

$$\forall \eta \in \mathcal{B}(0,\varepsilon) \quad \text{and} \quad \forall \xi \in \mathcal{T}^{-1}(\eta), \quad \mathrm{dist}_M(\xi, \mathcal{T}^{-1}(0)) \le a\|\eta\|_M,$$

where $\mathcal{B}(0,\varepsilon) = \{y \in \mathcal{Y} \mid \|y\| \le \varepsilon\}$. The constant a is called the error bound modulus associated with $\mathcal{T}$.

1. In many cases, $\mathcal{T}$ is a polyhedral multifunction [Robinson, 1981].
2. $\mathcal{T}_g$ (∂g) of LASSO, fused LASSO and elastic net regularized LS problems (piecewise linear-quadratic programming problems [J. Sun, PhD thesis, 1986] $+1$ ⇒ error bound).
3. $\mathcal{T}_g$ of ℓ_1 or elastic net regularized logistic regression [Luo and Tseng, 1992; Tseng and Yun, 2009].

PPDNA: Fast linear local convergence

Stopping criterion for the local convergence analysis

$$\text{(B)}\quad g_k(\beta^{k+1}) - \inf g_k$$
$$\leq \min\{1, (\sigma_k \delta_k^2/2)\}\|\beta^{k+1} - \beta^k\|_M^2, \quad \sum \delta_k < \infty.$$

Theorem

*Assume that the solution set Ω to (**P**) is nonempty. Assume that error bound condition holds for $\mathcal{T}_g$ with modulus l_g. Then, $\{\beta^k\}$ is convergent and, for all k sufficiently large,*

$$\text{dist}_M(\beta^{k+1}, \Omega) \leq \theta_k \text{dist}_M(\beta^k, \Omega),$$

where $\theta_k \approx \left(l_g(l_g^2 + \sigma_k^{-2})^{-1/2} + 2\delta_k\right) \to \theta_\infty = (l_g\sigma_\infty)/\sqrt{1 + (l_g\sigma_\infty)^2} < 1$ *as* $k \to \infty$.

Note that $\theta_\infty << 1$ when $l_g\sigma_\infty$ is close to zero. Thus, PPDNA can be treated as an approximate Newton's method!!! (arbitrary linear convergence rate, a name coined by M.J.D. Powell in 1969).

Summary: outer iterations and inner iterations

So far we have

1. Outer iterations (PPA): asymptotically superlinear (arbitrary rate of linear convergence)
2. Inner iterations (nonsmooth Newton): superlinear $+$ cheap

Essentially, we have a $"\mathrm{fast}^2"$ algorithm.

The mPPDNA for the nonconvex case: KL property

- Let $\eta > 0$ and Φ_η be a set of all concave functions $\phi : [0, \eta) \to \Re_+$ such that $\phi(0) = 0$, ϕ is continuous at 0 and continuously differentiable on $(0, \eta)$ and $\phi'(x) > 0$, for $\forall\ x \in (0, \eta)$.
- The function f is said to have the Kurdyka-Łojasiewicz (KL) property at $\bar{x}$ if there exists $\eta > 0$, a neighbourhood $\mathcal{U}$ of $\bar{x}$ and a concave function $\phi \in \Phi_\eta$ such that

  $$\phi'(f(x) - f(\bar{x}))\text{dist}(0, \partial f) \geq 1,\ \ \forall x \in \mathcal{U} \text{ and } f(\bar{x}) < f(x) < f(\bar{x}) + \eta,$$

  where $\text{dist}(x, C) := \min_{y \in C} \|y - x\|$ is the distance from a point x to a nonempty closed set C.

A function is said to have the KL property at $\bar{x}$ with an exponent α if the function ϕ in the definition of the KL property takes the form as $\phi(x) = \gamma x^{1-\alpha}$ with $\gamma > 0$ and $\alpha \in [0, 1)$.

Convergence analysis

Theorem

Suppose that the function $g(\cdot)$ is bounded below and that q is continuously differentiable near $\mathcal{B}^\infty$, the set of all cluster points of the sequence $\{\beta^k\}$ generated by mPPDNA. Then every cluster point in $\mathcal{B}^\infty$, if exists, is a d-stationary point of (3).

Theorem

Suppose that the function $g(\cdot)$ is bounded below and that q is continuously differentiable near $\mathcal{B}^\infty$, the set of all cluster points of the sequence $\{\beta^k\}$ generated by mPPDNA. If either one of the following two conditions holds,

(a) *$\mathcal{B}^\infty$ contains an isolated element;*

(b) *The sequence $\{\beta^k\}$ is bounded; for all $\beta \in \mathcal{B}^\infty$, $\nabla q(\cdot)$ is locally Lipschitz continuous near β; and the function g has the KL property at all $\beta \in \mathcal{B}^\infty$;*

then the whole sequence $\{\beta^k\}$ converges to a unique element of $\mathcal{B}^\infty$.

Convergence analysis

Theorem

Moreover, if the condition (b) is satisfied and $\{\beta^k\}$ *converges to* $\beta^\infty \in \mathcal{B}^\infty$*, the function* g *has the KL property at* β^∞ *with an exponent* $\alpha \in [0, 1)$*, then we have*

(i) *if* $\alpha = 0$*, then the sequence* $\{\beta^k\}$ *converges in a finite number of steps;*

(ii) *if* $\alpha \in (0, \frac{1}{2}]$*, then the sequence* $\{\beta^k\}$ *converges R-linearly, that is, for all* $k \geq 1$ *there exist* $\nu > 0$ *and* $\eta \in [0, 1)$ *such that* $\|\beta^k - \beta^\infty\| \leq \nu\eta^k$*;*

(iii) *if* $\alpha \in (\frac{1}{2}, 1)$*, then the sequence* $\{\beta^k\}$ *converges R-sublinearly, that is, for all* $k \geq 1$ *there exists* $\nu > 0$ *such that* $\|\beta^k - \beta^\infty\| \leq \nu k^{-\frac{1-\alpha}{2\alpha-1}}$*.*

Numerical experiments

- Our numerical experiments are implemented on a PC (Intel Core 2 Duo 2.6 GHz with 4 GB RAM).
- The parameter λ is defined by $\lambda = \lambda_c \Lambda$, $\Lambda = 1.1\Phi^{-1}(1 - 0.05/(2n))$ with Φ the cumulative normal distribution function.
- The number of nonzero elements of a vector is defined by the minimal k such that

$$\sum_{i=1}^{k} |\tilde{\beta}_i| \geq 0.9999\|\beta\|_1$$

where $\tilde{\beta}$ is obtained by sorting β such that $|\tilde{\beta}_1| \geq |\tilde{\beta}_2| \geq \ldots \geq |\tilde{\beta}_n|$

Stopping criteria for the convex square-root problems

The step 1 of the mPPNDA algorithm will be terminated if the relative KKT residual[1] satisfies

$$\eta_{kkt} := \frac{\left\| \beta - \mathrm{P}_{\lambda p}\left(\beta - \frac{X^*(X\beta - b)}{\|X\beta - b\|} \right) \right\|}{1 + \|\beta\| + \frac{\|X^*(X\beta - b)\|}{\|X\beta - b\|}} < 10^{-6}, \tag{6}$$

or the number of iterations reaches the maximum 200 while the ADMMs will be terminated if (6) is satisfied or the number of iterations reaches the maximum 5000.

[1] Whenever possible, try to avoid using the "fast convergence criteria" such as the relative distance of two consecutive iterates. Instead, try to design fast convergent algorithms independent of the "fast convergence criteria", which may only indicate that the employed algorithm is slow for an earlier termination.

Numerical results

Table: The performances of the Flare package and pADMM on synthetic datasets for the srLasso problem.

probname	λ_c	pobj		time	
m; n		Flare	pADMM	Flare	pADMM
exmp1 8000;800	1.0	3.8876+3	3.5799+3	11:26	12
	0.5	3.0501+3	1.9174+3	21:09	13
	0.1	1.0487+3	5.8738+2	28:42	16
exmp2 8000;800	1.0	2.2422+3	2.2419+3	14:09	19
	0.5	1.8050+3	1.2811+3	27:18	11
	0.1	5.6150+2	4.6013+2	27:37	09
exmp3 8000;400	1.0	2.4758+3	2.4569+3	10:05	07
	0.5	1.9819+3	1.9421+3	7:26	07
	0.1	1.4888+3	1.4438+3	7:14	05
exmp4 8000;4000	1.0	1.1210+4	1.1205+4	29:11	20:16
	0.5	1.0165+4	1.0165+4	1:43:48	21:48
	0.1	7.6846+3	3.4069+3	3:11:27	5:12

Numerical results

Table: The performances of the Flare package and pADMM on UCI datasets for the srLasso problem.

probname	λ_c	pobj		time	
m; n		Flare	pADMM	Flare	pADMM
abalone.scale.expanded7 4177;6435	1.0	–	2.3852+2	–	25:57
	0.5	–	2.0312+2	–	25:32
	0.1	–	1.5586+2	–	26:29
mpg.scale.expanded7 392;3432	1.0	2.3550+2	2.3544+2	1:00	04
	0.5	1.5856+2	1.5831+2	57	03
	0.1	7.8656+1	7.8616+1	1:06	03
space.ga.scale.expanded9 3107;5005	1.0	1.3113+1	1.3113+1	12:59	5:19
	0.5	2.2419+1	2.1607+1	9:01	2:00
	0.1	1.2950+1	1.1999+1	6:13	3:00

The "–" in the Table means that the Flare package fails to solve the problem due to being out of memory.

Numerical results

Table: The performances of ADMMs and PPDNA on synthetic datasets for the srLasso problem. In the table, "a"=PPDNA, "b"=pADMM, "c"=dADMM.

probname m; n	λ_c	nnz	η_{kkt}			η_G			pobj			time			testerror
			a	b	c	a	b	c	a	b	c	a	b	c	
exmp1 8000;800	0.127	499	9.4-7	9.9-7	9.9-7	1.9-8	3.1-7	1.7-8	6.6996+2	6.6996+2	6.6996+2	30	3:49	2:44	9.4085+0
exmp2 8000;800	0.081	627	9.8-7	9.9-7	9.9-7	2.6-9	1.8-7	7.2-9	4.1824+2	4.1824+2	4.1824+2	32	1:49	2:26	9.4807+0
exmp3 8000;400	0.124	298	7.4-7	9.1-7	9.9-7	4.8-9	1.1-7	3.7-9	1.4476+3	1.4476+3	1.4476+3	08	17	1:05	2.3420+2
exmp4 8000;4000	0.117	2845	7.7-7	9.9-7	9.5-7	8.6-10	5.2-7	5.3-7	3.6799+3	3.6799+3	3.6799+3	4:41	5:41	16:44	3.6759+2

Numerical results

Table: The performances of ADMMs and PPDNA on synthetic datasets for the srLasso problem. In the table, “a”=PPDNA, “b”=pADMM, “c”=dADMM.

probname m; n	λ_c	nnz	η_{kkt}	η_G	pobj	time	testerror
			$a \mid b \mid c$	$a \mid b \mid c$	$a \mid b \mid c$	$a \mid b \mid c$	
exmp1 8000;800	0.127	499	9.4-7 | 9.9-7 | 9.9-7	1.9-8 | 3.1-7 | 1.7-8	6.6996+2 | 6.6996+2 | 6.6996+2	30 | 3:49 | 2:44	9.4085+0
exmp2 8000;800	0.081	627	9.8-7 | 9.9-7 | 9.9-7	2.6-9 | 1.8-7 | 7.2-9	4.1824+2 | 4.1824+2 | 4.1824+2	32 | 1:49 | 2:26	9.4807+0
exmp3 8000;400	0.124	298	7.4-7 | 9.1-7 | 9.9-7	4.8-9 | 1.1-7 | 3.7-9	1.4476+3 | 1.4476+3 | 1.4476+3	08 | 17 | 1:05	2.3420+2
exmp4 8000;4000	0.117	2845	7.7-7 | 9.9-7 | 9.5-7	8.6-10 | 5.2-7 | 5.3-7	3.6799+3 | 3.6799+3 | 3.6799+3	4:41 | 5:41 | 16:44	3.6759+2

Numerical results

Table: The performances of ADMMs and PPDNA on UCI datasets for the srLasso problem. In the table, “a”=PPDNA, “b”=pADMM, “c”=dADMM.

probname m; n	λ_c	nnz	η_{kkt}	η_G	pobj	time
			$a \mid b \mid c$	$a \mid b \mid c$	$a \mid b \mid c$	$a \mid b \mid c$
E2006.test 3308;150358	0.019	1	5.7-7 | 3.2-7 | 9.7-7	2.5-7 | 3.8-8 | 8.1-9	2.1998+1 | 2.1998+1 | 2.1998+1	05 | 07 | 05
log1p.E2006.test 3308;1771946	0.260	201	7.9-7 | 1.2-4 | 1.2-3	2.8-6 | 3.0-3 | 3.9-5	2.1642+1 | 2.1713+1 | 2.1642+1	1:49 | 2:17:37 | 1:22:12
pyrim.scale.expanded5 74;201376	0.109	70	5.9-7 | 2.0-5 | 3.7-3	6.9-7 | 4.7-3 | 3.8-4	6.8301-1 | 6.9094-1 | 6.8308-1	18 | 20:31 | 12:10
abalone.scale.expanded7 4177;6435	0.004	82	9.6-7 | 9.9-7 | 8.8-7	1.0-9 | 3.0-7 | 6.4-9	1.3495+2 | 1.3495+2 | 1.3495+2	07 | 1:32 | 7:57
bodyfat.scale.expanded7 252;116280	0.012	51	7.4-7 | 1.1-6 | 9.9-7	2.2-8 | 6.3-5 | 3.5-9	8.5770-2 | 8.5834-2 | 8.5767-2	10 | 27:59 | 8:38

Stopping criteria for the nonconvex square-root problems

In our mPPDNA, the step 1 is used to generate an initial point for the step 2 and is stopped if $\eta_{kkt} < 10^{-4}$. The algorithms will be terminated if the relative KKT residual satisfies

$$\tilde{\eta}_{kkt} := \frac{\left\|\beta - \mathsf{P}_1(p-q)\left(\beta - \frac{X^*(X\beta-b)}{\|X\beta-b\|}\right)\right\|}{1 + \|\beta\| + \frac{\|X^*(X\beta-b)\|}{\|X\beta-b\|}\|} < 10^{-6}$$

Besides, the algorithms will also be stopped when they reach the pre-set maximum number of iterations (200 for the second step of mPPDNA and 5000 for ADMM).

Nonconvex regularizers: SCAD

For $\lambda > 0$, the SCAD regularization is defined by $r(\beta) = p(\beta) - q(\beta)$ with

$$p(\beta) = \lambda\|\beta\|_1$$

$$q(\beta) = \textstyle\sum_{i=1}^{n} \begin{cases} 0, & \text{if} \quad |\beta_i| < \lambda \\ \frac{(|\beta_i|-\lambda)^2}{2(a_s-1)}, & \text{if} \quad \lambda \leq |\beta_i| \leq a_s\lambda \\ \lambda|\beta_i| - \frac{a_s+1}{2}\lambda^2, & \text{if} \quad |x_i| > a_s\lambda \end{cases}$$

In our numerical experiments, we take $a_s = 3.7$.

Numerical results for the SCAD regularization

Table: The performances of ADMM and mPPDNA on synthetic datasets for the SCAD regularization. In the table, "a"=mPPDNA, "b"=ADMM.

probname m; n	λ_c	nnz	η_{kkt}	pobj	time	testerror
			$a \mid b$	$a \mid b$	$a \mid b$	
exmp1 8000;800	0.145	460	3.9-7 | 5.9-1	5.9368+2 | 5.9392+2	20 | 3:39	9.2406+0
exmp2 8000;800	0.087	616	5.9-7 | 1.0-1	4.0760+2 | 4.0777+2	28 | 3:33	9.3745+0
exmp3 8000;400	0.230	293	7.8-7 | 2.7-1	1.5486+3 | 1.5529+3	10 | 2:02	2.3629+2
exmp4 8000;4000	0.153	2554	5.4-7 | 6.8-1	3.1837+3 | 3.1940+3	4:21 | 16:41	3.4480+2

Numerical results for the SCAD regularization

Table: The performances of ADMM and mPPDNA on UCI datasets for the SCAD regularization. In the table, “a”=mPPDNA, “b”=ADMM.

probname m; n	λ_c	nnz	η_{kkt}	pobj	time
			a | b	a | b	a | b
E2006.test 3308;150358	0.071	1	2.2-8 | 9.0-7	2.2165+1 | 2.2165+1	08 | 12:51
log1p.E2006.test 3308;1771946	0.257	207	2.1-7 | 5.9-3	2.1613+1 | 2.1366+2	3:50 | 2:36:14
pyrim.scale.expanded5 74;201376	0.109	70	1.4-7 | 4.3-3	6.8301-1 | 7.2608-1	13 | 21:26
abalone.scale.expanded7 4177;6435	0.011	49	9.9-7 | 6.9-1	1.3292+2 | 1.3864+2	12 | 21:41
bodyfat.scale.expanded7 252;116280	0.201	2	3.9-8 | 7.6-2	9.4125-1 | 9.5136-1	06 | 25:51

Nonconvex regularizers: MCP

For two positive parameters $a_m > 2$ and λ, the MCP regularization can be defined as $r(\beta) = p(\beta) - q(\beta)$ with

$$p(\beta) = \lambda \|\beta\|_1$$

$$q(\beta) = \textstyle\sum_{i=1}^{n} \begin{cases} \frac{\beta_i^2}{a_m}, & \text{if } |\beta_i| \leq a_m \lambda, \\ 2\lambda|\beta_i| - a_m \lambda^2, & \text{if } |\beta_i| > a_m \lambda \end{cases}$$

The function $q(\cdot)$ is continuously differentiable with its derivative given by

$$\frac{\partial q(\beta)}{\partial \beta_i} = \begin{cases} \frac{2\beta_i}{a_m}, & \text{if } |\beta_i| \leq a_m \lambda, \\ 2\lambda \text{sign}(\beta_i), & \text{if } |\beta_i| > a_m \lambda \end{cases}$$

In our numerical experiments, we take $a_m = 3.7$.

Numerical results for the MCP regularization

Table: The performances of ADMM and mPPDNA on synthetic datasets for the MCP regularization. In the table, “a” =mPPDNA, “b” =ADMM.

probname m; n	λ_c	nnz	η_{kkt}		pobj		time		testerror
			a	b	a	b	a	b	
exmp1 8000;800	0.209	380	5.2-8	1.9-2	5.5695+2	5.6091+2	29	3:33	9.3483+0
exmp2 8000;800	0.151	535	2.7-7	1.3-1	4.5225+2	4.5414+2	38	3:30	9.4916+0
exmp3 8000;400	0.081	267	9.3-7	1.5-1	1.3590+3	1.3617+3	1:11	2:01	2.3613+2
exmp4 8000;4000	0.293	1821	9.4-7	6.9-2	4.1362+3	4.2741+3	5:33	16:37	3.8471+2

Numerical results for the MCP regularization

Table: The performances of ADMM and mPPDNA on UCI datasets for the MCP regularization. In the table, "a"=mPPDNA, "b"=ADMM.

probname m; n	λ_c	nnz	η_{kkt}		pobj		time	
			a	b	a	b	a	b
E2006.test 3308;150358	0.090	1	2.4-8	9.3-7	2.2077+1	2.2077+1	07	07
log1p.E2006.test 3308;1771946	0.261	187	8.2-7	2.2-3	2.1455+1	3.6500+1	4:09	2:20:46
pyrim.scale.expanded5 74;201376	0.221	43	9.9-7	7.0-3	1.1428+0	4.6112+0	18	19:36
abalone.scale.expanded7 4177;6435	0.012	55	7.1-7	1.6-5	1.3271+2	1.2693+2	09	21:32
bodyfat.scale.expanded7 252;116280	0.183	2	2.8-7	5.3-6	5.7347-1	5.8278-1	06	25:11

References

- X.D. Li, D.F. Sun, and K.-C. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems, SIAM J. on Optimization, 28 (2018), 433-458.
- Peipei Tang, Chengjing Wang, Defeng Sun, Kim-Chuan Toh, A sparse semismooth Newton based proximal majorization-minimization algorithm for nonconvex square-root-loss regression problems, arXiv:1903.11460, March 2019.
- Y. Gao and D. Sun, A Majorized Penalty Approach for Calibrating Rank Constrained Correlation Matrix Problems, Technical Report, Department of Mathematics, National University of Singapore, Singapore, revised May 2010.
- Y. Cui, J.S. Pang, and B. Sen, Composite difference-max programs for modern statistical estimation problems, SIAM J. on Optimization, 28 (2018), 3344-3374.
- B. Stucky, S. van der Geer, Sharp oracle inequalities for square root regularization, Journal of Machine Learning Research, 18 (2017), 1-29.