Sparse Semismooth Newton Methods and Big Data Composite Optimization

Defeng Sun

Department of Applied Mathematics, The Hong Kong Polytechnic University

August 17, 2018/Wuyishan

Based on joint works with: Kim-Chuan Toh, Houduo Qi, and many others

1. Let $\mathcal{X}, \mathcal{Y}$ be two finite-dimensional real Euclidean spaces
2. $F : \mathcal{X} \to \mathcal{Y}$ a locally Lipschitz continuous function.

Since F is almost everywhere differentiable [Rademacher, 1912], we can define

$$\partial_B F(x) := \left\{ \lim F'(x^k) : \; x^k \to x, \, x^k \in D_F \right\}.$$

Here D_F is the set of points where F is differentiable. Hence, Clarke's generalized Jacobian of F at x is given by

$$\partial F(x) = \operatorname{conv} \partial_B F(x).$$

Definition 1

Let $\mathcal{K} : \mathcal{X} \rightrightarrows \mathcal{L}(\mathcal{X}, \mathcal{Y})$ be a nonempty, compact valued and upper-semicontinous multifunction. We say that F is semismooth $x \in \mathcal{X}$ with respect to the multifunction $\mathcal{K}$ if (i) F is directionally differentiable at x; and (ii) for any $\Delta x \in \mathcal{X}$ and $V \in \mathcal{K}(x + \Delta x)$ with $\Delta x \to 0$,

$$F(x + \Delta x) - F(x) - V(\Delta x) = o(\|\Delta x\|). \tag{1}$$

Furthermore, if (1) is replaced by

$$F(x + \Delta x) - F(x) - V(\Delta x) = O(\|\Delta x\|^{1+\gamma}), \tag{2}$$

where $\gamma > 0$ is a constant, then F is said to be $\gamma\text{-order (strongly if } \gamma = 1) \text{ semismooth}$ at x with respect to $\mathcal{K}$. We say that F is a semismooth function with respect to $\mathcal{K}$ if it is semismooth everywhere in $\mathcal{X}$ with respect to $\mathcal{K}$.

Assume that $F(\bar{x}) = 0$.

Given $x^0 \in \mathcal{X}$. For $k = 0, 1, \ldots$

Main Step Choose an arbitrary $V_k \in \mathcal{K}(x^k)$. Solve

$$F(x^k) + V_k(x^{k+1} - x^k) = 0$$

Rates of Convergence: Assume that $\mathcal{K}(\bar{x})$ is nonsingular and that x^0 is sufficiently close to $\bar{x}$. If F is semismooth at $\bar{x}$, then

$$\|x^{k+1} - \bar{x}\| \;=\; \|V_k^{-1}[F(x^k) - F(\bar{x}) - V_k(x^k - \bar{x})]\| = o(\|x^k - \bar{x}\|).$$

It takes $o(\|x^k - \bar{x}\|^{1+\gamma})$ if F is γ-order semismooth at $\bar{x}$ [the directional differentiability of F is not needed in the above analysis]

1. The nonsmooth Newton approach is popular in the complementarity and variational inequalities (nonsmooth equations) community.
2. Kojima and Shindo (1986) investigated a piecewise smooth Newton's method.
3. Kummer (1988, 1992) gave a sufficient condition (1) to generalize Kojima and Shindo's work.
4. L. Qi and J. Sun (1993) proved what we know now.
5. Since then, many developments ...

Why nonsmooth Newton methods important in solving big data optimization?

Consider the nearest correlation matrix (NCM) problem:

$$\min\left\{\frac{1}{2}\|X-G\|_F^2 \mid X\succeq 0,\ X_{ii}=1,\ i=1,\ldots,n\right\}.$$

The dual of the above problem can be written as

$$\begin{array}{rl}\min & \frac{1}{2}\|\Xi\|^2-\langle b,\,y\rangle-\frac{1}{2}\|G\|^2\\ \text{s.t.} & S-\Xi+\mathcal{A}^*y=-G,\quad S\succeq 0\end{array}$$

or via eliminating Ξ and $S\succeq 0$, the following

$$\min\left\{\varphi(y):=\frac{1}{2}\|\Pi_{\succeq 0}(\mathcal{A}^*y+G)\|^2-\langle b,\,y\rangle-\frac{1}{2}\|G\|^2\right\}.$$

Test the second order nonsmooth Newton-CG method [H.-D. Qi & Sun 06] ([X,y] = CorrelationMatrix(G,b,tau,tol) in Matlab) and two popular first order methods (FOMs) [APG of Nesterov; ADMM of Glowinski (steplength 1.618)] all to the dual forms for the NCM with real financial data:

G: Cor3120, $n = 3,120$, obtained from [N. J. Higham & N. Strabić, SIMAX, 2016] [Optimal sol. rank $= 3,025$]

$n = 3,120$	SSNCG	ADMM	APG
Rel. KKT Res.	2.7-8	2.9-7	9.2-7
time (s)	26.8	246.4	459.1
iters	4	58	111
avg-time/iter	6.7	4.3	4.1

Newton's method only takes at most 40% time more than ADMM & APG per iteration. How is it possible?

We shall use simple vector cases to explain why:

(LASSO)

$$\min\left\{\frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1 \mid x\in\mathbb{R}^n\right\}$$

where $\lambda>0$, $A\in\mathbb{R}^{m\times n}$, and $b\in\mathbb{R}^m$.

(Fused LASSO)

$$\min\left\{\frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1+\lambda_2\|Bx\|_1\right\}$$

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

(Clustered LASSO)

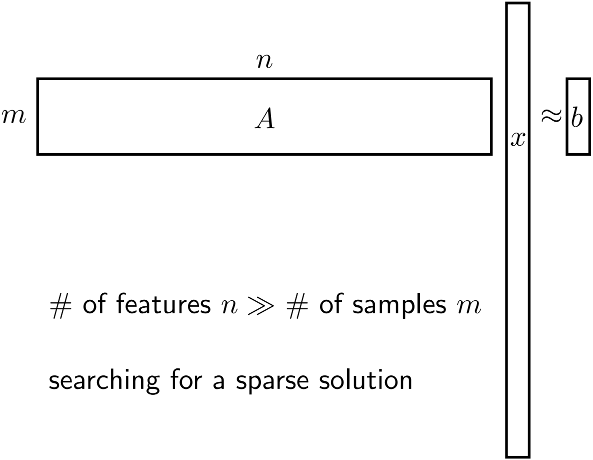
$$\min\left\{\frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1+\lambda_2\sum_{i=1}^{n}\sum_{j=i+1}^{n}|x_i-x_j|\right\}$$

(OSCAR)

$$\min\left\{\frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1+\lambda_2\sum_{i=1}^{n}\sum_{j=i+1}^{n}|x_i+x_j|+|x_i-x_j|\right\}$$

We are interested in n (number of features) large and/or m (number of samples) large

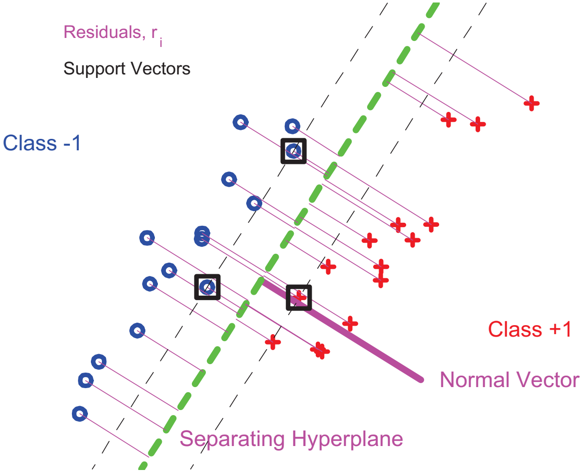
Sparse regression:



$\#$ of features $n \gg \#$ of samples m

searching for a sparse solution

Example: Support vector machine



Newton's method

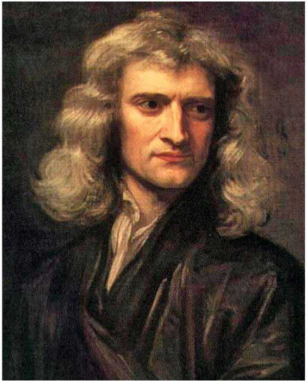


Figure: Sir Isaac Newton (Niu Dun) (4 January 1643 - 31 March 1727)

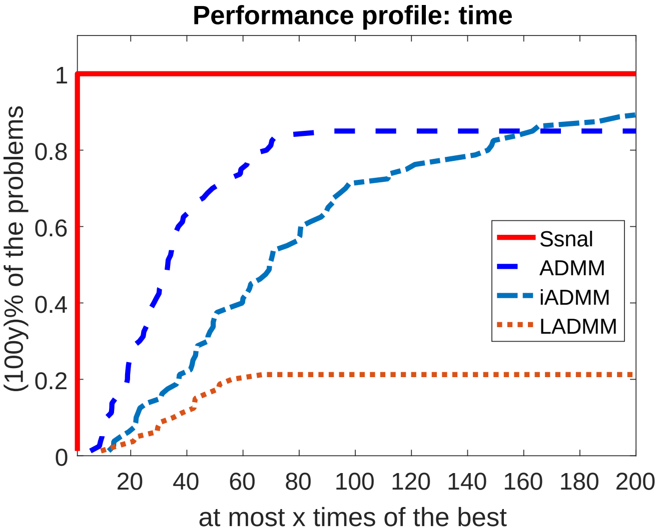
Which Newton's method?

(a) Snail (Niu)

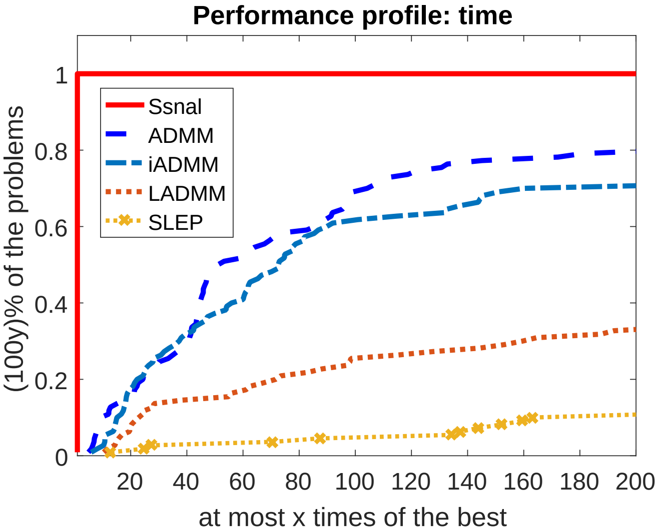
(b) Longhorn beetle (Niu)

(c) Charging Bull (Niu)

(d) Yak (Niu)



Performance profiles on biomedical datasets.



Performance profiles on UCI datasets.

Interior point methods

For the illustrative purpose, consider a simpler example

$$\min\left\{\frac{1}{2}\|Ax-b\|^2 \mid x \geq 0\right\}$$

and its dual

$$\max\left\{-\frac{1}{2}\|\xi\|^2 + \langle b,\, \xi\rangle \mid A^T\xi \leq 0\right\}$$

Interior-point based solver I: an $n \times n$ linear system

$$(D + A^T A)x = \mathrm{rhs}_1$$

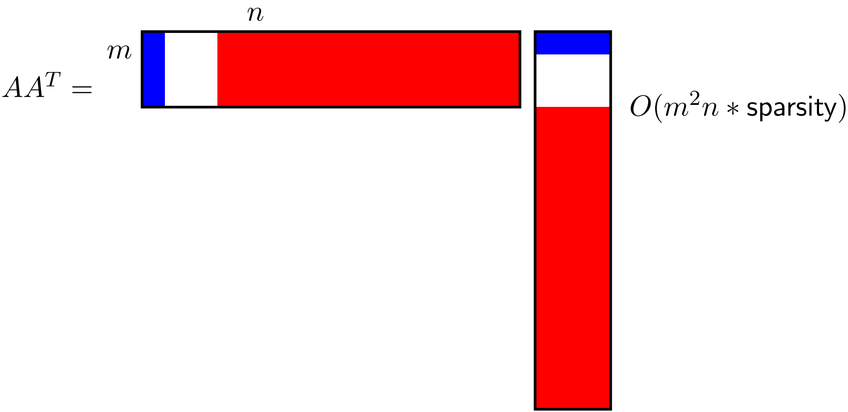
D: A Diagonal matrix with positive diagonal elements

Using PCG solver (e.g., matrix free interior point methods [K. Fountoulakis, J. Gondzio and P. Zhlobich, 2014])

Costly when n is large

Interior-point based solver II: an $m \times m$ linear system

$$(I_m + A{\color{blue}D^{-1}}A^T)\xi = \mathrm{rhs}_2$$



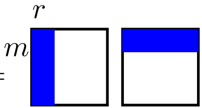
Our nonsmooth Newton's method

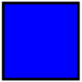
Our nonsmooth Newton's method: an $m \times m$ linear system

$$(I_m + APA^T)\xi = \mathrm{rhs}_2$$

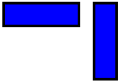
P: A Diagonal matrix with 0 or 1 diagonal elements

r: number of nonzero diagonal elements of P (second order sparsity)

$(AP)(AP)^T =$ 

$=$  $O(m^2 r *$ sparsity$)$

Sherman-Morrison-Woodbury formula:

$(AP)^T(AP) =$  $=$  $O(r^2 m *$ sparsity$)$

$$(\mathbf{P}) \quad \min\left\{f(x) := h(\mathcal{A}x) + p(x)\right\},$$

Real finite dimensional Hilbert spaces $\mathcal{X}$, $\mathcal{Y}$

Closed proper convex function $p : \mathcal{X} \to (-\infty, +\infty]$

Convex differentiable function $h : \mathcal{Y} \to \Re$

Linear map $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$

Dual problem

$$(\mathbf{D}) \quad \min\{h^*(\xi) + p^*(u) \mid \mathcal{A}^*\xi + u = 0\}$$

p^* and h^*: the Fenchel conjugate functions of p and h.

$p^*(z) = \sup\{\langle z,\, x\rangle - p(x)\}$.

Examples of smooth loss function h:

- Linear regression $h(y) = \|y - b\|^2$
- Logistic regression $h(y) = \log(1 + \exp(-yb))$
- many more ...

Examples of regularizer p:

- LASSO $p(x) = \|x\|_1$
- Fused LASSO $p(x) = \|x\|_1 + \sum_{i=1}^{n-1} |x_i - x_{i+1}|$
- Ridge $p(x) = \|x\|_2^2$
- Elastic net $p(x) = \|x\|_1 + \|x\|_2^2$
- Group LASSO
- Fused Group LASSO
- Clustered LASSO, OSCAR
- Ordered LASSO, etc

Assumptions on the loss function

Assumption 1 (Assumptions on h)

1. $h : \mathcal{Y} \rightarrow \Re$ *has a* $1/\alpha_h$*-Lipschitz continuous gradient:*

$$\|\nabla h(y_1) - \nabla h(y_2)\| \leq (1/\alpha_h)\|y_1 - y_2\|, \quad \forall y_1, y_2 \in \mathcal{Y}$$

2. h *is essentially locally strongly convex [Goebel and Rockafellar, 2008]: for any compact and convex set* $K \subset \operatorname{dom} \partial h$, $\exists\, \beta_K > 0$ *s.t.*

$$(1-\lambda)h(y_1)+\lambda h(y_2) \geq h((1-\lambda)y_1+\lambda y_2)+\frac{1}{2}\beta_K\lambda(1-\lambda)\|y_1-y_2\|^2$$

for all $\lambda \in [0,1]$, $y_1, y_2 \in K$

Under the assumptions on h, we know

a. h^*: strongly convex with constant α_h
b. h^*: essentially smooth[1]
c. ∇h^*: locally Lipschitz continuous on $\mathcal{D}_{h^*} := \text{int}\,(\text{dom}\, h^*)$
d. $\partial h^*(y) = \emptyset$ when $y \notin \mathcal{D}_{h^*}$.

Only need to focus on $\mathcal{D}_{h^*}$

[1] h^* is differentiable on $\text{int}\,(\text{dom}\, h^*) \neq \emptyset$ and $\lim_{i\to\infty} \|\nabla h^*(y_i)\| = +\infty$ whenever $\{y_i\} \subset \text{int}\,(\text{dom}\, h^*) \to y \in \text{bdry}(\text{int}\,(\text{dom}\, h^*))$.

The Lagrangian function for (**D**):

$$l(\xi, u; x) = h^*(\xi) + p^*(u) - \langle x, \mathcal{A}^*\xi + u\rangle, \quad \forall\, (\xi, u, x) \in \mathcal{Y} \times \mathcal{X} \times \mathcal{X}.$$

Given $\sigma > 0$, the augmented Lagrangian function for (**D**):

$$\mathcal{L}_\sigma(\xi, u; x) = l(\xi, u; x) + \frac{\sigma}{2}\|\mathcal{A}^*\xi + u\|^2, \quad \forall\, (\xi, u, x) \in \mathcal{Y} \times \mathcal{X} \times \mathcal{X}.$$

The proximal mapping $\mathrm{Prox}_p(x)$:

$$\mathrm{Prox}_p(x) = \arg\min_{u \in \mathcal{X}} \left\{ p(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

Assumption: $\mathrm{Prox}_{\sigma p}(x)$ is easy to compute given any x

Advantage of using (**D**): h^* is strongly convex; $\min_u\{\mathcal{L}_\sigma(\xi, u; x)\}$ is easy.

An inexact augmented Lagrangian method of multipliers.

Given $\sum \varepsilon_k < +\infty$, $\sigma_0 > 0$, choose $(\xi^0, u^0, x^0) \in \mathrm{int}(\mathrm{dom}\, h^*) \times \mathrm{dom}\, p^* \times \mathcal{X}$. For $k = 0, 1, \ldots$, iterate

Step 1. Compute

$$(\xi^{k+1}, u^{k+1}) \approx \arg\min\{\Psi_k(\xi, u) := \mathcal{L}_{\sigma_k}(\xi, u; x^k)\}.$$

To be solved via a nonsmooth Newton method.

Step 2. Compute $x^{k+1} = x^k - \sigma_k(\mathcal{A}^*\xi^{k+1} + u^{k+1})$ and update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$.

The stopping criterion for inner subproblem

$$\text{(A)} \quad \Psi_k(\xi^{k+1}, u^{k+1}) - \inf \Psi_k \leq \varepsilon_k^2 / 2\sigma_k, \quad \sum \varepsilon_k < \infty.$$

Theorem 2 (Global convergence)

*Suppose that the solution set to (**P**) is nonempty. Then, $\{x^k\}$ is bounded and converges to an optimal solution x^* of (**P**). In addition, $\{(\xi^k, u^k)\}$ is also bounded and converges to the unique optimal solution $(\xi^*, u^*) \in \mathrm{int}(\mathrm{dom}\, h^*) \times \mathrm{dom}\, p^*$ of (**D**).*

Assumption 2 (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.

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

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

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

Fast linear local convergence

Stopping criterion for the local convergence analysis

$$\text{(B)} \quad \Psi_k(\xi^{k+1}, u^{k+1}) - \inf \Psi_k$$
$$\leq \min\{1, (\delta_k^2/2\sigma_k)\}\|x^{k+1} - x^k\|^2, \quad \sum \delta_k < \infty.$$

Theorem 3

*Assume that the solution set Ω to (**P**) is nonempty. Suppose that Assumption 2 holds for $\mathcal{T}_f$ with modulus a_f. Then, $\{x^k\}$ is convergent and, for all k sufficiently large,*

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

where $\theta_k \approx \left(a_f(a_f^2 + \sigma_k^2)^{-1/2} + 2\delta_k\right) \to \theta_\infty = a_f/\sqrt{a_f^2 + \sigma_\infty^2} < 1$ *as $k \to \infty$. Moreover, the conclusions of Theorem 2 about $\{(\xi^k, y^k)\}$ are valid.*

ALM is an approximate Newton's method!!! (arbitrary linear convergence rate).

Fix $\sigma > 0$ and $\tilde{x}$, denote

$$\begin{aligned}\psi(\xi) &:= \inf_{u} \mathcal{L}_\sigma(\xi, u, \tilde{x}) \\ &= h^*(\xi) + p^*(\mathsf{Prox}_{p^*/\sigma}(\tilde{x}/\sigma - \mathcal{A}^*\xi)) + \frac{1}{2\sigma}\|\mathsf{Prox}_{\sigma p}(\tilde{x} - \sigma\mathcal{A}^*\xi)\|^2.\end{aligned}$$

$\psi(\cdot)$: strongly convex and continuously differentiable on $\mathcal{D}_{h^*}$ with

$$\nabla\psi(\xi) = \nabla h^*(\xi) - \mathcal{A}\,\mathsf{Prox}_{\sigma p}(\tilde{x} - \sigma\mathcal{A}^*\xi), \quad \forall \xi \in \mathcal{D}_{h^*}$$

Solving nonsmooth equation:

$$\nabla\psi(\xi) = 0, \quad \xi \in \mathcal{D}_{h^*}.$$

Denote for $\xi \in \mathcal{D}_{h^*}$:

$$\widehat{\partial}^2\psi(\xi) := \partial^2 h^*(\xi) + \sigma\mathcal{A}\partial\mathrm{Prox}_{\sigma p}(\tilde{x} - \sigma\mathcal{A}^*\xi)\mathcal{A}^*$$

$\partial^2 h^*(\xi)$: Clarke subdifferential of ∇h^* at ξ

$\partial\mathrm{Prox}_{\sigma p}(\tilde{x}-\sigma\mathcal{A}^*\xi)$: Clarke subdifferential of $\mathrm{Prox}_{\sigma p}(\cdot)$ at $\tilde{x}-\sigma\mathcal{A}^*\xi$

Lipschitz continuous mapping: ∇h^*, $\mathrm{Prox}_{\sigma p}(\cdot)$

From [Hiriart-Urruty et al., 1984],

$$\widehat{\partial}^2\psi(\xi)\,(d) = \partial^2\psi(\xi)\,(d), \quad \forall\, d \in \mathcal{Y}$$

$\partial^2\psi(\xi)$: the generalized Hessian of ψ at ξ. Define

$$V^0 := H^0 + \sigma\mathcal{A}U^0\mathcal{A}^*$$

with $H^0 \in \partial^2 h^*(\xi)$ and $U^0 \in \partial\mathrm{Prox}_{\sigma p}(\tilde{x} - \sigma\mathcal{A}^*\xi)$

$V^0 \succ 0$ and $V^0 \in \widehat{\partial}^2\psi(\xi)$

SSN($\xi^0, u^0, \tilde{x}, \sigma$). Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, and $\delta \in (0, 1)$. Choose $\xi^0 \in \mathcal{D}_{h^*}$. Iterate

Step 1. Find an approximate solution $d^j \in \mathcal{Y}$ to

$$V_j(d) = -\nabla\psi(\xi^j)$$

with $V_j \in \widehat{\partial}^2\psi(\xi^j)$ s.t.

$$\|V_j(d^j) + \nabla\psi(\xi^j)\| \le \min(\bar{\eta}, \|\nabla\psi(\xi^j)\|^{1+\tau}).$$

Step 2. (Line search) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\xi^j + \delta^m d^j \in \mathcal{D}_{h^*}$$

$$\psi(\xi^j + \delta^m d^j) \le \psi(\xi^j) + \mu\delta^m \langle \nabla\psi(\xi^j), d^j \rangle.$$

Step 3. Set $\xi^{j+1} = \xi^j + \alpha_j \, d^j$.

Theorem 4

Assume that $\nabla h^*(\cdot)$ *and* $\mathrm{Prox}_{\sigma p}(\cdot)$ *are strongly semismooth on* $\mathcal{D}_{h^*}$ *and* $\mathcal{X}$. *Then* $\{\xi^j\}$ *converges to the unique optimal solution* $\bar{\xi} \in \mathcal{D}_{h^*}$ *and*

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

Implementable stopping criteria: the stopping criteria (A) and (B) can be achieved via:

$$(\mathrm{A}') \quad \|\nabla \psi_k(\xi^{k+1})\| \leq \sqrt{\frac{\alpha_h}{\sigma_k}} \varepsilon_k$$

$$(\mathrm{B}') \quad \|\nabla \psi_k(\xi^{k+1})\| \leq \sqrt{\frac{\alpha_h}{\sigma_k}} \delta_k \min\{1, \sigma_k \|\mathcal{A}^* \xi^{k+1} + u^{k+1}\|\}$$

$(\mathrm{A}') \Rightarrow (\mathrm{A})$ & $(\mathrm{B}') \Rightarrow (\mathrm{B})$

Summary: outer iterations and inner iterations

So far we have

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

Essentially, we have a "fast + fast" algorithm.

LASSO: $\min\left\{\frac{1}{2}\|\mathcal{A}x - b\|^2 + \lambda_1\|x\|_1\right\}$

$h(y) = \frac{1}{2}\|y - b\|^2, \quad p(x) = \lambda_1\|x\|_1$

$\mathrm{Prox}_{\sigma p}(x)$: easy to compute $= \mathrm{sgn}(x) \circ \max\{|x| - \sigma\lambda_1, 0\}$

Newton System:

$$(\mathcal{I} + \sigma\mathcal{A}P\mathcal{A}^*)\xi = \text{rhs}$$

$P \in \partial\mathrm{Prox}_{\sigma p}(x^k - \sigma\mathcal{A}^*\xi)$: diagonal matrix with $0, 1$ entries. Most of these entries are 0 if the optimal solution x^{opt} is sparse.

Message: Nonsmooth Newton can fully exploit the second order sparsity (SOS) of solutions to solve the Newton system very efficiently!

Newton system for fused LASSO

Fused LASSO: $\min\left\{\frac{1}{2}\|\mathcal{A}x - b\|^2 + \lambda_1\|x\|_1 + \lambda_2\|\mathcal{B}x\|_1\right\}$

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

$h(y) = \frac{1}{2}\|y - b\|^2, \quad p(x) = \lambda_1\|x\|_1 + \lambda_2\|\mathcal{B}x\|_1$

Let $x_{\lambda_2}(v) := \arg\min_x \frac{1}{2}\|x - v\|^2 + \lambda_2\|\mathcal{B}x\|_1$.

Proximal mapping of p [Friedman et al., 2007]:

$$\mathrm{Prox}_p(v) = \mathrm{sign}(x_{\lambda_2}(v)) \circ \max(\mathrm{abs}(x_{\lambda_2}(v)) - \lambda_1, 0).$$

Efficient algorithms to obtain $x_{\lambda_2}(v)$: taut-string [Davies and Kovac, 2001], direct algorithm [Condat, 2013], dynamic programming [Johnson, 2013]

Dual approach to obtain $x_{\lambda_2}(v)$: denote

$$z(v) := \arg\min_z \left\{ \frac{1}{2}\|\mathcal{B}^* z\|^2 - \langle \mathcal{B}v,\, z\rangle \mid \|z\|_\infty \leq \lambda_2 \right\}$$

$\Rightarrow x(v) = v - \mathcal{B}^* z(v)$. Let $C = \{z \mid \|z\|_\infty \leq \lambda_2\}$, from optimality condition

$$z = \Pi_C(z - (\mathcal{B}\mathcal{B}^* z - \mathcal{B}v))$$

and the implicit function theorem $\Rightarrow$ Newton system for fused Lasso:

$(\mathcal{I} + \sigma \mathcal{A}\widehat{P}\mathcal{A}^*)\xi =$ rhs, `[P is Han-Sun Jacobian (JOTA, 1997)]`

$\widehat{P} = P(I - \mathcal{B}^*(I - \Sigma + \Sigma\mathcal{B}\mathcal{B}^*)^{-1}\Sigma\mathcal{B})$ (positive semidefinite)

$\Sigma \in \partial\Pi_C(z - (\mathcal{B}\mathcal{B}^* z - \mathcal{B}v))$

P, Σ: diagonal matrices with $0, 1$ entries. Most diagonal entries of P are 0 if x^{opt} is sparse. The red part is diagonal + low rank

Again, can use sparsity and the structure of the red part to solve the system efficiently

KKT residual:

$$\eta_{\rm KKT} := \frac{\|\tilde{x} - \text{Prox}_p[\tilde{x} - (\mathcal{A}\tilde{x} - b)]\|}{1 + \|\tilde{x}\| + \|\mathcal{A}\tilde{x} - b\|} \leq 10^{-6}.$$

Compare SSNAL with state-of-the-art solvers: mfIPM, ... [Fountoulakis et al., 2014] and APG [Liu et al. 2011]

$(\mathcal{A}, b)$ taken from 11 Sparco collections (all very easy problems) [Van Den Berg et al, 2009]

$\lambda = \lambda_c \|\mathcal{A}^* b\|_\infty$ with $\lambda_c = 10^{-3}$ and 10^{-4}

Add 60dB noise to b in MATLAB: `b = awgn(b,60,'measured')`

max. iteration number: 20,000 for APG
max. computation time: 7 hours

Numerical results for LASSO arising from compressed sensing

(a) our SSNAL
(b) mfIPM
(c) APG: Nesterov's accelerated proximal gradient method

$\lambda_c = 10^{-3}$		η_{KKT}	time (hh:mm:ss)
probname	$m;n$	a | b | c	a | b | c
srcsep1	29166;57344	1.6-7 | 7.3-7 | 8.7-7	5:44 | 42:34 | 1:56
soccer1	3200;4096	1.8-7 | 6.3-7 | 8.4-7	01 | 03 | 2:35
blurrycam	65536;65536	1.9-7 | 6.5-7 | 4.1-7	03 | 09 | 02
blurspike	16384;16384	3.1-7 | 9.5-7 | 9.9-7	03 | 05 | 03
$\lambda_c = 10^{-4}$			
srcsep1	29166;57344	9.8-7 | 9.5-7 | 9.9-7	9:28 | 3:31:08 | 2:50
soccer1	3200;4096	8.7-7 | 4.3-7 | 3.3-6	01 | 02 | 3:07
blurrycam	65536;65536	1.0-7 | 9.7-7 | 9.7-7	05 | 1:35 | 03
blurspike	16384;16384	3.5-7 | 7.4-7 | 9.8-7	10 | 08 | 05

Numerical results for LASSO arising from sparse regression

11 large scale instances $(\mathcal{A}, b)$ from LIBSVM [Chang and Lin, 2011]

$\mathcal{A}$: data normalized (with at most unit norm columns)

$\lambda_c = 10^{-3}$		$\eta_{\rm KKT}$	time (hh:mm:ss)
probname	$m; n$	a | b | c	a | b | c
E2006.train	16087; 150360	1.6-7 | 4.1-7 | 9.1-7	01 | 14 | 02
log1p.E2006.train	16087; 4272227	2.6-7 | 4.9-7 | 1.7-4	35 | 59:55 | 2:17:57
E2006.test	3308; 150358	1.6-7 | 1.3-7 | 3.9-7	01 | 08 | 01
log1p.E2006.test	3308; 4272226	1.4-7 | 9.2-8 | 1.6-2	27 | 30:45 | 1:29:25
pyrim5	74; 201376	2.5-7 | 4.2-7 | 3.6-3	05 | 9:03 | 8:25
triazines4	186; 635376	8.5-7 | 7.7-1 | 1.8-3	29 | 49:27 | 55:31
abalone7	4177; 6435	8.4-7 | 1.6-7 | 1.3-3	02 | 2:03 | 10:05
bodyfat7	252; 116280	1.2-8 | 5.2-7 | 1.4-2	02 | 1:41 | 12:49
housing7	506; 77520	8.8-7 | 6.6-7 | 4.1-4	03 | 6:26 | 17:00

For housing7, the computational costs in our SSNAL are as follows:

1. costs for Ax: 66 times, 0.11s in total;
2. costs for $A^T\xi$: 43 times, 2s in total;
3. costs for solving the inner linear systems: 43 times, 1.2s in total.

SSNAL has the ability to maintain the sparsity of x, the computational costs for calculating Ax are negligible comparing to other costs. In fact, each step of SSNAL is cheaper than many first order methods which need at least both Ax (x may be dense) and $A^T\xi$.

SOS is important for designing robust solvers!

SS-Newton equation can be solved very efficiently by exploiting the SOS property in solutions!

LassoNAL can generate solution path when λ varies

LassoNAL: start from $\lambda_{\max}$ to desired λ, each step $\lambda_{\text{new}} = 0.9\lambda_{\text{old}}$

$$\lambda_{\max} = \|\mathcal{A}^* b\|_\infty, \quad \lambda = 10^{-3}\lambda_{\max}$$

need to solve 66 lasso subproblems

Compare LassoNAL with SPAMS (SPArse Modeling Software by Julien Mairal et al.)

SPAMS: modified Lars or homotopy algorithm (solve problem via solution path)

Solution path of LassoNAL

(a) LassoNAL (one run with desired $\lambda = 10^{-3}\lambda_{\max}$)

(b) LassoNAL (solution path from $\lambda_{\max}$ to λ)

(c) SPAMS (solution path from $\lambda_{\max}$ to λ)

Randomly generated data

	time (ss)	λ NO.	ratio	nnz
$m; n$	a | b | c	b | c		
50; 1e4	0.4 | 3.5 | 1.5	66 | 75	8.75	46
50; 2e4	0.4 | 3.7 | 7.7	66 | 71	9.25	49
50; 3e4	0.4 | 5.1 | 17.4	66 | 71	12.75	46
50; 4e4	0.4 | 5.1 | 32.0	66 | 69	12.75	48
50; 5e4	0.5 | 8.4 | err	66 | err	16.30	49

SPAMS reports error when $n \geq 5 \times 10^4$

LassoNAL path: warm-start, $\mathrm{ratio} < 66$, for simple problems, running time almost independent with respect to n

Solution path of LassoNAL

(a) LassoNAL (one run with desired $\lambda = 10^{-3}\lambda_{\max}$)

(b) LassoNAL (solution path from $\lambda_{\max}$ to λ)

(c) SPAMS (solution path from $\lambda_{\max}$ to λ)

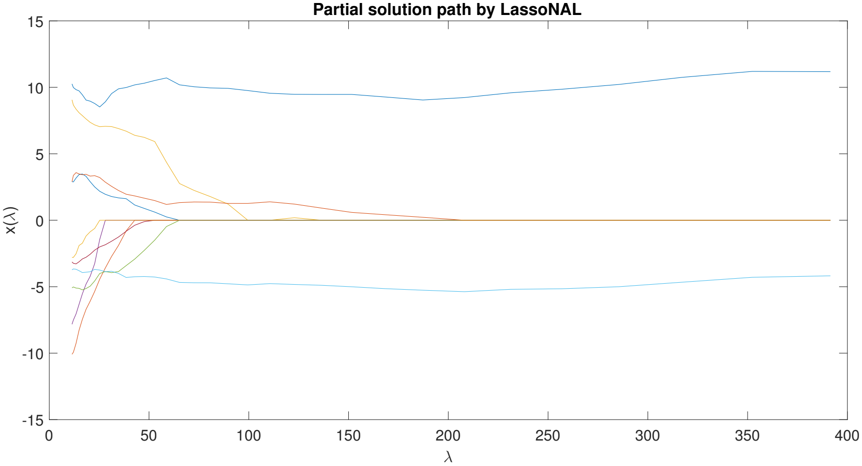
UCI data: truncated with $n \leq 4 \times 10^4$ (SPAMS reports error when n is large)

η: KKT residual

		time (ss)	λ NO.	ratio	nnz	η
Prob.	$m; n$	a | b | c	b | c		b| c	b| c
pyrim5	74; 4e4	0.4 | 10.9 | 37.9	66 | 1	27.25	56 | 0	2.5-7 | 9.9-1
triazines4	186; 4e4	1.4 | 33.9 | 38	66 | 1	24.21	136 | 0	4.4-7 | 9.9-1
housing7	506; 4e4	2.0 | 42.8 | 41.8	66 | 259	21.4	109 | 77	3.7-7 | 1.3-3

For difficult problems, SPAMS can not reach desired λ and may stop at $\lambda_{\max}$ (pyrim5 & triazines4)

Plot partial solution path for **housing7**, 10 largest nonzero elements in absolute values in the solution selected with $\lambda \in [10^{-3}\lambda_{\max}, 0.9^{33}\lambda_{\max}]$



Numerical results for fused LASSO

(a) our SSNAL
(b) APG based solver [Liu et al., 2011] (enhanced...)
(c1) ADMM (classical) (c2) ADMM (linearized)

Parameters: $\lambda_1 = \lambda_c \|\mathcal{A}^* y\|_\infty, \ \lambda_2 = 2\lambda_1, \ \text{tol} = 10^{-4}$

Problem: triazines 4, $m = 186$, $n = 635376$

Fused Lasso P.	iter	time (hh:mm:ss)
λ_c | nnz | η_C	a | b | c1 | c2	a | b | c1 | c2
10^{-1} ; 164; 2.4-2	10 | 6448 | 3461 | 8637	18 | 26:44 | 28:42 | 46:35
10^{-2} ; 1004; 1.7-2	13 | 11820 | 3841 | 19596	22 | 48:51 | 24:41 | 1:22:11
10^{-3} ; 1509; 1.2-3	16 | 20000 | 4532 | 20000	31 | 1:16:11 | 38:23 | 1:29:48
10^{-5} ; 2420; 6.4-5	24 | 20000 | 14384 | 20000	1:01 | 1:26:39 | 1:49:44 | 1:35:36

SSNAL is vastly superior to first-order methods: APG, ADMM (classical), ADMM (linearized)

ADMM (linearized) needs many more iterations than ADMM (classical)

When to choose SSNAL?

When Prox_p and its generalized (HS) Jacobian ∂Prox_p are easy to compute

Almost all of the LASSO models are suitable for SSNAL

When the problems are very easy, one may also consider APG or ADMM

Very complicated problems, in particular with many constraints, consider 2-phase approaches

Some remarks

1. For big optimization problems, our knowledge from the traditional optimization domain may be inadequate.

2. Belief: We do not know what we do not know. Always go to modern computers!

3. Big data optimization models provide many opportunities to test New and Old ideas. SOS is just one of them.

4. Many more need to be done such as the stochastic semismooth Newton methods (Andre Milzarek, Zaiwen Wen, ...), screening, sketching, parallelizing ...