Nonsmoothness can help:

sensitivity analysis and acceleration of proximal algorithms

Jérôme MALICK

CNRS, Laboratoire Jean Kuntzmann, Grenoble (France)

French-German-Swiss Conference on Optimization

FGS'2019 – September 2019 – Nice

Nonsmoothness: curse and blessing

Convex optimization

$$\min_{x\in\mathbb{R}^d} f(x) \qquad f\colon \mathbb{R}^d \to \mathbb{R} \text{ not differentiable everywhere} \quad \text{(though a.e.)}$$

Nonsmoothness is known to be a major difficulty for optimization ☹

Implicit nonsmoothness (e.g. robust/stoch. optim., Lagrangian/Benders decompositions,...)

$$f(x) = \sup_{u\in U} h(u,x) \qquad \text{with } h(u,\cdot) \text{ convex and } U \text{ arbitrary}$$

Nonsmoothness: curse and blessing

Convex optimization

$$\min_{x \in \mathbb{R}^d} \; f(x) \qquad f : \mathbb{R}^d \to \mathbb{R} \;\; \text{not differentiable everywhere} \;\; \text{(though a.e.)}$$

Nonsmoothness is known to be a major difficulty for optimization ☹

Implicit nonsmoothness (e.g. robust/stoch. optim., Lagrangian/Benders decompositions,...)

$$f(x) = \sup_{u \in U} h(u,x) \qquad \text{with} \;\; h(u,\cdot) \text{ convex and } U \text{ arbitrary}$$

In this talk: Nonsmoothness is sometimes a desirable property ☺

Chosen nonsmoothness (e.g. image processing, machine learning,...)

$$f(x) = F(x) + R(x) \qquad \text{with } F \text{ smooth and } R \text{ nonsmooth}$$

Nonsmoothness brings strong structure to optimization problems...

...offers extra-properties and can help in practice !

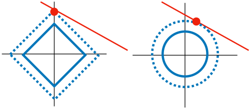
Example: ℓ_1-regularized least-squares & recovery

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Example: ℓ_1-regularized least-squares & recovery

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Nonsmoothness of $\|\cdot\|_1$
promotes sparse solutions
(many zero entries)

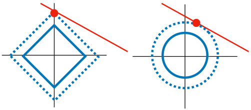


(opt. cond. ℓ_1 vs ℓ_2)

Example: ℓ_1-regularized least-squares & recovery

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2}\|Ax - y\|^2 \; + \; \lambda \|x\|_1 \qquad \text{(LASSO)}$$

Nonsmoothness of $\|\cdot\|_1$
promotes sparse solutions
(many zero entries)



(opt. cond. ℓ_1 vs ℓ_2)

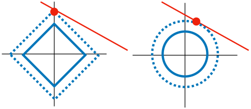
Recovery: compressed sensing

- Noisy observation $y = Ax_0 + w \in \mathbb{R}^n$ of a sparse $x_0 \in \mathbb{R}^d$
- Choosing ℓ_1-norm allows to recover x_0 and the support of x_0...
- ...when the problem is well-conditioned
  E.g. A gaussian + enough observations [Candès *et al* '05] [Dossal *et al* '11]
  model recovery when $P = \Omega(\|x_0\|_0 \log N)$
- A lot of research on recovery e.g. [Fuchs '04] [Grasmair '10] [Vaiter '14]...

Example: ℓ_1-regularized least-squares & recovery

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Nonsmoothness of $\|\cdot\|_1$
promotes sparse solutions
(many zero entries)



(opt. cond. ℓ_1 vs ℓ_2)

Recovery: compressed sensing

- Noisy observation $y = Ax_0 + w \in \mathbb{R}^n$ of a sparse $x_0 \in \mathbb{R}^d$
- Choosing ℓ_1-norm allows to recover x_0 and the support of x_0...
- ...when the problem is well-conditioned
  E.g. A gaussian + enough observations [Candès *et al* '05] [Dossal *et al* '11]
  model recovery when $P = \Omega(\|x_0\|_0 \log N)$
- A lot of research on recovery e.g. [Fuchs '04] [Grasmair '10] [Vaiter '14]...

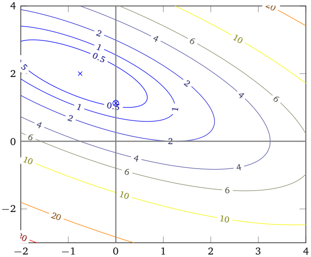
Nonsmoothness reveals underlying structure

Example: ℓ_1-regularized least-squares & stability

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2}\|Ax - y\|^2 \; + \; \lambda \|x\|_1 \qquad \text{(LASSO)}$$

Stability: the support of optimal solutions is stable under small perturbations

Illustration (on an instance with $d = 2$)

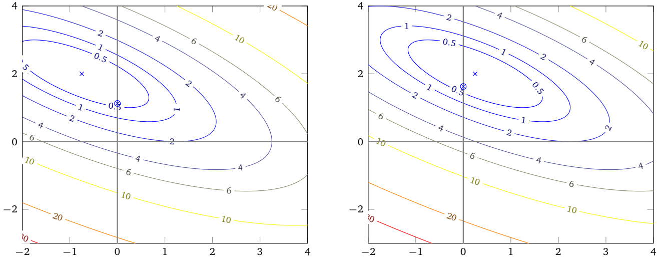


Example: ℓ_1-regularized least-squares & stability

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax - y\|^2 \; + \; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Stability: the support of optimal solutions is stable under small perturbations

Illustration (on an instance with $d = 2$)

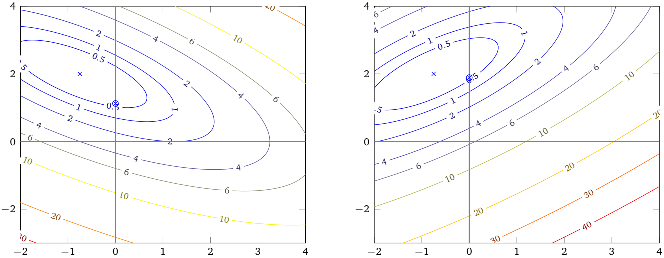


Example: ℓ_1-regularized least-squares & stability

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Stability: the support of optimal solutions is stable under small perturbations

Illustration (on an instance with $d=2$)

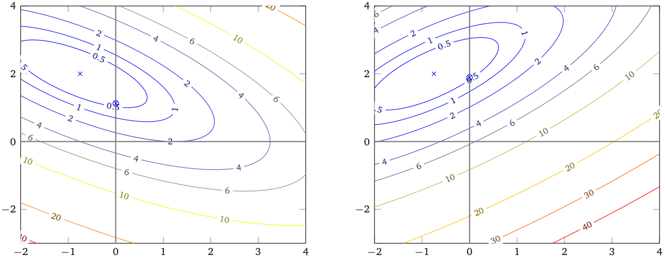


Example: ℓ_1-regularized least-squares & stability

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2}\|Ax - y\|^2 \; + \; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Stability: the support of optimal solutions is stable under small perturbations

Illustration (on an instance with $d = 2$)



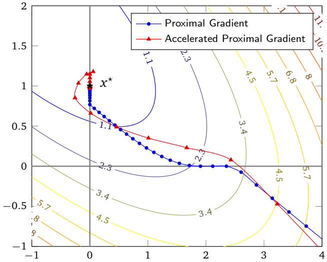
Nonsmoothness traps solutions in low-dimensional manifolds

Example: ℓ_1-regularized least-squares & identification

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Identification: (proximal-gradient) algorithms produce iterates...

...that eventually have the same support as the optimal solution



Runs of two proximal-gradient algos

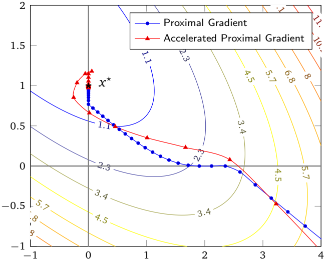
(same instance with $d = 2$)

Example: ℓ_1-regularized least-squares & identification

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \;+\; \lambda\|x\|_1 \qquad \text{(LASSO)}$$

Identification: (proximal-gradient) algorithms produce iterates...

...that eventually have the same support as the optimal solution



Runs of two proximal-gradient algos

(same instance with $d = 2$)

Nonsmoothness attracts (proximal) algorithms

Nonsmoothness can help...

To sum up on ℓ_1-regularized least-squares

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2}\|Ax - y\|^2 \; + \; \lambda \|x\|_1$$

$$\text{Nonsmoothness} \begin{cases} \text{reveals underlying structure} \quad \text{(recovery)} \\ \text{traps solutions in low-dimensional manifolds} \quad \text{(stability)} \\ \text{attracts (proximal) algorithms} \quad \text{(identification)} \end{cases}$$

Nonsmoothness can help...

To sum up on ℓ_1-regularized least-squares

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|Ax-y\|^2 \; + \; \lambda\|x\|_1$$

Nonsmoothness $\left\{ \begin{array}{l} \text{reveals underlying structure (recovery)} \\ \text{traps solutions in low-dimensional manifolds (stability)} \\ \text{attracts (proximal) algorithms (identification)} \end{array} \right.$

Beyong ℓ_1-norm: F smooth and many R nonsmooth

$$\min_{x\in\mathbb{R}^d} \quad F(x) \; + \; R(x)$$

In this talk

- Illustrate stability and identification
- 2 applications in machine learning
  - practical application: communication-efficient distributed proximal-gradient
  - theoretical application: model consistency for regularized least-squares
- High level: ideas on recent research (but skip details/maths + missing refs)

Outline

1. Introduction: nonsmoothness provides recovery, stability, identification
2. Stability of mirror-stratifiable regularizers
3. Identification of proximal algorithms
4. Application: communication-efficient distributed learning
5. Application: model consistency in supervised learning

Outline

Stability or sensitivity analysis

Nonsmoothness traps solutions in low-dimensional manifolds

Parameterized composite optimization problem (smooth + nonsmooth)

$$\min_{x\in\mathbb{R}^d} \quad F(x,p) + R(x),$$

Stability: Optimal solutions lie on a manifold: $x^\star(p) \in M$ for $p \sim p_0$

See [Lewis '02] sensitivity analysis of partly-smooth functions

Used/studied in e.g. [Hare Lewis '10] [Vaiter *et al* '15] [Liang *et al* '16]...

Example 1: $R = \|\cdot\|_1$, $\operatorname{supp}\big(x^\star(p)\big) = \operatorname{supp}\big(x^\star(p_0)\big)$

Stability or sensitivity analysis

Nonsmoothness traps solutions in low-dimensional manifolds

Parameterized composite optimization problem (smooth + nonsmooth)

$$\min_{x \in \mathbb{R}^d} \quad F(x,p) + R(x),$$

Stability: Optimal solutions lie on a manifold: $x^\star(p) \in M$ for $p \sim p_0$

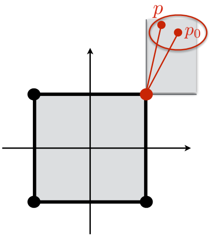
See [Lewis '02] sensitivity analysis of partly-smooth functions

Used/studied in e.g. [Hare Lewis '10] [Vaiter *et al* '15] [Liang *et al* '16]...

Example 1: $R = \|\cdot\|_1$, $\mathrm{supp}\big(x^\star(p)\big) = \mathrm{supp}\big(x^\star(p_0)\big)$

Example 2: $R = \iota_{\mathbb{B}_\infty}$ (indicator function)

projection onto the ℓ_∞ ball



Stability holds for many nonsmooth R...

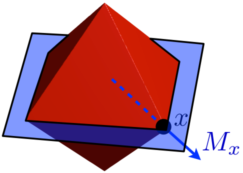
... let's exploit their strong structure !

Strong structure of nonsmooth regularizers

Many of the regularizers used in machine learning or image processing have a strong primal-dual structure – mirror-stratifiable [Fadili, M., Peyré '17]

Examples: (associated unit ball and low-dimensional manifold where x belongs)

- $R = \|\cdot\|_1$ (and $\|\cdot\|_\infty$ or other polyedral gauges)



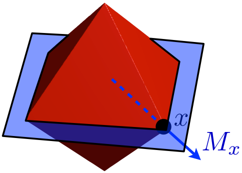
$M_x = \{z : \mathrm{supp}(z) = \mathrm{supp}(x)\}$

Strong structure of nonsmooth regularizers

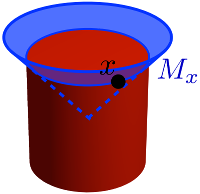
Many of the regularizers used in machine learning or image processing have a strong primal-dual structure – mirror-stratifiable [Fadili, M., Peyré '17]

Examples: (associated unit ball and low-dimensional manifold where x belongs)

- $R = \|\cdot\|_1$ (and $\|\cdot\|_\infty$ or other polyedral gauges)
- nuclear norm (aka trace-norm) $R(X) = \sum_i |\sigma_i(X)| = \|\sigma(X)\|_1$



$M_x = \{z : \mathrm{supp}(z) = \mathrm{supp}(x)\}$



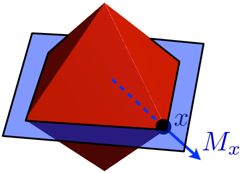
$M_x = \{z : \mathrm{rank}(z) = \mathrm{rank}(x)\}$

Strong structure of nonsmooth regularizers

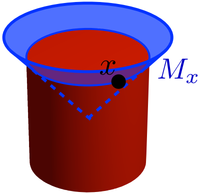
Many of the regularizers used in machine learning or image processing have a strong primal-dual structure – mirror-stratifiable [Fadili, M., Peyré '17]

Examples: (associated unit ball and low-dimensional manifold where x belongs)

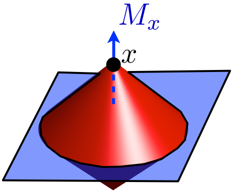
- $R = \|\cdot\|_1$ (and $\|\cdot\|_\infty$ or other polyedral gauges)
- nuclear norm (aka trace-norm) $R(X) = \sum_i |\sigma_i(X)| = \|\sigma(X)\|_1$
- group-ℓ_1 $R(x) = \sum_{b\in\mathcal{B}} \|x_b\|_2$ (e.g. $R(x) = \|x_{1,2}\| + |x_3|$)



$M_x = \{z : \mathrm{supp}(z) = \mathrm{supp}(x)\}$



$M_x = \{z : \mathrm{rank}(z) = \mathrm{rank}(x)\}$



$M_x = \{0\} \times \{0\} \times \mathbb{R}$

Recall on stratifications

A stratification of a set $D \subset \mathbb{R}^d$ is a (finite) partition $\mathcal{M} = \{M_i\}_{i \in I}$

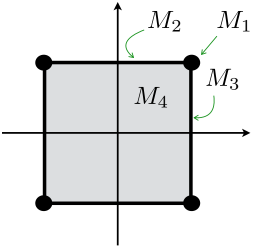
$$D = \bigcup_{i \in I} M_i$$

with so-called "strata" (e.g. smooth/affine manifolds) which fit nicely:

$$M \cap \mathrm{cl}(M') \neq \emptyset \quad \Longrightarrow \quad M \subset \mathrm{cl}(M')$$

Example: $\mathbb{B}_\infty$ the unit ℓ_∞-ball in $\mathbb{R}^2$

a stratification with 9 (affine) strata



Recall on stratifications

A stratification of a set $D \subset \mathbb{R}^d$ is a (finite) partition $\mathcal{M} = \{M_i\}_{i \in I}$

$$D = \bigcup_{i \in I} M_i$$

with so-called "strata" (e.g. smooth/affine manifolds) which fit nicely:

$$M \cap \mathrm{cl}(M') \neq \emptyset \quad \Longrightarrow \quad M \subset \mathrm{cl}(M')$$

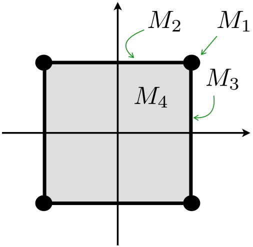
This relation induces a (partial) ordering $M \leqslant M'$

Example: $\mathbb{B}_\infty$ the unit ℓ_∞-ball in $\mathbb{R}^2$

a stratification with 9 (affine) strata

$$M_1 \leqslant M_2 \leqslant M_4$$

$$M_1 \leqslant M_3 \leqslant M_4$$



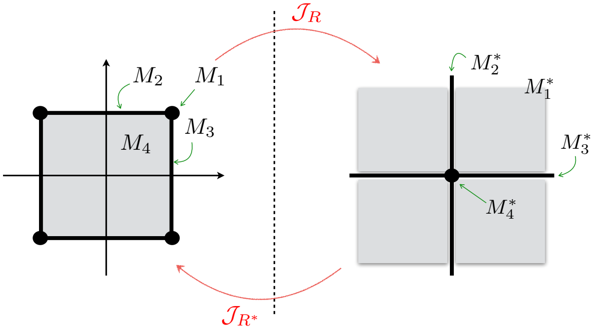
Mirror-stratifiable function

(primal) stratification $\mathcal{M} = \{M_i\}_{i\in I}$ and (dual) stratification $\mathcal{M}^* = \{M_i^*\}_{i\in I}$

in one-to-one decreasing correspondence

through the transfert operator $\mathcal{J}_R(S) = \bigcup_{x\in S} \mathrm{ri}(\partial R(x))$

Simple example: $R = \iota_{\mathbb{B}_\infty}$ $R^* = \|\cdot\|_1$



$\mathcal{J}_R(M_i) = \bigcup_{x\in M_i} \mathrm{ri}\, \partial R(x) = \mathrm{ri}\, N_{\mathbb{B}_\infty}(x) = M_i^*$ $M_i = \mathrm{ri}\, \partial \|x\|_1 = \bigcup_{x\in M_i^*} \mathrm{ri}\, \partial R^*(x) = \mathcal{J}_{R^*}(M_i^*)$

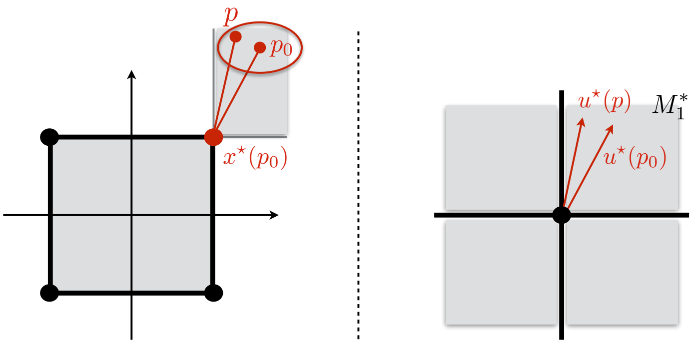
Enlarged stability illustrated

Simple problem

$$\begin{cases} \min & \frac{1}{2}\|x-p\|^2 \\ & \|x\|_\infty \leqslant 1 \end{cases} \qquad \begin{cases} \min & \frac{1}{2}\|u-p\|^2 + \|u\|_1 \\ & u \in \mathbb{R}^n \end{cases}$$

Non-degenerate case: $u^\star(p_0) = p_0 - x^\star(p_0) \;\in\; \mathrm{ri}\; N_{\mathbb{B}_\infty}(x^\star(p_0))$

$$\Longrightarrow \; M_1 = M_{x^\star(p_0)} = M_{x^\star(p)}$$ (in this case $x^\star(p) = x^\star(p_0)$)

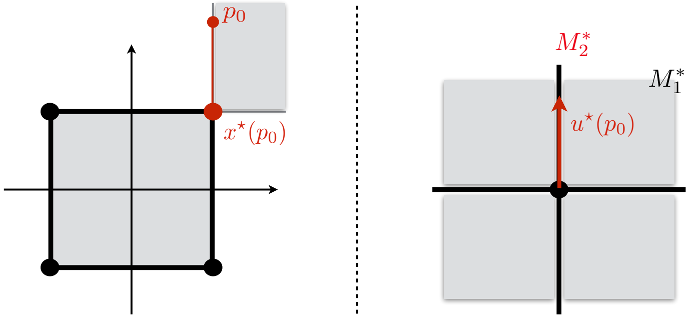


Enlarged stability illustrated

Simple problem

$$\begin{cases} \min & \frac{1}{2}\|x-p\|^2 \\ & \|x\|_\infty \leqslant 1 \end{cases} \qquad\qquad \begin{cases} \min & \frac{1}{2}\|u-p\|^2 + \|u\|_1 \\ & u \in \mathbb{R}^n \end{cases}$$

General case: $u^\star(p_0) = p_0 - x^\star(p_0) \;\in\; \cancel{ri}\, N_{\mathbb{B}_\infty}(x^\star(p))$



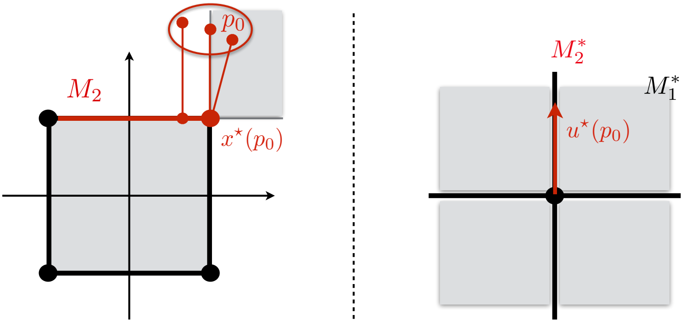
Enlarged stability illustrated

Simple problem

$$\begin{cases} \min & \frac{1}{2}\|x-p\|^2 \\ & \|x\|_\infty \leqslant 1 \end{cases} \qquad\qquad \begin{cases} \min & \frac{1}{2}\|u-p\|^2 + \|u\|_1 \\ & u \in \mathbb{R}^n \end{cases}$$

General case: $u^\star(p_0) = p_0 - x^\star(p_0) \;\in\; \cancel{\mathrm{ri}}\, N_{\mathbb{B}_\infty}(x^\star(p))$

$$\Longrightarrow\; M_1 = M_{x^\star(p_0)} \leqslant M_{x^\star(p)} \leqslant \mathcal{J}_{R^*}(M^*_{u^\star(p_0)}) = M_2$$



Enlarged sensitivity result

Theorem (Fadili, M., Peyré '17)

For the composite optimization problem (smooth + nonsmooth)

$$\min_{x\in\mathbb{R}^d} \quad F(x,p) + R(x),$$

satisfying mild assumptions (unique minimizer $x^\star(p_0)$ at p_0 and objective uniformly level-bounded in x)*, if R is mirror-stratifiable, then for $p \sim p_0$,*

$$M_{x^\star(p_0)} \leqslant M_{x^\star(p)} \leqslant \mathcal{J}_{R^*}(M^*_{u^\star(p_0)})$$

If $R = \|\cdot\|_1$, then $\quad \mathrm{supp}(x^\star(p_0)) \subseteq \mathrm{supp}(x^\star(p)) \subseteq \{i : |u^\star(p_0)_i| = 1\}$

Enlarged sensitivity result

Theorem (Fadili, M., Peyré '17)

For the composite optimization problem (smooth + nonsmooth)

$$\min_{x\in\mathbb{R}^d} \quad F(x,p)+R(x),$$

satisfying mild assumptions (unique minimizer $x^\star(p_0)$ at p_0 and objective uniformly level-bounded in x)*, if R is mirror-stratifiable, then for $p\sim p_0$,*

$$M_{x^\star(p_0)} \leqslant M_{x^\star(p)} \leqslant \mathcal{J}_{R^*}(M^*_{u^\star(p_0)})$$

If $R=\|\cdot\|_1$, then $\quad \mathrm{supp}(x^\star(p_0)) \subseteq \mathrm{supp}(x^\star(p)) \subseteq \{i : |u^\star(p_0)_i| = 1\}$

Remark: Optimality conditions for a primal-dual solution $(x^\star(p), u^\star(p))$

$$u^\star(p) = -\nabla F(x^\star(p),p) \;\in\; \partial R(x^\star(p))$$

In the non-degenerate case: $\quad u^\star(p_0) \in \mathrm{ri}\left(\partial R(x^\star(p_0))\right)$

$$M_{x^\star(p_0)} = M_{x^\star(p)} \quad \left(= \mathcal{J}_{R^*}(M^*_{u^\star(p_0)})\right)$$

we retrieve exactly the active strata ([Lewis '02] for partly-smooth functions)

Enlarged sensitivity result

Theorem (Fadili, M., Peyré '17)

For the composite optimization problem (smooth + nonsmooth)

$$\min_{x\in\mathbb{R}^d} \quad F(x,p) + R(x),$$

satisfying mild assumptions (unique minimizer $x^\star(p_0)$ at p_0 and objective uniformly level-bounded in x), *if R is mirror-stratifiable, then for $p \sim p_0$,*

$$M_{x^\star(p_0)} \leqslant M_{x^\star(p)} \leqslant \mathcal{J}_{R^*}(M^*_{u^\star(p_0)})$$

If $R = \|\cdot\|_1$, then $\quad \operatorname{supp}(x^\star(p_0)) \subseteq \operatorname{supp}(x^\star(p)) \subseteq \{i : |u^\star(p_0)_i| = 1\}$

Remark: Optimality conditions for a primal-dual solution $(x^\star(p), u^\star(p))$

$$u^\star(p) = -\nabla F(x^\star(p), p) \ \in \ \partial R(x^\star(p))$$

In the non-degenerate case: $\quad u^\star(p_0) \in \operatorname{ri}\left(\partial R(x^\star(p_0))\right)$

$$M_{x^\star(p_0)} = M_{x^\star(p)} \quad \left(= \mathcal{J}_{R^*}(M^*_{u^\star(p_0)})\right)$$

we retrieve exactly the active strata ([Lewis '02] for partly-smooth functions)

$$\boxed{\text{Nonsmoothness traps solutions in low-dimensional manifolds}}$$

Outline

Activity identification

Nonsmoothness attracts (proximal) algorithms

Composite optimization problem (smooth + nonsmooth)

$$\min_{x \in \mathbb{R}^d} \quad F(x) + R(x)$$

Proximal-gradient algorithm (aka forward-backward algorithm)

$$x_{k+1} = \operatorname{prox}_{\gamma R}\big(x_k - \gamma \nabla F(x_k)\big)$$

$$\operatorname{prox}_{\gamma R}(x) \;=\; \operatorname*{argmin}_{y} \; R(y) + \frac{1}{2\gamma}\|y - x\|^2$$

Identification: beyond convergence

after a finite moment of time K, all iterates x_k $(k \geqslant K)$ lie in an active set M

Well-studied, [Bertsekas '76], [Wright '96], [Lewis Drusvyatskiy '13]...

Enlarged activity identification

Theorem (Fadili, M., Peyré '17)

Under convergence assumptions, if R is mirror-stratifiable, then for $k \geqslant K$

$$M_{x^\star} \leqslant M_{x_k} \leqslant \mathcal{J}_{R^*}(M^*_{-\nabla F(x^\star)})$$

- Optimality condition $\quad -\nabla F(x^\star) \in \partial R(x^\star)$

  In the non-degenerate case: $\quad -\nabla F(x^\star) \in \mathrm{ri}\,\big(\partial R(x^\star)\big)$

  we have exact identification $M_{x^\star} = M_{x_k} \; \big(= \mathcal{J}_{R^*}(M^*_{-\nabla F(x^\star)})\big)$ [Liang *et al* 15]

Enlarged activity identification

Theorem (Fadili, M., Peyré '17)

Under convergence assumptions, if R is mirror-stratifiable, then for $k \geqslant K$

$$M_{x^\star} \leqslant M_{x_k} \leqslant \mathcal{J}_{R^*}(M^*_{-\nabla F(x^\star)})$$

- Optimality condition $\quad -\nabla F(x^\star) \in \partial R(x^\star)$

  In the non-degenerate case: $\quad -\nabla F(x^\star) \in \mathrm{ri}\,(\partial R(x^\star))$

  we have exact identification $M_{x^\star} = M_{x_k} \ \left(= \mathcal{J}_{R^*}(M^*_{-\nabla F(x^\star)})\right)$ [Liang *et al* 15]

- In the general case: δ quantifies the degeneracy of the problem

$$\delta = \dim(\mathcal{J}_{R^*}(M^*_{-\nabla F(x^\star)})) - \dim(M_{x^\star})$$

  $\delta = 0$: weak degeneracy (fast convergence and identification)

  δ large : strong degeneracy (slow convergence and identification)

- Note: δ and K are not computable beforehand in general...

Illustration with nuclear norm

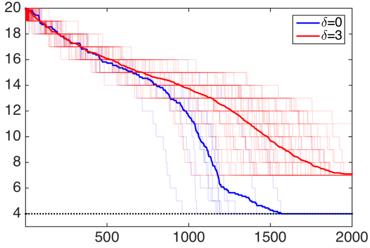
Matrix least-squares regularized by nuclear norm $(\|X\|_* = \|\sigma(X)\|_1)$

$$\min_{X\in\mathbb{R}^{d=m\times m}} \quad \frac{1}{2}\|A(X)-y\|^2 \ + \ \lambda\|X\|_*$$

Generate many random problems (with $m = 20$ and $n = 300$), solve them

Select those with $\operatorname{rank}(X^\star) = 4$ and $\delta = 0$ or 3 $\quad (\delta = \#\{i : |\sigma_i(U^\star)| = 1\} - \operatorname{rank}(X^\star))$

Plot the decrease of $\operatorname{rank}(X_k)$ with $X_{k+1} = \operatorname{prox}_{\gamma\|\cdot\|_*}\big(X_k - \gamma A^*(A(X_k) - y)\big)$



$\delta = 0$: weak degeneracy vs. $\delta = 3$: strong degeneracy

Illustration with nuclear norm

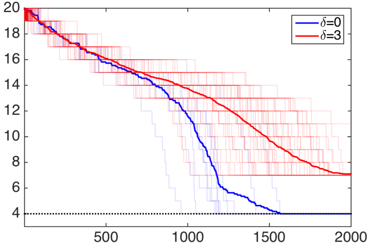
Matrix least-squares regularized by nuclear norm $(\|X\|_* = \|\sigma(X)\|_1)$

$$\min_{X\in\mathbb{R}^{d=m\times m}} \quad \frac{1}{2}\|A(X)-y\|^2 \;+\; \lambda\|X\|_*$$

Generate many random problems (with $m = 20$ and $n = 300$), solve them

Select those with $\mathrm{rank}(X^\star)=4$ and $\delta = 0$ or 3 $\quad (\delta = \#\{i : |\sigma_i(U^\star)| = 1\} - \mathrm{rank}(X^\star))$

Plot the decrease of $\mathrm{rank}(X_k)$ with $X_{k+1} = \mathrm{prox}_{\gamma\|\cdot\|_*}\big(X_k - \gamma A^*(A(X_k) - y)\big)$



$\delta = 0$: weak degeneracy vs. $\delta = 3$: strong degeneracy

Nonsmoothness attracts (proximal) algorithms

Outline

Machine learning in a nutschell

Supervised learning set-up

- Data $(a_j, y_j)_{j=1,\dots,n}$, prediction $h(\cdot, x)$, model parameters $x \in \mathbb{R}^d$
- (Regularized) empirical risk minimization (learning is optimizing !)

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{j=1}^{n} \ell\big(y_j, h(a_j, x)\big) \quad (+ \;\; \lambda R(x))$$

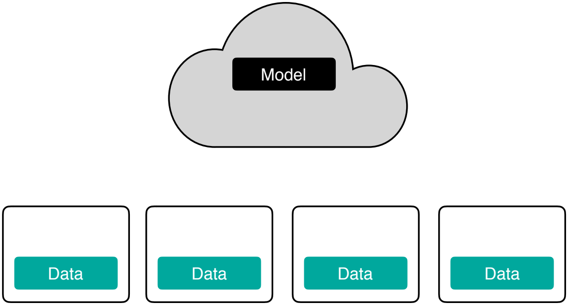
Machine learning in a nutschell

Supervised learning set-up

- Data $(a_j, y_j)_{j=1,\ldots,n}$, prediction $h(\cdot, x)$, model parameters $x \in \mathbb{R}^d$
- (Regularized) empirical risk minimization (learning is optimizing !)

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{j=1}^{n} \ell\big(y_j, h(a_j, x)\big) \quad (+ \ \lambda R(x))$$

(Standard) centralized learning



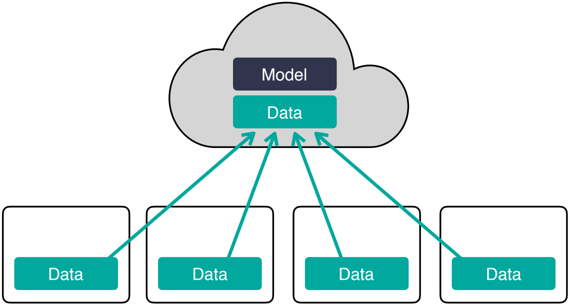
Machine learning in a nutschell

Supervised learning set-up

- Data $(a_j, y_j)_{j=1,\dots,n}$, prediction $h(\cdot, x)$, model parameters $x \in \mathbb{R}^d$
- (Regularized) empirical risk minimization (learning is optimizing !)

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \ell\big(y_j, h(a_j, x)\big) \quad (+ \ \lambda R(x))$$

(Standard) centralized learning



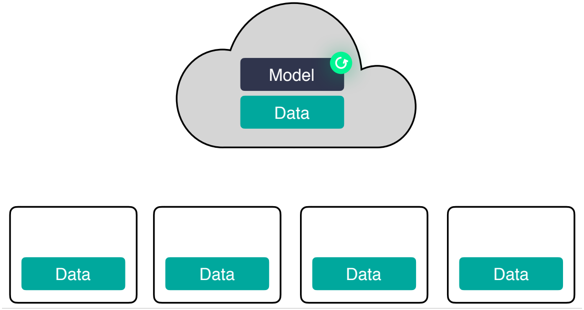
Machine learning in a nutschell

Supervised learning set-up

- Data $(a_j, y_j)_{j=1,\ldots,n}$, prediction $h(\cdot, x)$, model parameters $x \in \mathbb{R}^d$
- (Regularized) empirical risk minimization (learning is optimizing !)

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{j=1}^{n} \ell\big(y_j, h(a_j, x)\big) \quad (+ \ \lambda R(x))$$

(Standard) centralized learning



Machine learning in a nutschell

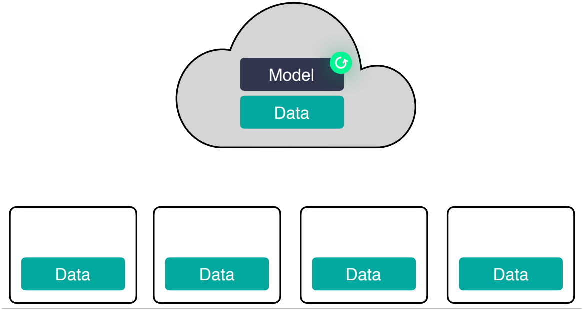
Supervised learning set-up

- Data $(a_j, y_j)_{j=1,\dots,n}$, prediction $h(\cdot, x)$, model parameters $x \in \mathbb{R}^d$
- (Regularized) empirical risk minimization (learning is optimizing !)

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{j=1}^{n} \ell\big(y_j, h(a_j, x)\big) \quad (+ \ \lambda R(x))$$

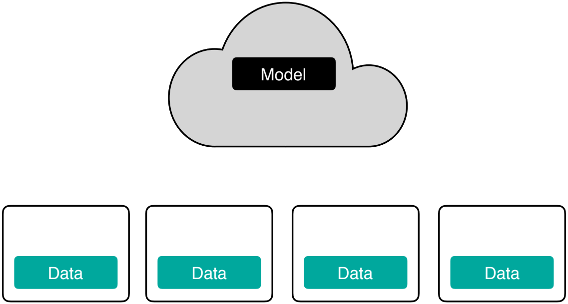
(Standard) centralized learning

- needs of lot of storage ☹
- is highly privacy invasive ☹



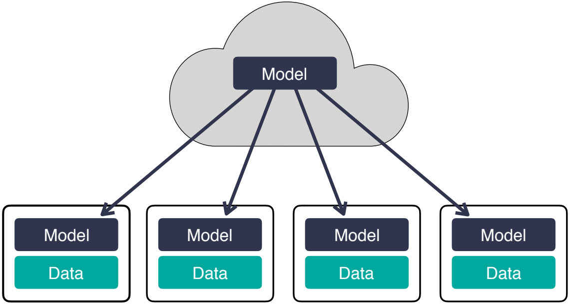
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up



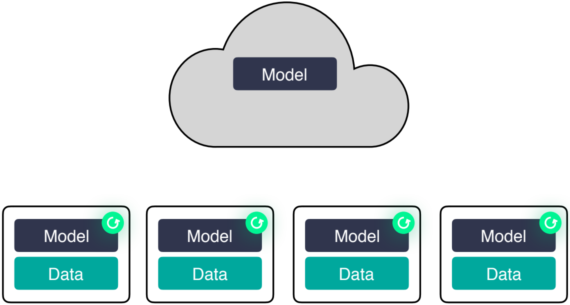
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up



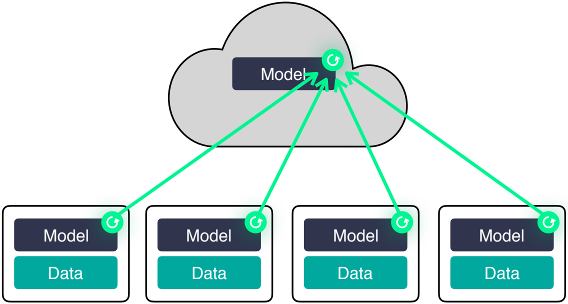
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up



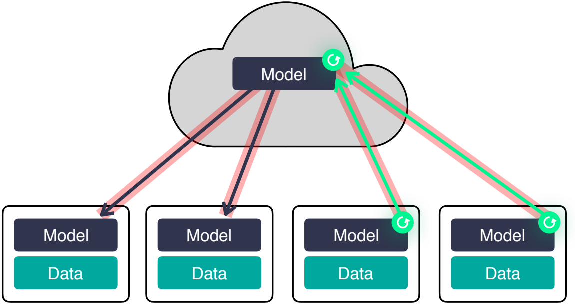
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up



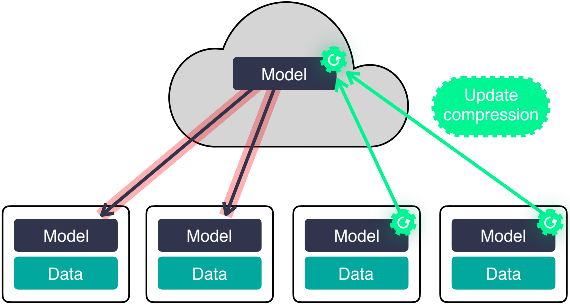
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up   Communication is the bottleneck ☹



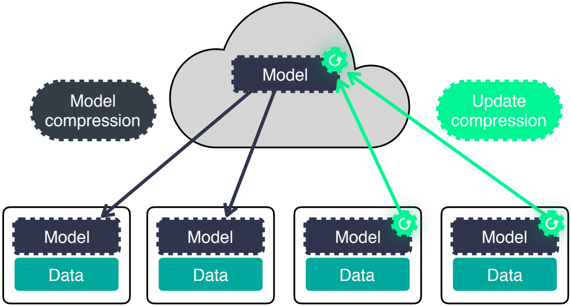
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up   Communication is the bottleneck ☹



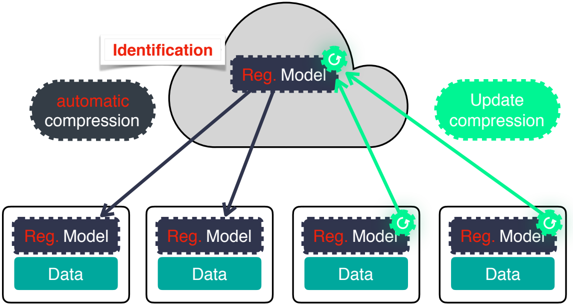
Nonsmooth regularization for distributed learning

Distributed (or federative) set-up   Communication is the bottleneck ☹



Nonsmooth regularization for distributed learning

Distributed (or federative) set-up   Communication is the bottleneck ☹

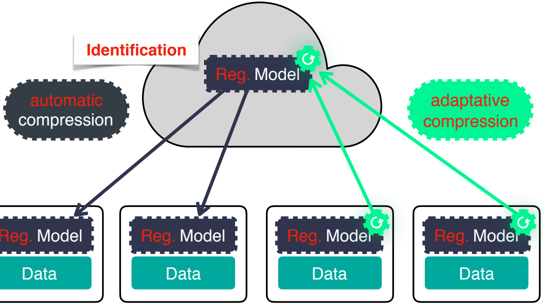


- Observation: identification gives automatic model compression

  e.g. for $R = \|\cdot\|_1$, model becomes sparse... just communicate nonzero entries!

Nonsmooth regularization for distributed learning

Distributed (or federative) set-up   Communication is the bottleneck ☹



- Observation: identification gives automatic model compression

  e.g. for $R = \|\cdot\|_1$, model becomes sparse... just communicate nonzero entries!

- [Grishchenko, Iutzeler, M. '19] uses again identification for update comp.

  Project update onto $\quad M_{x_k} \quad + \quad$ randomly selected M

  e.g. for $R = \|\cdot\|_1$, select current support + random entries

- Algo with intricate convergence analysis due to non-uniform selection...

Illustration of communication-efficient proximal method

On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\langle a_j,x\rangle\right) \;+\; \lambda\,\mathrm{TV}(x)$$

Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual distributed proximal-gradient (black)
- Adaptive distributed proximal-subspace descent (red)

  for different selections M_{x_k} + random others

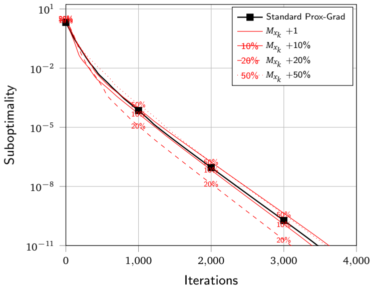


Illustration of communication-efficient proximal method

On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

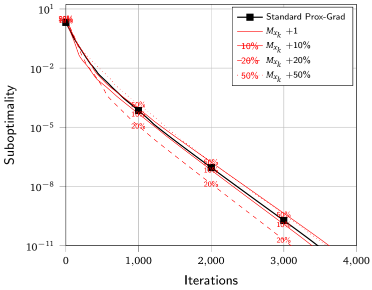
$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\langle a_j, x\rangle\right) \;+\; \lambda\,\mathrm{TV}(x)$$

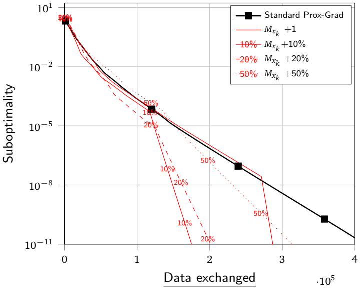
Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual distributed proximal-gradient (black)
- Adaptive distributed proximal-subspace descent (red)

for different selections M_{x_k} + random others





Acceleration... with respect to data-exchanged !

Illustration of communication-efficient proximal method

On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

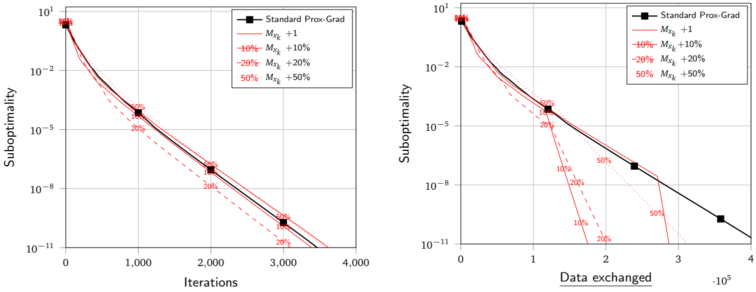
$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\langle a_j, x\rangle\right) \;+\; \lambda\,\mathrm{TV}(x)$$

Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual distributed proximal-gradient (black)
- Adaptive distributed proximal-subspace descent (red)

  for different selections M_{x_k} + random others



Acceleration... with respect to data-exchanged !

Tradeoff between compression (less comm.) and identification (faster cv)

Outline

Supervised learning: model consistency ?

- Assume data $(a_i, y_i)_{i=1,\dots,n}$ are sampled from linear model

  $y = \langle a, x_0 \rangle + w$ with random (a, w) (of unknown probability measure ρ)

- Structure assumption: x_0 has a low-complexity for R

  $$x_0 = \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ R(x) : x \in \operatorname{argmin}_{z \in \mathbb{R}^d} \mathbb{E}_\rho \left[(\langle a, z \rangle - y)^2 \right] \right\}$$

- Regularized least-squares (if $R = \|\cdot\|_1$, this is LASSO)

  $$\min_{x \in \mathbb{R}^d} \frac{1}{2n} \sum_{i=1}^{n} (\langle a_i, x \rangle - y_i)^2 + \lambda_n R(x)$$

- Stochastic (proximal-)gradient algorithms (at iteration k, pick randomly $i(k)$)

  $$x_{k+1} = \operatorname{prox}_{\gamma_k \lambda_n R} \left(x_k - \gamma_k \big((\langle a_{i(k)}, x_k \rangle - y_{i(k)}) \, a_{i(k)} + \varepsilon_k \big) \right)$$

  E.g. SGD, SAGA [Delfazio *et al* '14], SVRG [Xiao-Zhang '14]

Supervised learning: model consistency ?

- Assume data $(a_i, y_i)_{i=1,\dots,n}$ are sampled from linear model

  $y = \langle a, x_0 \rangle + w$ with random (a, w) (of unknown probability measure ρ)

- Structure assumption: x_0 has a low-complexity for R

  $x_0 = \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ R(x) : x \in \operatorname{argmin}_{z \in \mathbb{R}^d} \mathbb{E}_\rho \left[(\langle a, z \rangle - y)^2 \right] \right\}$

- Regularized least-squares (if $R = \|\cdot\|_1$, this is LASSO)

$$\min_{x \in \mathbb{R}^d} \frac{1}{2n} \sum_{i=1}^{n} (\langle a_i, x \rangle - y_i)^2 + \lambda_n R(x)$$

- Stochastic (proximal-)gradient algorithms (at iteration k, pick randomly $i(k)$)

$$x_{k+1} = \operatorname{prox}_{\gamma_k \lambda_n R} \Big(x_k - \gamma_k \big((\langle a_{i(k)}, x_k \rangle - y_{i(k)}) \, a_{i(k)} + \varepsilon_k \big) \Big)$$

  E.g. SGD, SAGA [Delfazio *et al* '14], SVRG [Xiao-Zhang '14]

- Do we have model recovery/consistency i.e. $x_k \in M_{x_0}$?
  (when number of observations $n \to +\infty$)

Enlarged identification of stochastic algorithms

Theorem (Garrigos, Fadili, M., Peyré '18)

Take $\lambda_n \to 0$ *with* $\lambda_n \sqrt{n/(\log\log n)} \to +\infty$. *If* n *large enough and for*

$$x_{k+1} = \mathrm{prox}_{\gamma_k \lambda_n R}\left(x_k - \gamma_k \big((\langle a_{i(k)}, x_k \rangle - y_{i(k)})\, a_{i(k)} + \varepsilon_k \big) \right)$$

with mild assumptions on errors ε_k *and stepsizes* γ_k. *Then, for* k *large, a.s.*

$$M_{x_0} \leqslant M_{x_k} \leqslant \mathcal{J}_{R^*}(M^*_{\eta_0})$$

with $\eta_0 = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\{ \eta^\top C^\dagger \eta \,:\, \eta \in \partial R(w_0) \cap \mathrm{Im}\, C \right\}$ *and* $C = \mathbb{E}_\rho\left[a a^\top \right]$

Comments:

- key dual object $\eta_0 \in \partial R(x_0)$ [Vaiter *et al* '16]
- λ_n decreases to 0, but not too fast
- SAGA and SVRG satisfy the "mild" assumption [Poon *et al* '18]
- (Prox-)SGD does not – and does not identify (e.g. [Lee Wright '12])

Enlarged identification of stochastic algorithms

Theorem (Garrigos, Fadili, M., Peyré '18)

Take $\lambda_n \to 0$ *with* $\lambda_n \sqrt{n/(\log\log n)} \to +\infty$. *If* n *large enough and for*

$$x_{k+1} = \mathrm{prox}_{\gamma_k \lambda_n R}\left(x_k - \gamma_k \left(\left(\langle a_{i(k)}, x_k \rangle - y_{i(k)} \right) a_{i(k)} + \varepsilon_k \right) \right)$$

with mild assumptions on errors ε_k *and stepsizes* γ_k. *Then, for* k *large, a.s.*

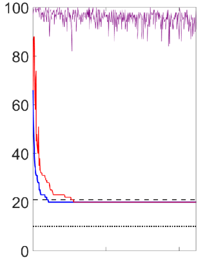
$$M_{x_0} \leqslant M_{x_k} \leqslant \mathcal{J}_{R^*}(M^*_{\eta_0})$$

with $\eta_0 = \underset{\eta \in \mathbb{R}^p}{\mathrm{argmin}} \left\{ \eta^\top C^\dagger \eta \,:\, \eta \in \partial R(w_0) \cap \mathrm{Im}\, C \right\}$ *and* $C = \mathbb{E}_\rho\left[a a^\top \right]$

Comments:

- key dual object $\eta_0 \in \partial R(x_0)$ [Vaiter *et al* '16]
- λ_n decreases to 0, but not too fast
- SAGA and SVRG satisfy the "mild" assumption [Poon *et al* '18]
- (Prox-)SGD does not – and does not identify (e.g. [Lee Wright '12])

(on a LASSO instance)



Conclusion, perspectives

Take-home message

- Nonsmooth regularizers are useful in models, in theory, and in practice
- Compressed communinations by adaptative dimension reduction
- Better understanding of optim. algos (beyond convergence)
- Enlarged localization results (explaining observed phenomena)

Extensions

- Many possible refinements of sensitivity results
  other data fidelity terms, a priori control on strata dimension, explaining transition curves...
- Use identification to accelerate convergence
  interplay between identification and acceleration
- Subspace descent algorithms generalizing coordinate descent
  for nonseparable functions

Conclusion, perspectives

Take-home message

- Nonsmooth regularizers are useful in models, in theory, and in practice
- Compressed communinations by adaptative dimension reduction
- Better understanding of optim. algos (beyond convergence)
- Enlarged localization results (explaining observed phenomena)

Extensions

- Many possible refinements of sensitivity results
  other data fidelity terms, a priori control on strata dimension, explaining transition curves...
- Use identification to accelerate convergence
  interplay between identification and acceleration
- Subspace descent algorithms generalizing coordinate descent
  for nonseparable functions

thanks !!