Harnessing Structure in Optimization for Machine Learning

Franck Iutzeler LJK, Univ. Grenoble Alpes

Optimization for Machine Learning
CIRM – 9-13 March 2020

Structure	Regularization
sparsity	$r = \|\cdot\|_1$
anti-sparsity	$r = \|\cdot\|_\infty$
low rank	$r = \|\cdot\|_*$
$\vdots$	$\vdots$

Linear inverse problems: for a chosen regularization, we seek

$$x^\star \in \arg\min_x r(x) \qquad \text{such that } Ax = b$$

Regularized **Empirical Risk Minimization** problem:

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \mathcal{R}\left(x; \{a_i, b_i\}_{i=1}^m\right) \quad + \quad \lambda \; r(x)$$

obtained from statistical modeling — chosen regularization

$$\text{e.g. Lasso: Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^m \frac{1}{2}(a_i^\top x - b_i)^2 \quad + \quad \lambda \; \|x\|_1$$

Regularization can improve statistical properties (generalization, stability, ...).

◇ Tibshirani: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (1996)
◇ Tibshirani *et al.*: Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society (2004)
◇ Vaiter, Peyré, Fadili: Model consistency of partly smooth regularizers. IEEE Trans. on Information Theory (2017)

Composite minimization

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \mathcal{R}\left(x; \{a_i, b_i\}_{i=1}^m\right) \quad + \quad \lambda\ r(x)$$

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad f(x) \quad + \quad g(x)$$

smooth non-smooth

- f: differentiable surrogate of the empirical risk ⇒ **Gradient**
  non-linear smooth function that depends on all the data
- g: non-smooth but chosen regularization ⇒ **Proximity operator**
  non-differentiability on some manifolds implies structure on the solutions

$$\mathbf{prox}_{\gamma g}(u) = \arg\min_{y\in\mathbb{R}^n} \left\{ g(y) + \frac{1}{2\gamma} \|y - u\|_2^2 \right\}$$

closed form/easy for many regularizations:

- $g(x) = \|x\|_1$
- $g(x) = TV(x)$
- $g(x) = indicator_C(x)$

Natural optimization method: **proximal gradient**

$$\begin{cases} u_{k+1} = x_k - \gamma \nabla f(x_k) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases}$$

and its stochastic variants: proximal sgd, etc.

Example: LASSO

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \mathcal{R}\left(x; \{a_i, b_i\}_{i=1}^m\right) \quad + \quad \lambda\ r(x)$$

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \frac{1}{2}\|Ax-b\|_2^2 \quad + \quad \lambda\|x\|_1$$

smooth non-smooth

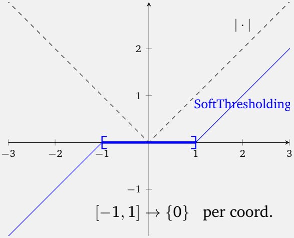
Coordinates	**Structure**	$\leftrightarrow$	**Optimality conditions**
$\forall i$	$x_i^\star = 0$	$\Leftrightarrow$	$A_i^\top(Ax^\star - b) \in [-\lambda, \lambda]$

Proximity Operator: per coordinate

$$\left[\mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u)\right]_i = \begin{cases} u_i - \lambda\gamma & \text{if } u_i > \lambda\gamma \\ 0 & \text{if } u_i \in [-\lambda\gamma; \lambda\gamma] \\ u_i + \lambda\gamma & \text{if } u_i < -\lambda\gamma \end{cases}$$

Proximal Gradient (aka ISTA):

$$\begin{cases} u_{k+1} = x_k - \gamma A^\top(Ax_k - b) \\ x_{k+1} = \mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u_{k+1}) \end{cases}$$



>>> Structure, Non-differentiability, and Proximity operator

Example: LASSO

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \mathcal{R}\left(x; \{a_i, b_i\}_{i=1}^m\right) \quad + \quad \lambda\ r(x)$$

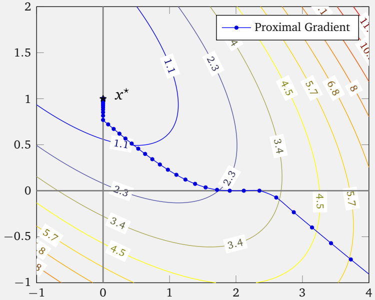
$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \underbrace{\tfrac{1}{2}\|Ax-b\|_2^2}_{\text{smooth}} \quad + \quad \underbrace{\lambda\|x\|_1}_{\text{non-smooth}}$$

Coordinates	**Structure**	$\leftrightarrow$	**Optimality conditions**	$\leftrightarrow$	**Proximity operation**
$\forall i$	$x_i^\star = 0$	$\Leftrightarrow$	$A_i^\top(Ax^\star - b) \in [-\lambda, \lambda]$	$\Leftrightarrow$	$\left[\mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u^\star)\right]_i = 0$, $u^\star = x^\star - \gamma A^\top(Ax^\star - b)$

$$\left[\mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u)\right]_i = \begin{cases} u_i - \lambda\gamma & \text{if } u_i > \lambda\gamma \\ 0 & \text{if } u_i \in [-\lambda\gamma; \lambda\gamma] \\ u_i + \lambda\gamma & \text{if } u_i < -\lambda\gamma \end{cases}$$

Proximal Gradient (aka ISTA):

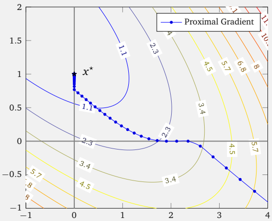
$$\begin{cases} u_{k+1} = x_k - \gamma A^\top(Ax_k - b) \\ x_{k+1} = \mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u_{k+1}) \end{cases}$$



Iterates (x_k) reach the **same structure as $x^\star$** in **finite** time!

Proximal Algorithms:

$$\begin{cases} u_{k+1} = x_k - \gamma \nabla f(x_k) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases}$$



> **project on manifolds**

Let $\mathcal{M}$ be a manifold and u_k such that

$$x_k = \mathbf{prox}_{\gamma g}(u_k) \in \mathcal{M} \quad \text{and} \quad \frac{u_k - x_k}{\gamma} \in \operatorname{ri} \partial g(x_k)$$

If g is partly smooth at x_k relative to $\mathcal{M}$, then

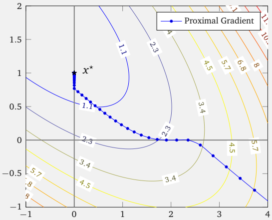
$$\mathbf{prox}_{\gamma g}(u) \in \mathcal{M}$$

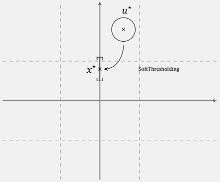
for any u close to u_k.

◇ Hare, Lewis: Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis (2004)
◇ Daniilidis, Hare, Malick: Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. Optimization (2006)

Proximal Algorithms:

$$\begin{cases} u_{k+1} = x_k - \gamma \nabla f(x_k) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases}$$





- > **project on manifolds**
- > **identify the optimal structure**

Let (x_k) and (u_k) be a pair of sequences such that

$$x_k = \mathbf{prox}_{\gamma g}(u^k) \to x^\star = \mathbf{prox}_{\gamma g}(u^\star)$$

and $\mathcal{M}$ be a manifold. If $x^\star \in \mathcal{M}$ and

$$\exists \varepsilon > 0 \text{ such that for all } u \in \mathcal{B}(u^\star, \varepsilon),\ \mathbf{prox}_{\gamma g}(u) \in \mathcal{M} \qquad \text{(QC)}$$

holds, then, **after some finite but unknown time, $x_k \in \mathcal{M}$**.

◇ Lewis: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization (2002)
◇ Fadili, Malick, Peyré: Sensitivity analysis for mirror-stratifiable convex functions. SIAM Journal on Optimization (2018)

> **Nonsmoothness** is actively studied in Numerical Optimization...

Subgradients, Partial Smoothness/prox-regularity, Bregman metrics, Error Bounds/Kurdyka-Łojasiewicz, etc.

◇ Hare, Lewis: Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis (2004)
◇ Lemarechal, Oustry, Sagastizabal: The U-Lagrangian of a convex function. Transactions of the AMS (2000)
◇ Bolte, Daniilidis, Lewis: The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization (2007)
◇ Chen, Teboulle: A proximal-based decomposition method for convex minimization problems. Mathematical Programming (1994)

>>> “Nonsmoothness can help”

- **Nonsmoothness** is actively studied in Numerical Optimization...

  Subgradients, Partial Smoothness/prox-regularity, Bregman metrics, Error Bounds/Kurdyka-Łojasiewicz, etc.

- ...but **often suffered** because of lack of structure/expression.

  Bundle methods, Gradient Sampling, Smoothing, Inexact proximal methods, etc.

- Nesterov: Smooth minimization of non-smooth functions. Mathematical Programming (2005)
- Burke, Lewis, Overton: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM Journal on Optimization (2005)
- Solodov, Svaiter: A hybrid projection-proximal point algorithm. Journal of convex analysis (1999)
- de Oliveira, Sagastizábal: Bundle methods in the XXIst century: A bird’s-eye view. Pesquisa Operacional (2014)

>>> “Nonsmoothness can help”

- > **Nonsmoothness** is actively studied in Numerical Optimization...

  Subgradients, Partial Smoothness/prox-regularity, Bregman metrics, Error Bounds/Kurdyka-Łojasiewicz, etc.

- > ...but **often suffered** because of lack of structure/expression.

  Bundle methods, Gradient Sampling, Smoothing, Inexact proximal methods, etc.

- > For **Machine Learning objectives**, it can often be **harnessed**
  - Explicit/“proximable” regularizations ℓ_1, nuclear norm
  - We *know* the *expressions* and *activity* of sought structures sparsity, rank

See the talks of ...

◇ Bach, et al.: Optimization with sparsity-inducing penalties. Foundations and Trends in Machine Learning (2012)
◇ Massias, Salmon, Gramfort: Celer: a fast solver for the lasso with dual extrapolation. ICML (2018)
◇ Liang, Fadili, Peyré: Local linear convergence of forward–backward under partial smoothness. NeurIPS (2014)
◇ O’Donoghue, Candes: Adaptive restart for accelerated gradient schemes. Foundations of computational mathematic (2015)

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \mathcal{R}\left(x; \{a_i, b_i\}_{i=1}^m\right) \quad + \quad \lambda\ r(x)$$

$$\text{Find} \quad x^\star \in \arg\min_{x\in\mathbb{R}^n} \quad \underbrace{f(x)}_{\text{smooth}} \quad + \quad \underbrace{g(x)}_{\text{non-smooth}}$$

A reason why the nonsmoothness of ML problems can be leveraged is their **noticeable structure**, that is:

We can design a *lookout collection* $\mathsf{C} = \{\mathcal{M}_1, .., \mathcal{M}_p\}$ of closed sets such that:

(i) we have a projection mapping $\mathrm{proj}_{\mathcal{M}_i}$ onto $\mathcal{M}_i$ for all i;

(ii) $\mathbf{prox}_{\gamma g}(u)$ is a singleton and can be computed explicitly for any u and γ;

(iii) upon computation of $x = \mathbf{prox}_{\gamma g}(u)$, we know if $x \in \mathcal{M}_i$ or not for all i.

$\Rightarrow$ **Identification can be directly harnessed**.

Example: Sparse structure and $g = \|\cdot\|_1, \|\cdot\|_{0.5}^{0.5}, \|\cdot\|_0, \ldots$

$$\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\}$$

lookout collection $\mathsf{C} = \{\mathcal{M}_1, .., \mathcal{M}_p\}$ of closed sets such that:

(i) we have a projection mapping $\text{proj}_{\mathcal{M}_i}$ onto $\mathcal{M}_i$ for all i;
(ii) $\mathbf{prox}_{\gamma g}(u)$ is a singleton and can be computed explicitly for any u and γ;
(iii) upon computation of $x = \mathbf{prox}_{\gamma g}(u)$, we know if $x \in \mathcal{M}_i$ or not for all i.

(QC) $\exists \varepsilon > 0$ such that for all $u \in \mathcal{B}(u^\star, \varepsilon)$, $\mathbf{prox}_{\gamma g}(u) \in \mathcal{M}$

Take any proximal algorithm

$$\begin{cases} u_{k+1} & = \mathsf{Update}\left(f; \{x_\ell\}_{\ell \leq k}; \{u_\ell\}_{\ell \leq k}; \gamma\right) \\ x_{k+1} & = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases} \qquad (\mathbf{prox} - \mathsf{Update})$$

such that (u_k) converges almost surely to a point $u^\star$
with $x^\star = \mathbf{prox}_{\gamma g}(u^\star)$ a solution of the problem.

Let's use the structure
What can we do on the way to identification/when screening is inefficient?

not close to $x^\star$, no explicit or bad dual (non-convex), $\mathbf{prox}_{\gamma g}(\mathcal{U}_k)$ difficult to evaluate *lookout collection* $\mathsf{C} = \{\mathcal{M}_1, .., \mathcal{M}_p\}$ of closed sets such that:

(i) we have a projection mapping $\text{proj}_{\mathcal{M}_i}$ onto $\mathcal{M}_i$ for all i;
(ii) $\mathbf{prox}_{\gamma g}(u)$ is a singleton and can be computed explicitly for any u and γ;
(iii) upon computation of $x = \mathbf{prox}_{\gamma g}(u)$, we know if $x \in \mathcal{M}_i$ or not for all i.

(QC) $\exists \varepsilon > 0$ such that for all $u \in \mathcal{B}(u^\star, \varepsilon)$, $\mathbf{prox}_{\gamma g}(u) \in \mathcal{M}$

Take any proximal algorithm

$$\begin{cases} u_{k+1} &= \mathsf{Update}\left(f; \{x_\ell\}_{\ell \le k}; \{u_\ell\}_{\ell \le k}; \gamma\right) \\ x_{k+1} &= \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases} \qquad (\mathbf{prox}-\mathsf{Update})$$

such that (u_k) converges almost surely to a point $u^\star$
with $x^\star = \mathbf{prox}_{\gamma g}(u^\star)$ a solution of the problem.

Define $\mathcal{M}_k = \mathbb{R}^n \bigcap_{i: x_k \in \mathcal{M}_i} \mathcal{M}_i$ and $\mathcal{M}^\star := \mathbb{R}^n \bigcap_{i: x^\star \in \mathcal{M}_i} \mathcal{M}_i$, then:

$\mathcal{M}_k \subset \mathbb{R}^n$ partial identif/screening and $\mathcal{M}_k = \mathcal{M}^\star$ after some finite time identification

1– Observing $\mathcal{M}_k$ can help reduce the dimension of the problem on the way
Can we efficiently restrict Update **using** $\mathcal{M}_k$**?**

2– The uncovered structure along the way bears valuable information
Does accelerated proximal gradient identify as well as vanilla?

- **Adaptive Subspace Descent**

Interplay between Acceleration and Identification

Disclaimer: This part talk assumes that the identified manifolds are linear subspaces eg: $\|Dx\|_1$.

$$\begin{cases} y_k &= x_k - \gamma \nabla f(x_k) \\ z_k &= \qquad\qquad y_k \\ x_{k+1} &= \mathbf{prox}_{\gamma g}(z_k) \end{cases}$$

- Vanilla Proximal gradient identifies but does not use it
  full gradient computed at each iteration

Example: Sparse structure and $g = \|\cdot\|_1$

$$\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\}$$
$$\mathcal{M}_k = \{x \in \mathbb{R}^n : x_i = x_{i,k} \qquad\qquad \}$$

Disclaimer: This part talk assumes that the identified manifolds are linear subspaces eg: $\|Dx\|_1$.

$$\begin{cases} \quad \text{Observe } \mathcal{M}_k = \mathbb{R}^n \bigcap_{i:x_k \in \mathcal{M}_i} \quad \mathcal{M}_i \\ y_k = x_k - \gamma \nabla f(x_k) \\ z_k = \quad \operatorname{proj}_{\mathcal{M}_k}(\quad y_k) + \operatorname{proj}^{\perp}_{\mathcal{M}_k}(z_{k-1}) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(z_k) \end{cases}$$

> Direct Use of Identification may not converge
eg: starting with 0

Example: Sparse structure and $g = \|\cdot\|_1$

$$\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\}$$
$$\mathcal{M}_k = \{x \in \mathbb{R}^n : x_i = x_{i,k} \quad \}$$

Disclaimer: This part talk assumes that the identified manifolds are linear subspaces eg: $\|Dx\|_1$.

$$\begin{cases} & \text{Observe } \mathcal{M}_k = \mathbb{R}^n \bigcap_{i:x_k \in \mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n) \text{ for } \xi_{k,i} \sim \mathcal{B}(p) \\ y_k & = x_k - \gamma \nabla f(x_k) \\ z_k & = \quad \mathrm{proj}_{\mathcal{M}_k}(\quad y_k) + \mathrm{proj}^{\perp}_{\mathcal{M}_k}(z_{k-1}) \\ x_{k+1} & = \mathbf{prox}_{\gamma g}(z_k) \end{cases}$$

> Mixing Identification and Randomized coordinate descent biases gradient convergence issues

Example: Sparse structure and $g = \|\cdot\|_1$

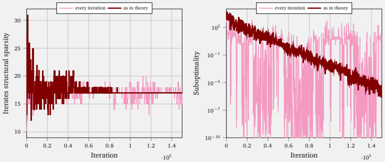
$$\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\}$$
$$\mathcal{M}_k = \{x \in \mathbb{R}^n : x_i = x_{i,k} \text{for some } i\}$$

Disclaimer: This part talk assumes that the identified manifolds are linear subspaces eg: $\|Dx\|_1$.

$$\left\{\begin{aligned} & \text{Observe } \mathcal{M}_k = \mathbb{R}^n \bigcap_{i:x_k \in \mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n) \text{ for } \xi_{k,i} \sim \mathcal{B}(p) \\ y_k &= x_k - \gamma \nabla f(x_k) \\ z_k &= Q_k^{-1}(\text{proj}_{\mathcal{M}_k}(Q_k y_k) + \text{proj}^{\perp}_{\mathcal{M}_k}(z_{k-1})) \\ x_{k+1} &= \mathbf{prox}_{\gamma g}(z_k) \end{aligned}\right.$$

> With $Q_k := (\mathbb{E}\text{proj}_{\mathcal{M}_k})^{-1/2}$, this works *after identification*
but before... no, which prevents identification...

TV-regularized logistic regression:

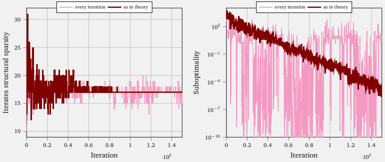


Disclaimer: This part talk assumes that the identified manifolds are linear subspaces eg: $\|Dx\|_1$.

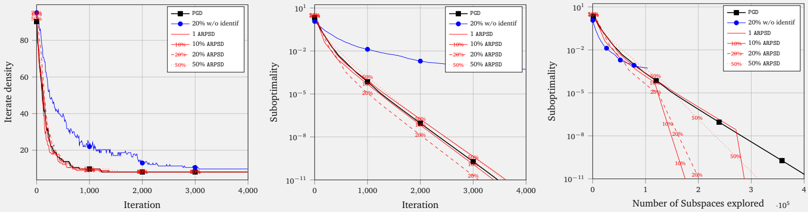
$$
\begin{cases}
\text{Observe } \mathcal{M}_k = \mathbb{R}^n \bigcap_{i:x_\ell \in \mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n) \text{ for } \xi_{k,i} \sim \mathcal{B}(p) \\
y_k = x_k - \gamma \nabla f(x_k) \\
z_k = Q_k^{-1}(\mathrm{proj}_{\mathcal{M}_k}(Q_k y_k) + \mathrm{proj}^{\perp}_{\mathcal{M}_k}(z_{k-1})) \\
x_{k+1} = \mathbf{prox}_{\gamma g}(z_k) \\
\text{Check if an adaptation can be performed, if so } \ell \leftarrow k+1
\end{cases}
$$

> Generalized Support adaptation can be performed at *some* iterations
depends on the *amount of change* $\|Q_k Q_{k+1}^{-1}\|$ and *harshness* of the sparsification $\lambda_{\min}(Q_k)$

TV-regularized logistic regression:



TV-reg. logistic regression on a1a (1605×143), 90% final *jump* sparsity



- > Iterate structure enforced by nonsmooth regularizers can be used to adapt the selection probabilities of coordinate descent/sketching;
- > Before identification, adaptation *has to be moderate*.

▷ Grishchenko, I., & Malick: *Proximal Gradient Methods with Adaptive Subspace Sampling*, in revision for Mathematics of Operation Research

available on my webpage, more details at SMAI MODE

Adaptive Subspace Descent

- **Interplay between Acceleration and Identification**

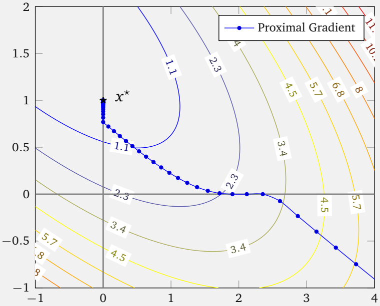
$$\begin{cases} u_{k+1} = y_k - \gamma \nabla f(y_k) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(u_{k+1}) \\ y_{k+1} = x_{k+1} + \underbrace{\alpha_{k+1}(x_{k+1} - x_k)}_{\text{inertia/acceleration}} \end{cases}$$

> $\alpha_{k+1} \equiv 0$: vanilla Proximal Gradient
> $\alpha_{k+1} = \frac{k-1}{k+3}$: **accelerated** Proximal Gradient (aka FISTA)
  Optimal rate for composite problems (coefficients may vary a little)

	PG	Accel. PG
$F(x_k) - F^\star$	$\mathcal{O}(1/k)$	$\mathcal{O}(1/k^2)$
iterates convergence	yes	yes
monotone functional decrease	yes	no
Fejér-monotone iterates	yes	no

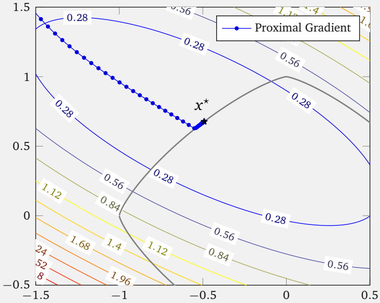
◇ Nesterov: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Dokladi A.N. Sssr (1983)
◇ Beck, Teboulle: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on Imaging Sciences (2009)
◇ Chambolle, Dossal: On the convergence of the iterates of "FISTA". Journal of Optimization theory and Applications (2015)
◇ I., Malick: On the Proximal Gradient Algorithm with Alternated Inertia. Journal of Optimization Theory and Applications (2018)

$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2 + \lambda g(x)$$



$g(x) = \|x\|_1$

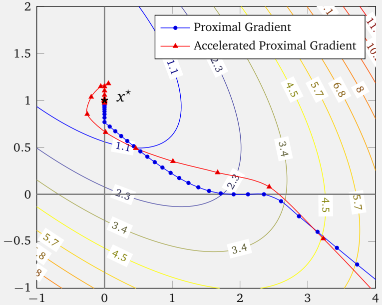
1-norm regularization



$g(x) = \max(\|x\|_{1.3} - 1; 0)$

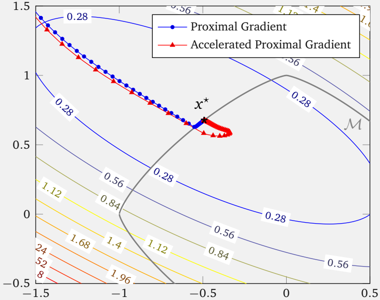
distance to 1.3-norm unit ball

$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2 + \lambda g(x)$$



$g(x) = \|x\|_1$

1-norm regularization



$g(x) = \max(\|x\|_{1.3} - 1; 0)$

distance to 1.3-norm unit ball

- PG identifies well;
- Accelerated PG explores well, identifies eventually, but erratically.

Can we converge fast ***and*** *identify well?*

T is a boolean function of past iterates; decides whether to accelerate or not.

$$\begin{cases} u_{k+1} = y_k - \gamma \nabla f(y_k) \\ x_{k+1} = \mathbf{prox}_{\gamma g}(u_{k+1}) \\ y_{k+1} = \begin{cases} x_{k+1} + \alpha_{k+1}(x_{k+1} - x_k) & \text{if } \mathsf{T} = 1 \\ x_{k+1} & \text{if } \mathsf{T} = 0 \end{cases} \end{cases}$$

Proposed tests: We use our *lookout collection* C

1. No Acceleration *i.e.* $\mathsf{T}^1 = 0$ when reaching a new one:

$$x_{k+1} \in \mathcal{M} \text{ and } x_k \notin \mathcal{M}$$

for some $\mathcal{M} \in \mathsf{C}$.

2. No Acceleration *i.e.* $\mathsf{T}^2 = 0$ if this means leaving:

$$\mathcal{T}_\gamma(x_{k+1}) \in \mathcal{M} \text{ and } \mathcal{T}_\gamma(x_{k+1} + \alpha_{k+1}(x_{k+1} - x_k)) \notin \mathcal{M}$$

for some $\mathcal{M} \in \mathsf{C}$.

where $\mathcal{T}_\gamma := \mathbf{prox}_{\gamma g}(\cdot - \gamma \nabla f(\cdot))$ is the proximal gradient operator.

For analysis reasons, we allow no acceleration only when $\|\mathcal{T}_\gamma(y_k) - y_k\|^2 \leq \delta$ and $F(\mathcal{T}_\gamma(y_k)) \leq F(x_0)$.

Theorem

Let f, g be two convex functions such that f is L-smooth, g is lower semi-continuous, and $f+g$ is semi-algebraic with a minimizer. Take $\gamma \in (0, 1/L]$. Then, the iterates of the proposed methods with test T^1 or T^2 verify

$$F(x_{k+1}) - F^\star \leq \frac{9\|x_0 - x^\star\|^2}{2\gamma(k+2)^2} + \frac{9k\mathsf{R}}{2\gamma(k+2)^2} = \mathcal{O}\left(\frac{1}{k}\right)$$

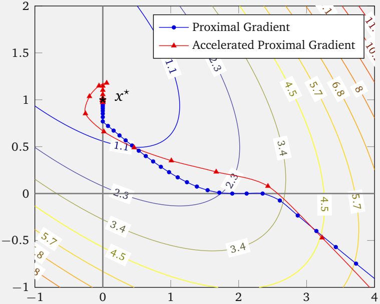
for some $\mathsf{R} > 0$.
Furthermore, if the problem has a unique minimizer $x^\star$ and the qualifying constraint (QC) holds, then the iterates sequence (x_k) converges, finite-time identification happens and

$$F(x_{k+1}) - F(x^\star) \leq \frac{9\|x_0 - x^\star\|^2}{2\gamma(k+2)^2} + \frac{9K\mathsf{R}}{2\gamma(k+2)^2} = \mathcal{O}\left(\frac{1}{k^2}\right).$$

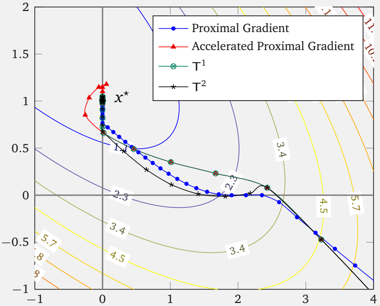
for some finite $K > 0$.

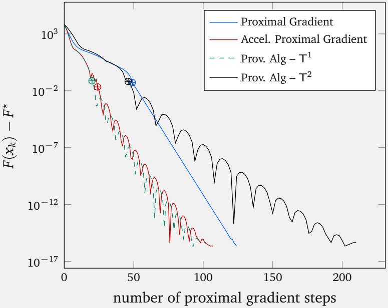
L-smooth means that f is differentiable and ∇f is L-Lipschitz continuous.

$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$



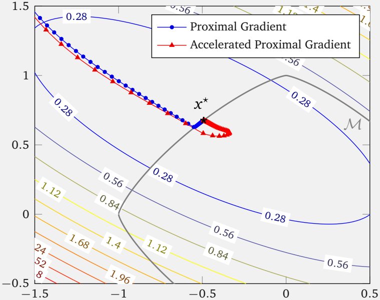
$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$



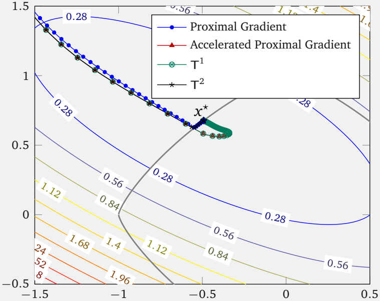


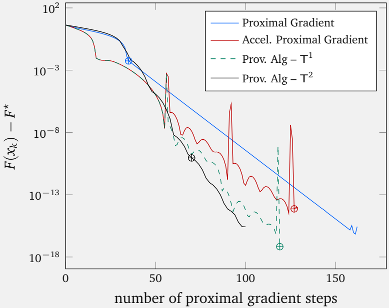
$\oplus$ marks identification time

$$\min_{x\in\mathbb{R}^2} \|Ax-b\|_2^2 + \lambda \max(|x\|_{1.3} - 1; 0)$$



$$\min_{x\in\mathbb{R}^2} \|Ax-b\|_2^2 + \lambda \max(\|x\|_{1.3} - 1; 0)$$

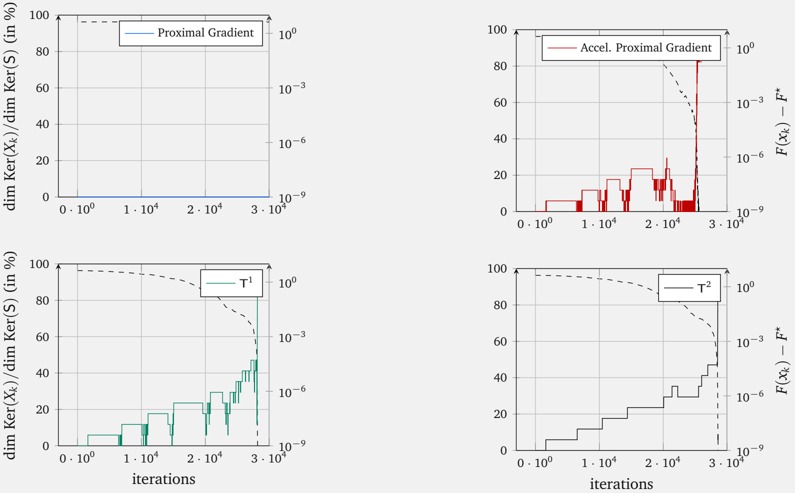




$\oplus$ marks identification time

$$\min_{X \in \mathbb{R}^{20\times 20}} \|AX - B\|_F^2 + \lambda \|X\|_*$$

- $\mathsf{S} \in R^{20\times 20}$ is a **rank 3** matrix;
- $A \in \mathbb{R}^{(16\times 16)\times(20\times 20)}$ is drawn from the normal distribution;
- $B = A\mathsf{S} + E$ with E drawn from the normal distribution with variance .01



- acceleration can hurt identification for the proximal gradient algorithm;
- we proposed a method with stable identification behavior, maintaining an accelerated convergence rate.

▷ Bareilles & I.: *On the Interplay between Acceleration and Identification for the Proximal Gradient algorithm*. arXiv:1909.08944
Try it in Julia on `https://github.com/GillesBareilles/Acceleration-Identification`

- Machine Learning problems often have a *noticeable structure*;
- We can design a *lookout collection* $\mathsf{C} = \{\mathcal{M}_1, .., \mathcal{M}_p\}$ of sets: (i) with easy projections; (ii) identified by proximity operations; (iii) we *know* if these sets are identified or not;
- This structure can/should be harnessed but may be tricky before identification.

▷ Malick & I.: *Nonsmoothness can help! on the Specific Structure of Machine Learning problems*, review/pedagogical paper coming hopefully soon
thanks to this week at CIRM but it also depends whether we go hiking/running in the calanques which may very well be the case

Thanks to ANR JCJC *STROLL* & IDEX UGA IRS *DOLL* & PGMO

Thank you! – Franck IUTZELER `http://www.iutzeler.org`