On the Interplay between Acceleration and Identification for the Proximal Gradient algorithm

Franck Iutzeler LJK, Univ. Grenoble Alpes

IFIP TC7

Regularized **Empirical Risk Minimization** problem:

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

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

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

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 $\Rightarrow$ **Gradient**
  non-linear smooth function that depends on all the data
- g: non-smooth but chosen regularization $\Rightarrow$ **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:

– $g(x) = \|x\|_1$
– $g(x) = TV(x)$
– $g(x) = indicator_C(x)$
– see `http://proximity-operator.net/`

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

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

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

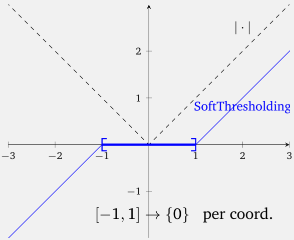
Coordinates	**Structure**	$\leftrightarrow$	**Optimality conditions**
$\forall i$	$x_i^\star = 0$	$\Leftrightarrow$	$A_i^\top(Ax^\star - b) \in [-\lambda, \lambda]$

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):

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



Example: LASSO

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

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

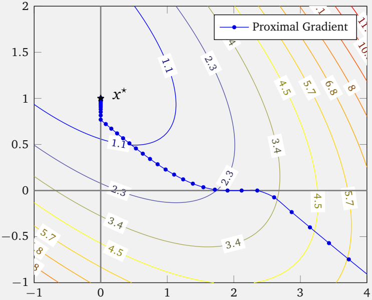
smooth   non-smooth

Coordinates	**Structure**	$\leftrightarrow$	**Optimality conditions**	$\leftrightarrow$	**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):

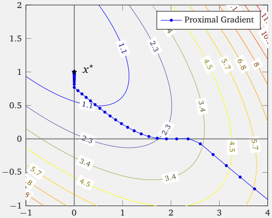
$$\begin{cases} u_{k+1} = x_k - \gamma A^\top(Ax_k - b) \\ x_{k+1} = \mathbf{prox}_{\gamma\lambda\|\cdot\|_1}(u_{k+1}) \end{cases}$$

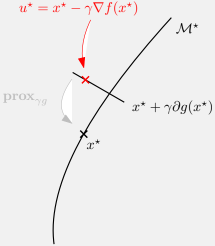


Iterates (x_k) reach the **same structure as $x^\star$** in **finite** time!

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 $u^\star$ such that

$$x^\star = \mathbf{prox}_{\gamma g}(u^\star) \in \mathcal{M} \quad \text{and} \quad \frac{u^\star - x^\star}{\gamma} \in \operatorname{ri} \partial g(x^\star)$$

If g is *partly smooth* at $x^\star$ relative to $\mathcal{M}^\star$ locally smooth along $\mathcal{M}$ and nonsmooth across, then

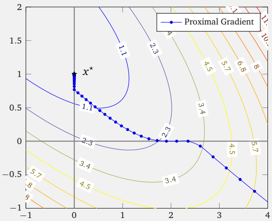
$$\mathbf{prox}_{\gamma g}(u) \in \mathcal{M}^\star$$

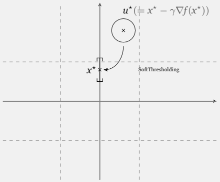
for any u close to $u^\star$.

◇ 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 the *qualification condition*

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

“structure is stable under small perturbation of the data”

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 geometry, etc.

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

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

  Subgradients, Partial Smoothness/prox-regularity, Bregman geometry, 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 J. on Optim. (2005)
◇ Solodov, Svaiter: A hybrid projection-proximal point algorithm. J. of Conv. Analysis (1999)
◇ de Oliveira, Sagastizábal: Bundle methods in the XXIst century: A bird's-eye view. Pesquisa Operacional (2014)

- **Nonsmoothness** is actively studied in Numerical Optimization...
  Subgradients, Partial Smoothness/prox-regularity, Bregman geometry, 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**
  Feature selection, Screening, Faster rates, etc.

◇ Bach, et al.: Optimization with sparsity-inducing penalties. FnT 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 Comp. Math. (2015)

- **Nonsmoothness** is actively studied in Numerical Optimization...
  Subgradients, Partial Smoothness/prox-regularity, Bregman geometry, 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**
  Feature selection, Screening, Faster rates, etc.
- Why?
  - *Explicit/“proximable”* regularizations ℓ_1, nuclear norm
  - We *know* the *expressions* and *activity* of sought structures sparsity, rank
  - Any converging proximal algorithm will *identify* the *optimal structure* of the problem.

▷ I. & Malick: *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications*, review/pedagogical paper, Set-Valued and Variational Analysis, 2020, `https://arxiv.org/abs/2010.00848`

Thanks to the *Optimization for Machine Learning* week at CIRM in March 2020!

Let us solve a *composite problem* problem with a *proximal algorithm*

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

with $x_k = \mathbf{prox}_{\gamma g}(u_k) \longrightarrow x^\star = \mathbf{prox}_{\gamma g}(u^\star)$

- The proximity operator gives a *current structure* $\mathcal{M}_k \subset \mathbb{R}^n$
  partial identif/screening
- We know that *eventually* $\mathcal{M}_k = \mathcal{M}^\star$ after some finite time
  identification

1– **Does faster minimization means faster identification ?**

2– **Can we efficiently restrict our update to $\mathcal{M}_k$?**

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

$\mathcal{M}^\star$ represents the points with the same support as $x^\star$ (ie. non-selected features are put to zero).
$\mathcal{M}_k = \{x \in \mathbb{R}^n : \text{supp}(x_i) = \text{supp}(x_i)\}$ is the current structure.

Interplay between Acceleration and Identification

Newton acceleration on identified manifolds

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

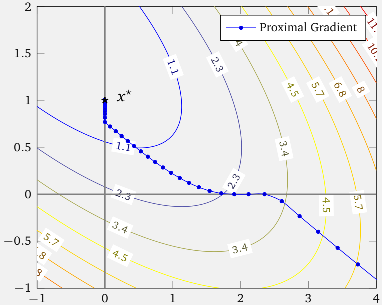
> $\alpha_{k+1} = 0$: vanilla Proximal Gradient

> $\alpha_{k+1} = \frac{k-1}{k+3}$: **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

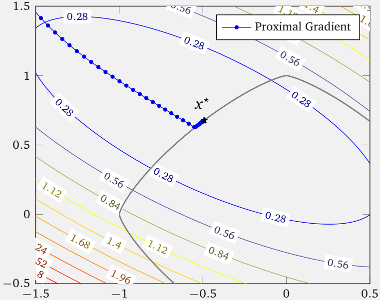
◇ Nesterov: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Sov. Dok. (1983)
◇ Beck, Teboulle: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. on Imag. Sci. (2009)
◇ Chambolle, Dossal: On the convergence of the iterates of "FISTA". J. of Optim. Theory and App. (2015)
◇ I., Malick: On the Proximal Gradient Algorithm with Alternated Inertia. J. of Optim. Theory and App. (2018)

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



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

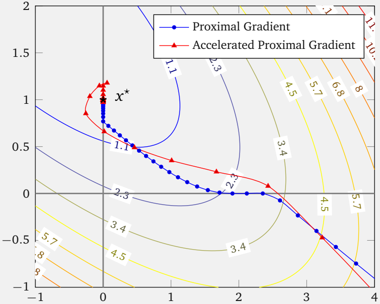
1-norm regularization



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

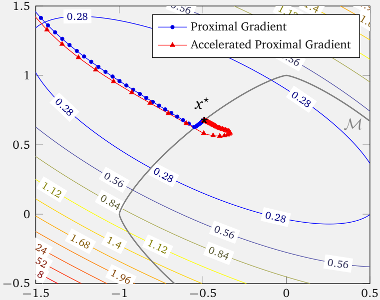
distance to 1.3-norm unit ball

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



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

1-norm regularization



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

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 pre-define a collection $\mathsf{C} = \{\mathcal{M}_1, .., \mathcal{M}_p\}$ of *sought structures*

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

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

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

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

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

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.

Examples of sought structures: sparsity supports, rank.

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 satisfy

$$F(x_{k+1}) - F^\star = \mathcal{O}\left(\frac{1}{k}\right)$$

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

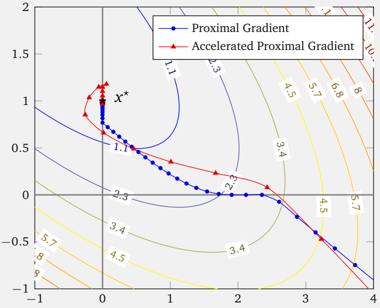
$$F(x_{k+1}) - F(x^\star) = \mathcal{O}\left(\frac{1}{k^2}\right).$$

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

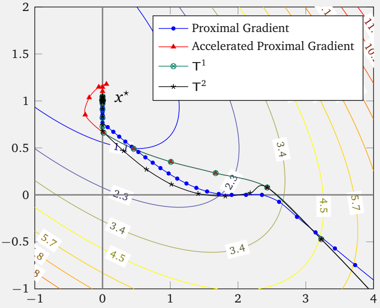
$$\exists \varepsilon > 0 \text{ such that for all } u \in \mathcal{B}(x^\star - \gamma \nabla f(x^\star), \varepsilon),\ \mathbf{prox}_{\gamma g}(u) \in \mathcal{M}^\star \qquad \text{(QC)}$$

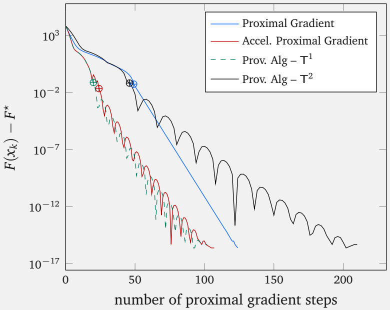
For the ℓ_1 norm, this means this means $-\nabla_i f(x^\star) \in (-\lambda; \lambda)$.

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



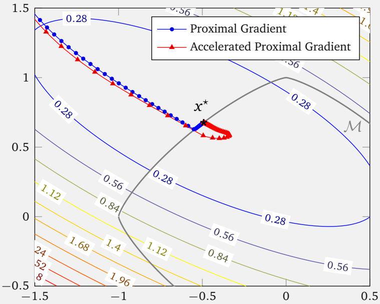
$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$



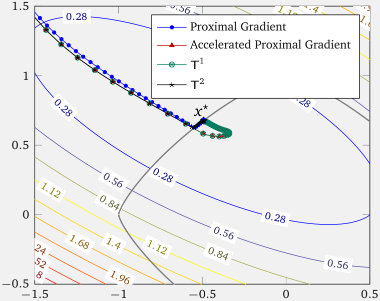


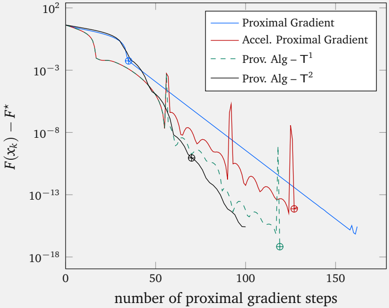
$\oplus$ marks identification time

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



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

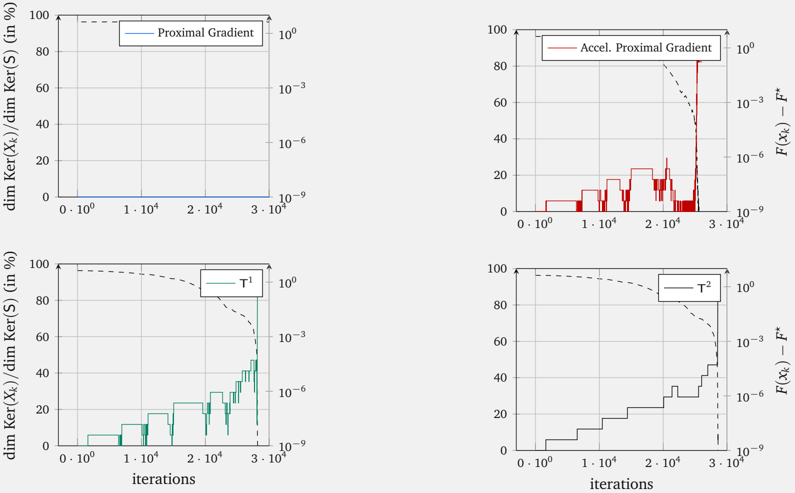




⊕ marks identification time

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

- $\mathsf{S} \in R^{20\times 20}$ 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
  - ⇒ Faster convergence does not means faster structure identification
  - ⇒ Accuracy vs. Structure tradeoff for the learning problem

> We propose a method with stable identification behavior, maintaining an accelerated convergence rate

> General ideas:
  - ⇒ keep a list of the possible structures sparsity patterns, rank
  - ⇒ look at their activity at the output of the proximity operator

▷ Bareilles & I.: *On the Interplay between Acceleration and Identification for the Proximal Gradient algorithm*, Computational Optimization and Applications, 2020, `https://arxiv.org/abs/1909.08944`. Try it in Julia on `https://github.com/GillesBareilles/Acceleration-Identification`

Interplay between Acceleration and Identification

- Newton acceleration on identified manifolds

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

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

Recall that when solving a *composite optimization* problem with proximal gradient

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

the proximity operator outputs a *current structure* $\mathcal{M}_k \subset \mathbb{R}^n$ ($x_k \in \mathcal{M}_k$) and *eventually* $\mathcal{M}_k = \mathcal{M}^\star$.

Reminder: Think of $\mathcal{M}_k$ as a sparsity pattern or a rank in matrix regression.

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

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

Recall that when solving a *composite optimization* problem with proximal gradient

$$\begin{cases} \text{Observe } \mathcal{M}_k \text{, then } y_{k+1} & = \text{RiemannianStep}_{f+g}(x_k, \mathcal{M}_k) \\ u_{k+1} & = y_k - \gamma \nabla f(y_k) \\ x_{k+1} & = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases}$$

the proximity operator outputs a *current structure* $\mathcal{M}_k \subset \mathbb{R}^n$ ($x_k \in \mathcal{M}_k$) and *eventually* $\mathcal{M}_k = \mathcal{M}^\star$.

Reminder: Think of $\mathcal{M}_k$ as a sparsity pattern or a rank in matrix regression.

Predictor-Corrector methods: perform a Riemannian step on $\mathcal{M}_k$, then a proximal step to correct the structure, and so on.

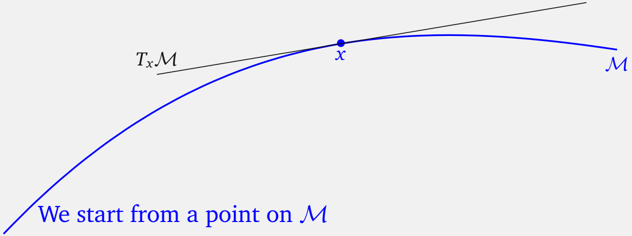
◇ Lemaréchal, Oustry, Sagastizábal: The U-Lagrangian of a convex function. Trans. of the AMS (2000)
◇ Daniilidis, Hare, Malick: Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. Optimization (2006)

- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

  eg. Riemannian gradient step:

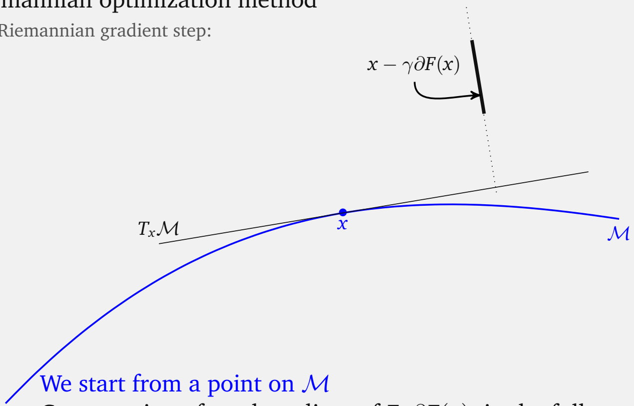
- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

  eg. Riemannian gradient step:



- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

  eg. Riemannian gradient step:

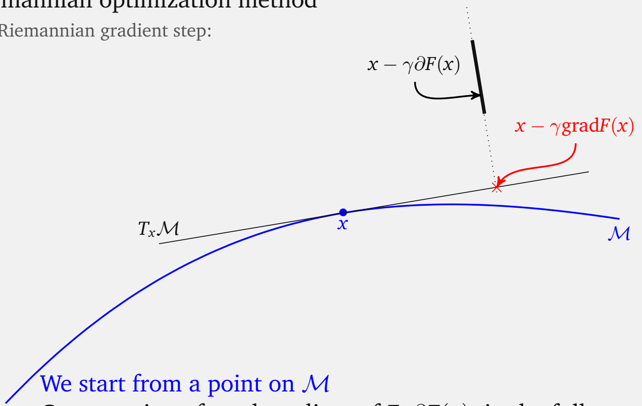


We start from a point on $\mathcal{M}$

Computation of a subgradient of F, $\partial F(x)$, in the full space

- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

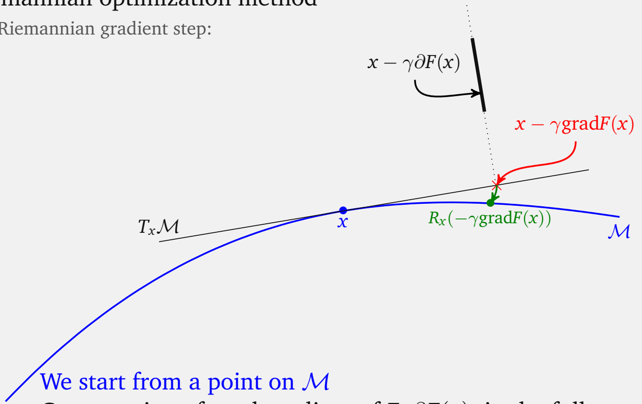
  eg. Riemannian gradient step:



We start from a point on $\mathcal{M}$
Computation of a subgradient of F, $\partial F(x)$, in the full space
Projection on the tangent plane to get a Riemannian *gradient*

- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

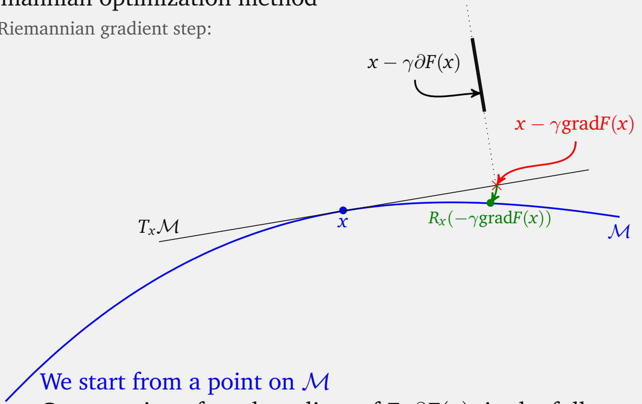
  eg. Riemannian gradient step:



We start from a point on $\mathcal{M}$
Computation of a subgradient of F, $\partial F(x)$, in the full space
Projection on the tangent plane to get a Riemannian *gradient*
Retraction on the manifold to perform a Riemannian gradient step
(Test different γ to decrease F)

- $F = f + g$ is nonsmooth on $\mathbb{R}^n$ but **smooth along $\mathcal{M}$** nonsmooth across
- Riemannian optimization method

  eg. Riemannian gradient step:



We start from a point on $\mathcal{M}$
Computation of a subgradient of F, $\partial F(x)$, in the full space
Projection on the tangent plane to get a Riemannian *gradient*
Retraction on the manifold to perform a Riemannian gradient step

- 1st and 2nd order optimization methods can be implemented on manifolds (see `https://www.manopt.org/` in Matlab, Python, Julia)
- Tractable for linear spaces (sparsity), fixed rank, etc.

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

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

$$\begin{cases} \text{Observe} & \mathcal{M}_k \\ \quad y_{k+1} & = \text{RiemannianNewton}_{f+g}(x_k, \mathcal{M}_k) \\ \quad u_{k+1} & = y_k - \gamma \nabla f(y_k) \\ \quad x_{k+1} & = \mathbf{prox}_{\gamma g}(u_{k+1}) \end{cases}$$

> Intuition from the sparse/ℓ_1 case:

We temporarily restrict to vectors with the same sparsity pattern as x_k

Compute the gradient and Hessian *for these coordinates*

Perform a Newton step possible since it is locally smooth

The proximal gradient step after will ensure the structure validity

Theorem

*Provided that the minimum $x^\star$ lies on some manifold $\mathcal{M}$ and is **qualified**, alternating:*

i) a proximal gradient step with $\gamma < 1/L$

ii) a Riemannian Newton step on the identified manifold *with backtracking line-search*

generates iterates that

a) belong to $\mathcal{M}$ in finite time

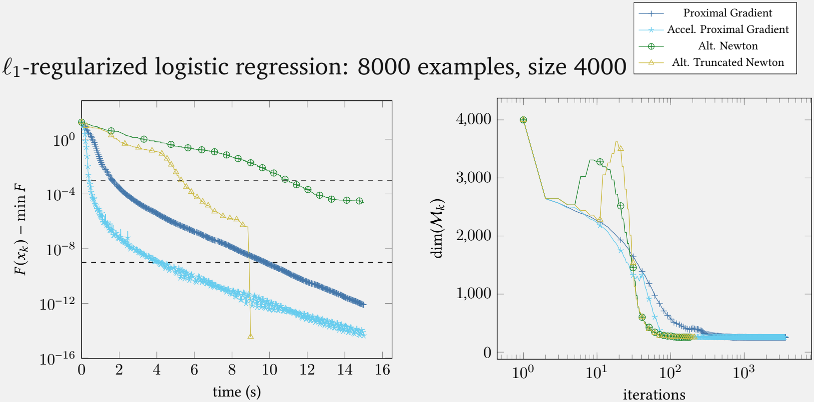
b) converge quadratically to $x^\star$:

$$\mathrm{dist}_{\mathcal{M}}(x_{k+1}, x^\star) \leq \mathrm{dist}_{\mathcal{M}}(x_k, x^\star)^2$$

> Qualification is needed as before for identification...
(QC) + partial smoothness at $x^\star$ for $\mathcal{M}$

> ... and for quadratic convergence of Newton
Riemannian Hessian positive definite at $x^\star$

ℓ_1-regularized logistic regression: 8000 examples, size 4000

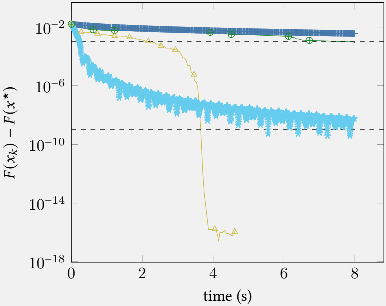


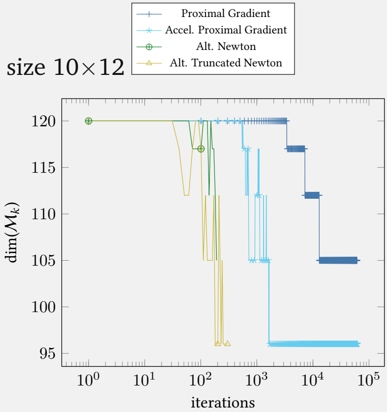
Algorithm	Tolerance	$F(x_k) - \min F$	#prox. grad. steps	# Riemannian steps	$\#HessF(\cdot)[\cdot]$	$\#f$	$\#g$
Prox. Gradient	$1 \cdot 10^{-3}$	0.0009963198229036019	357	–	–	779	358
Prox. Gradient	$1 \cdot 10^{-9}$	9.965078207052613e-10	2306	–	–	4677	2307
Accel. Prox. Gradient	$1 \cdot 10^{-3}$	0.0009257766624239938	90	–	–	246	91
Accel. Prox. Gradient	$1 \cdot 10^{-9}$	9.899422392933843e-10	953	–	–	1972	954
Alt. Newton	$1 \cdot 10^{-3}$	0.0009759231753842523	62	61	6303	556	427
Alt. Newton	$1 \cdot 10^{-9}$	–	–	–	–	–	–
Alt. Truncated Newton	$1 \cdot 10^{-3}$	0.0009557819627238895	51	50	2616	437	321
Alt. Truncated Newton	$1 \cdot 10^{-9}$	3.774758283725532e-15	105	105	5091	742	572

> Newton is too costly without a low dimensional structure

> Truncated Newton offers a good compromise approximate Newton equation

low rank matrix regression: 60 matrices, size 10×12





Algorithm	Tolerance	$F(x_k) - \min F$	#prox. grad. steps	# Riemannian steps	$\#HessF(\cdot)[\cdot]$	$\#f$	$\#g$
Prox. Gradient	$1 \cdot 10^{-3}$	–	–	–	–	–	–
Prox. Gradient	$1 \cdot 10^{-9}$	–	–	–	–	–	–
Accel. Prox. Gradient	$1 \cdot 10^{-3}$	0.00099894916795637	1489	–	–	3073	1490
Accel. Prox. Gradient	$1 \cdot 10^{-9}$	9.858174276899945e-10	43283	–	–	86661	43284
Alt. Newton	$1 \cdot 10^{-3}$	0.0009833250032105778	93	93	28063	873	687
Alt. Newton	$1 \cdot 10^{-9}$	–	–	–	–	–	–
Alt. Truncated Newton	$1 \cdot 10^{-3}$	0.0009695009931029591	76	76	16342	738	568
Alt. Truncated Newton	$1 \cdot 10^{-9}$	2.2716245551279712e-11	128	128	27786	1101	879

> Stable structure identification & much less iterative algorithm

- The structure of some composite optimization problems can be harnessed by Riemannian methods

  Thanks to the local smooth along the structure manifold
- Proximal steps have to be intertwined to ensure identification

  Prox. grad. = identification step – Riemannian Newton = efficent step
- Non-convex regularizations can work

  you may use ℓ_0 semi norm, rank for a matrix

▷ Bareilles, I., Malick: *Newton acceleration on manifolds identified by proximal-gradient methods*, `https://arxiv.org/abs/2012.12936`

- Machine Learning problems often have a *noticeable structure*;
  sparsity, low rank
- This structure is identified progressively by *proximal methods*;
  + CD, Var. Red., Distributed methods, etc.
- For most problem, we *do not know* if the identified structure is optimal;
  adaptivity is key
- Nevertheless, it can be used to boost numerical performance;
  low complexity model
- *Structure vs. Optimality* tradeoff in Optimization for ML.
  structure is better than overfitting

▷ I., Malick: *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications*, Set Valued & Variational Analysis, 2020, https://arxiv.org/abs/2010.00848

▷ Bareilles, I.: *On the Interplay between Acceleration and Identification for the Proximal Gradient algorithm*, Computation Optimization and Applications, 2020, https://arxiv.org/abs/1909.08944.

▷ Bareilles, I., Malick: *Newton acceleration on manifolds identified by proximal-gradient methods*, https://arxiv.org/abs/2012.12936

Thanks to ANR JCJC *STROLL*

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