Harnessing the Structure of some Optimization Problems

Habilitation defense

Franck Iutzeler

December 15th, 2021

Univ. Grenoble Alpes

Introduction

Musée des Beaux-Arts et d'Archéologie de Besançon

Mosaic of Neptune (IInd century)

Introduction

Ph.D. in 2013
Optimization on graphs

Post-docs in Supélec
& Louvain-la-Neuve

Since Sept. 2015
Assistant Professor at UGA

- Research interests:
  - Numerical Optimization
  - Machine Learning
  - Multi-agent systems
- In this defense:
  - Selection of works from 2018-2021
  - In collaboration with 5 PhD & master students
  - Current interests and perspectives for future research

Harnessing the Structure of some Optimization Problems

Hieronymus Bosch

The Garden of Earthly Delights, open (1490-1500)

- In Data Science, one seeks a model that fits the observed data parametrized model P_x, data $\{\mathbf{a}_j, b_j\}_{j=1}^m$, loss ℓ while ensuring some structure on the parameter/model for generalization and stability regularizer Ω

central thread of this presentation

Regularized
Emprirical Risk Minimization

$$\min_{x \in \mathbb{R}^n} \quad \underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(b_i, P_x(\mathbf{a}_i))}_{\substack{=:f(x) \\ \text{minimizes the risk}}} \quad + \quad \underbrace{\lambda\, \Omega(x)}_{\substack{=:g(x) \\ \text{regularizes the solutions}}}$$

Mathematical Optimization in Data Science

- In Data Science, one seeks a model that fits the observed data parametrized model P_x, data $\{\mathbf{a}_j, b_j\}_{j=1}^m$, loss ℓ
  while ensuring some structure on the parameter/model for generalization and stability regularizer Ω

central thread of this presentation

Regularized
Emprirical Risk Minimization

$$\min_{x\in\mathbb{R}^n} \quad \underbrace{\frac{1}{m}\sum_{i=1}^m \ell(b_i, P_x(\mathbf{a}_i))}_{=:f(x)} \quad + \quad \underbrace{\lambda\,\Omega(x)}_{=:g(x)}$$

minimizes the risk

regularizes the solutions

Brings valuable structure – Part A

- In Data Science, one seeks a model that fits the observed data parametrized model P_x, data $\{\mathbf{a}_j, b_j\}_{j=1}^m$, loss ℓ while ensuring some structure on the parameter/model for generalization and stability regularizer Ω

central thread of this presentation

Computations can be split – Part B

Regularized
Emprirical Risk Minimization

$$\min_{x \in \mathbb{R}^n} \quad \underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(b_i, P_x(\mathbf{a}_i))}_{=: f(x)} \quad + \quad \underbrace{\lambda\, \Omega(x)}_{=: g(x)}$$

minimizes the risk

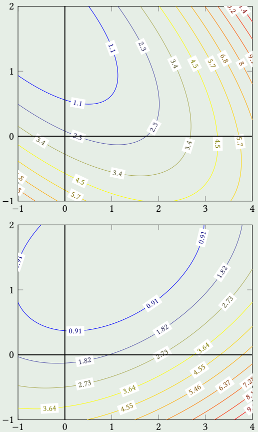
regularizes the solutions

Brings valuable structure – Part A

Harnessing the Structure of some Optimization Problems

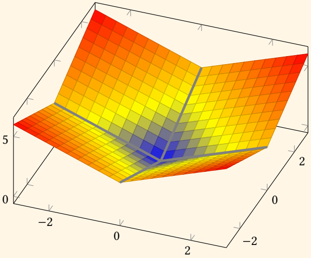
A – Structure Identification in Data Science

Quadratic risk



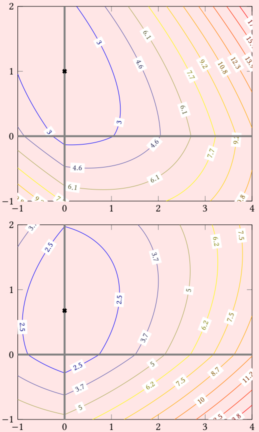
+

ℓ_1 norm



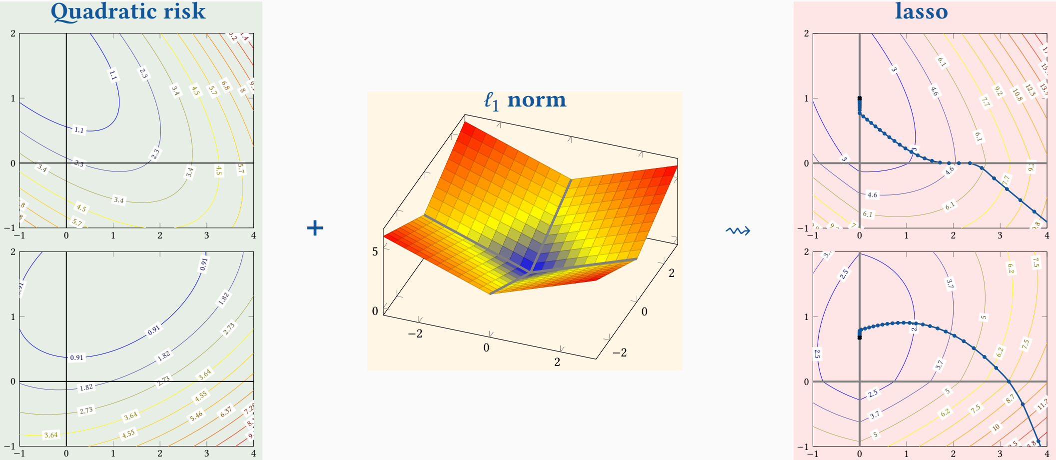
$\rightsquigarrow$

lasso



- Non-differentiability patterns of regularizers
  - can trap the problems' solutions small changes in the data may not change the sparsity structure

Structure & Identification for the lasso



- ▹ Non-differentiability patterns of regularizers
  - ◇ can trap the problems' solutions small changes in the data may not change the sparsity structure
  - ◇ attract the iterates of some optimization methods but not all

Optimization for the lasso

Quadratic risk

$$\frac{1}{m}\sum_{j=1}^{m}\Big(\langle x, \mathbf{a}_j\rangle - b_j\Big)^2$$

smooth function f

but **costly** to evaluate

+

ℓ_1 norm

$$\lambda\|x\|_1$$

nonsmooth function g

but **simple** to minimize

$\rightsquigarrow$

lasso

$$\frac{1}{m}\sum_{j=1}^{m}\Big(\langle x, \mathbf{a}_j\rangle - b_j\Big)^2 + \lambda\|x\|_1$$

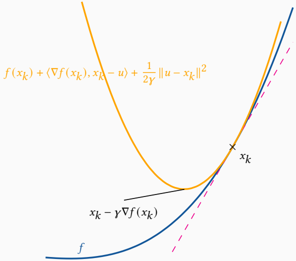
composite function $f + g$

To minimize $f + g$

- Iteratively approximate f by a quadratic smoothness

$$x_{k+1} = \operatorname{argmin}_u \left\{ f(x_k) + \langle \nabla f(x_k), u - x_k\rangle + \frac{1}{2\gamma}\|u - x_k\|^2 + g(x) \right\}$$

$$= \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\Big\|u - \big(x_k - \gamma\nabla f(x_k)\big)\Big\|^2 \right\}$$



◇ Tibshirani: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (1996)
◇ Bach, Jenatton, Mairal, Obozinski: Optimization with sparsity-inducing penalties. Foundation and Trends in Machine Learning (2012)

Optimization for the lasso

Quadratic risk

$$\frac{1}{m}\sum_{j=1}^{m}\Big(\langle x, \mathbf{a}_j\rangle - b_j\Big)^2$$

smooth function f

but **costly** to evaluate

$+$

ℓ_1 norm

$$\lambda\|x\|_1$$

nonsmooth function g

but **simple** to minimize

$\rightsquigarrow$

lasso

$$\frac{1}{m}\sum_{j=1}^{m}\Big(\langle x, \mathbf{a}_j\rangle - b_j\Big)^2 + \lambda\|x\|_1$$

composite function $f + g$

To minimize $f + g$

Proximal gradient ISTA

- Iteratively approximate f by a quadratic smoothness
- Use the proximity operator of g prox simple

$$\begin{aligned} x_{k+1} &= \operatorname{argmin}_u \left\{ f(x_k) + \langle \nabla f(x_k), u - x_k\rangle + \frac{1}{2\gamma}\|u - x_k\|^2 + g(x)\right\} \\ &= \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\Big\|u - \big(x_k - \gamma\nabla f(x_k)\big)\Big\|^2\right\} \\ &= \mathbf{prox}_{\gamma g}\big(x_k - \gamma\nabla f(x_k)\big) \end{aligned}$$

$$\mathbf{prox}_{\gamma g}(y) := \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\|u - y\|^2\right\}$$

for the ℓ_1 norm: *soft-thresholding* per coordinate

$$\mathbf{prox}^{[i]}_{\gamma\lambda\|\cdot\|_1}(y) = \begin{cases} y^{[i]} + \gamma\lambda & \text{if} \quad y^{[i]} < -\gamma\lambda \\ 0 & \text{if} \quad -\gamma\lambda \le y^{[i]} \le \gamma\lambda \\ y^{[i]} - \gamma\lambda & \text{if} \quad y^{[i]} > \gamma\lambda \end{cases}$$

- Tibshirani: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (1996)
- Bach, Jenatton, Mairal, Obozinski: Optimization with sparsity-inducing penalties. Foundation and Trends in Machine Learning (2012)

Proximal Identification for the lasso

Solving the lasso problem $\min_{x\in\mathbb{R}^n} \overbrace{\frac{1}{m}\sum_{j=1}^{m}\left(\langle x, \mathbf{a}_j\rangle - b_j\right)^2}^{f(x)} + \overbrace{\lambda\|x\|_1}^{g(x)}$

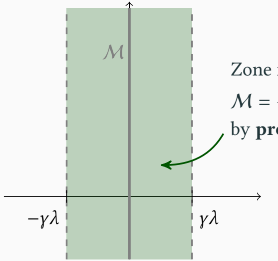
by proximal gradient $x_{k+1} = \mathbf{prox}_{\gamma g}\left(x_k - \gamma\nabla f(x_k)\right)$

$$\mathbf{prox}_{\gamma g}(y) := \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\|u-y\|^2 \right\}$$

for the ℓ_1 norm: *soft-thresholding* per coordinate

$$\mathbf{prox}^{[i]}_{\gamma\lambda\|\cdot\|_1}(y) = \begin{cases} y^{[i]} + \gamma\lambda & \text{if} \quad y^{[i]} < -\gamma\lambda \\ 0 & \text{if} \quad -\gamma\lambda \le y^{[i]} \le \gamma\lambda \\ y^{[i]} - \gamma\lambda & \text{if} \quad y^{[i]} > \gamma\lambda \end{cases}$$

The sequence (x_k) converges to a solution $x^\star$.



Zone mapped to
$\mathcal{M} = \{x \in \mathbb{R}^n : x_i = 0\}$
by $\mathbf{prox}_{\gamma\lambda\|\cdot\|_1}$

◇ Liang, Fadili, Peyré: Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. SIAM Journal on Optimization (2017)

Proximal Identification for the lasso

Solving the lasso problem $\min_{x\in\mathbb{R}^n} \overbrace{\frac{1}{m}\sum_{j=1}^{m}\left(\langle x, \mathbf{a}_j\rangle - b_j\right)^2}^{f(x)} + \overbrace{\lambda\|x\|_1}^{g(x)}$

by proximal gradient $x_{k+1} = \mathbf{prox}_{\gamma g}\left(x_k - \gamma\nabla f(x_k)\right)$

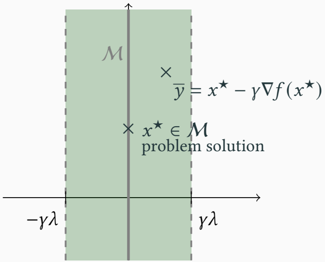
$$\mathbf{prox}_{\gamma g}(y) := \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\|u - y\|^2 \right\}$$

for the ℓ_1 norm: *soft-thresholding* per coordinate

$$\mathbf{prox}^{[i]}_{\gamma\lambda\|\cdot\|_1}(y) = \begin{cases} y^{[i]} + \gamma\lambda & \text{if} \quad y^{[i]} < -\gamma\lambda \\ 0 & \text{if} \quad -\gamma\lambda \le y^{[i]} \le \gamma\lambda \\ y^{[i]} - \gamma\lambda & \text{if} \quad y^{[i]} > \gamma\lambda \end{cases}$$

The sequence (x_k) converges to a solution $x^\star$.

If $x^\star \in \mathcal{M}$



◇ Liang, Fadili, Peyré: Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. SIAM Journal on Optimization (2017)

Proximal Identification for the lasso

Solving the lasso problem $\min_{x\in\mathbb{R}^n} \overbrace{\frac{1}{m}\sum_{j=1}^m \left(\langle x, \mathbf{a}_j\rangle - b_j\right)^2}^{f(x)} + \overbrace{\lambda\|x\|_1}^{g(x)}$

by proximal gradient $x_{k+1} = \mathbf{prox}_{\gamma g}\left(x_k - \gamma\nabla f(x_k)\right)$

$$\mathbf{prox}_{\gamma g}(y) := \operatorname{argmin}_u \left\{g(u) + \frac{1}{2\gamma}\|u-y\|^2\right\}$$

for the ℓ_1 norm: *soft-thresholding* per coordinate

$$\mathbf{prox}^{[i]}_{\gamma\lambda\|\cdot\|_1}(y) = \begin{cases} y^{[i]} + \gamma\lambda & \text{if} \quad y^{[i]} < -\gamma\lambda \\ 0 & \text{if} \quad -\gamma\lambda \le y^{[i]} \le \gamma\lambda \\ y^{[i]} - \gamma\lambda & \text{if} \quad y^{[i]} > \gamma\lambda \end{cases}$$

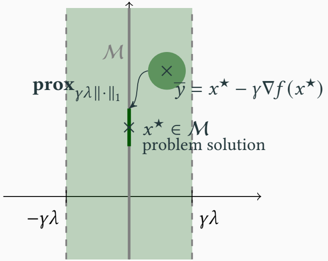
The sequence (x_k) converges to a solution $x^\star$.

If $x^\star \in \mathcal{M}$ and a **Qualifying Condition** holds

$\bar{y}$ is in the relative interior of the green zone

$\Leftrightarrow \nabla^{[i]} f(x^\star) \in (-\lambda, \lambda)$

$\Leftrightarrow \frac{1}{m}\sum_{j=1}^m \mathbf{a}_j^{[i]}(\langle x^\star, \mathbf{a}_j\rangle - b_j) \in (-\lambda, \lambda)$



◇ Liang, Fadili, Peyré: Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. SIAM Journal on Optimization (2017)

Proximal Identification for the lasso

Solving the lasso problem $\min_{x\in\mathbb{R}^n} \overbrace{\frac{1}{m}\sum_{j=1}^{m}\left(\langle x, \mathbf{a}_j\rangle - b_j\right)^2}^{f(x)} + \overbrace{\lambda\|x\|_1}^{g(x)}$

by proximal gradient $x_{k+1} = \mathbf{prox}_{\gamma g}\left(x_k - \gamma \nabla f(x_k)\right)$

$$\mathbf{prox}_{\gamma g}(y) := \operatorname{argmin}_u \left\{ g(u) + \frac{1}{2\gamma}\|u - y\|^2 \right\}$$

for the ℓ_1 norm: *soft-thresholding* per coordinate

$$\mathbf{prox}^{[i]}_{\gamma\lambda\|\cdot\|_1}(y) = \begin{cases} y^{[i]} + \gamma\lambda & \text{if} \quad y^{[i]} < -\gamma\lambda \\ 0 & \text{if} \quad -\gamma\lambda \le y^{[i]} \le \gamma\lambda \\ y^{[i]} - \gamma\lambda & \text{if} \quad y^{[i]} > \gamma\lambda \end{cases}$$

The sequence (x_k) converges to a solution $x^\star$.

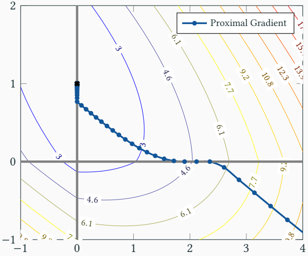
If $x^\star \in \mathcal{M}$ and a **Qualifying Condition** holds

$\bar{y}$ is in the relative interior of the green zone

$\Leftrightarrow \nabla^{[i]} f(x^\star) \in (-\lambda, \lambda)$

$\Leftrightarrow \frac{1}{m}\sum_{j=1}^{m} \mathbf{a}_j^{[i]}(\langle x^\star, \mathbf{a}_j\rangle - b_j) \in (-\lambda, \lambda)$

Then, the iterates belong to $\mathcal{M}$ in finite time.



◇ Liang, Fadili, Peyré: Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. SIAM Journal on Optimization (2017)

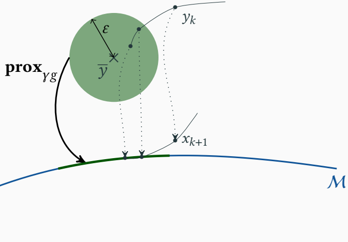
Proximal Identification Theory

Finite-time identification holds for a *proximal method* as long as

- the iterates are well-defined and **converge** $y_k \to \overline{y}$
- g is **nonsmooth across** the smooth structure manifold but **smooth along** it
- some **Qualifying Condition** (QC) holds

$$\begin{cases} y_k &= \dots \\ x_{k+1} &= \mathbf{prox}_{\gamma g}(y_k) \end{cases}$$

$$\exists\, \varepsilon > 0 \text{ such that for all } y \in \mathcal{B}(\overline{y}, \varepsilon),\ \ \mathbf{prox}_g(y) \in \mathcal{M} \qquad \text{(QC)}$$



- Well-grounded theory in nonsmooth analysis partial smoothness, nonconvex proximal methods

- Lewis: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization (2002)
- 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)
- Vaiter, Peyré, Fadili: Model consistency of partly smooth regularizers. IEEE Trans. on Information Theory (2017)
- Fadili, Malick, Peyré: Sensitivity analysis for mirror-stratifiable convex functions. SIAM Journal on Optimization (2018)

Proximal Identification in Data Science

- In Data Science problems, the regularizer is often *chosen* to have
  - an **explicit** proximity operator proximal methods are possible
  - which is also a **structure oracle** the structure of the output is known

ℓ_1 norm $\rightsquigarrow$ sparsity

```
def prox_g(x,gamma):
    p = np.zeros(n)
    for i in range(n):
        if x[i] > gamma:
            p[i] = x[i] - gamma
        elif x[i] < - gamma:
            p[i] = x[i] + gamma
    return p
```

nuclear norm $\rightsquigarrow$ low-rank

soft thresholding the singular values

1D total variation $\rightsquigarrow$ change sparsity

dynamic programming

Finite time Identification and Current structure *proximity operator* but we never know if the structure is final

I, Malick: *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications*, Set-Valued and Variational Analysis, 2020.

- Does faster minimization means faster identification?
- Can we leverage the current structure numerically?

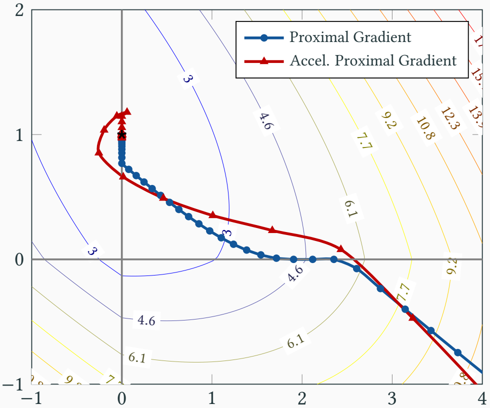
Harnessing the Structure
of some Optimization Problems

A – Structure Identification in Data Science

▹ **Does faster minimization means faster identification?**

Interplay between Acceleration and Identification

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



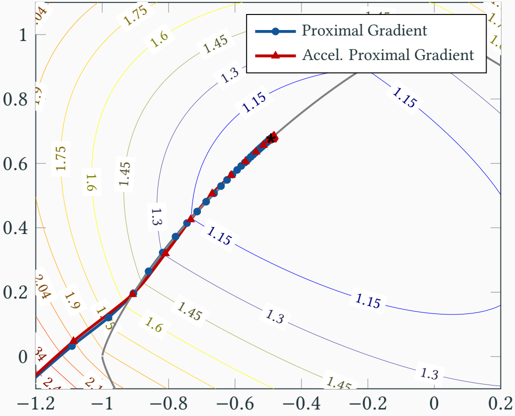
Accelerated proximal gradient (FISTA)

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

- ▷ faster in practice and worst case rates
- ✓ *exploratory* behavior
- ✗ overshooting

Interplay between Acceleration and Identification

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



Accelerated proximal gradient (FISTA)

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

- ▹ faster in practice and worst case rates
- ✓ *exploratory* behavior
- ✗ overshooting
- ✗ misfit to curved structure

Structure-adapted acceleration

Idea Pre-define a *collection* $\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\}$ of *sought structures* eg. sparsity patterns, rank, constraint activity

and condition the acceleration to a *structure test*
$$\begin{cases} y_k = \begin{cases} x_k & \text{if } \mathsf{T}_k = 0 \\ x_k + \alpha_k (x_k - x_{k-1}) & \text{otherwise} \end{cases} \\ x_{k+1} = \mathbf{prox}_{\gamma g}(y_k - \gamma \nabla f(y_k)) \end{cases}$$

T^1: **counter overshooting**

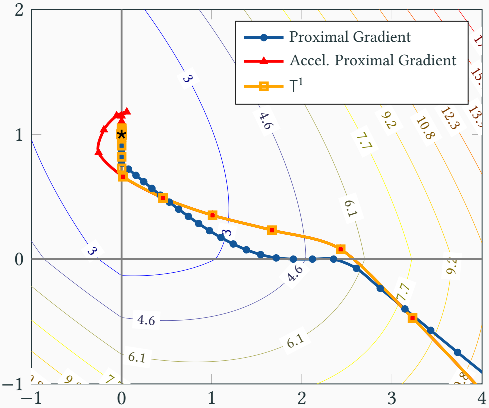
$$\mathsf{T}^1_k = 0 \quad \text{(no acceleration)} \quad \text{if } \begin{cases} x_k \in \mathcal{M} \\ x_{k-1} \notin \mathcal{M} \end{cases} \text{ for some } \mathcal{M} \in \mathcal{C}$$

Theorem The accelerated rate $\mathcal{O}(1/k^2)$ is maintained if the qualification condition (QC) holds.

Bareilles & **I**: *On the Interplay between Acceleration and Identification for the Proximal Gradient algorithm*, Computational Optimization and Applications, 2020

Effect in practice

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

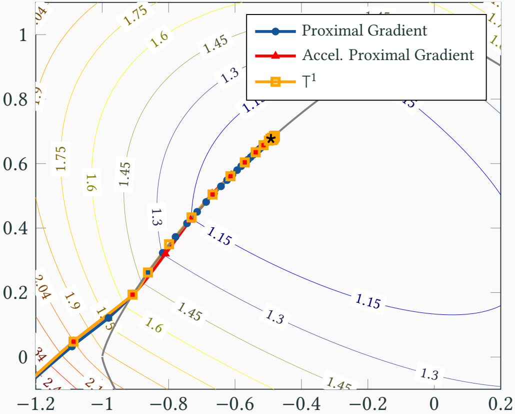


Conditioning acceleration to structure test T^1

- no overshooting
- similar suboptimality
- structure is more stable

Effect in practice

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



Conditioning acceleration to structure test T^1

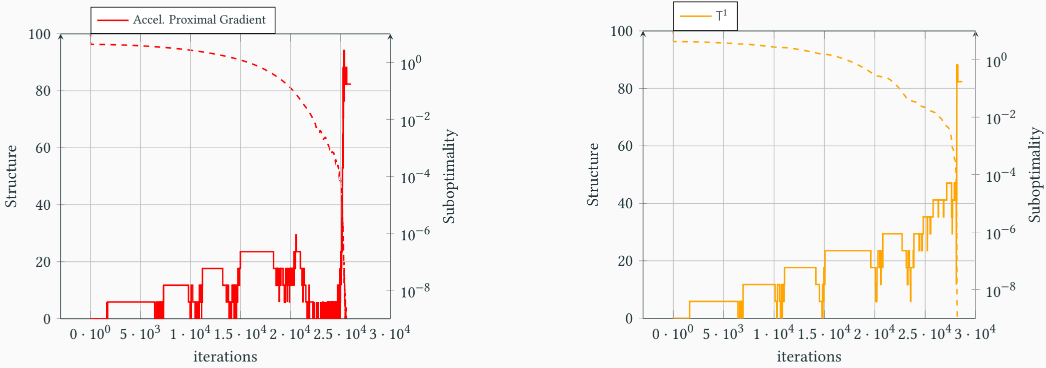
- no overshooting
- similar suboptimality
- structure is more stable

Why is structure information important?

Low-rank matrix regression
$Y = AS + E$ with $\mathrm{rank}(S) = 3$

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

- Structure (plain line), recovered percentage
  $0\% \rightsquigarrow \mathrm{rank}(x_k) = 20$
  $100\% \rightsquigarrow \mathrm{rank}(x_k) = \mathrm{rank}(S) = 3$
- Suboptimality (dashed)



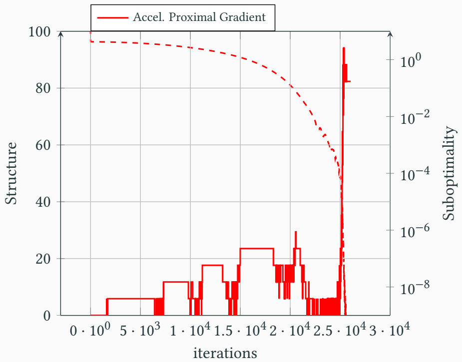
- Does faster minimization means faster identification ?
  - Not always, **but** a compromised can be reached by structure-aware acceleration
  - Valuable structure can be completely lost even if the suboptimality is low

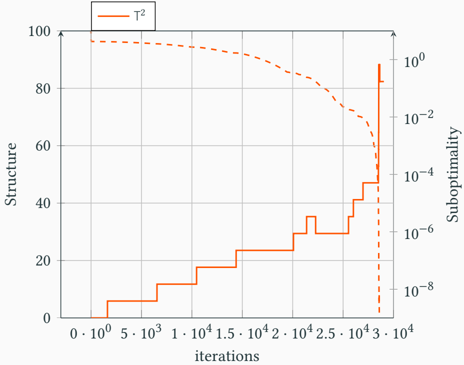
Why is structure information important?

Low-rank matrix regression
$Y = AS + E$ with $\mathrm{rank}(S) = 3$

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

- Structure (plain line), recovered percentage
  $0\% \rightsquigarrow \mathrm{rank}(x_k) = 20$
  $100\% \rightsquigarrow \mathrm{rank}(x_k) = \mathrm{rank}(S) = 3$
- Suboptimality (dashed)





- Does faster minimization means faster identification ?
  - Not always, **but** a compromised can be reached by structure-aware acceleration
  - Valuable structure can be completely lost even if the suboptimality is low

Harnessing the Structure
of some Optimization Problems

A – Structure Identification in Data Science

▹ **Can we leverage the current structure numerically?**

Leveraging the structure numerically

$$
\begin{aligned}
y_k &= x_k - \gamma \nabla f(x_k) && \text{gradient step} \rightsquigarrow \text{most of the computational cost} \\
x_{k+1} &= \mathbf{prox}_{\gamma g}(y_k) && \text{proximity operator} \rightsquigarrow \text{gives structure}
\end{aligned}
$$

- Proximal gradient identifies structure but does not use it

Leveraging the structure numerically

Observe $\mathcal{S}_k = \bigcap_{i : x_k \in \mathcal{M}_i} \mathcal{M}_i$

$$
\begin{aligned}
y_k &= \mathrm{proj}_{\mathcal{S}_k}\left(x_k - \gamma \nabla f(x_k) \right) + \mathrm{proj}^{\perp}_{\mathcal{S}_k}(y_{k-1}) && \text{gradient step} \rightsquigarrow \text{most of the computational cost} \\
x_{k+1} &= \mathbf{prox}_{\gamma g}(y_k) && \text{proximity operator} \rightsquigarrow \text{gives structure}
\end{aligned}
$$

Idea Project using the output of $\mathbf{prox}_{\gamma g}$ and the pre-defined *collection* $\mathsf{C} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\}$

- For sparsity patterns: $\mathcal{M}_i = \{x \in \mathbb{R}^n : x^{[i]} = 0\}$ and $\mathcal{S}_k = \{x \in \mathbb{R}^n : \mathrm{supp}(x) = \mathrm{supp}(x_k)\}$
- Direct use of the structure fails No correctness guarantee, contrary to screening methods

◇ Ndiaye, Fercoq, Gramfort, Salmon: Gap-safe screening rules for sparsity enforcing penalties. Journal of Machine Learning Research (2017)

Leveraging the structure numerically

Observe $\mathcal{S}_k = \bigcap_{i:x_k\in\mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n)$ for $\xi_{k,i} \sim \mathcal{B}(p)$ **additional randomness**

$$
\begin{aligned}
y_k &= \mathrm{proj}_{\mathcal{S}_k}(\quad x_k - \gamma\nabla f(x_k) \quad) + \mathrm{proj}^{\perp}_{\mathcal{S}_k}(y_{k-1}) && \text{gradient step} \rightsquigarrow \text{most of the computational cost}\\
x_{k+1} &= \mathbf{prox}_{\gamma g}(\quad y_k) && \text{proximity operator} \rightsquigarrow \text{gives structure}
\end{aligned}
$$

Idea Project on a **random** space comprising the current structure so that the whole space is spanned

- For sparsity patterns: sort of “coordinate descent” on the support + random ones
- Mixing randomized coordinate descent with identification induces a biais convergence issues

◇ Friedman, Hastie, Tibshirani: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software (2010)
◇ Massias, Gramfort, Salmon: Celer: a Fast Solver for the Lasso with Dual Extrapolation. ICML (2018)

Leveraging the structure numerically

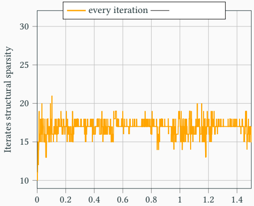
Observe $\mathcal{S}_k = \bigcap_{i: x_k \in \mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n)$ for $\xi_{k,i} \sim \mathcal{B}(p)$

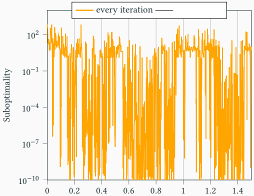
and compute $\mathbf{P}_k = \mathbb{E}\,\mathrm{proj}_{\mathcal{S}_k}$ and $\mathbf{Q}_k = (\mathbf{P}_k)^{-1/2}$

$$y_k = \mathrm{proj}_{\mathcal{S}_k}\left(\mathbf{Q}_k\left(x_k - \gamma\nabla f(x_k)\right)\right) + \mathrm{proj}^{\perp}_{\mathcal{S}_k}(y_{k-1})$$ gradient step ⇝ most of the computational cost

$$x_{k+1} = \mathbf{prox}_{\gamma g}(\mathbf{Q}_k^{-1} y_k)$$ proximity operator ⇝ gives structure

- ▹ We restrict ourselves to **affine subspaces** ℓ_1/ℓ_2-group lasso, 1D TV-fused lasso, g may not be separable
- ▹ Unbiasing with $\mathbf{Q}_k$ works *after identification* but not before which prevents identification...





Leveraging the structure numerically

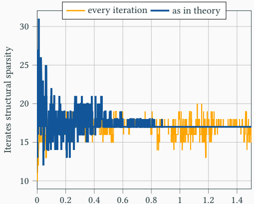
Observe $\mathcal{S}_k = \bigcap_{i:\mathbf{x}_\ell \in \mathcal{M}_i} (\xi_{k,i}\mathcal{M}_i + (1-\xi_{k,i})\mathbb{R}^n)$ for $\xi_{k,i} \sim \mathcal{B}(p)$

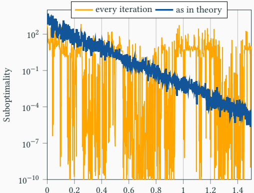
and compute $\mathbf{P}_k = \mathbb{E}\,\mathrm{proj}_{\mathcal{S}_k}$ and $\mathbf{Q}_k = (\mathbf{P}_k)^{-1/2}$ **the reference point $\mathbf{x}_\ell$ only changes *if possible***

$$y_k = \mathrm{proj}_{\mathcal{S}_k}\left(\mathbf{Q}_k\left(x_k - \gamma\nabla f(x_k)\right)\right) + \mathrm{proj}^{\perp}_{\mathcal{S}_k}(y_{k-1})$$ gradient step ⇝ most of the computational cost

$$x_{k+1} = \mathbf{prox}_{\gamma g}(\mathbf{Q}_k^{-1} y_k)$$ proximity operator ⇝ gives structure

- Structure adaptation can be performed only at **some** iterations
- The *amount of change* $\|\mathbf{Q}_{k-1}\mathbf{Q}_k^{-1}\|$ and *harshness* of the sparsification $\lambda_{\min}(\mathbf{Q}_k)$ has to be tampered





Convergence result for strongly convex problems

Theorem There is an explicit adaptation strategy such that the iterates of the previous method satisfy

$$\mathbb{E}\,\|x_k - x^\star\|^2 = \mathcal{O}\left(\left(1 - \lambda\frac{\gamma\mu L}{\mu + L}\right)^{a_k}\right)$$

where a_k is the number of *adaptations* performed before k and $\lambda = \inf_k \lambda_{\min}(\mathbb{E}\,\mathrm{proj}_{\mathcal{S}_k})$.
Furthermore, if the qualifying constraint (QC) holds, finite-time identification happens and the rate improves

$$\|x_k - x^\star\|^2 = \mathcal{O}_{\mathbb{P}}\left(\left(1 - 2\lambda_{\min}(\mathbb{E}\,\mathrm{proj}_{\mathcal{S}^\star})\frac{\gamma\mu L}{\mu + L}\right)^{k}\right).$$

Example for sparsity patterns: We sample s coordinates at random outside of the support.
If $k = k_{\ell-1}$ is an adaptation time, the current support can be used after

$$\mathbf{c}_\ell = \left\lceil \frac{\log\left(\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2\right) + \log(1/(1 - 2\gamma\mu L/(n(\mu+L))))}{\log(1/(1 - 2s\gamma\mu L/(\mathrm{null}(x_{\ell-1})(\mu+L))))} \right\rceil \text{ iterations}$$

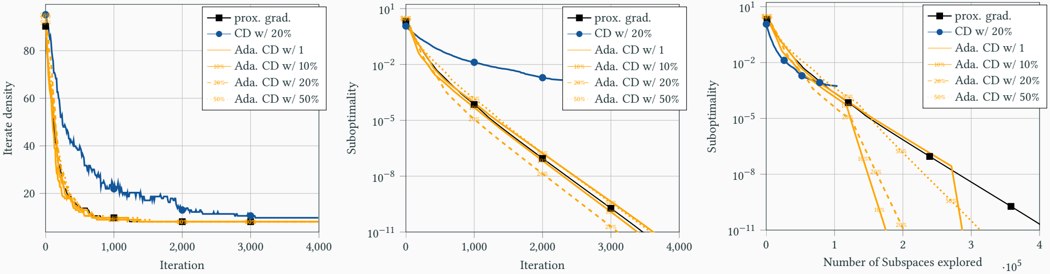
Grishchenko, **I**, Malick: *Proximal Gradient Methods with Adaptive Subspace Sampling*, Mathematics of Operations Research, 2021

Numerical illustration

logistic regression with 1D total variation regularization

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{j=1}^{m} \log\left(1+\exp\left(-b_j \mathbf{a}_j^\top x\right)\right) + \frac{\lambda_2}{2}\|x\|_2^2 + \lambda \mathrm{TV}(x)$$

- $n = 123$
- the solutions has 13 jumps



- Can we leverage the current structure numerically ?
  - For coordinate descent methods and with affine structures, the sampling strategy can be adapted to the uncovered structure

Harnessing the Structure
of some Optimization Problems

B – Distributed Structure & Asynchrony

Computations can be split – Part B

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{j=1}^{m} \ell(b_j, P_x(\mathbf{a}_j)) + \lambda\,\Omega(x) \rightsquigarrow \min_{x\in\mathbb{R}^n} \frac{1}{M}\sum_{i=1}^{M} f^i(x) + g(x)$$

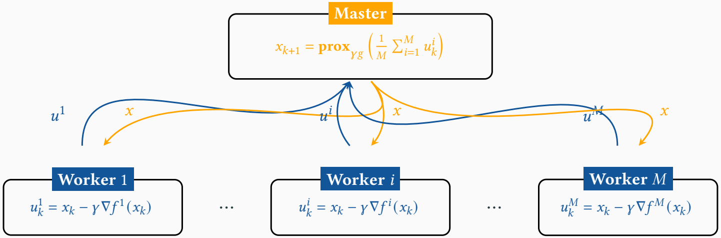
Local data/loss

$$f^i(x) = \frac{M}{m}\sum_{j\in\mathcal{D}^i} \ell(b_j, P_x(\mathbf{a}_j))$$

Common regularizer

$$g(x) = \frac{\lambda}{M}\Omega(x)$$

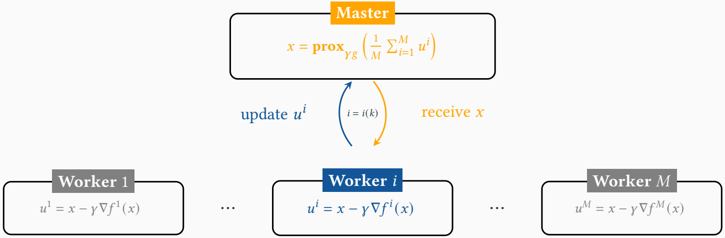
M workers + Coordinator



- Distributed Proximal Gradient leads to **communication bottlenecks**
- We can implement **asynchronous *send/receive*** with the coordinator

  using the MPI standard in Python/C/C++, or the Channel objects in Julia along with the Distributed library

Asynchronous Proximal Gradient



- Proximal gradient with **asynchronous** communications... with no further assumptions on the system

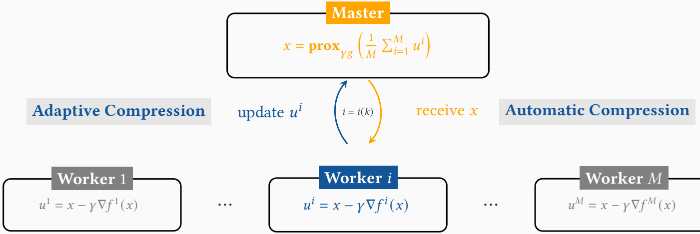
Idea Introduction of an *epoch sequence*: $k_{\ell+1} = \min\{k : \text{ each machine made at least 2 updates in } [k_\ell, k]\}$

Showing that $\max_{k\in[k_\ell,k_{\ell+1})} \left\|\overline{x}_k - \overline{x}^\star\right\|^2 \leq (1-\beta)^2 \max_{k'\in[k_{\ell-1},k_\ell)} \left\|\overline{x}_{k'} - \overline{x}^\star\right\|^2$

Mishchenko, **I**, Malick, Amini: *A delay-tolerant proximal-gradient algorithm for distributed learning*, ICML, 2018
—, —, —: *A Distributed Flexible Delay-tolerant Proximal Gradient Algorithm*, SIAM Journal on Optimization, 2020

◇ Bertsekas, Tsitsiklis. Parallel and distributed computation: numerical methods. (2015)

Asynchronous Proximal Gradient



- Proximal gradient with **asynchronous** communications... with no further assumptions on the system
- ... that can be further **sparsified** using **identification** for ℓ_1 regularization

Idea Coordinate descent as presented before only works for well-conditioned problems due to asynchronicity

Iterative reconditionning *à la* Catalyst

Grishchenko, **I**, Malick, Amini: *Distributed Learning with Sparse Communications by Identification*, SIAM Journal on Mathematics of Data Science, 2021

◇ Lin, Mairal, Harchaoui: Catalyst acceleration for first-order convex optimization: from theory to practice. Journal of Machine Learning Research (2018)

Two Current Directions & Perspectives

Pierre Soulages

Peinture 222 x 314 cm, 24 février 2008

Two Current Directions & Perspectives

α – Structure Identification Cont'

Identification of Smooth Manifolds

$$\min_{x\in\mathbb{R}^n} F(x) = \underset{\text{smooth}}{\overset{\text{risk}}{f(x)}} + \underset{\text{nonsmooth}}{\overset{\text{regularization}}{g(x)}}$$

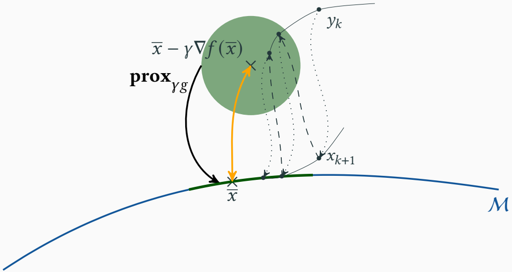
$$\begin{cases} y_k &= x_k - \gamma\nabla f(x_k) \\ x_{k+1} &= \mathbf{prox}_{\gamma g}(y_k) \end{cases}$$

Provided that:

- a **Qualifying Condition** holds around a **critical** limit point $\overline{x} \in \mathcal{M}$
- g is **nonsmooth across** the manifold $\mathcal{M}$ but **smooth along** it

After some finite time:

- the proximal gradient map $x \mapsto \mathbf{prox}_{\gamma g}(x - \gamma\nabla f(x))$ is **$\mathcal{M}$-valued** and **Lipschitz-continuous**
- $F = f + g$ is **smooth locally on $\mathcal{M}$**



- Poliquin and Rockafellar: Prox-regular functions in variational analysis. Transactions of the American Mathematical Society (1996)
- Lewis: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization (2002)
- Hare, Lewis: Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis (2004)

Optimization on Smooth Manifolds

$$\min_{x \in \mathbb{R}^n} F(x) = \underbrace{\overset{\text{risk}}{f(x)}}_{\text{smooth}} + \underbrace{\overset{\text{regularization}}{g(x)}}_{\text{nonsmooth}}$$

$$x_{k+1} = \mathrm{RiemannianGradient}_{\mathcal{M}}(x_k)$$

Since after some time:

- x_k **belongs to $\mathcal{M}$**
- $F = f + g$ is **smooth locally on $\mathcal{M}$**

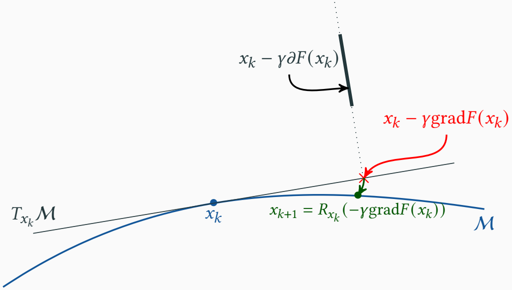
Riemannian optimization steps can be performed:

- Tractable for many regularizers

  linear spaces (sparsity), fixed rank, etc.

- 1st and 2nd order methods can be implemented

  Toolbox ManOpt in Matlab, Python, Julia



- This is useful **only if** we are on the "right" manifold and we never know that

◇ Boumal, Mishra, Absil, Sepulchre: Manopt, a Matlab toolbox for optimization on manifolds. The Journal of Machine Learning Research (2014)
◇ Boumal: An introduction to optimization on smooth manifolds (2020)

Newton acceleration

Idea Alternate proximal gradient steps and Riemannian Newton steps

$$\begin{cases} x_{k+1} = \mathbf{prox}_{\gamma g}(u_k - \gamma \nabla f(u_k)) & \textbf{identifies the current structure} \\ \text{Observe } \mathcal{M}_{k+1} \ni x_{k+1} & \\ u_{k+1} = \mathrm{RiemannianNewton}_{\mathcal{M}_{k+1}}(x_{k+1}) & \textbf{updates on the corresponding manifold} \end{cases}$$

Theorem Provided that the minimizers of the function are *qualified*, the method converges quadratically.

Bareilles, **I**, Malick: *Newton acceleration on manifolds identified by proximal-gradient methods*, ArXiv, 2020

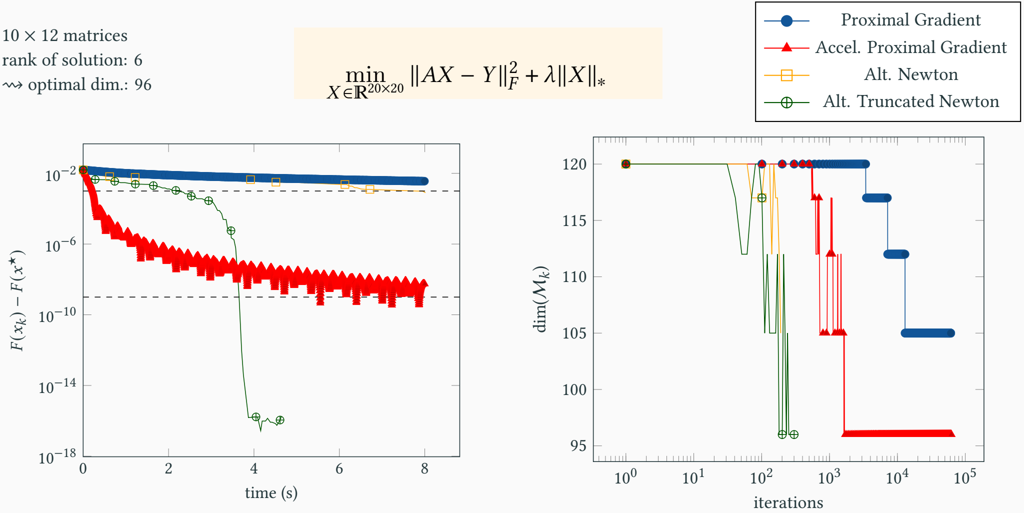
Perspective Providing "structure stability" guarantees for statistical models eg. by estimating the radius of the qualification, tuning the regularization by bi-level programming. This motivates high-accuracy objectives for nonsmooth solvers

- Lemaréchal, Oustry, Sagastizábal: The $\mathcal{U}$-Lagrangian of a convex function. Trans. of the AMS (2000)
- Mifflin, Sagastizábal: A $\mathcal{VU}$-algorithm for convex minimization. Mathematical programming (2005)
- Daniilidis, Hare, Malick: Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. Optimization (2006)

Illustration on trace norm regression

10 × 12 matrices
rank of solution: 6
⇝ optimal dim.: 96

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



Perspective Understanding the relation between structure identification and statistical simplicity

eg. the links between qualification and RIP-like properties

Another interesting structure: composition of a smooth map and a nonsmooth function

$$\min_{x\in\mathbb{R}^n} g(c(x))$$

$$\mathbb{R}^n \xrightarrow[c]{\text{smooth map}} \underset{\text{intermediate space}}{\mathbb{R}^m} \xrightarrow[g]{\text{nonsmooth function}} \mathbb{R}$$

max. of functions
$\max(f_1(x), \ldots, f_N(x))$

maximal eigenvalue
$\lambda_{\max}(A_0 + \sum_{i=1}^n x^{[i]} A_i)$

- Identification by $\mathbf{prox}_{\gamma g}$ holds in the *intermediate* space structure is lost when restoring the feasibility

Idea Minimize $g \circ c$ *along* a tentative structure even if the current point is not on it

- Use $\mathbf{prox}_{\gamma g}(c(x_k))$ to find structure in the intermediate space, defined by $h_k(u) = 0$
- Translate the structure to the input space as $s_k(x) = h_k(c(x)) = 0$ in general $s_k(x_k) \neq 0$
- Perform a SQP step on $\min_{x\in\mathbb{R}^n} g(c(x))$ s.t. $s_k(x) = 0$ smooth minimization along, Newton-Raphson across

Perspective Writing a generic optimizer for composite problems when $\mathbf{prox}_{\gamma g}$ is explicit
with automatic differentiation for the map and benchmark against other nonsmooth methods

- Overton. On minimizing the maximum eigenvalue of a symmetric matrix. SIAM Journal on Matrix Analysis and Applications. (1988)
- Oustry. A second-order bundle method to minimize the maximum eigenvalue function. Mathematical Programming. (1999)
- Lewis and Wright. A proximal method for composite minimization. Mathematical Programming (2016)
- Lewis and Wylie. A simple newton method for local nonsmooth optimization. (2019)
- Bolte, Chen, Pauwels. The multiproximal linearization method for convex composite problems. Mathematical Programming. (2020)
- Han and Lewis. Survey Descent: A Multipoint Generalization of Gradient Descent for Nonsmooth Optimization. preprint (2021)

Two Current Directions & Perspectives

β – Optimization beyond Minimization

Some optimization problems beyond minimization

Empirical Risk Minimization optimizes the *average* loss under $\mathsf{P}_m = \frac{1}{m}\sum_{j=1}^{m} \delta_{(\mathbf{a}_j, b_j)}$

$$\min_{x \in \mathbb{R}^n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} \ell(b_j, P_x(\mathbf{a}_j))}_{f(x)} = \mathbb{E}_{\xi \sim \mathsf{P}_m}\left[\ell_x(\xi)\right]$$

1. Online learning data is revealed in a sequential order

$$\min_{x \in \mathbb{R}^n} f_t(x) \text{ for } t = 1, \ldots, T$$

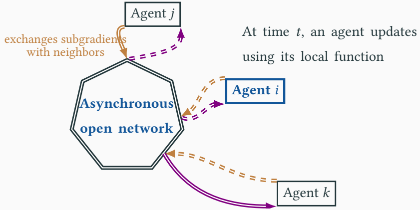
2. Saddle-point problems Nash equilibria, adversarial examples, variational inequalities

$$\min_{x_1 \in C_1} \max_{x_2 \in C_2} f(x_1, x_2)$$

3. Robust risk minimization distribution shifts between training and testing

$$\min_{x \in \mathbb{R}^n} \max_{\mu \in \mathcal{A}} \mathbb{E}_{\xi \sim \mu}\left[\ell_x(\xi)\right]$$

1. Optimization in Open Networks as an Online Problem



Agents can join and leave so minimizing the *current loss* is out of reach

$$f_t(x) = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} f^i(x)$$

$\mathcal{V}_t$ are the agents at time t

Goal: minimize the running loss

$$\mathbf{Loss}(T) = \frac{1}{\sum_{t=1}^{T} |\mathcal{V}_t|} \sum_{t=1}^{T} \sum_{i \in \mathcal{V}_t} f^i(x_t^{\mathrm{ref}})$$

x_t^{ref} is the value of any agent at time t

Idea Use the framework of *online optimization* to analyze (offline) minimization over *open networks*

- with subgradient exchanges, we obtain $\mathbf{Loss}(T) = \mathcal{O}(1/\sqrt{T})$ without a global clock or current network state
- by extending *dual averaging* $x_t = \mathrm{proj}_C\left(x_1 - \gamma_{i,t} \sum_{s \in \mathcal{S}_{i,t}} g_{j,s}\right)$ to incorporate all gradients "equally"

Hsieh, **I**, Malick, Mertikopoulos: *Optimization in Open Networks via Dual Averaging*, CDC 2021.
–: *Multi-Agent Online Optimization with Delays: Asynchronicity, Adaptivity, and Optimism*, preprint Dec. 2020.

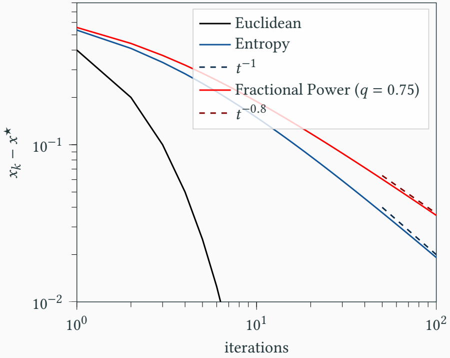
Perspective Examining the behavior of a flock of agents wishing to regroup and learn simultaneously

Find $x^\star \in C$ such that
$\langle v(x^\star), x - x^\star \rangle \geq 0$ for all $x \in C$
v is Lipschitz and strongly monotone

$$x_{k+1} = \operatorname{argmin}_{u \in C} \left\{ -\gamma \langle v(x_k), x_k - u \rangle + D^h(u, x_k) \right\}$$
with $D^h(u, x) = h(u) - h(x) - \langle \nabla h(x), u - x \rangle$

- If $x^\star$ is on the border of C, the observed rate depends on the regularizer h

$v(x) = x - x^\star$
$C = [0, +\infty]$



2. Rates of Mirror Descent for Border Solutions in Variational Inequalities

Find $x^\star \in C$ such that
$$\langle v(x^\star), x - x^\star \rangle \geq 0 \text{ for all } x \in C$$
v is Lipschitz and strongly monotone

$$x_{k+1} = \operatorname{argmin}_{u \in C} \left\{ -\gamma \langle v(x_k), x_k - u \rangle + D^h(u, x_k) \right\}$$
with $D^h(u, x) = h(u) - h(x) - \langle \nabla h(x), u - x \rangle$

- If $x^\star$ is on the border of C, the observed rate depends on the regularizer h

Idea Upper-bound the Bregman divergence *locally* around a border solution

Find the smallest $\beta^\star \in [0,1]$ such that $D^h(x^\star, x) \leq \frac{\kappa}{2} \|x - x^\star\|^{2(1-\beta^\star)}$ for all x close to $x^\star$ in C

For border solutions, the convergence rate of $D^h(x^\star, x_k)$ for Mirror Descent depends on the value of $\beta^\star$

	Domain (C)	Regularizer (h)	Legendre Exponent ($\beta^\star$)	Convergence Rate
Euclidean	arbitrary	$x^2/2$	0	$\exp(-O(t))$
Entropic	$[0, \infty)$	$x \log x$	$1/2$	$O(1/t)$
Tsallis	$[0, \infty)$	$[q(1-q)]^{-1}(x - x^q)$	$\max\{0, 1 - q/2\}$	$O(1/t^{q/(2-q)})$
Hellinger	$[-1, 1]$	$-\sqrt{1-x^2}$	$3/4$	$O(1/t^{1/3})$

- Can be extended to Mirror Prox, Optimistic Mirror Descent, and stochastic variants

Perspective Exploiting the structure of constraint sets in variational inequalities

Distributionally robust risk minimization

$$\min_{x\in\mathbb{R}^n} \quad \sup_{\mu\in\mathcal{U}(\mathsf{P}_m)} \quad \mathbb{E}_{\xi\sim\mu}\left[\ell_x(\xi)\right]$$

- Ambiguity set $\mathcal{U}(\mathsf{P}_m)$: distributions in a neighborhood of the observed samples P_m discrete distribution
- This neighborhood depends on a *chosen metric* on distributions
  - Wasserstein distance has many good properties includes continuous distributions, statistical guarantees
  - but the resulting problem may be difficult to optimize dual approach

Idea Regularize the Wasserstein distance with Kullback-Liebler divergences for a more tractable objective

Perspective Going towards statistical guarantees & non-convex objectives not only for neural networks!

- Esfahani and Kuhn: Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming (2018)
- Blanchet, Murthy, Zhang. Optimal transport-based distributionally robust optimization: Structural properties and iterative schemes. Mathematics of Operations Research (2021)
- Gao and Kleywegt. Distributionally Robust Stochastic Optimization with Wasserstein Distance. preprint (2016)

Conclusion

Hokusai

Fine Wind, Clear Morning (Gaifū kaisei) in Thirty-six Views of Mount Fuji (1830-1832)

Conclusion

- Data Science problems offer a vast playground for optimizers
- I particularly enjoy the theory & practice of
  - Structure stability
  - Distributional robustness
  - Resilience in multi-agent systems

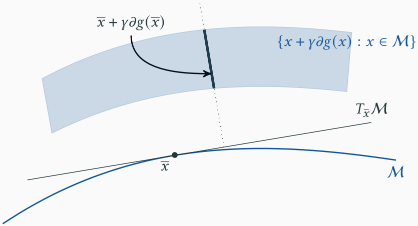
- Many thanks to all the collegues, students, and friends that made all this possible.

Thank you!

Motivating the Riemannian manifold nature of the observed structures: partial smoothness

A function g is (C^2-)*partly smooth* at a point $\bar{x}$ relative to the C^2 manifold $\mathcal{M}$ around $\bar{x}$ if:

- (smoothness) the restriction of g to $\mathcal{M}$ is a C^2 function near $\bar{x}$;
- (regularity) g is (Clarke) regular at all points $x \in \mathcal{M}$ near $\bar{x}$, with $\partial g(x) \neq \emptyset$;
- (sharpness) the affine span of $\partial g(\bar{x})$ is a translate of $N_{\bar{x}}\mathcal{M}$;
- (sub-continuity) the set-valued mapping ∂g restricted to $\mathcal{M}$ is continuous at $\bar{x}$.

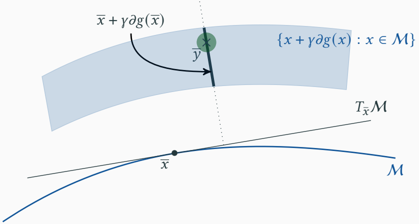


- Lewis: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization (2002)
- Hare, Lewis: Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis (2004)
- Daniilidis, Hare, Malick: Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. Optimization (2006)

Motivating the Riemannian manifold nature of the observed structures: partial smoothness

A function g is (C^2-)*partly smooth* at a point $\bar{x}$ relative to the C^2 manifold $\mathcal{M}$ around $\bar{x}$ if:

- (smoothness) the restriction of g to $\mathcal{M}$ is a C^2 function near $\bar{x}$;
- (regularity) g is (Clarke) regular at all points $x \in \mathcal{M}$ near $\bar{x}$, with $\partial g(x) \neq \emptyset$;
- (sharpness) the affine span of $\partial g(\bar{x})$ is a translate of $N_{\bar{x}}\mathcal{M}$;
- (sub-continuity) the set-valued mapping ∂g restricted to $\mathcal{M}$ is continuous at $\bar{x}$.



If g is p.s. and $\frac{\overline{y}-\overline{x}}{\gamma} \in \operatorname{ri} \partial g(\overline{x})$,
then for all y close to $\overline{y}$, $\mathbf{prox}_{\gamma g}(y) \in \mathcal{M}$

In the non-convex case,
additional conditions are needed on $\mathit{prox}_{\gamma g}$

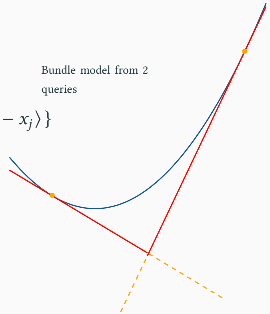
- Lewis: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization (2002)
- Hare, Lewis: Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis (2004)
- Daniilidis, Hare, Malick: Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. Optimization (2006)

Asynchronous Level Bundle

$$F(x) = x^2 + 2x + 1$$

$$\check{F}_k(x) := \max_{j \in \mathcal{J}_k} \{F(x_j) + \langle v_j, x - x_j \rangle\}$$

$\mathcal{J}_k \subset \{1, 2, \ldots, k\}$: set of indices of points at which the oracle was called

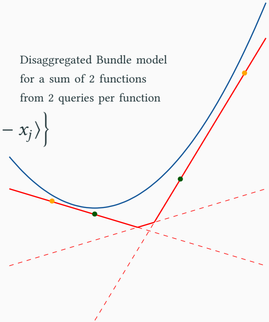


$$F^1(x) = x^2/2$$
$$F^2(x) = x^2/2 + 2x + 1$$

$$\check{F}^i_k(x) := \max_{j \in \mathcal{J}^i_k} \left\{F^i(x_j) + \langle v^i_j, x - x_j \rangle\right\}$$

$\mathcal{J}^i_k \subset \{1, 2, \ldots, k\}$: set of indices of points at which oracle i was called

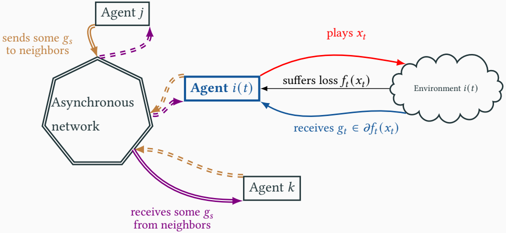
$$\check{F}^{\mathrm{d}}_k(x) := \sum_{i=1}^{M} \check{F}^i_k$$



- The disaggregated bundle accumulates information asynchronously no need to query all functions at all points
- We can design a level bundle method to minimize F

I, Malick, de Oliveira: *Asynchronous level bundle methods*, Mathematical Programming, 2020.

Multi-agent Online Optimization with Delays



At time t:

- an agent $i(t)$ becomes actives
- plays a point x_t
- suffers loss $f_t(x_t)$
- receives feedback $g_t \in \partial f_t(x_t)$

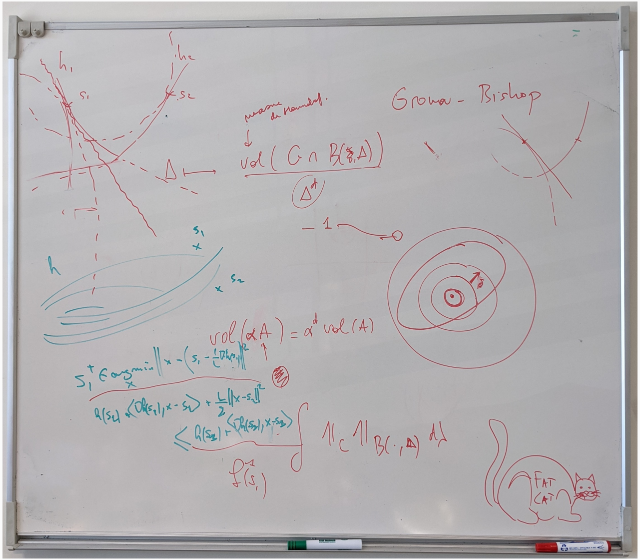
Asynchronously: agents exchange feedback vectors g_s
$\rightsquigarrow$ delay bounded by τ

Goal: minimize the regret $\mathbf{Reg}_T(u) = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u)$

Idea active agent $i(t)$ only has some subgradients $\{g_s : s \in \mathcal{S}_t\}$ at time t

- we extend *dual averaging* to incorporate all gradients "equally" $x_t = \operatorname{argmin}_{x \in C} \left\{ \sum_{s \in \mathcal{S}_t} \langle g_s, x \rangle + \frac{\|x\|^2}{2\gamma_t} \right\}$
- even without a global clock, we obtain $\mathbf{Reg}_T(u) = \mathcal{O}(\sqrt{T \times \tau})$

Hsieh, **I**, Malick, Mertikopoulos: *Multi-Agent Online Optimization with Delays: Asynchronicity, Adaptivity, and Optimism*, ArXiv 2012.11579, 2020.


h1
h2
s1
s2
Gromov - Bishop
vol(C ∩ B(s,Δ))
Δ^d
vol(αA) = α^d vol(A)
h
s1
s2
FAT
CAT