UniGe | MaLGa

Efficient approaches to regularized inverse problems

Dante seminar

Mathurin Massias

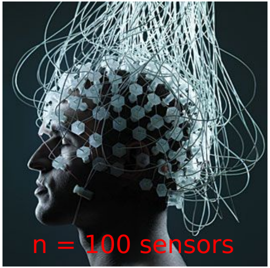
https://mathurinm.github.io

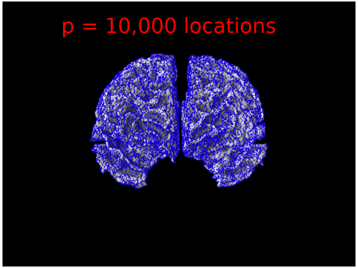
MaLGa - Machine Learning Genova Center
Università di Genova

Joint works with C. Molinari, L. Rosasco, S. Villa, J. Salmon, A. Gramfort, S. Vaiter, O. Fercoq, Q. Bertrand

A linear inverse problem (my PhD)

- observe magnetoelectric field outside the scalp (100 sensors)
- reconstruct cerebral activity inside the brain (10,000 locations)





- linear relationship by Maxwell equations
- $n \ll p$: ill-posed problem $\hookrightarrow$ regularization, sparsity

General setup: overparametrization & linear models

Overparametrized models are common in DL, signal processing, sparse learning, etc.
In linear models, they take the form:

$$Xw = y$$

where we know $y \in \mathcal{Y}$, X bounded linear operator between Hilberts $\mathcal{X}$ and $\mathcal{Y}$.[1]
$\hookrightarrow$ overparametrized: $\ker X$ non zero, infinitely many solutions

In the noiseless case (y exact), favor a particular one by solving:

$$\boxed{w^\star = \underset{Xw=y}{\arg\min}\, J(w)}$$

where the convex $J : \mathcal{X} \to \mathbb{R}$ promotes *something* (sparsity, low-rank, edges with $\|\cdot\|_1, \|\cdot\|_{\text{nuc}}, \|\cdot\|_{\text{TV}}$)

UniGe | MaLGa

[1] finite dim: think $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, and $p > n$

Corruption by noise

Usually true data y is available only through $y^\delta \approx y$, e.g. with "deterministic noise":

$$\|y - y^\delta\| \leq \delta$$

and generally

$$w^\star = \arg\min_{Xw=y} J(w) \neq \arg\min_{Xw=y^\delta} J(w) = w^\star_\delta$$

$\hookrightarrow$ it is pointless to solve $\arg\min_{Xw=y^\delta} J(w)$ *exactly*. Actually if $\dim \mathcal{X} = +\infty$, $X^\dagger$ need not be bounded and $w^\star_\delta$ may be arbitrarily far from $w^\star$[2]

[2] think in $\ell_2(\mathbb{N})$, $Xe_n = e_n/n$ linear bounded, take $y^\delta = y + \delta e_n$ for n arbitrarily large

Plan

Focus on **overparametrized linear inverse problems**:

$$Xw = y, \qquad \text{with infinitely many solutions}$$

- when we want to pick a "**simple**" solution amongst all possible through J
- in the case where only a **noisy** version of y is available

For that there are two main approaches

1. relaxation
2. iterative regularization

$\hookrightarrow$ we propose efficient methods for both

Outline

Fast solvers for sparse ML problems

Iterative regularization for non strongly convex regularizers

Addressing the noise issue

Focus on finite dimension for the first part, $y \in \mathbb{R}^n$ and X explicit matrix in $\mathbb{R}^{n \times p}$

- **1st approach: relaxation/penalization**, solve

$$\arg\min_{w \in \mathbb{R}^p} \frac{1}{2} \|Xw - y^\delta\|^2 + \lambda J(w)$$

  and tune $\lambda > 0 \hookrightarrow$ regularized ERM, e.g. Lasso (other datafits possible, logreg)

Classical selection procedure: cross-validation on a predetermined grid of λ's

Many problems must be solved, we need efficient solvers for them

Composite optimization

$$\min_{w \in \mathbb{R}^p} f(Xw) + \lambda J(w)$$

Lasso, group Lasso, elastic net, multitask Lasso... penalties are neither strongly convex, nor smooth, but they all have a computable *proximal operator*

$$\text{prox}_J(x) = \arg\min_y \frac{1}{2}\|x - y\|^2 + J(y)$$

Workhorse for composite "smooth + proximable" pbs: Forward-Backward (ISTA for lasso)

$$w^{(t+1)} \triangleq \text{prox}_{\frac{\lambda}{L} J}\Big(w^{(t)} - \tfrac{1}{L} X^\top \nabla f(Xw^{(t)})\Big)$$

can be slow in ML setting

Exploiting separability with coordinate descent (Tseng and Yun, 2009)

Applies to *separable* penalty J (and L-smooth f):

$$\min_{w \in \mathbb{R}^p} f(Xw) + \lambda \sum_{j=1}^{p} g_j(w_j)$$

(Proximal) coordinate descent: update only coordinate j per iteration

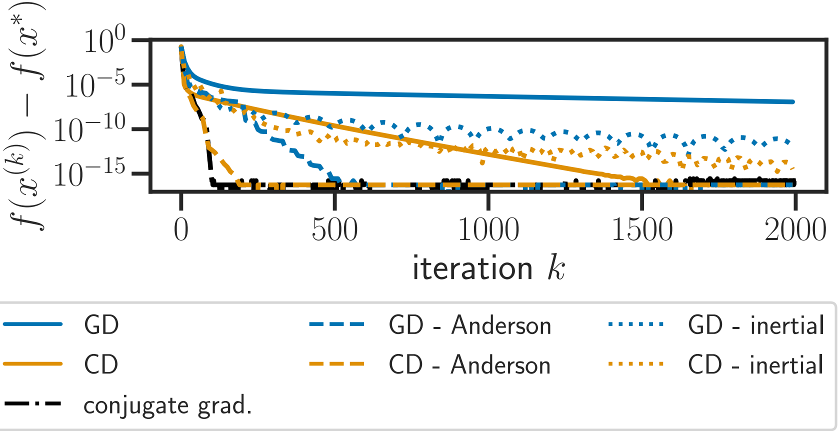
$$w_j^{(t+1)} = \operatorname{prox}_{\frac{\lambda}{L_j} g_j} \left(w_j^{(t)} - \frac{1}{L_j} X_{:j}^\top \nabla f(Xw^{(t)}) \right)$$

this update of one coordinate costs n (np for Forward-Backward)

Both FB and CD can be accelerated inertially, but tricky practice for CD (Nesterov, 1983; Beck and Teboulle, 2009; Fercoq and Richtárik, 2015)

We propose instead *Anderson* acceleration for CD and prox-CD

Anderson vs inertial acceleration on OLS (rcv1 dataset)



Anderson acceleration idea : extrapolation for 1D sequences

If $(x_k)_{k\in\mathbb{N}}$ follows a converging autoregressive process (AR):

$$x_k = a x_{k-1} + b \qquad (|a| < 1, b \in \mathbb{R}) \quad \text{with} \ \lim_{k\to\infty} x_k = x^*$$

Going to the limit yields $x^* = a x^* - b$, hence:

$$x_k - x^* = a(x_{k-1} - x^*)$$

Aitken's Δ^2: 2 unknowns, so 2 equations/3 points x_{k+1}, x_k, x_{k-1} are enough to find x^*:

$$x_{k+1} - x^* = a(x_k - x^*)$$
$$x_k - x^* = a(x_{k-1} - x^*)$$

Aitken (1926) uses, after computing r_{k+1}, the extrapolated point:

$$\Delta_k^2 = x_k + \frac{1}{\frac{1}{x_{k+1}-x_k} - \frac{1}{x_k - x_{k-1}}}$$

UniGe | MaLGa

Aitken's Δ^2 interest: works for approximate AR sequence

$$\lim_{k\to\infty} \sum_{i=0}^{k} \frac{(-1)^i}{2i+1} = \frac{\pi}{4} = 0.785398\ldots$$

k	$\sum_{i=0}^{k} \frac{(-1)^i}{2i+1}$	Δ_k^2
0	1.0000	–
1	0.66667	–
2	0.86667	**0.7**9167
3	**0.7**2381	**0.78**333
4	0.83492	**0.78**631
5	**0.7**4401	**0.78**492
6	0.82093	**0.785**68
7	**0.7**5427	**0.785**22
8	0.81309	**0.785**52
9	**0.7**6046	**0.7853**1

UniGe | MaLGa

Extrapolation for vectors?

Target sequences satisfying:

$$\mathbf{x}^{(k+1)} - \mathbf{x}^* = \mathbf{T}(\mathbf{x}^{(k)} - \mathbf{x}^*) \quad (\mathbf{T} \in \mathbb{R}^{d\times d})$$

Many names:

- Anderson extrapolation (Anderson, 1965)
- Approximate Minimal Polynomial Extrapolation (Wynn, 1962)
- Eddy-Mesina method (Eddy, 1979)
- etc (Smith et al., 1987), reviews: Sidi (2017); Brezinski et al. (2018)

Recent interest in ML following Scieur et al. (2016): Mai and Johansson (2019); Massias et al. (2019); Poon and Liang (2019); Fu et al. (2019)

Approximate Minimal Polynomial Extrapolation (AMPE)

$$\mathbf{x}^{(k+1)} - \mathbf{x}^* = \mathbf{T}(\mathbf{x}^{(k)} - \mathbf{x}^*)$$

For any K coefficients c_k:

$$\sum_{k=1}^{K} c_k(\mathbf{x}^{(k)} - \mathbf{x}^*) = \sum_{k=1}^{K} c_k \mathbf{T}^k(\mathbf{x}^{(0)} - \mathbf{x}^*)$$

Approximate Minimal Polynomial Extrapolation (AMPE)

$$\mathbf{x}^{(k+1)} - \mathbf{x}^* = \mathbf{T}(\mathbf{x}^{(k)} - \mathbf{x}^*)$$

For any K coefficients c_k:

$$\sum_{k=1}^{K} c_k(\mathbf{x}^{(k)} - \mathbf{x}^*) = \sum_{k=1}^{K} c_k \mathbf{T}^k(\mathbf{x}^{(0)} - \mathbf{x}^*)$$

If one has $\sum_{k=1}^{K} c_k = 1$, we get:

$$\sum_{k=1}^{K} c_k \mathbf{x}^{(k)} - \mathbf{x}^* = \left(\sum_{k=1}^{K} c_k \mathbf{T}^k\right)(\mathbf{x}^{(0)} - \mathbf{x}^*)$$

Approximate Minimal Polynomial Extrapolation (AMPE)

$$\mathbf{x}^{(k+1)} - \mathbf{x}^* = \mathbf{T}(\mathbf{x}^{(k)} - \mathbf{x}^*)$$

For any K coefficients c_k:

$$\sum_{k=1}^{K} c_k(\mathbf{x}^{(k)} - \mathbf{x}^*) = \sum_{k=1}^{K} c_k \mathbf{T}^k(\mathbf{x}^{(0)} - \mathbf{x}^*)$$

If one has $\sum_{k=1}^{K} c_k = 1$, we get:

$$\sum_{k=1}^{K} c_k \mathbf{x}^{(k)} - \mathbf{x}^* = \left(\sum_{k=1}^{K} c_k \mathbf{T}^k\right)(\mathbf{x}^{(0)} - \mathbf{x}^*)$$

For $K = d$ and c_k the coefficients of the minimal polynomial of $\mathbf{T}$, the RHS vanishes.

Idea: approximate $\mathbf{x}^*$ by an affine combination of $\mathbf{x}^{(k)}$'s

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k(\mathbf{x}^{(k)} - \mathbf{x}^*) \right\|, \text{ where } \mathbf{1}_K = (1, \ldots, 1)^\top \in \mathbb{R}^K$$

AMPE continued

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k (\mathbf{x}^{(k)} - \mathbf{x}^*) \right\|$$ cannot be solved due to $\mathbf{x}^*$

Proxy: try to make the extrapolated point a fixed point by solving:

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k \mathbf{x}^{(k)} - \mathbf{T} \sum_{k=1}^{K} c_k \mathbf{x}^{(k)} - \mathbf{b} \right\|$$
$$= \min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} \right) \right\|$$

AMPE: closed-form

Anderson extrapolation coefficients are found by solving:

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k \big(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\big) \right\|$$

Introducing $U = (\mathbf{x}^{(1)} - \mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(K)} - \mathbf{x}^{(K-1)}) \in \mathbb{R}^{d \times K}$:

$$\min_{c^\top \mathbf{1}_K = 1} \|Uc\|^2$$

Closed-form (cost: $K^3 + K^2 d$):

$$c = \frac{(U^\top U)^{-1} \mathbf{1}_K}{\mathbf{1}_K^\top (U^\top U)^{-1} \mathbf{1}_K}$$

AMPE: to what does it apply?

Need *linear iterations, vector autoregressive structure*:

$$\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{b}, \qquad \text{with } \lambda_{\max}(\mathbf{T}) < 1$$

Prototypical example: GD on least squares $\frac{1}{2}\|\mathbf{A}x - y\|^2$

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - \gamma \mathbf{A}^\top(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}) \\ &= (\mathrm{Id} - \gamma \mathbf{A}^\top \mathbf{A})\mathbf{x}^{(k)} - \gamma \mathbf{A}^\top \mathbf{y}\end{aligned}$$

Also valid for coordinate descent ($\mathbf{T}$ not symmetrical, Bertrand and Massias (2020)):

$$\mathbf{x}^{(k+1)} = \left(\mathrm{Id}_p - \frac{e_n e_n^\top}{\mathbf{A}_{nn}}\mathbf{A}\right) \dots \left(\mathrm{Id}_p - \frac{e_1 e_1^\top}{\mathbf{A}_{11}}\mathbf{A}\right)\mathbf{x}^{(k)} + \mathbf{b}^{\mathtt{CD}}$$

Anderson: offline algorithm

Algorithm is easy to implement:

Algorithm 1 Offline Anderson extrapolation

init: $\mathbf{x}^{(0)} \in \mathbb{R}^n$

for $k = 1, \ldots$ **do**

  $\mathbf{x}^{(k)} = \mathbf{A}\mathbf{x}^{(k-1)} + b$ // regular linear iteration

  $U = [\mathbf{x}^{(1)} - \mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}] \in \mathbb{R}^{n \times k}$ // grows in size with k

  $c = (U^\top U)^{-1}\mathbf{1}_k / \mathbf{1}_k^\top (U^\top U)^{-1}\mathbf{1}_k \in \mathbb{R}^k$ // gets harder and harder to solve

  $\mathbf{x}_{\mathrm{e-off}}^{(k)} = \sum_{i=1}^k c_i \mathbf{x}^{(i)}$ // does not affect base sequence $\mathbf{x}^{(k)}$

return $\mathbf{x}_{\mathrm{e-off}}^{(k)}$

UniGe | MaLGa

Online Anderson

2 goals: improve base sequence convergence, make cost of Anderson fixed.

Algorithm 2 Online Anderson extrapolation

init: $\mathbf{x}^{(0)} \in \mathbb{R}^n$

for $k = 1, \ldots$ **do**

  $\mathbf{x}^{(k)} = \mathbf{A}\mathbf{x}^{(k-1)} + b$ `// regular iteration`

  if $k = 0 \mod K$ **then**

    $U = [\mathbf{x}^{(k-K+1)} - \mathbf{x}^{(k-K)}, \ldots, \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}]$ `// only` K `last ones`

    $c = (U^\top U)^{-1}\mathbf{1}_K / \mathbf{1}_K^\top (U^\top U)^{-1}\mathbf{1}_K \in \mathbb{R}^K$ `// fixed (small) size`

    $\mathbf{x}_{\mathrm{e-on}}^{(k)} = \sum_{i=1}^{K} c_i \mathbf{x}^{(k-K+i)}$

    $\mathbf{x}^{(k)} = \mathbf{x}_{\mathrm{e-on}}^{(k)}$ `// base sequence changes`

return $\mathbf{x}^{(k)}$

Compute $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)} \hookrightarrow$ extrapolate, change $\mathbf{x}^{(K)}$ to extrapolated point, from here compute $\mathbf{x}^{(K)}, \mathbf{x}^{(K+1)}, \ldots, \mathbf{x}^{(2K)}$, extrapolate, etc.

Convergence rates in the symmetric case (Scieur, 2019)

Theorem
Let the iteration matrix $\mathbf{A}$ be symmetric semi-definite positive, with spectral radius $\rho = \rho(\mathbf{A}) < 1$.
Let $\mathbf{x}^*$ be the limit of the sequence $(\mathbf{x}^{(k)})$.
Let $\zeta = \rho/(1+\sqrt{1-\rho})^2 < \rho$, let $\mathbf{B} = (\mathrm{Id} - \mathbf{A})^2$.

Then the iterates of offline Anderson acceleration satisfy:

$$\|\mathbf{x}^{(k)}_{\text{e-off}} - \mathbf{x}^*\|_{\mathbf{B}} \leq \frac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}} \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_{\mathbf{B}} \ ,$$

and thus those of online extrapolation satisfy:

$$\|\mathbf{x}^{(k)}_{\text{e-on}} - \mathbf{x}^*\|_{\mathbf{B}} \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1}}\right)^{k/K} \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_{\mathbf{B}} \ .$$

Way harder: convergence rates for non-symmetric $\mathbf{A}$

Bollapragada et al. (2018): when $\mathbf{A}$ is not symmetric, and its spectral radius $\rho(\mathbf{A}) < 1$,

$$\|\mathbf{x}^{(k)}_{\text{e-off}} - \mathbf{A}\mathbf{x}^{(k)}_{\text{e-off}} - \mathbf{b}\| \leq \|\mathrm{Id} - \rho(\mathbf{A} - \mathrm{Id})\|_2 \, \|P^*(\mathbf{A})(\mathbf{x}^{(0)} - \mathbf{A}\mathbf{x}^{(0)} - \mathbf{b})\|$$

where the unavailable polynomial P^* minimizes $\left\|P(\mathbf{A})(\mathbf{x}^{(0)} - \mathbf{A}\mathbf{x}^{(0)} - \mathbf{b})\right\|$ amongst all polynomials P of degree exactly $k-1$ whose coefficients sum to 1.

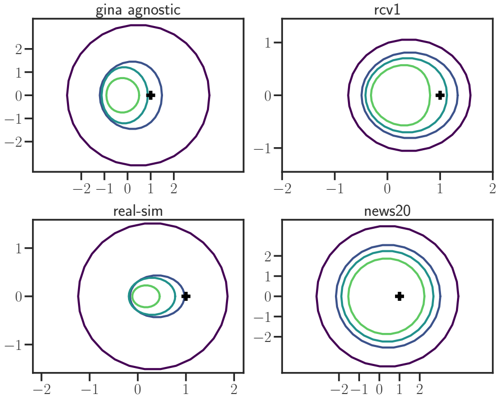
The quality of the bound (in particular, its eventual convergence to 0) crucially depends on $\|P(\mathbf{A})\|$. Using the Crouzeix conjecture Crouzeix (2004) they managed to bound $\|P(\mathbf{A})\|$, with P a polynomial:

$$\|P(\mathbf{A})\| \leq c \max_{z \in W(\mathbf{A})} |P(z)| \;,$$

with $c \geq 2$, and $W(\mathbf{A})$ the numerical range:

$$W(\mathbf{A}) := \{\mathbf{u}^*\mathbf{A}\mathbf{u} : \|\mathbf{u}\|_2 = 1, \mathbf{u} \in \mathbb{C}^p\} \;.$$

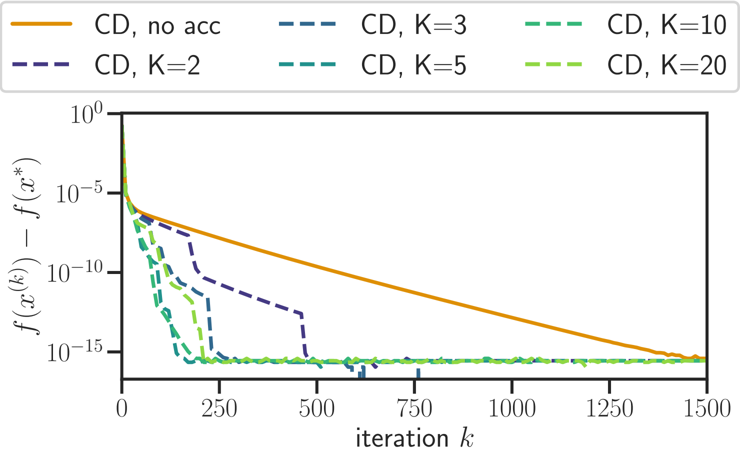
Numerical range in practice



For the bound to be useful, you need K to be large enough such that $(1, 0)$ lies outside the range $\hookrightarrow$ not satisfied in practice

for CD, we provide a convergence rate like in the symmetric case, with a pseudo-symmetrization trick

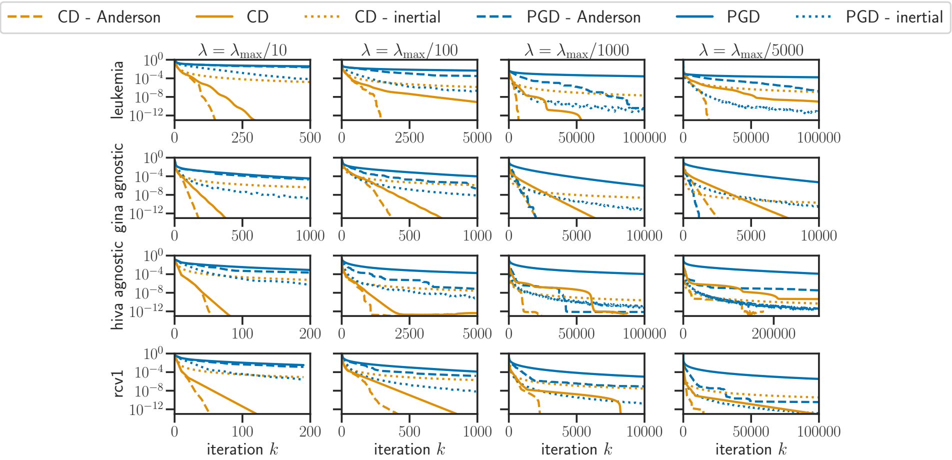
Parameter tuning: $K \hookrightarrow$ 5, 10 good default



$\hookrightarrow$ makes additional cost weak: $K^3 = 125$, $K^2 d = 25d$.

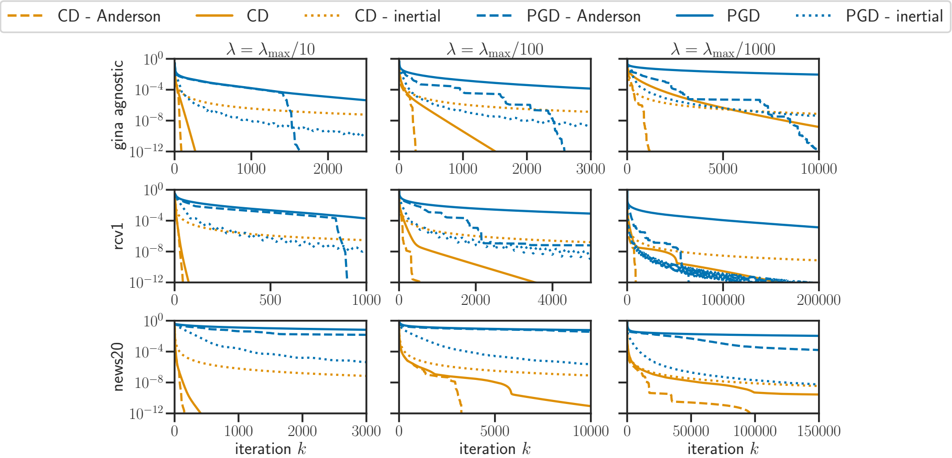
UniGe | MaLGa

Experiments: Lasso (non symmetric iteration matrix)



(suboptimality, lasso)

Sparse logistic regression (non symmetric, noisy linear iterations)



(suboptimality, sparse logistic regression)

Recap

- Anderson acceleration is a cheap, easy to implement, alternative to inertial acceleration
- on quadratics everything is fine
- experimentally, on other problems it works fine too (Lasso, ℓ_1 or ℓ_2 logistic regression, group lasso)
- for non symmetric iteration matrices, current bound seems very pessimistic

Most of all, getting rid of the dependency on the dual allows to generalize to least squares and non convex problems.

Outline

Fast solvers for sparse ML problems

Iterative regularization for non strongly convex regularizers

Addressing the noise issue

2nd approach, instead of relaxing the constraint $Xw = y^\delta$:

- early stopping/iterative regularization: use iterative algorithm to solve

$$\arg\min_{Xw = y^\delta} J(w)$$

  but **stop before convergence** (save computation while obtaining a point close/closer to $w^\star$ with *semiconvergence*, Engl et al. 1996)

$\hookrightarrow$ This part: when, and why does it make sense?

Iterative regularization & known results

For J strongly convex, known results with Bregman iterations/Mirror descent, or application of (accelerated) gradient descent to the dual (Matet et al., 2017; Gunasekar et al., 2018).

we address the case where J is **only convex**

Roadmap:

- find an algorithm to solve $\min_{Xw=y} J(w)$
- derive convergence rates (in which quantity?)

1) The algorithm

Our problem is of the form[3] $\min_w f(w) + g(Xw)$, both functions non-smooth:

$$\min_{Xw=y} J(w) \Leftrightarrow \min_{w \in \mathbb{R}^p} J(w) + \iota_{\{y\}}(Xw)$$

Introduce the Lagrangian $\mathcal{L}(w, \theta) \triangleq J(w) + \langle Xw, \theta \rangle - \langle y, \theta \rangle$

Why the Lagrangian?

$$\min_{w \in \mathbb{R}^p} J(w) + \iota_{\{y\}}(Xw) \Leftrightarrow \min_{w \in \mathbb{R}^p} \max_{\theta \in \mathbb{R}^n} J(w) + \langle Xw - y, \theta \rangle$$

Inverting $\min$ and $\max$ yields *dual* problem:

$$\max_{\theta \in \mathbb{R}^n} \min_{w \in \mathbb{R}^p} J(w) + \langle Xw - y, \theta \rangle \Leftrightarrow \max_{\theta \in \mathbb{R}^n} -J^\star(-X^\top \theta) - \langle y, \theta \rangle$$

[3] $\iota_C(x) = 0$ if $x \in C$, $+\infty$ otherwise
Fenchel-Legendre conjugate: $f^\star(u) = \sup_x(\langle x, u \rangle - f(x))$

Chambolle-Pock algorithm (Chambolle and Pock, 2011)

Primal-dual equivalence: if $w^\star$ is a solution of the primal and $-X^\top \theta^\star \in \partial J(w^\star)$, then $\theta^\star$ solves the dual and $(w^\star, \theta^\star)$ is a saddle point of the Lagrangian

$$\boxed{\begin{aligned} \mathcal{L}(w, \theta) = J(w) + \langle Xw, \theta \rangle - \langle y, \theta \rangle \\ \mathcal{L}(w^\star, \theta) \leq \mathcal{L}(w^\star, \theta^\star) \leq \mathcal{L}(w, \theta^\star) \end{aligned}}$$

$\hookrightarrow$ do prox-gradient step to minimize $\mathcal{L}$ in w (primal), then to maximize $\mathcal{L}$ in θ (dual):

$$\begin{cases} w_{k+1} = \mathrm{prox}_{\tau J}(w_k - \tau X^\top \theta_k) \\ \theta_{k+1} = \mathrm{prox}_{\sigma \iota_{\{y\}}}(\theta_k + \sigma X w_k) = \theta_k + \sigma(X w_{k+1} - y) \end{cases}$$

Chambolle-Pock algorithm (Chambolle and Pock, 2011)

Primal-dual equivalence: if $w^\star$ is a solution of the primal and $-X^\top\theta^\star \in \partial J(w^\star)$, then $\theta^\star$ solves the dual and $(w^\star, \theta^\star)$ is a saddle point of the Lagrangian

$$\boxed{\begin{aligned}\mathcal{L}(w,\theta) = J(w) + \langle Xw, \theta\rangle - \langle y, \theta\rangle \\ \mathcal{L}(w^\star,\theta) \le \mathcal{L}(w^\star,\theta^\star) \le \mathcal{L}(w,\theta^\star)\end{aligned}}$$

$\hookrightarrow$ do prox-gradient step to minimize $\mathcal{L}$ in w (primal), then to maximize $\mathcal{L}$ in θ (dual):

$$\begin{cases} w_{k+1} = \operatorname{prox}_{\tau J}(w_k - \tau X^\top(2\theta_k - \theta_{k-1})) \\ \theta_{k+1} = \operatorname{prox}_{\sigma \iota_{\{y\}}}(\theta_k + \sigma X w_k) = \theta_k + \sigma(X w_{k+1} - y) \end{cases}$$

⚠ not a symmetric algorithm, we need interpolation in the primal

Key to our analysis: view θ update as an *inexact prox* when y^δ replaces y (Rasch and Chambolle, 2020).

UniGe | MaLGa

2) How to measure optimality?

Duality gap is equal to Bregman divergence of J here:

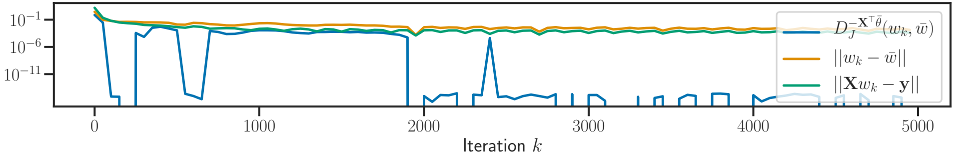
$$\begin{aligned}\mathcal{L}(w, \theta^\star) - \mathcal{L}(w^\star, \theta) &= J(w) + \langle \theta^\star, Xw - y\rangle - J(w^\star) - \langle \theta, \cancel{Xw^\star - y}\rangle \\ &= J(w) - J(w^\star) - \langle -X^\top \theta^\star, w - w^\star\rangle \\ &= D_J^{-X^\top \theta^\star}(w, w^\star)\end{aligned}$$

(Bregman divergence = difference at w between convex function J and its linear lower bound at $w^\star$)

$$D_J^\eta(w, w') = J(w) - J(w') - \langle \eta, w - w'\rangle, \qquad \eta \in \partial J(w')$$

Bregman is not point-separating for non-strongly convex functions

Not a real distance, zero duality gap is not enough for optimality with $J = \|\cdot\|_1$:



How to measure optimality? (cont'd)

Vanishing duality gap/Bregman divergence does not mean optimality
but when combined with feasability, it is enough!

Lemma:
If $(w^\star, \theta^\star)$ is a saddle point, $D_J^{-X^\top\theta^\star}(w, w^\star) = 0$ **AND** $Xw = y$, then w solves the primal.

$\hookrightarrow$ we derive bounds on both $D_J^{-X^\top\theta^\star}(w, w^\star) = 0$ and $\|Xw - y\|$.

Our results in the noisy case

Results on the ergodic[4] iterates of the **noisy** problem (CP with y^δ)

Theorem (Molinari et al., 2020)
If the stepsizes σ, τ satisfy $\sigma\tau < \varepsilon||X||_{\text{op}}$ with $0 < \varepsilon < 1$:

$$\mathcal{L}\left(\bar{w}^k, \theta^\star\right) - \mathcal{L}\left(w^\star, \bar{\theta}^k\right) \leqslant \frac{1}{k}(c_1 + c_2\delta k)^2$$

$$\left\|X\bar{w}^k - y\right\|^2 \leq c_3\left[c_4\delta + c_5\delta^2 + c_6\delta^2 k + \frac{1}{k}c_7\right]$$

Corollary
For $k = c/\delta$:

$$\mathcal{L}\left(\bar{w}^k, \theta^\star\right) - \mathcal{L}\left(w^\star, \bar{\theta}^k\right) \leq C\delta; \quad \left\|X\bar{w}^k - y\right\|^2 \leq C'\delta + C''\delta^2$$

[4] $\bar{w}^k \triangleq \frac{1}{k}\sum_{t=1}^{k} w_t$

Specialization for sparse recovery ($J = \|\cdot\|_1$)

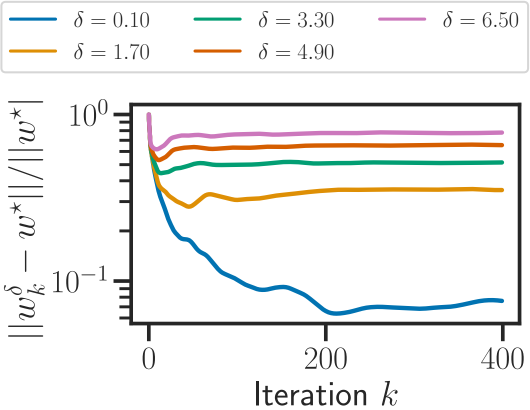
Theorem (Molinari et al., 2020)
With $\Gamma := \{j \in [p] : \left|X_{:j}^\top \theta^\star\right| = 1\}$, assume that X_Γ (X restricted to columns whose indices lie in Γ) is injective. Let $m := \max_{j \notin \Gamma} \left|X_{:j}^\top \theta^\star\right| < 1$. Then:

$$\|w - w^\star\| \leqslant \left\|X_\Gamma^{-1}\right\|_{\mathrm{op}} \|Xw - y\| + \frac{1 + \left\|X_\Gamma^{-1}\right\|_{\mathrm{op}} \|X\|_{\mathrm{op}}}{1 - m} D_{\|\cdot\|_1}^{-X^\top \theta^\star}(w, w^\star)$$

Corollary
For $k = c/\delta$:

$$\left\|\bar{w}^k - w^\star\right\| \leqslant C'\sqrt{\delta} + C''\delta$$

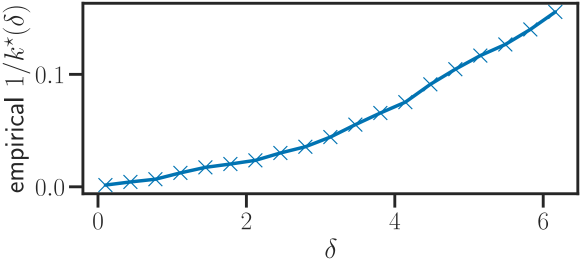
Experiment 1: There exists a stopping time ($J = \|\cdot\|_*$)



UniGe | MaLGa

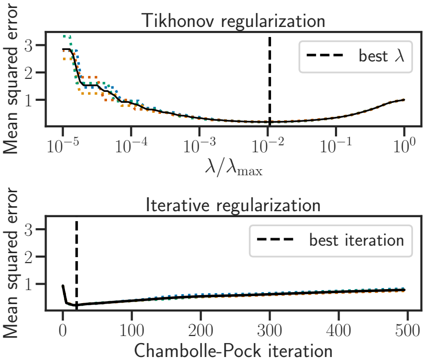
Experiment 2: Stopping time dependency is linear ($J = \|\cdot\|_1$)

empirical $k^\star(\delta) = \arg\min_k \left\|w_k^\delta - w^\star\right\|$



(The empirical stopping time is stronger than what we prove. Our analysis is valid for infinite dim. where there may not be a noisy solution, and early stopping is mandatory)

Experiment 3: Computation gains



Regularization path and optimization path of Chambolle-Pock on rcv1 ($n, p = 20$k) with 4-fold CV.
Minimal value reached: 0.19 (top), 0.21 (bottom).
Computation time up to optimal parameter: 50 s (top); 0.5 s (bottom).

Reproducibility

Online open source code, tested, documented, with code for experiments automatically run and published in the doc

- Anderson acceleration of CD: `https://mathurinm.github.io/andersoncd`
- Fast sklearn-like solver for sparse problems (can now handle non-convex penalties): `https://mathurinm.github.io/celer`
- Iterative sklearn-like solver, automated early stopping on left out data: `https://lcsl.github.io/iterreg`

Summary of interests

Fast CD approaches for regularized inverse problems:

- M. Massias, A. Gramfort, and J. Salmon. *Celer: a fast solver for the Lasso with dual extrapolation. ICML*, 2018
- M. Massias, S. Vaiter, A. Gramfort, and J. Salmon. *Dual extrapolation for sparse GLMs. to appear in JMLR*, 2020
- Q. Bertrand and M. Massias. *Anderson acceleration of coordinate descent. submitted*, 2020

Iterative regularization:

- C. Molinari, M. Massias, L. Rosasco, S. Villa. *Iterative regularization for convex regularizers. submitted*, 2020

Handling of correlated noise and repeated measurements, smoothing techniques:

- M. Massias, O. Fercoq, A. Gramfort, and J. Salmon. *Smoothed generalized concomitant Lasso for sparse multimodal regression. AISTATS*, 2018
- Q. Bertrand*, M. Massias*, A. Gramfort, and J. Salmon. *Handling correlated and repeated measurements with the smoothed multivariate square-root Lasso. NeurIPS*, 2019
- M. Massias*, Q. Bertrand*, A. Gramfort, and J. Salmon. *Support recovery and sup-norm convergence rates for sparse pivotal estimation. AISTATS*, 2020

Deep unfolding of proximal algorithms:

- T. Moreau, P. Ablin, M. Massias, A. Gramfort. *Learning stepsizes for the iterative soft-thresholding algorithm. NeurIPS*, 2019

Current interests: non-convex penalties, hyperparameter setting, non linear inverse problems, average case analysis of CD (random design), structure identification

References I

Aitken, A. (1926). On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305.

Anderson, D. G. (1965). Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202.

Bertrand, Q. and Massias, M. (2020). Anderson acceleration of coordinate descent.

Bollapragada, R., Scieur, D., and d'Aspremont, A. (2018). Nonlinear acceleration of momentum and primal-dual algorithms. *arXiv preprint arXiv:1810.04539*.

Brezinski, C., Redivo-Zaglia, M., and Saad, Y. (2018). Shanks sequence transformations and anderson acceleration. *SIAM Review*, 60(3):646–669.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145.

References II

Crouzeix, M. (2004). Bounds for analytical functions of matrices. *Integral Equations and Operator Theory*, 48(4):461–477.

Eddy, R. P. (1979). Extrapolating to the limit of a vector sequence. In *Information linkage between applied mathematics and industry*, pages 387–396. Elsevier.

Engl, H. W., Heinz, W., Hanke, M., and Neubauer, A. (1996). *Regularization of inverse problems*, volume 375. Springer Science & Business Media.

Fercoq, O. and Richtárik, P. (2015). Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023.

Fu, A., Zhang, J., and Boyd, S. (2019). Anderson accelerated Douglas-Rachford splitting. *arXiv preprint arXiv:1908.11482*.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. (2018). Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*.

Mai, V. V. and Johansson, M. (2019). Anderson acceleration of proximal gradient methods. In *ICML*.

References III

Massias, M., Vaiter, S., Gramfort, A., and Salmon, J. (2019). Dual extrapolation for sparse generalized linear models. *arXiv preprint arXiv:1907.05830*.

Matet, S., Rosasco, L., Villa, S., and Vu, B. L. (2017). Don't relax: early stopping for convex regularization. *arXiv preprint arXiv:1707.05422*.

Molinari, C., Massias, M., Rosasco, L., and Villa, S. (2020). Iterative regularization for convex regularizers. *arxiv preprint*.

Nesterov, Y. (1983). A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547.

Poon, C. and Liang, J. (2019). Trajectory of alternating direction method of multipliers and adaptive acceleration. In *NeurIPS*, pages 7357–7365.

Rasch, J. and Chambolle, A. (2020). Inexact first-order primal–dual algorithms. *Computational Optimization and Applications*, 76(2):381–430.

Scieur, D. (2019). Generalized framework for nonlinear acceleration. *arXiv preprint arXiv:1903.08764*.

References IV

Scieur, D., d'Aspremont, A., and Bach, F. (2016). Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720.

Sidi, A. (2017). *Vector extrapolation methods with applications*. SIAM.

Smith, D. A., Ford, W. F., and Sidi, A. (1987). Extrapolation methods for vector sequences. *SIAM review*, 29(2):199–233.

Tseng, P. and Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423.

Wynn, P. (1962). Acceleration techniques for iterated vector and matrix problems. *Mathematics of Computation*, 16(79):301–322.