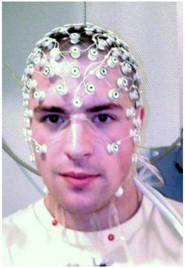
Hyperparameter selection for high dimensional sparse learning: application to neuroimaging

PhD defense

Quentin Bertrand (Inria)
https://QB3.github.io

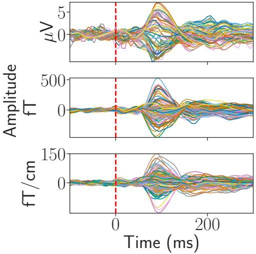
- J. Fadili (rapporteur)
- M. Pontil (rapporteur)
- C.-B. Schoënlib (examinatrice)
- K. Lounici (examinateur)
- P. Ochs (examinateur)
- J. Salmon (codirecteur)
- A. Gramfort (codirecteur)

Neuroimaging data: EEG[1] and MEG[2]



EEG

MEG



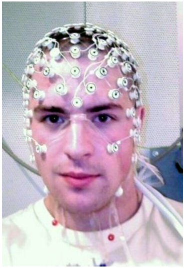
M/EEG data Y

- **Data** Y: electric and magnetic fields at the head surface

[1] H. Berger. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

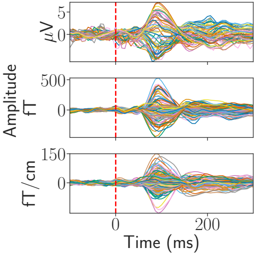
[2] D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

Neuroimaging data: EEG[1] and MEG[2]



EEG

MEG



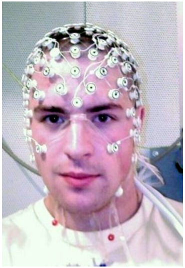
M/EEG data Y

- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?

(1) H. Berger. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

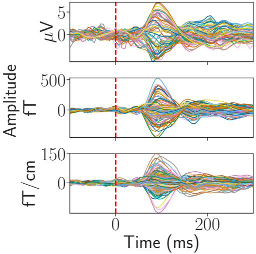
(2) D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

Neuroimaging data: EEG[1] and MEG[2]



EEG

MEG



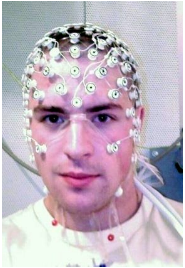
M/EEG data Y

- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?
- **Applications**: clinical and cognitive experiments

[1] H. Berger. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

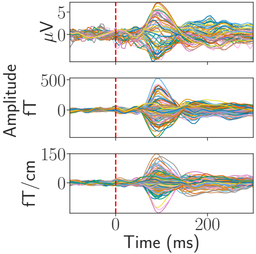
[2] D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

Neuroimaging data: EEG[1] and MEG[2]



EEG

MEG



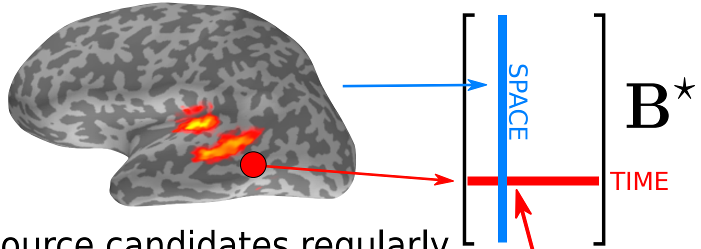
M/EEG data Y

- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?
- **Applications**: clinical and cognitive experiments

(1) H. Berger. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

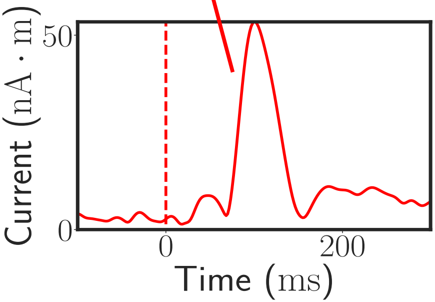
(2) D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

Source modeling

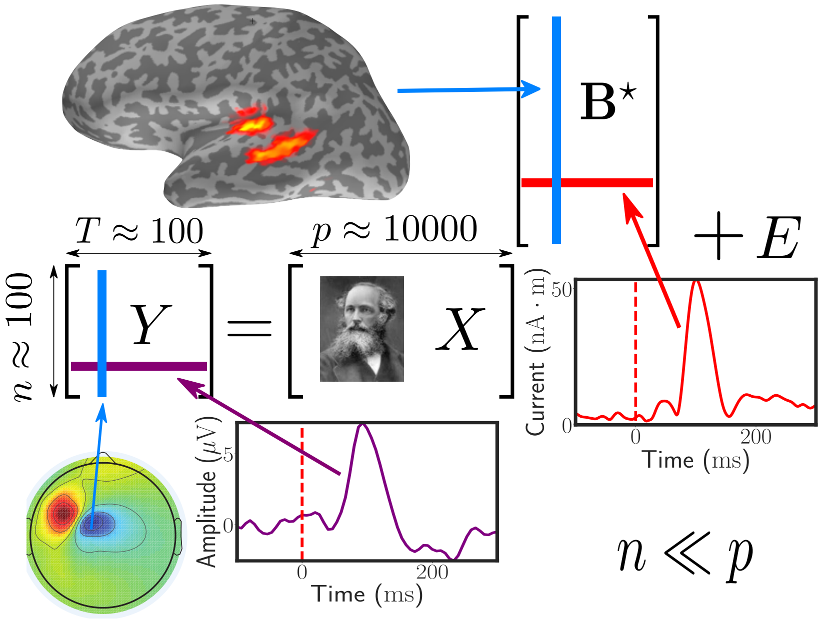


Source candidates regularly spaced in the brain (e.g., every 5mm)

$$\mathbf{B}^\star \in \mathbb{R}^{p \times T}$$



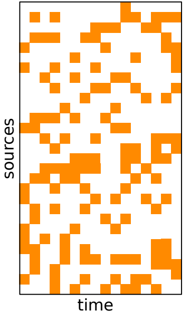
The M/EEG inverse problem



Multitask penalties[3][4]

Popular convex penalties:

$$\hat{\mathrm{B}} \in \mathop{\arg\min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \| Y - X\mathrm{B} \|_F^2 + \lambda \Omega(\mathrm{B}) \right)$$



Parameter $\hat{\mathrm{B}} \in \mathbb{R}^{p \times T}$

Sparse support: no structure

Penalty: **Lasso**

$$\Omega(\mathrm{B}) = \|\mathrm{B}\|_1 = \sum_{j=1}^{p} \sum_{k=1}^{T} |\mathrm{B}_{j,k}|$$

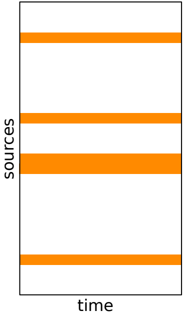
[3] A. Argyriou, T. Evgeniou, and M. Pontil. "Convex multi-task feature learning". In: *Machine Learning* (2008).

[4] A. Gramfort, M. Kowalski, and M. Hämäläinen. "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods". In: *Phys. Med. Biol.* (2012).

Multitask penalties[3][4]

Popular convex penalties: multitask Lasso (MTL)

$$\hat{\mathrm{B}} \in \mathop{\arg\min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \|Y - X\mathrm{B}\|_F^2 + \lambda \Omega(\mathrm{B}) \right)$$



Parameter $\hat{\mathrm{B}} \in \mathbb{R}^{p \times T}$

Sparse support: group structure ✓

Penalty: **Group-Lasso**

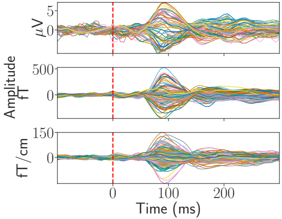
$$\Omega(\mathrm{B}) = \|\mathrm{B}\|_{2,1} = \sum_{j=1}^{p} \|\mathrm{B}_{j,:}\|_2$$

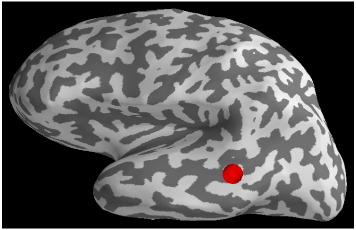
where $\mathrm{B}_{j,:}$ the j-th row of B

[3] A. Argyriou, T. Evgeniou, and M. Pontil. "Convex multi-task feature learning". In: *Machine Learning* (2008).

[4] A. Gramfort, M. Kowalski, and M. Hämäläinen. "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods". In: *Phys. Med. Biol.* (2012).

Summary of the problem setting



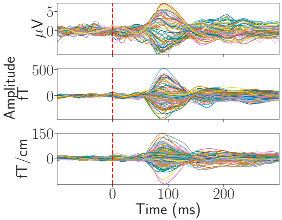


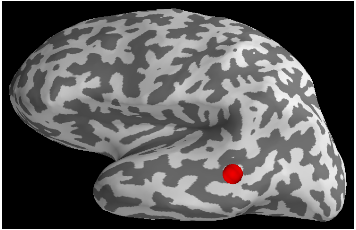
What you have: $Y \in \mathbb{R}^{n \times T}$ What you want: $\mathrm{B} \in \mathbb{R}^{p \times T}$

This is typically done using optimization based estimators

$$\hat{\mathrm{B}} \in \underset{\mathrm{B} \in \mathbb{R}^{p \times T}}{\arg\min} \left(\frac{1}{2nT} \|Y - X\mathrm{B}\|_F^2 + \lambda \Omega(\mathrm{B}) \right)$$

Summary of the problem setting





What you have: $Y \in \mathbb{R}^{n \times T}$ What you want: $\mathrm{B} \in \mathbb{R}^{p \times T}$

This is typically done using optimization based estimators

$$\hat{\mathrm{B}} \in \operatorname*{arg\,min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \| Y - X\mathrm{B} \|_F^2 + \lambda \Omega(\mathrm{B}) \right)$$

Summary of contributions

$$\hat{\mathrm{B}} \in \mathop{\arg\min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \|Y - X\mathrm{B}\|_F^2 + \lambda \Omega(\mathrm{B}) \right)$$

Covered in this presentation

- How to efficiently solve this optimization problem?(5)
- How to efficiently select the regularization parameter λ?(6),(7)

Not covered in this presentation(8),(9),(10)

---

(5) **Q. Bertrand** and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.

(6) **Q. Bertrand** et al. "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML* (2020).

(7) **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

(8) **Q. Bertrand** et al. "Handling correlated and repeated measurements with the smoothed Multivariate square-root Lasso". In: *NeurIPS* (2019).

(9) M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. "Support recovery and sup-norm convergence rates for sparse pivotal estimation". In: *AISTATS*. 2020.

(10) Q. Klopfenstein, **Q. Bertrand** et al. "Model identification and local linear convergence of coordinate descent". In: *arXiv preprint arXiv:2010.11825* (2020).

Table of Contents

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$

[11] J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

[12] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

[13] R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

[14] M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Elastic net $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2$

(11) J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

(12) F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

(13) R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

(14) M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Elastic net $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2$
- Dual SVM $\arg\min_{w \in \mathbb{R}^n} \frac{1}{2}\|(y \odot X)^\top w\|^2 - \sum w_i + \iota_{0 \leq \cdot \leq C}(w)$

(11) J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

(12) F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

(13) R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

(14) M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Elastic net $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2$
- Dual SVM $\arg\min_{w \in \mathbb{R}^n} \frac{1}{2}\|(y \odot X)^\top w\|^2 - \sum w_i + \iota_{0 \leq \cdot \leq C}(w)$
- Minimax concave penalty (MCP)

(11) J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

(12) F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

(13) R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

(14) M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Elastic net $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2$
- Dual SVM $\arg\min_{w \in \mathbb{R}^n} \frac{1}{2}\|(y \odot X)^\top w\|^2 - \sum w_i + \iota_{0 \leq \cdot \leq C}(w)$
- Minimax concave penalty (MCP)

---

(11) J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

(12) F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

(13) R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

(14) M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

Why (proximal) coordinate descent?

Efficient algorithms to solve

$$\underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{f(X\beta)}_{\text{smooth}} + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\text{separable}}$$

Examples:

- Lasso $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Elastic net $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2$
- Dual SVM $\arg\min_{w \in \mathbb{R}^n} \frac{1}{2}\|(y \odot X)^\top w\|^2 - \sum w_i + \iota_{0 \leq \cdot \leq C}(w)$
- Minimax concave penalty (MCP)

Default solver in ML packages[11],[12],[13],[14]

---

[11] J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* (2007).

[12] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

[13] R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).

[14] M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

CD on least squares

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 \quad , X \in \mathbb{R}^{n \times p}, y \in \mathbb{R}^n$$

Algorithm: Gradient descent

init : $\beta \in \mathbb{R}^p$

for $k = 0, 1, \dots,$ **do**

$\quad \beta \leftarrow \beta - \frac{X^\top(X\beta - y)}{\|X\|_2^2}$

return β

Algorithm: CD

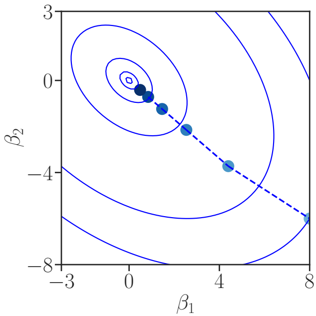
init : $\beta \in \mathbb{R}^p$

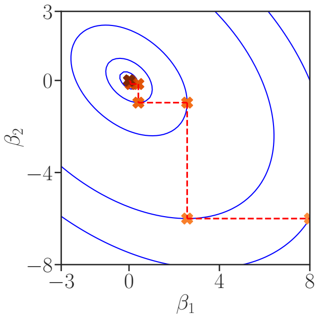
for $k = 0, 1, \dots,$ **do**

$\quad$ Select $j \in [p]$

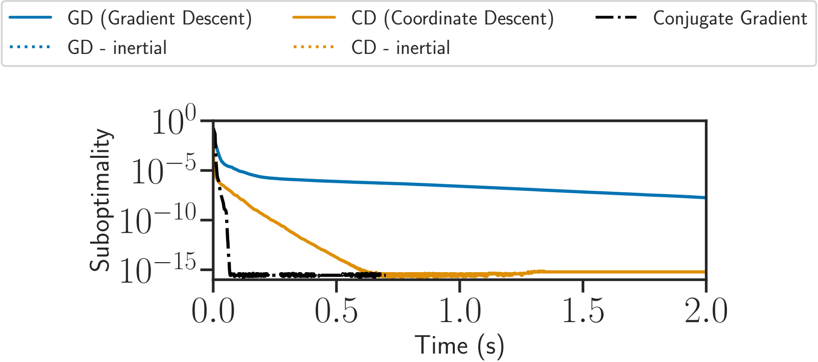
$\quad \beta_j \leftarrow \beta_j - \frac{X_{:j}^\top(X\beta - y)}{\|X_{:j}\|^2}$

return β





CD Acceleration (toy example)



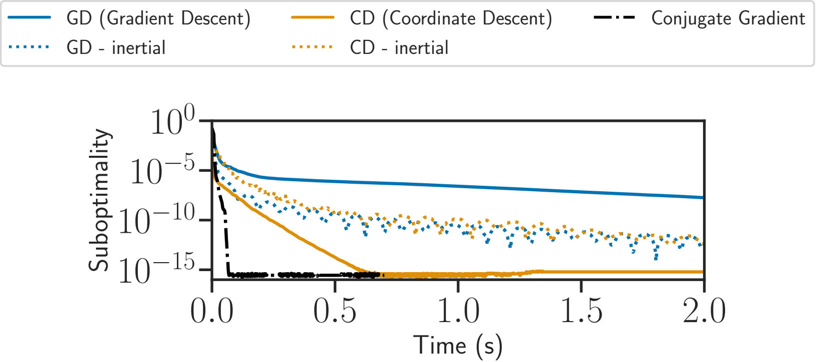
Least squares on *rcv1* ($n = p \approx 20$k)

Nesterov-like **inertial CD**[(15),(16)] may **slow down** convergence

(15) Q. Lin, Z. Lu, and L. Xiao. "An Accelerated Proximal Coordinate Gradient Method". In: *NeurIPS*. Citeseer, 2014.

(16) O. Fercoq and P. Richtárik. "Accelerated, parallel and proximal coordinate descent". In: *SIAM J. Optim.* (2015).

CD Acceleration (toy example)



Least squares on *rcv1* ($n = p \approx 20$k)

Nesterov-like **inertial CD**[(15),(16)] may **slow down** convergence

---

(15) Q. Lin, Z. Lu, and L. Xiao. "An Accelerated Proximal Coordinate Gradient Method". In: *NeurIPS*. Citeseer, 2014.

(16) O. Fercoq and P. Richtárik. "Accelerated, parallel and proximal coordinate descent". In: *SIAM J. Optim.* (2015).

Anderson acceleration: intuition

How to accelerate fixed point algorithms $\beta^{(k+1)} = T\beta^{(k)} + b$?

Idea: search a fixed point of the form $\hat{\beta} = \sum_{i=1}^{k} c_i \beta^{(i-1)}$

Anderson acceleration: intuition

How to accelerate fixed point algorithms $\beta^{(k+1)} = T\beta^{(k)} + b$?

Idea: search a fixed point of the form $\hat{\beta} = \sum_{i=1}^{k} c_i \beta^{(i-1)}$

One should have

$$\sum_{i=1}^{k} c_i \beta^{(i-1)} \approx T \sum_{i=0}^{k-1} c_i \beta^{(i-1)} + b$$

Anderson acceleration: intuition

How to accelerate fixed point algorithms $\beta^{(k+1)} = T\beta^{(k)} + b$?

Idea: search a fixed point of the form $\hat{\beta} = \sum_{i=1}^{k} c_i \beta^{(i-1)}$

One should have

$$\sum_{i=1}^{k} c_i \beta^{(i-1)} \approx T \sum_{i=0}^{k-1} c_i \beta^{(i-1)} + b$$

Choose c_i such that

$$c \in \underset{\sum_i c_i = 1}{\arg\min} \| \sum_{i=1}^{k} c_i \beta^{(i-1)} - T \sum_{i=1}^{k} c_i \beta^{(i-1)} - b \|^2$$

$$\in \underset{\sum_i c_i = 1}{\arg\min} \| \sum_{i=1}^{k} c_i \beta^{(i-1)} - \sum_{i=1}^{k} c_i \beta^{(i)} \|^2 = \| \sum_{i=1}^{k} c_i (\beta^{(i-1)} - \beta^{(i)}) \|^2$$

Anderson acceleration: intuition

How to accelerate fixed point algorithms $\beta^{(k+1)} = T\beta^{(k)} + b$?

Idea: search a fixed point of the form $\hat{\beta} = \sum_{i=1}^{k} c_i \beta^{(i-1)}$

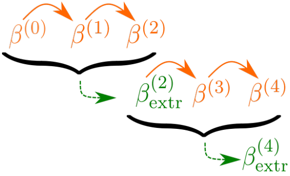
One should have

$$\sum_{i=1}^{k} c_i \beta^{(i-1)} \approx T \sum_{i=0}^{k-1} c_i \beta^{(i-1)} + b$$

Choose c_i such that

$$\begin{aligned} c &\in \underset{\sum_i c_i = 1}{\arg\min} \Big\| \sum_{i=1}^{k} c_i \beta^{(i-1)} - T \sum_{i=1}^{k} c_i \beta^{(i-1)} - b \Big\|^2 \\ &\in \underset{\sum_i c_i = 1}{\arg\min} \Big\| \sum_{i=1}^{k} c_i \beta^{(i-1)} - \sum_{i=1}^{k} c_i \beta^{(i)} \Big\|^2 = \Big\| \sum_{i=1}^{k} c_i (\beta^{(i-1)} - \beta^{(i)}) \Big\|^2 \end{aligned}$$

Anderson acceleration: algorithm[17],[18],[19]



init : $\beta^{(0)} \in \mathbb{R}^p$

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

- $\beta^{(k)} = T\beta^{(k-1)} + b$ // regular epoch
- **if** $k = 0 \mod K$ **then**
  - $U = [\beta^{(k-K+1)} - \beta^{(k-K)}, \dots,]$
  - $c = (U^\top U)^{-1} \mathbf{1}_K$ // linear system
  - $\beta^{(k)}_{\text{extr}} = \sum_i^K c_i \beta^{(k-K+i)} / \sum_i c_i$
  - $\beta^{(k)} = \beta^{(k)}_{\text{extr}}$ // sequence changes

return $\beta^{(k)}$

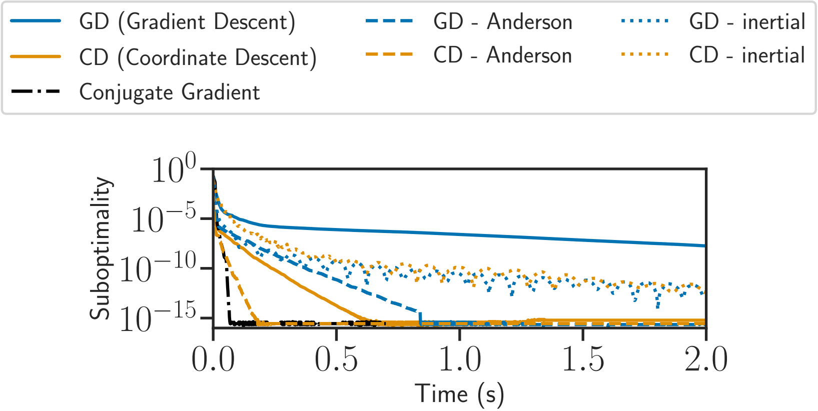
- For CD, T corresponds to one update of all the coordinates

[17] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* (1965).

[18] D. Scieur, A. d'Aspremont, and F. Bach. "Regularized nonlinear acceleration". In: *NeurIPS*. 2016.

[19] A. Sidi. *Vector extrapolation methods with applications*. SIAM, 2017.

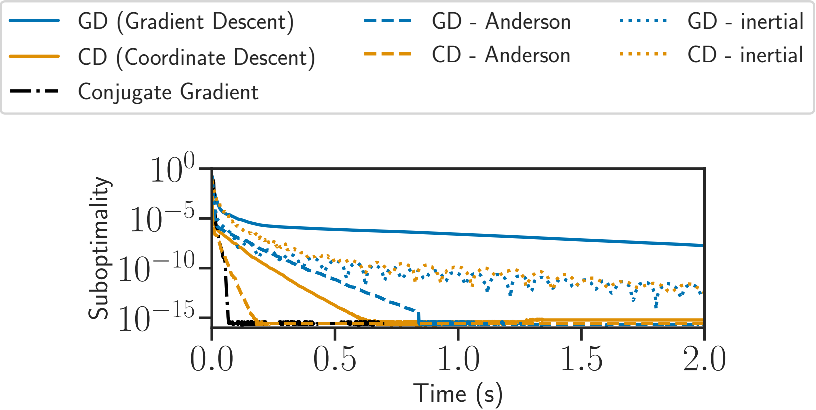
Acceleration of CD (toy example) II



Least squares on *rcv1* ($n = p \approx 20$k)

- Anderson acceleration provides speedups for CD

Acceleration of CD (toy example) II



Least squares on *rcv1* ($n = p \approx 20$k)

- Anderson acceleration provides speedups for CD

Theoretical properties

$$\beta^{(k+1)} = T\beta^{(k)} + b$$

Symmetric T

Let the iteration matrix T be symmetric semi-definite positive, with spectral radius $\rho = \rho(T) < 1$. Let $\hat{\beta}$ be the limit of the sequence $(\beta^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$. Then the iterates of Anderson acceleration satisfy ,[20] with $B = (\mathrm{Id} - T)^2$:

$$\|\beta_{\mathrm{extr}}^{(k)} - \hat{\beta}\|_B \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|\beta^{(0)} - \hat{\beta}\|_B \quad .$$

Symmetric T: gradient descent ✓
Coordinate descent?

[20] D. Scieur. "Generalized Framework for Nonlinear Acceleration". In: *arXiv preprint arXiv:1903.08764* (2019).

Theoretical properties

$$\beta^{(k+1)} = T\beta^{(k)} + b$$

Symmetric T

Let the iteration matrix T be symmetric semi-definite positive, with spectral radius $\rho = \rho(T) < 1$. Let $\hat{\beta}$ be the limit of the sequence $(\beta^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$. Then the iterates of Anderson acceleration satisfy ,[20] with $B = (\mathrm{Id} - T)^2$:

$$\|\beta_{\mathrm{extr}}^{(k)} - \hat{\beta}\|_B \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|\beta^{(0)} - \hat{\beta}\|_B \ .$$

Symmetric T: gradient descent ✓
Coordinate descent?

[20] D. Scieur. "Generalized Framework for Nonlinear Acceleration". In: *arXiv preprint arXiv:1903.08764* (2019).

Coordinate descent (CD)

- Quadratic problem, with $b \in \mathbb{R}^p$, $H \in \mathbb{S}^p_{++}$, $H \succ 0$:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \beta^\top H \beta + \langle b, \beta \rangle$$

- Updates of coordinate descent: for all $j \in 1, \ldots, p$:

$$\beta_j \leftarrow \beta_j - (H_{j:} \beta + b_j) / H_{jj}$$

- Updating all the coordinates yields a **fixed point iteration**

$$\beta^{(k+1)} = T \beta^{(k)} + v$$

with a **nonsymmetric** iteration matrix T ✗

Theoretical properties

Weaker theoretical properties for AA with non-symmetric T[21]

Non-symmetric T

Let T be the iteration matrix of pseudo-symmetric coordinate descent: $T = H^{-1/2} S H^{1/2}$, with S the symmetric positive semidefinite matrix

$$S = \left(\mathrm{Id}_p - H^{1/2} \frac{e_1 e_1^\top}{H_{11}} H^{\frac{1}{2}} \right) \times \cdots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \frac{e_p e_p^\top}{H_{pp}} H^{\frac{1}{2}} \right)$$
$$\times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \frac{e_p e_p^\top}{H_{pp}} H^{\frac{1}{2}} \right) \times \cdots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \frac{e_1 e_1^\top}{H_{11}} H^{\frac{1}{2}} \right) .$$

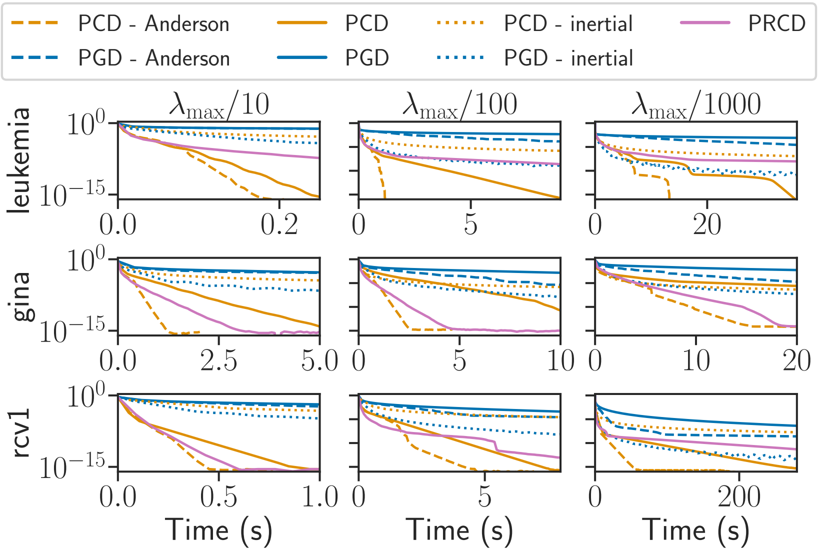
Let $\hat{\beta}$ be the limit of the sequence $(\beta^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$. Then $\rho = \rho(T) = \rho(S) < 1$ and the iterates of online extrapolation satisfy[22]:

$$\|\beta^{(k)}_{\text{extr}} - \hat{\beta}\|_B \leq \left(\sqrt{\kappa(H)} \frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}} \right)^{k/K} \|\beta^{(0)} - \hat{\beta}\|_B \ .$$

[21] R. Bollapragada, D. Scieur, and A. d'Aspremont. "Nonlinear acceleration of momentum and primal-dual algorithms". In: *AISTATS* (2018).

[22] **Q. Bertrand** and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.

Lasso

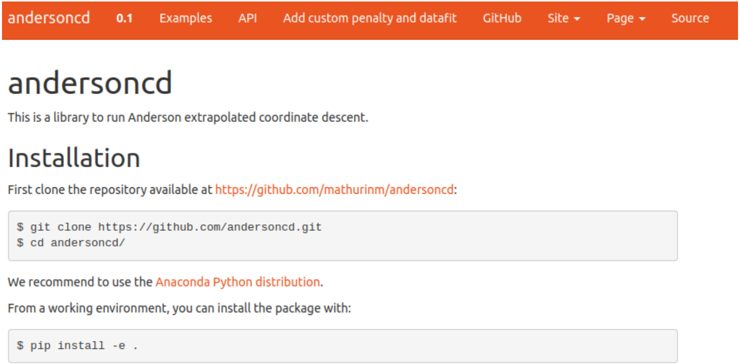


PCD - Anderson
PCD
PCD - inertial
PRCD
PGD - Anderson
PGD
PGD - inertial
$\lambda_{\max}/10$
$\lambda_{\max}/100$
$\lambda_{\max}/1000$
leukemia
gina
rcv1
10^{0}
10^{-15}
Time (s)

Dissemination

Combined with working sets strategies[23]:

- **Default solver** for sparsity based estimators in the most popular brain signal processing package `MNE`[24]
- Open source and **modular** package `andersoncd`



andersoncd 0.1 Examples API Add custom penalty and datafit GitHub Site Page Source

andersoncd

This is a library to run Anderson extrapolated coordinate descent.

Installation

First clone the repository available at https://github.com/mathurinm/andersoncd:

```
$ git clone https://github.com/andersoncd.git
$ cd andersoncd/
```

We recommend to use the Anaconda Python distribution.

From a working environment, you can install the package with:

```
$ pip install -e .
```

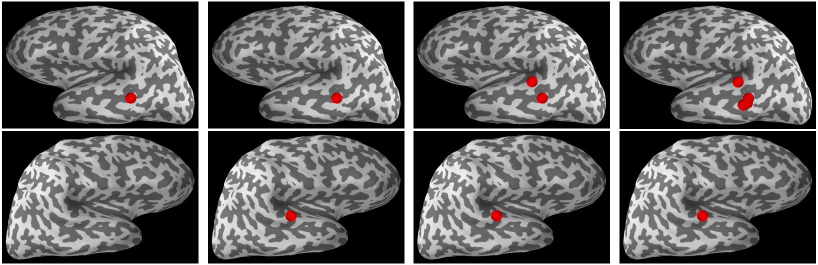
[23] J. Fan and J. Lv. “Sure independence screening for ultrahigh dimensional feature space”. In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* (2008).

[24] A. Gramfort et al. “MNE software for processing MEG and EEG data”. In: *NeuroImage* (2014).

Table of Contents

Which λ to pick?

$$\hat{B} \in \arg\min_{B \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \|Y - XB\|_F^2 + \lambda \|B\|_{2,1} \right)$$



$\lambda = 0.85\lambda_{\max}$ $\lambda = 0.82\lambda_{\max}$ $\lambda = 0.80\lambda_{\max}$ $\lambda = 0.75\lambda_{\max}$

Real MEEG data. Brain source reconstruction using multitask Lasso with multiple λ. Which λ to pick? How to *automatically* select λ?

- When $\lambda \geq \lambda_{\max}$, $\hat{B} = 0$ no sources are recovered

Model selection techniques

- Statistical route[25],[26]:
  assumptions on the design matrix X
- Bayesian statistics[27],[28]:
  prior on λ
- Hyperparameter optimization[29],[30]:
  minimize a given criterion $\mathcal{C}(\hat{\beta}^{(\lambda)})$

---

[25] K. Lounici. "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators". In: *Electron. J. Stat.* (2008).

[26] K. Lounici et al. "Taking Advantage of Sparsity in Multi-Task Learning". In: *arXiv preprint arXiv:0903.1468* (2009).

[27] M. E. Tipping. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of Machine Learning Research* (2001).

[28] M. Figueiredo. "Adaptive Sparseness Using Jeffreys Prior.". In: *NeurIPS*. 2001.

[29] R. Kohavi and G. H. John. "Automatic parameter selection by minimizing estimated error". In: *Machine Learning Proceedings*. 1995.

[30] F. Hutter, J. Lücke, and L. Schmidt-Thieme. "Beyond manual tuning of hyperparameters". In: *KI-Künstliche Intelligenz* (2015).

Hyperparameter optimization (HO)

Possible selection criterion:

- Good generalization[31],[32] of $\hat{\beta}^{(\lambda)}$
- AIC/BIC,[33] SURE[34] that controls model complexity

---

[31] L. R. A. Stone and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.

[32] K. Lounici, K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).

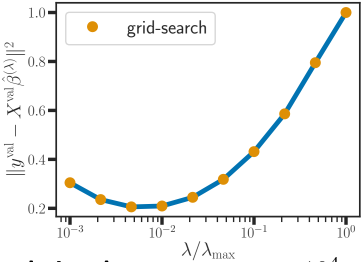
[33] W. Liu, Y. Yang, et al. "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.

[34] C. M. Stein. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.

Hyperparameter optimization (HO)

Possible selection criterion:

- Good generalization[31],[32] of $\hat{\beta}^{(\lambda)}$
- AIC/BIC,[33] SURE[34] that controls model complexity



Real-sim dataset, $n \approx p \approx 10^4$
Validation loss as a function of λ.

Example

Model: Lasso

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{\|y^{\text{train}} - X^{\text{train}}\beta\|^2}{2n} + \lambda\|\beta\|_1$$

Criterion: held-out loss

$$\arg\min_{\lambda} \|y^{\text{val}} - X^{\text{val}}\hat{\beta}^{(\lambda)}\|^2$$

[31] L. R. A. Stone and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.

[32] K. Lounici, K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).

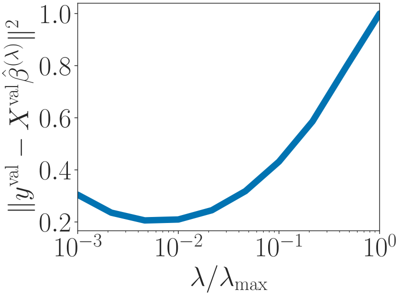
[33] W. Liu, Y. Yang, et al. "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.

[34] C. M. Stein. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.

HO as a bilevel optimization problem[35][36]

$$\overbrace{\underset{\lambda \in \mathbb{R}}{\arg\min} \left\{ \mathcal{L}(\lambda) := \| y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)} \|^2 \right\}}^{\text{outer optimization problem}}$$

$$\text{s.t. } \underbrace{\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \| \beta \|_1}_{\text{inner optimization problem}}$$



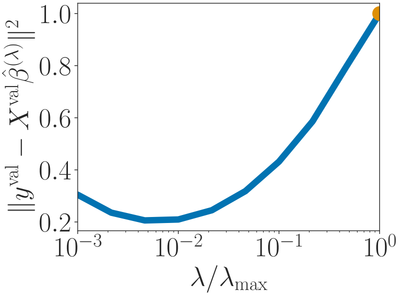
[35] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[36] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem[35][36]

$$\overbrace{\underset{\lambda \in \mathbb{R}}{\arg\min} \left\{ \mathcal{L}(\lambda) := \| y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)} \|^2 \right\}}^{\text{outer optimization problem}}$$

$$\text{s.t. } \underbrace{\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$



[35] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[36] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

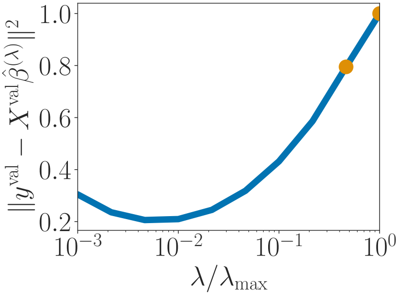
HO as a bilevel optimization problem[35][36]

outer optimization problem

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \| y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)} \|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \|\beta\|_1$$

inner optimization problem



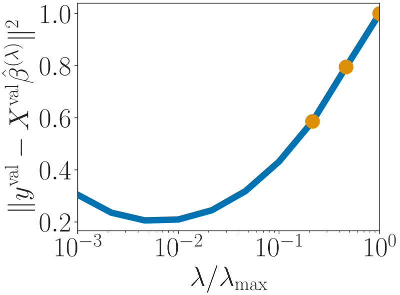
[35] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[36] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem[35][36]

$$\underbrace{\overbrace{\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}^{\text{outer optimization problem}}}$$

$$\text{s.t. } \underbrace{\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$



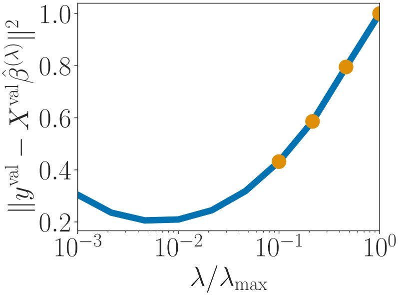
[35] P. Ochs et al. “Bilevel optimization with nonsmooth lower level problems”. In: *SSVM*. 2015.

[36] F. Pedregosa. “Hyperparameter optimization with approximate gradient”. In: *ICML*. 2016.

HO as a bilevel optimization problem[35][36]

$$\overbrace{\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}^{\text{outer optimization problem}}$$

$$\text{s.t. } \underbrace{\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$



(35) P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

(36) F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

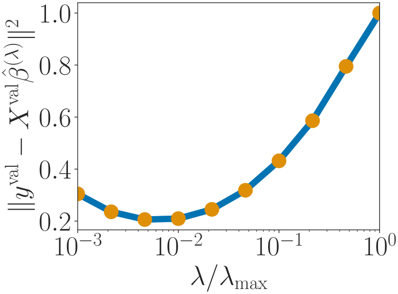
HO as a bilevel optimization problem[35][36]

outer optimization problem

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \| y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)} \|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \|\beta\|_1$$

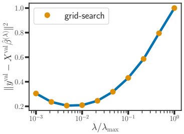
inner optimization problem



[35] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[36] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

Grid-search as a 0-order optimization method



$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

- Grid-search, random-search,[37] SMBO[38]: 0-order methods to solve bilevel optimization problem
- **Idea:** if $\mathcal{L}$ is differentiable, use first-order optimization, *i.e.*, compute $\nabla_\lambda \mathcal{L}$
- Once $\nabla_\lambda \mathcal{L}(\lambda)$ is computed, use gradient descent[39]:

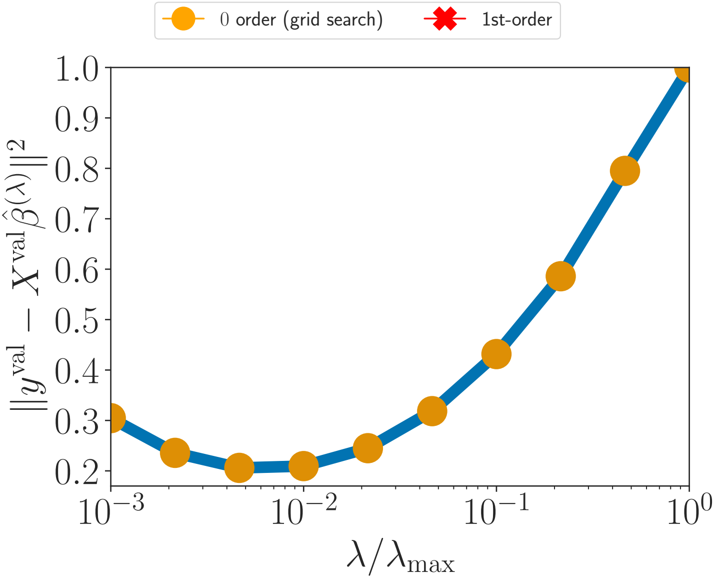
$$\lambda^{(t+1)} = \lambda^{(t)} - \rho \nabla_\lambda \mathcal{L}(\lambda^{(t)}) \quad \text{with } \rho > 0$$

(37) J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* (2012).

(38) E. Brochu, V. M. Cora, and N. De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).

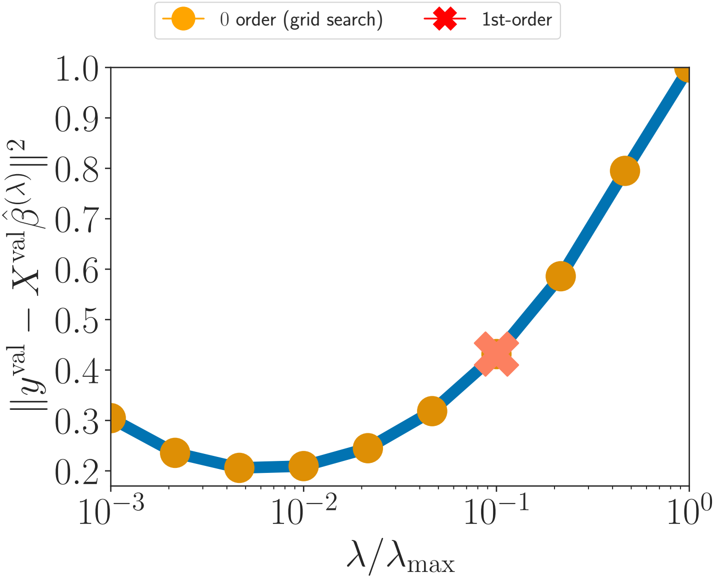
(39) F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

First-order optimization in λ, Lasso



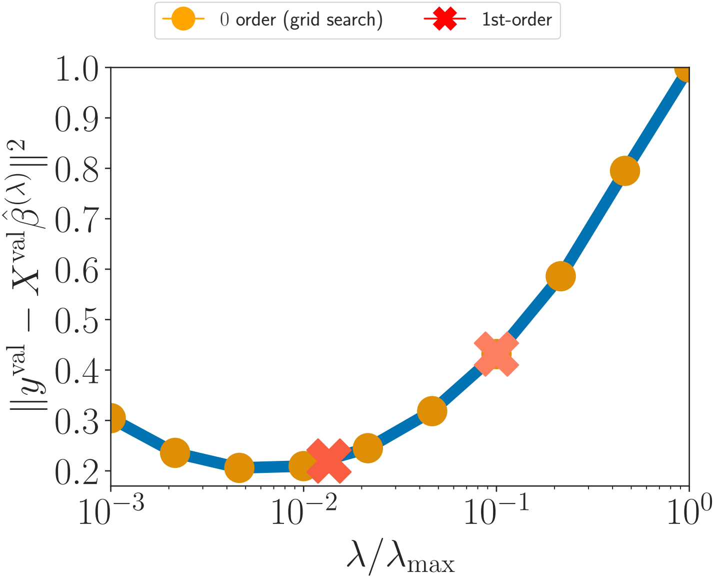
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



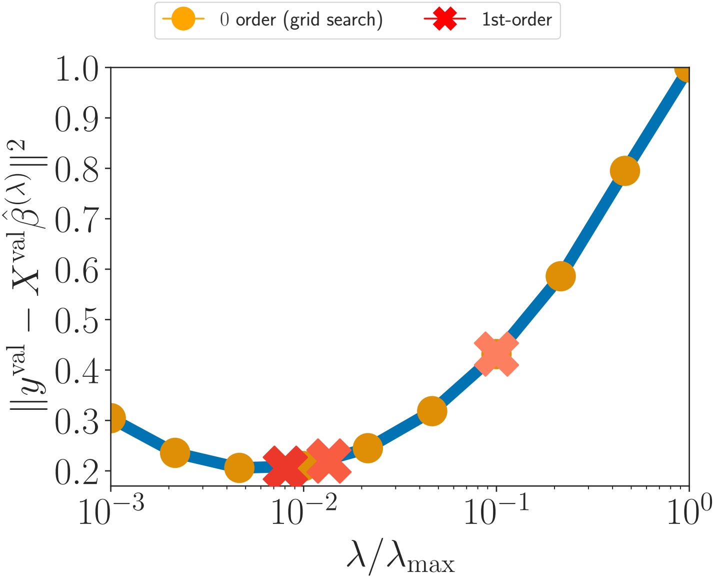
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



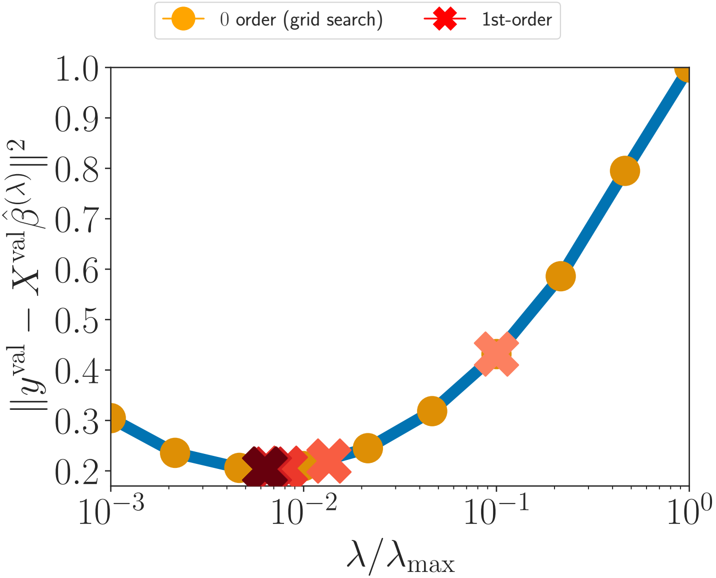
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



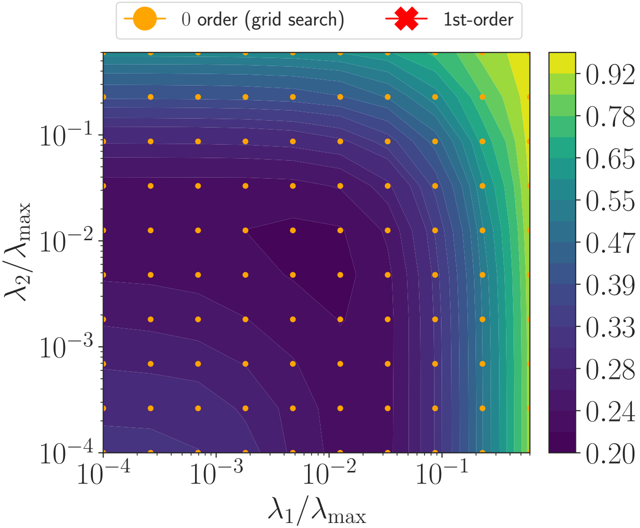
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

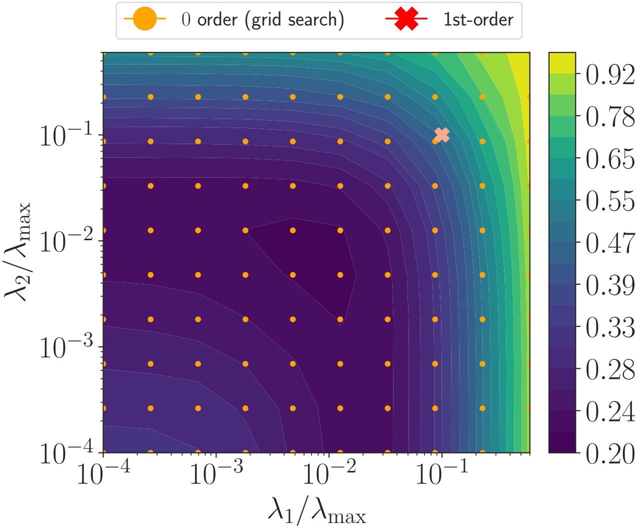
First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$$

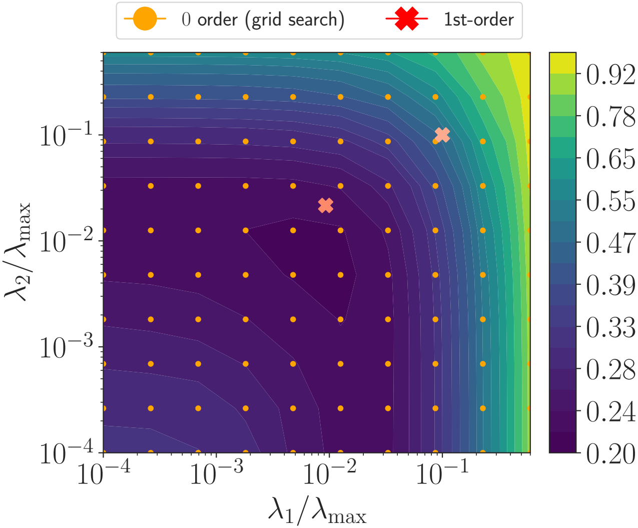
First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$$

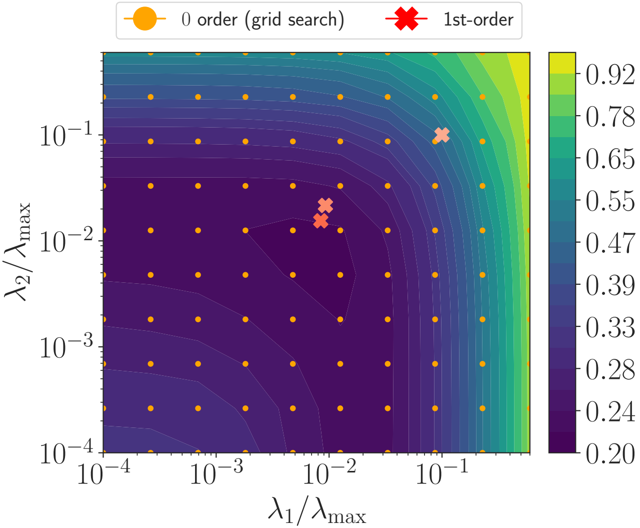
First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$$

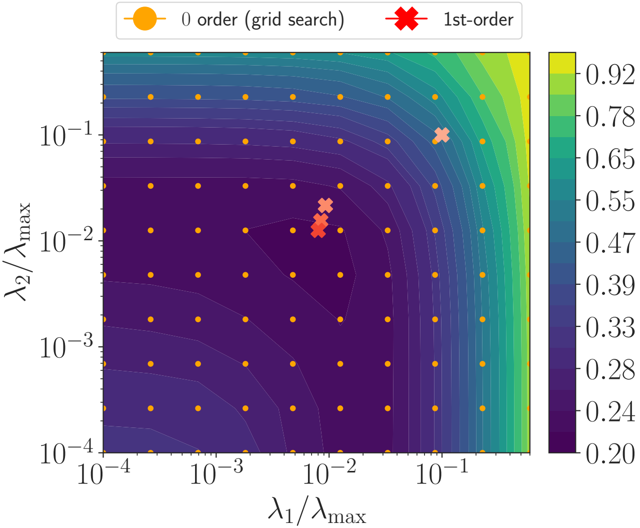
First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$$

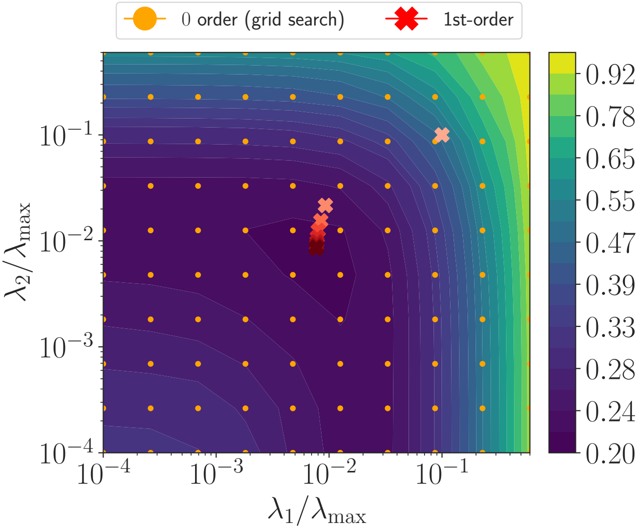
First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$$

First-order optimization in λ, Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|^2$

What's hard? Computing $\nabla_\lambda \mathcal{L}(\lambda)$

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Once $\nabla_\lambda \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- Line-search[40]
- L-BFGS[41]
- Gradient descent

[40] J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

[41] D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

What's hard? Computing $\nabla_\lambda \mathcal{L}(\lambda)$

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\mathsf{val}} - X^{\mathsf{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\mathsf{train}} - X^{\mathsf{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Once $\nabla_\lambda \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- Line-search[40]
- L-BFGS[41]
- Gradient descent

Main challenge: compute $\nabla_\lambda \mathcal{L}(\lambda)$ for a given λ

---

(40) J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

(41) D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

What's hard? Computing $\nabla_\lambda \mathcal{L}(\lambda)$

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\mathsf{val}} - X^{\mathsf{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\mathsf{train}} - X^{\mathsf{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Once $\nabla_\lambda \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- Line-search[40]
- L-BFGS[41]
- Gradient descent

Main challenge: compute $\nabla_\lambda \mathcal{L}(\lambda)$ for a given λ

[40] J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

[41] D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

How to compute $\nabla_\lambda \mathcal{L}(\lambda)$?

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Chain rule:

$$\nabla_\lambda \mathcal{L}(\lambda) = \underbrace{\hat{\mathcal{J}}_{(\lambda)}^\top}_{:=(\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})} \nabla_\beta C(\hat{\beta}^{(\lambda)})$$

$\rightarrow$main challenge

- Boils down to:

  how to compute the Jacobian $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times 1}$ efficiently?

How to compute $\nabla_\lambda \mathcal{L}(\lambda)$?

$$\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Chain rule:

$$\nabla_\lambda \mathcal{L}(\lambda) = \underbrace{\hat{\mathcal{J}}_{(\lambda)}^\top}_{\substack{:=(\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)}) \\ \to \text{main challenge}}} \nabla_\beta C(\hat{\beta}^{(\lambda)})$$

- Boils down to:

  how to compute the Jacobian $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times 1}$ efficiently?

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})^\top$?

$$\underbrace{\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \frac{\lambda}{2} \| \beta \|^2}_{\text{inner optimization problem}}$$

Smooth inner optimization problems, **well studied**:

- *Implicit differentiation* (**closed-form** formula)[42] [43]: need to solve a $p \times p$ linear system ($p = \#$features)
- *Automatic differentiation*, *reverse*[44] or *forward*[45] mode

---

[42] J. Larsen et al. "Design and regularization of neural networks: the optimal use of a validation set". In: *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. 1996.

[43] Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

[44] J. Domke. "Generic methods for optimization-based modeling". In: *AISTATS*. vol. 22. 2012.

[45] L. Franceschi et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_\lambda \hat\beta_1^{(\lambda)}, \dots, \nabla_\lambda \hat\beta_p^{(\lambda)})^\top$?

$$\underbrace{\hat\beta^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

Nonsmooth inner optimization problems, **scarcer literature**:

- *Smooth the nonsmooth term*[46]
- Use algorithms with differentiable updates[47],[48] (Bregman)

Our contributions:

- Iterative differentiation can be applied on proximal algorithms
- $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_\lambda \hat\beta_1^{(\lambda)}, \dots, \nabla_\lambda \hat\beta_p^{(\lambda)})^\top$ shares $\hat\beta^{(\lambda)}$'s **sparsity pattern**

[46] G. Peyré and J. Fadili. "Learning analysis sparsity priors". In: *Sampta*. 2011.

[47] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[48] J. Frecon, S. Salzo, and M. Pontil. "Bilevel learning of the group lasso structure". In: *NeurIPS*. 2018.

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})^\top$?

$$\underbrace{\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

Nonsmooth inner optimization problems, **scarcer literature**:

- *Smooth the nonsmooth term*[46]
- Use algorithms with differentiable updates[47],[48] (Bregman)

Our contributions:

- Iterative differentiation can be applied on proximal algorithms
- $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})^\top$ shares $\hat{\beta}^{(\lambda)}$'s **sparsity pattern**

---

[46] G. Peyré and J. Fadili. "Learning analysis sparsity priors". In: *Sampta*. 2011.

[47] P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

[48] J. Frecon, S. Salzo, and M. Pontil. "Bilevel learning of the group lasso structure". In: *NeurIPS*. 2018.

Forward-mode differentiation[49],[50] of PGD

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Proximal gradient descent PGD

init : $\beta = 0_p, \quad , L$

for iter $= 1, \dots,$ **do**

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ `// gradient step`

$\beta \leftarrow \text{prox}_{\lambda g / L}(z)$ `// proximal step`

return β

[49] R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

[50] C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation[49],[50] of PGD

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for iter $= 1, \dots,$ **do**

- $z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ `// gradient step`
- $dz \leftarrow \left(\mathrm{Id}_p - \frac{1}{L} \nabla^2 f(\beta)\right) \mathcal{J}$ `// diff w.r.t. λ: chain rule`
- $\beta \leftarrow \mathrm{prox}_{\lambda g / L}(z)$ `// proximal step`

return β

[49] R. E. Wengert. “A simple automatic derivative evaluation program”. In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

[50] C.-A. Deledalle et al. “Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection”. In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation[49],[50] of PGD

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for iter $= 1, \dots,$ **do**

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ // gradient step

$dz \leftarrow \left(\mathrm{Id}_p - \frac{1}{L} \nabla^2 f(\beta)\right) \mathcal{J}$ // diff *w.r.t.* λ: chain rule

$\beta \leftarrow \mathrm{prox}_{\lambda g / L}(z)$ // proximal step

$\mathcal{J} \leftarrow \partial_z \mathrm{prox}_{\lambda g / L}(z) dz$ // diff *w.r.t.* λ: chain rule

return β, $\mathcal{J}$

[49] R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

[50] C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation[49],[50] of PGD

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for $\mathrm{iter} = 1, \dots,$ **do**

$z \leftarrow \beta - \frac{1}{L}\nabla f(\beta)$ // gradient step

$dz \leftarrow \left(\mathrm{Id}_p - \frac{1}{L}\nabla^2 f(\beta)\right)\mathcal{J}$ // diff $w.r.t.$ λ: chain rule

$\beta \leftarrow \mathrm{prox}_{\lambda g/L}(z)$ // proximal step

$\mathcal{J} \leftarrow \partial_z \,\mathrm{prox}_{\lambda g/L}(z) dz$ // diff $w.r.t.$ λ: chain rule

$\quad + \partial_\lambda \,\mathrm{prox}_{\lambda g/L}(z)$ // do not forget this term!

return β, $\mathcal{J}$

[49] R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

[50] C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

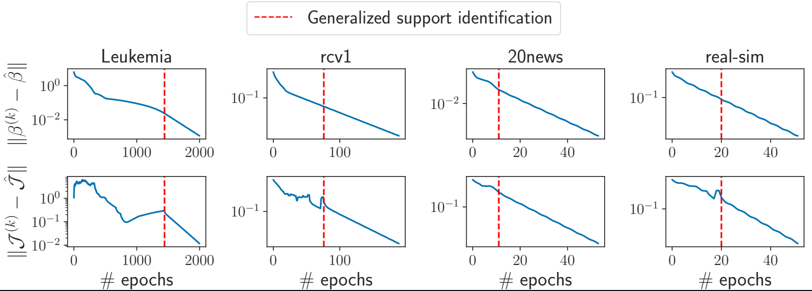
Local linear convergence of the Jacobian

Forward diff. PCD convergence, Lasso

Assume

- The sequence $(\beta^{(k)})$ generated by PCD converges to $\hat{\beta}$
- The problem is not degenerated: $-X^\top(X\hat{\beta} - y) \in \mathrm{ri}(\lambda\partial\|\cdot\|_1)$
- Restricted injectivity holds: $X_{:\mathcal{S}}^\top X_{:\mathcal{S}} \succ 0$

Then the Jacobian sequence based on forward diff. of PCD converges to the true Jacobian. Once the support (the non-zeros coefs.) has been identified, convergence is linear.[51]



[51] **Q. Bertrand** et al. "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML* (2020).

Implicit differentiation (smooth ψ)[52]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_\beta \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

[52] Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)[52]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \psi\left(\beta, \lambda\right)$$

$$\nabla_\beta \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

$$\nabla^2_{\beta,\lambda} \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}^\top_{(\lambda)} \nabla^2_\beta \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

[52] Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)[52]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_\beta \psi \left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

$$\nabla^2_{\beta,\lambda} \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}^\top_{(\lambda)} \nabla^2_\beta \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

$$\hat{\mathcal{J}}^\top_{(\lambda)} = -\nabla^2_{\beta,\lambda} \psi \left(\hat{\beta}^{(\lambda)}, \lambda\right) \underbrace{\left(\nabla^2_\beta \psi(\beta^{(\lambda)}, \lambda)\right)}_{p \times p}^{-1}$$

- Need to solve a linear **system of size** p
- Prohibitive for large p

[52] Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)[52]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_\beta \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

$$\nabla^2_{\beta,\lambda} \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}_{(\lambda)}^\top \nabla^2_\beta \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

$$\hat{\mathcal{J}}_{(\lambda)}^\top = -\nabla^2_{\beta,\lambda} \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) \underbrace{\left(\nabla^2_\beta \psi(\beta^{(\lambda)}, \lambda)\right)}_{p \times p}^{-1}$$

- Need to solve a linear **system of size p**
- Prohibitive for large p

[52] Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation $(f + \lambda \sum_j |\beta_j|)$[53]

$$\hat\beta^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat\beta^{(\lambda)} = \mathrm{ST}\left(\hat\beta^{(\lambda)} - \frac{1}{L}\nabla f(\hat\beta^{(\lambda)}), \frac{\lambda}{L}\right)$$

[53] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

Implicit differentiation $(f + \lambda \sum_j |\beta_j|)$[53]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

$$\hat{\mathcal{J}} = \partial_\beta \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)\left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}}$$
$$+ \partial_\lambda \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

[53] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

Implicit differentiation $\left(f + \lambda \sum_j |\beta_j|\right)$[53]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

$$\hat{\mathcal{J}} = \partial_\beta \, \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)\left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} + \partial_\lambda \, \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$:

$$\partial_\beta \, \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right) = 0 = \partial_\lambda \, \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

[53] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

Implicit differentiation $(f + \lambda \sum_j |\beta_j|)$[53]

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

$$\begin{aligned}\hat{\mathcal{J}} = {} & \partial_\beta \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)\left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} \\ & + \partial_\lambda \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)\end{aligned}$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$:

$$\partial_\beta \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right) = 0 = \partial_\lambda \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

With $\mathcal{S} = \left\{j \in [p] : \hat{\beta}_j^{(\lambda)} = 0\right\}$ we have $\hat{\mathcal{J}}_{\mathcal{S}^c} = 0$

$$\hat{\mathcal{J}}_{\mathcal{S}} = \partial_\beta \mathrm{ST}(\hat{\beta}^{(\lambda)} - \tfrac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \tfrac{\lambda}{L})_{\mathcal{S}} \hat{\mathcal{J}}_{\mathcal{S}} + \partial_\lambda \mathrm{ST}(\hat{\beta}_j^{(\lambda)} - \tfrac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \tfrac{\lambda}{L})_{\mathcal{S}}$$

---

[53] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

Implicit differentiation $(f + \lambda \sum_j |\beta_j|)$[53]

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

$$\hat{\mathcal{J}} = \partial_\beta \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)\left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} + \partial_\lambda \mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$:

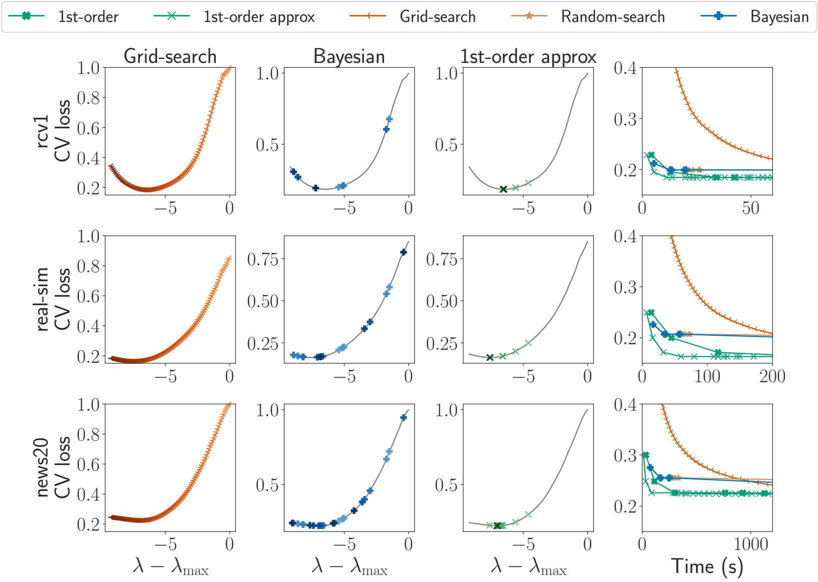
$$\partial_\beta \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right) = 0 = \partial_\lambda \mathrm{ST}\left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

With $\mathcal{S} = \left\{j \in [p] \, : \, \hat{\beta}_j^{(\lambda)} = 0\right\}$ we have $\hat{\mathcal{J}}_{\mathcal{S}^c} = 0$

$$\hat{\mathcal{J}}_{\mathcal{S}} = \partial_\beta \mathrm{ST}(\hat{\beta}^{(\lambda)} - \tfrac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \tfrac{\lambda}{L})_{\mathcal{S}} \hat{\mathcal{J}}_{\mathcal{S}} + \partial_\lambda \mathrm{ST}(\hat{\beta}_j^{(\lambda)} - \tfrac{1}{L}\nabla_j f(\hat{\beta}^{(\lambda)}), \tfrac{\lambda}{L})_{\mathcal{S}}$$

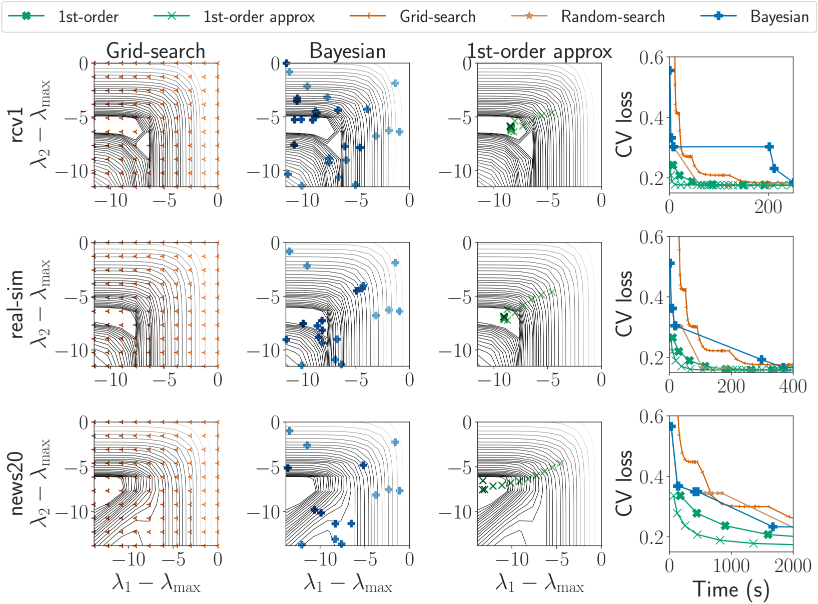
[53] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

Experiments I - Lasso cross-validation



$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + e^{\lambda}\|\beta\|_1$$

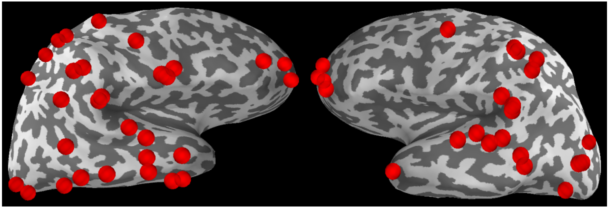
Experiments II - Enet cross-validation



$$\arg\min_\beta \frac{1}{2n}\|y^{\text{train}} - X^{\text{train}}\beta\|^2 + e^{\lambda_1}\|\beta\|_1 + \frac{e^{\lambda_2}}{2}\|\beta\|^2$$

Experiments III - Real MEEG data

- **Outer criterion:** FDMC SURE[54]
- **Inner problems:** vanilla Lasso



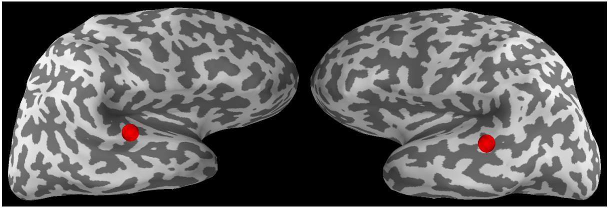
Real M/EEG data, vanilla Lasso (1 hyperparameter λ)

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} ||y - X\beta||_2^2 + e^{\lambda} \|\beta\|_1$$

[54] C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Experiments III - Real MEEG data

- **Outer criterion:** FDMC SURE[54]
- **Inner problems:** weighted Lasso



Real M/EEG data, weighted Lasso (p hyperparameters)

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} ||y - X\beta||_2^2 + \sum_{j=1}^{p} e^{\lambda_j} |\beta_j|$$

[54] C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Limitations

- Specific parametrization e^{λ}
- Need a **differentiable criterion**: cannot use 0/1-loss
- Need a **continuous estimator** *w.r.t.* data and hyperparameters: does not apply yet to **non-convex** penalties[55] [56]
- Optimized function often **non-convex**: possibly multiple local minima
- Hard to calibrate **nested *for* loops**

(55) P. Breheny and J. Huang. "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection". In: *Ann. Appl. Stat.* (2011).

(56) E. J. Candès, M. B. Wakin, and S. P. Boyd. "Enhancing Sparsity by Reweighted l_1 Minimization". In: *J. Fourier Anal. Applicat.* (2008).

Contributions and dissemination

- **Local linear convergence** of the Jacobian
- **Leverage sparsity** to speed up hypergradient computation
- Open source package
  https://github.com/QB3/sparse-ho

sparse-ho   0.1.dev   Examples   API   GitHub   Site

sparse-ho

build passing   codecov 79%

sparse-ho stands for "sparse hyperparameter optimization". This package implements efficient hyperparameter tuning for sparse machine learning models. It supports models such as the Lasso, the Weighted Lasso, multiclass sparse Logistic regression, SVM, etc.

Relying on a first order algorithm for bilevel optimization, `sparse-ho`'s performances scales gracefully with the number of hyperparameters to tune.

In order to benchmark performances, the package also implements alternatives such as forward or backward differentiation.

Documentation

Please visit `https://qb3.github.io/sparse-ho`_ for the latest version of the documentation.

Install

To run the code you first need to clone the repository, and then run, in the folder containing the `setup.py` file (root folder):

```
pip install -e .
```

Summary of contributions

$$\hat{B} \in \arg\min_{B \in \mathbb{R}^{p \times T}} \left(\frac{1}{2nT} \|Y - XB\|_F^2 + \lambda \Omega(B) \right)$$

Covered in this presentation

- How to efficiently solve this optimization problem?[57]
- How to efficiently select the regularization parameter λ?[58][59]

Not covered in this presentation[60][61][62]

---

[57] **Q. Bertrand** and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.

[58] **Q. Bertrand** et al. "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML* (2020).

[59] **Q. Bertrand** et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR* (2021).

[60] **Q. Bertrand** et al. "Handling correlated and repeated measurements with the smoothed Multivariate square-root Lasso". In: *NeurIPS* (2019).

[61] M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. "Support recovery and sup-norm convergence rates for sparse pivotal estimation". In: *AISTATS*. 2020.

[62] Q. Klopfenstein, **Q. Bertrand** et al. "Model identification and local linear convergence of coordinate descent". In: *arXiv preprint arXiv:2010.11825* (2020).

Perspectives I, `andersoncd`

Name	Fast	Modular	sk API	Nncvx	Language
`glmnet`[63]	+	✗	✗	✗	Fortran
`scikit-learn`[64]	+	✗	✓	✗	cython
`lightning`[65]	+	✓	✓	✗	cython
`celer`[66]	++	✗	✓	✗	cython
`picasso`[67]	++	✗	✗	✓	C++
`pyGLMnet`[68]	−	✓	✓	✗	python
remains to be done	++	✓	✓	✓	python

Existing packages for linear models

[63] J. Friedman, T. J. Hastie, and R. Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *J. Stat. Softw.* (2010).

[64] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011).

[65] M. Blondel and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.

[66] M. Massias et al. "Dual Extrapolation for Sparse Generalized Linear Models". In: *Journal of Machine Learning Research* (2020).

[67] J. Ge et al. "Picasso: A sparse learning library for high dimensional data analysis in R and Python". In: *The Journal of Machine Learning Research* (2019).

[68] M. Jas et al. "Pyglmnet: Python implementation of elastic-net regularized generalized linear models". In: *Journal of Open Source Software* (2020).

Perspectives II, bilevel optimization

- For smooth inner problems, HO packages exist[69][70] ....
- But practitioners mostly rely on 0-order methods[71][72]

Main problems

- Hard to tune *hyperhyperparameters*
- Hard to calibrate nested *for* loops

What I propose

- Study bilevel optimization through the lens of games theory[73]

---

(69) F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

(70) L. Franceschi et al. "Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning". In: *arXiv preprint arXiv:1806.04941* (2018).

(71) L. Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *Journal of Machine Learning Research* (2017).

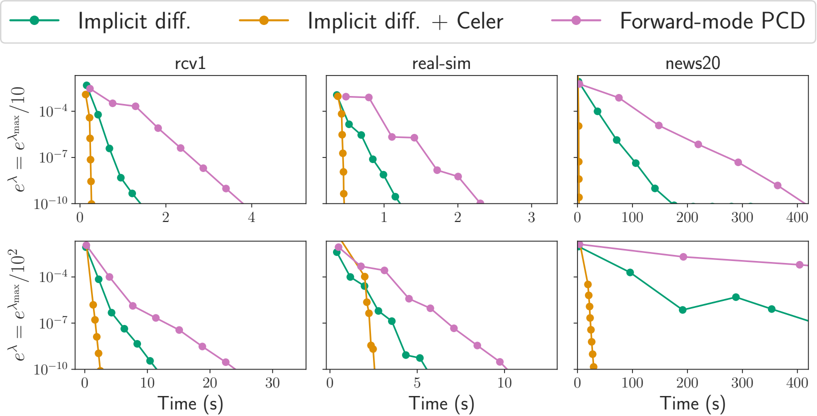
(72) T. Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.

(73) H. von Stackelberg. *Marktform und Gleichgewicht*. J. Springer, 1934.

Thank you!

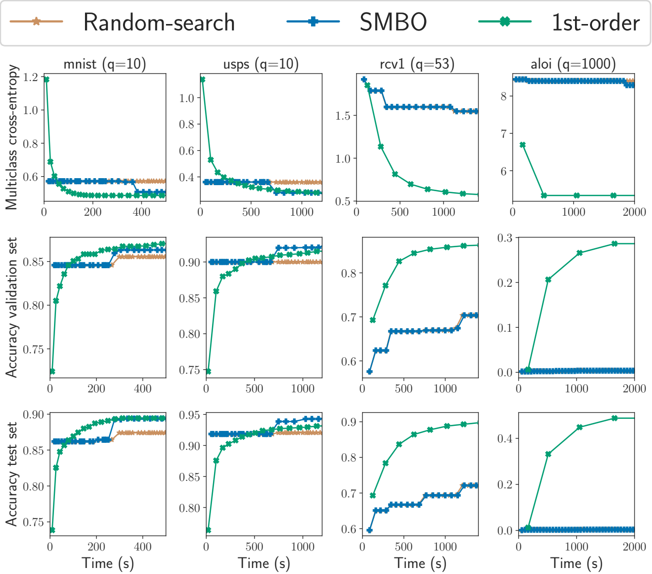
Alexandre, Joseph, Samuel, Mathieu, Mathurin, Quentin K. and Pierre-Antoine

Backup - Implicit vs forward-mode



Lasso with hold-out criterion: absolute difference between the exact hypergradient (using $\hat{\beta}$) and the iterate hypergradient (using $\beta^{(k)}$) of the Lasso as a function of time.

Backup - Multiclass logistic regression



Multiclass sparse logistic regression hold-out, time comparison (# classes = # hyperparameters).

Backup - Outer procedure

Algorithm: OUTER PROCEDURE

input : $\lambda \in \mathbb{R}^r, (\epsilon_i)$
init : use_adaptive_step_size = True
for $i = 1, \dots, \text{iter}$ **do**
    $\lambda^{\text{old}} \leftarrow \lambda$
    // compute the value and the gradient
    $\mathcal{L}(\lambda), \nabla \mathcal{L}(\lambda) \leftarrow$ Implicit diff$(X, y, \lambda, \epsilon_i)$
    if use_adaptive_step_size **then**
        $\alpha = 1 / \|\nabla \mathcal{L}(\lambda)\|$
    // gradient step
    $\lambda -= \alpha \nabla \mathcal{L}(\lambda)$
    if $\mathcal{L}(\lambda) > \mathcal{L}(\lambda^{\text{old}})$ **then**
        use_adaptive_step_size = False
        $\alpha /= 10$
return λ

- Akiba, T. et al. “Optuna: A next-generation hyperparameter optimization framework”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.
- Anderson, D. G. “Iterative procedures for nonlinear integral equations”. In: *Journal of the ACM* (1965).
- Argyriou, A., T. Evgeniou, and M. Pontil. “Convex multi-task feature learning”. In: *Machine Learning* (2008).
- Bengio, Y. “Gradient-based optimization of hyperparameters”. In: *Neural computation* (2000).
- Berger, H. “Über das elektroenkephalogramm des menschen”. In: *Archiv für psychiatrie und nervenkrankheiten* (1929).
- Bergstra, J. and Y. Bengio. “Random search for hyper-parameter optimization”. In: *Journal of Machine Learning Research* (2012).
- **Bertrand, Q.** and M. Massias. “Anderson acceleration of coordinate descent”. In: *AISTATS*. 2021.

- **Bertrand**, **Q.** et al. “Handling correlated and repeated measurements with the smoothed Multivariate square-root Lasso”. In: *NeurIPS* (2019).
- **Bertrand**, **Q.** et al. “Implicit differentiation for fast hyperparameter selection in non-smooth convex learning”. In: *Submitted to JMLR* (2021).
- **Bertrand**, **Q.** et al. “Implicit differentiation of Lasso-type models for hyperparameter optimization”. In: *ICML* (2020).
- Blondel, M. and F. Pedregosa. *Lightning: large-scale linear classification, regression and ranking in Python*. 2016.
- Bollapragada, R., D. Scieur, and A. d’Aspremont. “Nonlinear acceleration of momentum and primal-dual algorithms”. In: *AISTATS* (2018).
- Breheny, P. and J. Huang. “Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection”. In: *Ann. Appl. Stat.* (2011).

- Brochu, E., V. M. Cora, and N. De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).
- Candès, E. J., M. B. Wakin, and S. P. Boyd. "Enhancing Sparsity by Reweighted l_1 Minimization". In: *J. Fourier Anal. Applicat.* (2008).
- Cohen, D. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).
- Deledalle, C.-A. et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).
- Domke, J. "Generic methods for optimization-based modeling". In: *AISTATS*. Vol. 22. 2012.
- Fan, J. and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* (2008).

- Fercoq, O. and P. Richtárik. “Accelerated, parallel and proximal coordinate descent”. In: *SIAM J. Optim.* (2015).
- Figueiredo, M. “Adaptive Sparseness Using Jeffreys Prior.”. In: *NeurIPS*. 2001.
- Franceschi, L. et al. “Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning”. In: *arXiv preprint arXiv:1806.04941* (2018).
- Franceschi, L. et al. “Forward and reverse gradient-based hyperparameter optimization”. In: *ICML*. 2017.
- Frecon, J., S. Salzo, and M. Pontil. “Bilevel learning of the group lasso structure”. In: *NeurIPS*. 2018.
- Friedman, J., T. J. Hastie, and R. Tibshirani. “Regularization paths for generalized linear models via coordinate descent”. In: *J. Stat. Softw.* (2010).
- Friedman, J. et al. “Pathwise coordinate optimization”. In: *Ann. Appl. Stat.* (2007).

- Ge, J. et al. “Picasso: A sparse learning library for high dimensional data analysis in R and Python”. In: *The Journal of Machine Learning Research* (2019).
- Gramfort, A., M. Kowalski, and M. Hämäläinen. “Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods”. In: *Phys. Med. Biol.* (2012).
- Gramfort, A. et al. “MNE software for processing MEG and EEG data”. In: *NeuroImage* (2014).
- Hutter, F., J. Lücke, and L. Schmidt-Thieme. “Beyond manual tuning of hyperparameters”. In: *KI-Künstliche Intelligenz* (2015).
- Jas, M. et al. “Pyglmnet: Python implementation of elastic-net regularized generalized linear models”. In: *Journal of Open Source Software* (2020).
- Kohavi, R. and G. H. John. “Automatic parameter selection by minimizing estimated error”. In: *Machine Learning Proceedings*. 1995.

- Larsen, J. et al. “Design and regularization of neural networks: the optimal use of a validation set”. In: *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. 1996.
- Li, L. et al. “Hyperband: A novel bandit-based approach to hyperparameter optimization”. In: *Journal of Machine Learning Research* (2017).
- Lin, Q., Z. Lu, and L. Xiao. “An Accelerated Proximal Coordinate Gradient Method”. In: *NeurIPS*. Citeseer, 2014.
- Liu, D. C. and J. Nocedal. “On the limited memory BFGS method for large scale optimization”. In: *Mathematical programming* (1989).
- Liu, W., Y. Yang, et al. “Parametric or nonparametric? A parametricness index for model selection”. In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.
- Lounici, K. “Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators”. In: *Electron. J. Stat.* (2008).

- Lounici, K., K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).
- Lounici, K. et al. "Taking Advantage of Sparsity in Multi-Task Learning". In: *arXiv preprint arXiv:0903.1468* (2009).
- M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. "Support recovery and sup-norm convergence rates for sparse pivotal estimation". In: *AISTATS*. 2020.
- Massias, M. et al. "Dual Extrapolation for Sparse Generalized Linear Models". In: *Journal of Machine Learning Research* (2020).
- Mazumder, R., J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". In: *Journal of the American Statistical Association* (2011).
- Nocedal, J. and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006.

- Ochs, P. et al. “Bilevel optimization with nonsmooth lower level problems”. In: *SSVM*. 2015.
- Pedregosa, F. “Hyperparameter optimization with approximate gradient”. In: *ICML*. 2016.
- Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* (2011).
- Peyré, G. and J. Fadili. “Learning analysis sparsity priors”. In: *Sampta*. 2011.
- Q. Klopfenstein, **Q. Bertrand** et al. “Model identification and local linear convergence of coordinate descent”. In: *arXiv preprint arXiv:2010.11825* (2020).
- Scieur, D. “Generalized Framework for Nonlinear Acceleration”. In: *arXiv preprint arXiv:1903.08764* (2019).
- Scieur, D., A. d’Aspremont, and F. Bach. “Regularized nonlinear acceleration”. In: *NeurIPS*. 2016.
- Sidi, A. *Vector extrapolation methods with applications*. SIAM, 2017.

- Stackelberg, H. von. *Marktform und Gleichgewicht*. J. Springer, 1934.
- Stein, C. M. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.
- Stone, L. R. A. and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.
- Tipping, M. E. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of Machine Learning Research* (2001).
- Wengert, R. E. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.