Coordinate descent for Slope

Mathurin Massias

https://mathurinm.github.io

Inria Lyon, team OCKHAM

Joint work with J. Larsson, Q. Klopfenstein and J. Wallin (accepted at AISTATS 2023)

https://arxiv.org/abs/2210.14780

The impact of sparsity

Seminal convex estimator for joint regression and feature selection: Lasso

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \left\| y - X\beta \right\|^2 + \lambda \left\| \beta \right\|_1$$

Key property if λ not too small: $\#\{j : \hat{\beta}_j \neq 0\} \ll p$, by nonsmoothness of $\left\| \cdot \right\|_1$

Statisticians love it (Candès et al., 2006; Donoho, 2006; Hastie et al., 2015):

- provable recovery guarantees if real model is sparse + good properties on X
- basically same error rate as least squares but handles $p \gg n$

What about computing the Lasso?

Computing the Lasso estimator

Initially a hard problem (non-smoothness), but optimizers now love it too.

$$\min_{\beta \in \mathbb{R}^p} f(\beta) + g(\beta) \qquad \text{prox}_g(x) = \arg\min_y \frac{1}{2} \|x - y\|^2 + g(y)$$

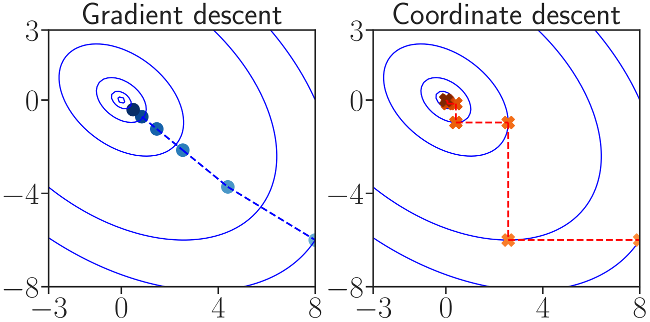
- "smooth + proximable" problem, amenable to proximal splitting methods (Combettes and Wajs, 2005) e.g. FISTA (Beck and Teboulle, 2009)

$$\beta^{k+1} = \text{prox}_{\tau g}(\beta^k - \tau \nabla f(\beta^k))$$

- from curse to blessing of non-smoothness (Iutzeler and Malick, 2020): leverage sparsity of iterates with screening or working sets (Ndiaye et al., 2017)
- even faster algorithm: coordinate descent

(Proximal) coordinate descent

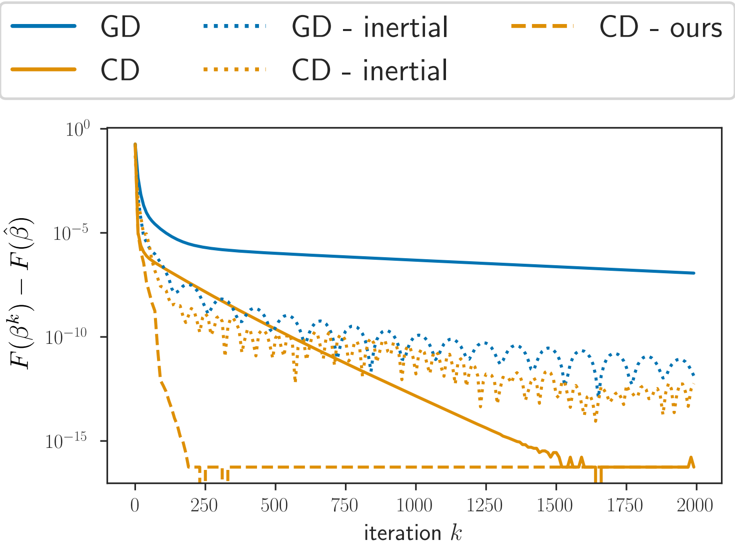
- Do proximal gradient descent steps on *one coordinate at a time*
- Should not converge... but does for smooth functions, smooth + separable



Lasso is the prototypical problem solvable by coordinate descent!

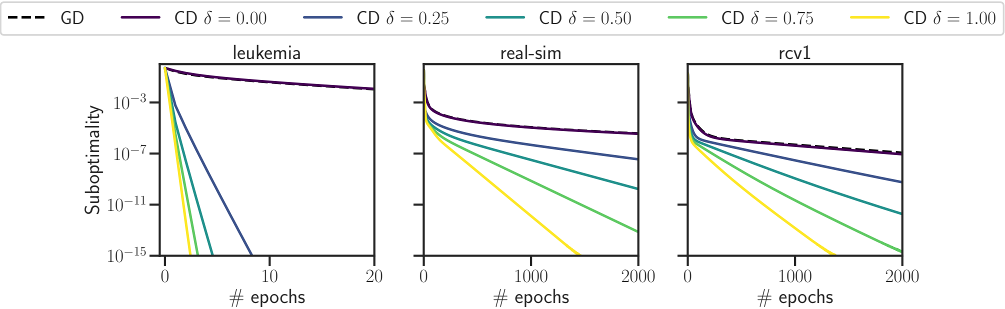
$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

CD for Lasso can be quite fast (Bertrand and Massias, 2021)



Main reason for success of CD

- One full update of β not more costly than one gradient in general: $\mathcal{O}(np)$
- Much larger stepsizes than GD ($1/L_j$ vs $1/L$, coordinatewise vs global gradient Lipschitz constant)



In pratice, CD can be at least one order of magnitude faster than FISTA

Impact on practitioners

- With efficient implementations of Lasso solvers such as Celer (Massias et al., 2020) it is possible to solve problems with millions of variables in a few seconds
- Interpretable models are popular among practitioners
- Large scale applications in biology, neuroscience, geophysics... (Muir and Zhan, 2021; Kim et al., 2021; Reidenbach et al., 2021)

So are we done? Why this talk?

Lasso has limitations

- Amplitude bias (Zhang and Huang, 2008)
- Difficulty to deal with correlated coefficients (Zou and Hastie, 2005)
- Many false positive, false positive occur even for strong regularization (Su et al., 2017)

Potential solution: non convex penalties (ℓ_q, MCP, SCAD, log) for which efficient solvers such as `skglm` also exist (Bertrand et al., 2022)...

... but convexity is lost and so far you're never sure of what you get in the end.

We'll take the convex road!

A convex alternative: SLOPE

Sorted L-One Penalized Estimator, based on the *sorted ℓ_1 norm* (Bogdan et al., 2013; Zeng and Figueiredo, 2014):

$$\lambda_1 \geq \ldots \geq \lambda_p \geq 0$$

$$J(\beta) = \sum_{j=1}^{p} \lambda_j |\beta_{(j)}| = \sum_{j=1}^{p} \lambda_{(j)^-} |\beta_j|$$

where $(\cdot)$ reorders β by descending magnitude ($(\cdot)^-$ its inverse):

$$|\beta_{(1)}| \geq \ldots \geq |\beta_{(p)}|$$

$\hookrightarrow$ largest coefficients are more penalized

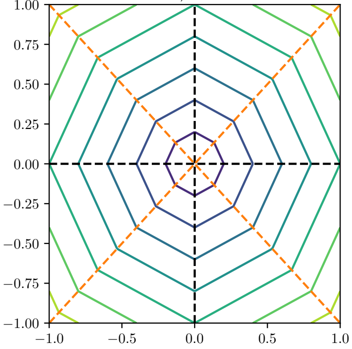
Generalization of two peculiar instances:

- $\lambda_1 = \ldots = \lambda_p \rightarrow$ Lasso penalty
- $\lambda_2 = \ldots = \lambda_p = 0 \rightarrow \ell_\infty$ penalty

SLOPE properties

- convex (pointwise supremum of affine hence convex functions)
- non differentiable along axes AND when coefficients are equal in magnitude

$\lambda_1 = 1, \lambda_2 = 0.5$ $\lambda_1 = 1, \lambda_2 = 0.3$

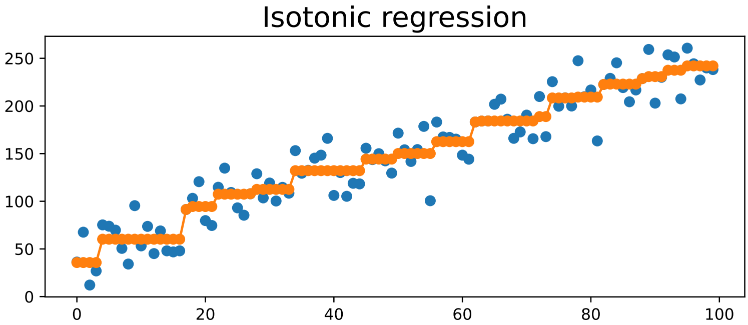


SLOPE solves some of the Lasso's problem

- false discovery rate control (Bogdan et al., 2015; Kos and Bogdan, 2020)
- coefficient clustering (Figueiredo and Nowak, 2016; Schneider and Tardivel, 2020): $|\beta_j|$ takes m distinct values $c_1 > c_2 > \cdots > c_m \geq 0$, on sets of indices $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$
- sparsity and ordering patterns recovery (Bogdan et al., 2022)

The Optimizer's point of view

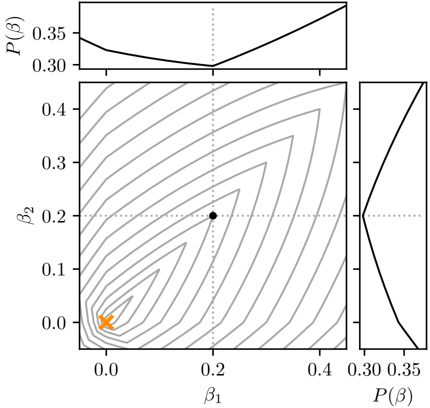
The prox of SLOPE is (surprisingly?) known, based on isotonic regression



$\hookrightarrow$ ISTA, FISTA can be used

Could we still use proximal CD?

CD cannot be applied for lack of separability



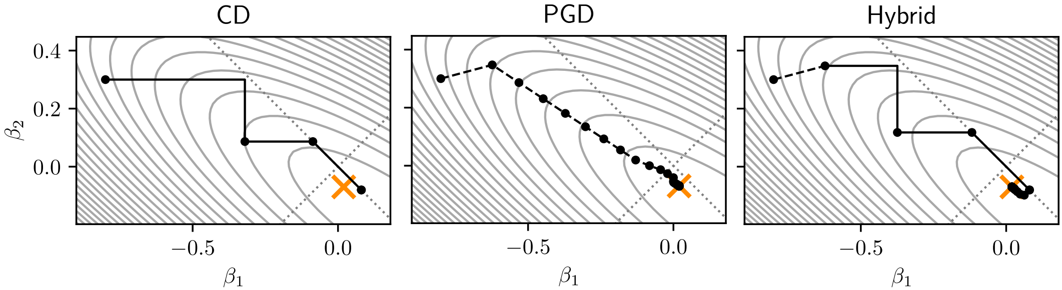
CD can only move along the dashed line and thus stays there

Key issue: clusters are not known

If clusters $\mathcal{C}_1, \dots, \mathcal{C}_{m^*}$ of the solution β^* are known, the penalty becomes separable (Dupuis and Tardivel, 2022) and one can solve:

$$\min_{z \in \mathbb{R}^{m^*}} \left(\frac{1}{2} \Big\| y - X \sum_{i=1}^{m^*} \sum_{j \in \mathcal{C}_i^*} z_i \operatorname{sign}(\beta_j^*) e_j \Big\|^2 + \sum_{i=1}^{m^*} |z_i| \sum_{j \in \mathcal{C}_i^*} \lambda_j \right).$$

Idea: alternate between cluster identification steps and fast CD step



Why relying on PGD for cluster identification?

Def: J is said to be *partly smooth* at x relative to a set $\mathcal{M}$ containing x if:

1. $\mathcal{M}$ is a C^2-manifold around x and J restricted to $\mathcal{M}$ is C^2 around x.
2. The tangent space of $\mathcal{M}$ at x is the orthogonal of the parallel space of $\partial J(x)$.
3. ∂J is continuous at x relative to $\mathcal{M}$.

Prop: The SLOPE is partly smooth at any x w.r.t. $\mathcal{M} =$ "vectors with same support, signs and clusters as x" (linear manifold)

(links with polyhedral norms (Vaiter et al., 2017))

$\hookrightarrow$ PGD identifies the clusters in a finite number of iterations (Liang et al., 2014)

Minimization on a single cluster

When we update the value taken by β on its cluster $\mathcal{C}_k$ we let:

$$\beta_i(z) = \begin{cases} \operatorname{sign}(\beta_i) z\,, & \text{if } i \in \mathcal{C}_k\,, \\ \beta_i\,, & \text{otherwise}\,. \end{cases}$$

Minimizing the objective in this direction amounts to solving the following one-dimensional problem:

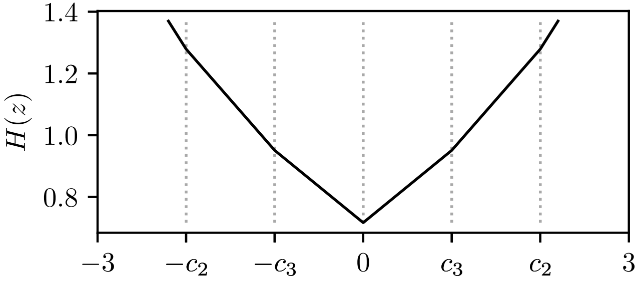
$$\min_{z \in \mathbb{R}} \left(G(z) = \frac{1}{2} \|y - X\beta(z)\|^2 + H(z) \right),$$

where

$$H(z) = |z| \sum_{j \in \mathcal{C}_k} \lambda_{(j)^-_z} + \sum_{j \notin \mathcal{C}_k} |\beta_j| \lambda_{(j)^-_z}$$

is the *partial sorted ℓ_1 norm* with respect to the k-th cluster and $\lambda_{(j)^-_z}$ means that the inverse sorting permutation $(j)^-_z$ is defined with respect to $\beta(z)$.

The partial sorted ℓ_1 norm



The partial sorted ℓ_1 norm with $\beta = [-3, 1, 3, 2]^T$, $k = 1$, and so $c_1, c_2, c_3 = (3, 2, 1)$.

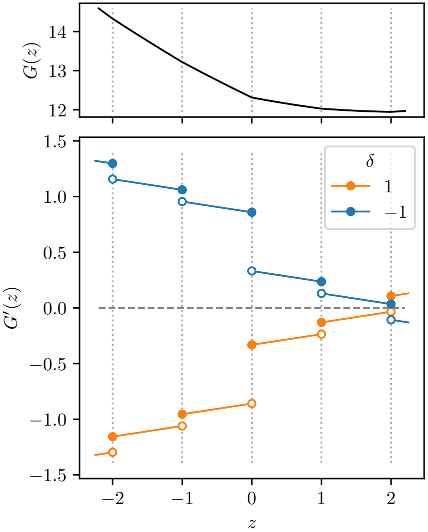
How do we solve the minimization for one cluster?

1D minimization pb, optimality condition:

$$\forall \delta \in \{-1, 1\}, \quad G'(z; \delta) \geq 0$$

$$G'(z; \delta) = \delta \sum_{j \in \mathcal{C}_k} X_{:j}^\top (X\beta(z) - y) + H'(z; \delta)$$

and H is the partial sorted L1 norm.



Expression for the directional derivative

Thm: Let $c^{\setminus k}$ be the set containing all elements of c except the k-th one: $c^{\setminus k} = \{c_1, \dots c_{k-1}, c_{k+1}, \dots, c_m\}$. Let $\varepsilon_c > 0$ such that

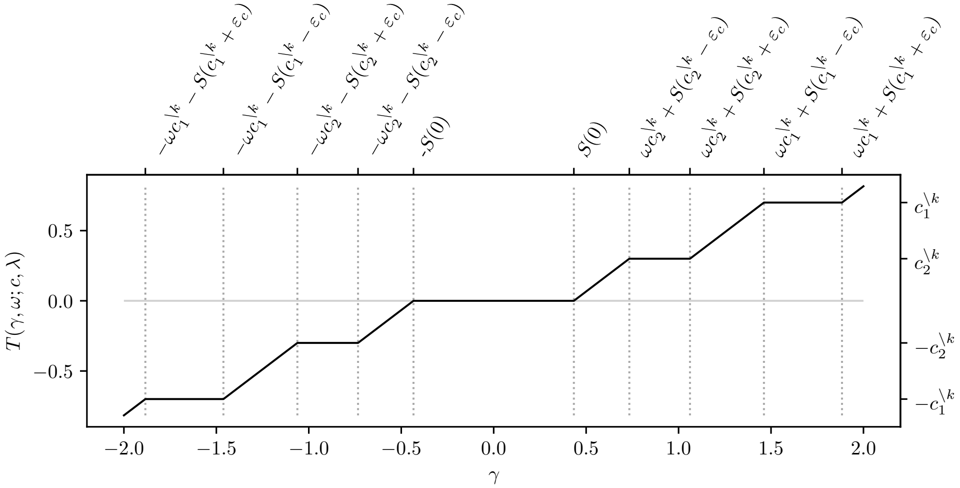
$$\varepsilon_c < \left|c_i - c_j\right|, \quad \forall\, i \neq j \text{ and } \varepsilon_c < c_m \text{ if } c_m \neq 0\,.$$

The directional derivative of the partial sorted ℓ_1 norm with respect to the k-th cluster, H, in the direction δ is

$$H'(z;\delta) = \begin{cases} \sum_{j \in C(\varepsilon_c)} \lambda_{(j)^-_{\varepsilon_c}} & \text{if } z = 0\,, \\ \operatorname{sign}(z)\delta \sum_{j \in C(z+\varepsilon_c\delta)} \lambda_{(j)^-_{z+\varepsilon_c\delta}} & \text{if } |z| \in c^{\setminus k} \setminus \{0\}, \\ \operatorname{sign}(z)\delta \sum_{j \in C(z)} \lambda_{(j)^-_z} & \text{otherwise}\,. \end{cases}$$

Solution of update given by the "SLOPE thresholding operator"

Thm: $\arg\min_z G(z) = T(c_k \|\tilde{x}\|^2 - \tilde{x}^T(X\beta - y); \|x\|^2, c^{\setminus k}, \lambda)$



The full algorithm

input: $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$, $\lambda \in \{\mathbb{R}^p : \lambda_1 \geq \lambda_2 \geq \cdots > 0\}$, $v \in \mathbb{N}$, $\beta \in \mathbb{R}^p$

for $t \leftarrow 0, 1, \ldots$ **do**

  if $t \bmod v = 0$ **then**

    $\beta \leftarrow \operatorname{prox}_{J/\|X\|_2^2}\left(\beta - \frac{1}{\|X\|_2^2} X^T (X\beta - y)\right)$

    Update c, $\mathcal{C}$

  else

    $k \leftarrow 1$

    while $k \leq |\mathcal{C}|$ **do**

      $\tilde{x}_k \leftarrow X_{\mathcal{C}_k} \operatorname{sign}(\beta_{\mathcal{C}_k})$

      $z \leftarrow T(c_k \|\tilde{x}\|^2 - \tilde{x}^T (X\beta - y); \|x\|^2, c^{\setminus k}, \lambda)$

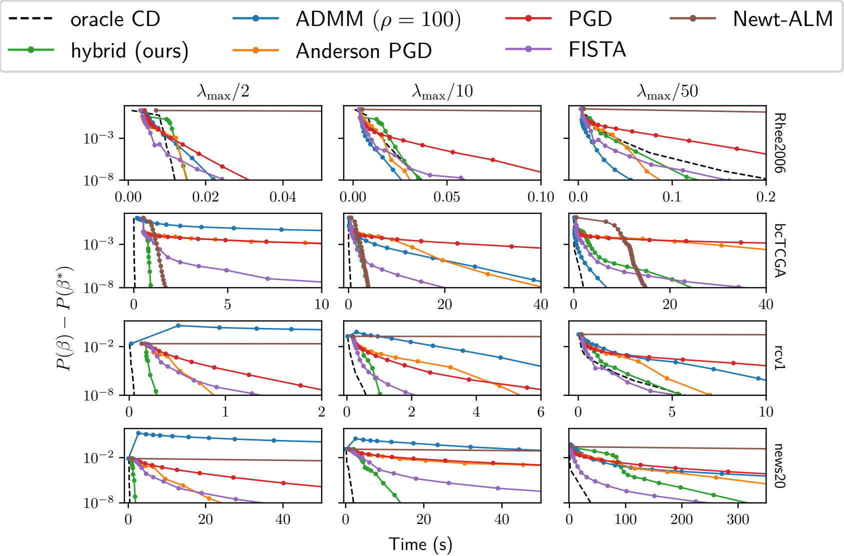
      $\beta_{\mathcal{C}_k} \leftarrow z \operatorname{sign}(\beta_{\mathcal{C}_k})$

      Update c, $\mathcal{C}$

      $k \leftarrow k + 1$

return β

Benchmarks



oracle CD
hybrid (ours)
ADMM ($\rho = 100$)
Anderson PGD
PGD
FISTA
Newt-ALM
$\lambda_{\max}/2$
$\lambda_{\max}/10$
$\lambda_{\max}/50$
Rhee2006
bcTCGA
rcv1
news20
$P(\beta) - P(\beta^*)$
Time (s)

Part II: easier and better benchmarks with Benchopt

“*Benchopt: Reproducible, efficient and collaborative optimization benchmarks*”, NeurIPS 2022.

`https://benchopt.github.io/`

Benchmarking algorithms is a pain

Machine Learning research relies on numerical validation.

Pain points of a benchmark:

- competitors' methods do not work out of the box.
- re-code methods and tools to integrate a new method.
- hard to extend with new settings.

all of this started from scratch by every submission!

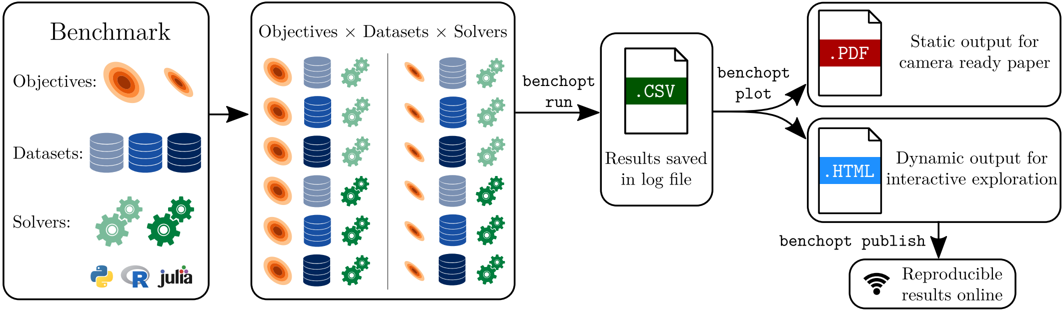
`Benchopt` produces **open**, **reproducible**, **extendable** benchmarks

How does Benchopt do it?

Benchopt is a framework to organize and run benchmarks:

- one repository per benchmark
- one base open source Python CLI to run them

3 components: Objective, Dataset, Solver



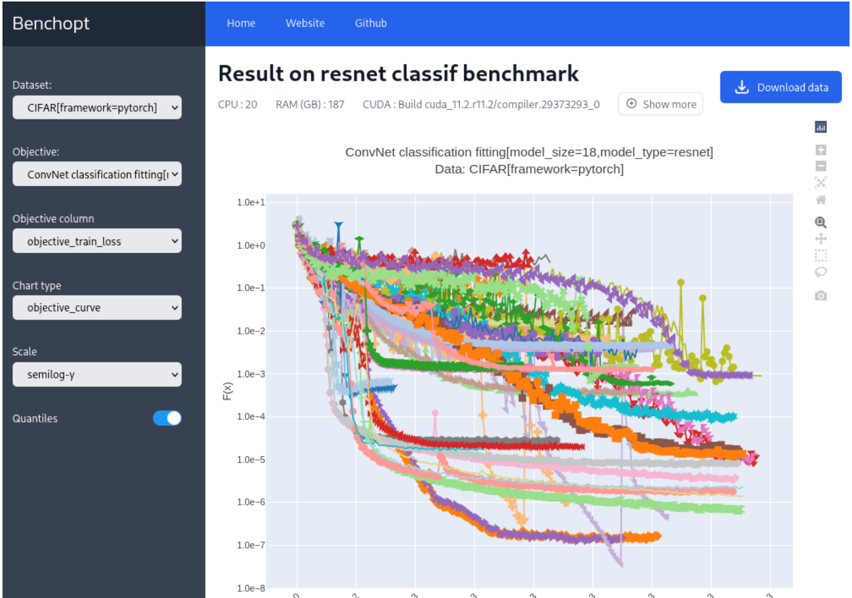
Structure of a benchmark

```
benchmark/
├── objective.py
├── datasets/
│   ├── dataset1.py
│   └── dataset2.py
└── solvers/
    ├── solver1.py
    └── solver2.py
```

Modular & extendable

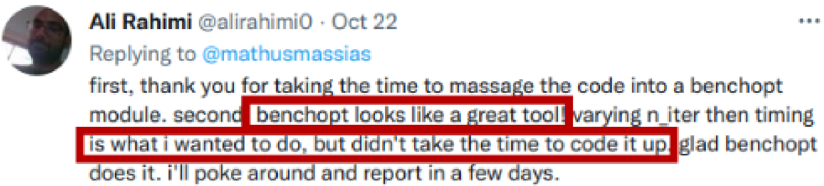
New solver? add a file
New dataset? add a file
New metric? modify objective

Interactive results exploration



Benchopt **makes your life easy**

- build on previous benchmarks
- use solvers in Python, R, Julia, binaries...
- monitor any metric you want altogether (test/train loss, ...)
- add parameters to solvers
- share and publish HTML results
- run all benchmarks in parallel
- cache results
- and much more!



Existing benchmarks

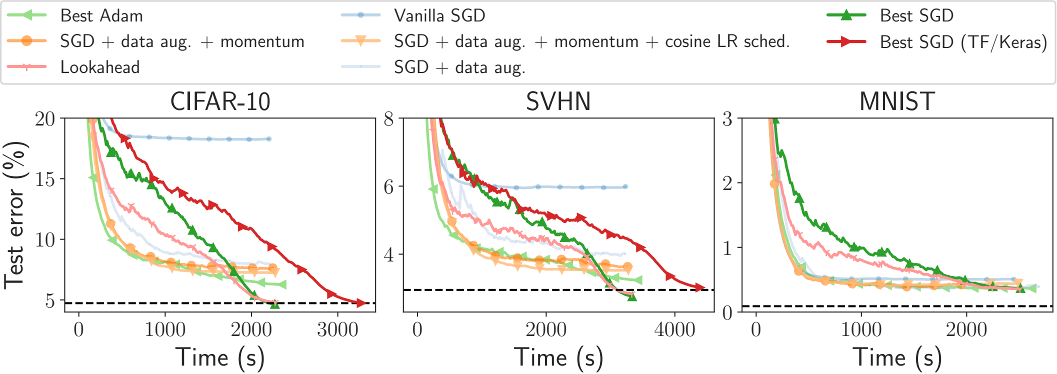
Examples of existing benchmarks:

- Resnet18
- Lasso
- Slope
- MCP
- Logistic regression
- ICA
- Total Variation
- Ordinary Least Squares
- Non convex sparse regression
- linear SVM

Start yours with `https://github.com/benchopt/template_benchmark`!

Example: Resnet benchmark

- image classification with resnet18
- various optimization strategies
- compare `pytorch` and `tensorflow`
- publish reproducible SOTA for baselines



https://github.com/benchopt/benchmark_resnet_classif/

E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2): 489–509, 2006.

D. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143:143, 2015.

Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale modeling & simulation*, 4(4):1168–1200, 2005.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

F. Iutzeler and J. Malick. Nonsmoothness in machine learning: specific structure, proximal identification, and applications. *Set-Valued and Variational Analysis*, 28(4):661–678, 2020.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *The Journal of Machine Learning Research*, 18(1):4671–4703, 2017.

Q. Bertrand and M. Massias. Anderson acceleration of coordinate descent. In *AISTATS*, pages 1288–1296. PMLR, 2021.

Mathurin Massias, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Dual extrapolation for sparse generalized linear models. *Journal of Machine Learning Research*, 21(234):1–33, 2020.

J. B. Muir and Z. Zhan. Seismic wavefield reconstruction using a pre-conditioned wavelet–curvelet compressive sensing approach. *Geophysical Journal International*, 227(1):303–315, 2021.

Y. J. Kim, N. Brackbill, E. Batty, J. Lee, C. Mitelut, W. Tong, EJ Chichilnisky, and L. Paninski. Nonlinear decoding of natural images from large-scale primate retinal ganglion recordings. *Neural Computation*, 33(7):1719–1750, 2021.

D. A. Reidenbach, A. Lal, L. Slim, O. Mosafi, and J. Israeli. Gepsi: A python library to simulate gwas phenotype data. *bioRxiv*, 2021.

Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

H. Zou and T. J. Hastie. Regularization and variable selection via the elastic net. 67(2):301–320, 2005.

Weijie Su, Małgorzata Bogdan, and Emmanuel Candes. False discoveries occur early on the lasso path. *The Annals of statistics*, pages 2133–2150, 2017.

Quentin Bertrand, Quentin Klopfenstein, Pierre-Antoine Bannier, Gauthier Gidel, and Mathurin Massias. Beyond l1: Faster and better sparse models with skglm. *arXiv preprint arXiv:2204.07826*, 2022.

Małgorzata Bogdan, Ewout van den Berg, Weijie Su, and Emmanuel Candès. Statistical estimation and testing via the sorted L1 norm. 2013.

Xiangrong Zeng and Mario Figueiredo. The ordered weighted ℓ_1 norm: Atomic formulation, projections, and algorithms, 2014.

Małgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel Candès. SLOPE - adaptive variable selection via convex optimization. 9(3):1103–1140, 2015.

Michał Kos and Małgorzata Bogdan. On the asymptotic properties of SLOPE. 82(2):499–532, 2020.

Mario Figueiredo and Robert Nowak. Ordered weighted L1 regularized regression with strongly correlated covariates: Theoretical aspects. In *AISTATS*, pages 930–938, 2016.

Ulrike Schneider and Patrick Tardivel. The Geometry of Uniqueness, sparsity and clustering in penalized estimation, 2020. URL http://arxiv.org/abs/2004.09106.

Małgorzata Bogdan, Xavier Dupuis, Piotr Graczyk, Bartosz Kołodziejek, Tomasz Skalski, Patrick Tardivel, and Maciej Wilczyński. Pattern recovery by SLOPE. 2022. URL http://arxiv.org/abs/2203.12086.

Xavier Dupuis and Patrick Tardivel. Proximal operator for the sorted l1 norm: Application to testing procedures based on slope. *Journal of Statistical Planning and Inference*, 221:1–8, 2022.

Samuel Vaiter, Charles Deledalle, Jalal Fadili, Gabriel Peyré, and Charles Dossal. The degrees of freedom of partly smooth regularizers. *Annals of the Institute of Statistical Mathematics*, 69(4): 791–832, 2017.

Jingwei Liang, Jalal Fadili, and Gabriel Peyré. Local linear convergence of forward–backward under partial smoothness. *Advances in neural information processing systems*, 27, 2014.

Definition of the SLOPE thresholding operator

Define $S(x) = \sum_{j \in C(x)} \lambda_{(j)^-_x}$ and let

$$T(\gamma; \omega, c, \lambda) = \begin{cases} 0 & \text{if } |\gamma| \leq S(\varepsilon_c), \\ \operatorname{sign}(\gamma) c_i & \text{if } \omega c_i + S(c_i - \varepsilon_c) \\ & \quad \leq |\gamma| \leq \\ & \quad \omega c_i + S(c_i + \varepsilon_c), \\ \frac{\operatorname{sign}(\gamma)}{\omega}\big(|\gamma| - S(c_i + \varepsilon_c)\big) & \text{if } \omega c_i + S(c_i + \varepsilon_c) \\ & \quad < |\gamma| < \\ & \quad \omega c_{i-1} + S(c_{i-1} - \varepsilon_c), \\ \frac{\operatorname{sign}(\gamma)}{\omega}\big(|\gamma| - S(c_1 + \varepsilon_c)\big) & \text{if } |\gamma| \geq \omega c_1 + S(c_1 + \varepsilon_c). \end{cases}$$

with ε_c such that $\varepsilon_c < \left|c_i - c_j\right|, \quad \forall\, i \neq j$ and $\varepsilon_c < c_m$ if $c_m \neq 0$.
Let $\tilde{x} = X_{\mathcal{C}_k} \operatorname{sign}(\beta_{\mathcal{C}_k})$ and $r = y - X\beta$. Then

$$T\left(c_k \left\|\tilde{x}\right\|^2 + \tilde{x}^T r; \left\|x\right\|^2, c^{\setminus k}, \lambda\right) = \underset{z \in \mathbb{R}}{\arg\min}\, G(z)\,.$$