High dimensional penalized linear models with interactions using graphics card

Internship Master 2 Biostatistics (Now SSD)

Tanguy Lefort
IMAG, Univ Montpellier, CNRS
LIRMM, Inria

Joseph Salmon
IMAG, Univ Montpellier, CNRS
Institut Universitaire de France (IUF)

Benjamin Charlier
IMAG, Univ Montpellier, CNRS

We denote $X = [x_1, \dots, x_p] \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^p$ such that $y \simeq X\beta$

Ordinary least squares:

$$\hat{\beta}^{ls} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2 \iff \hat{\beta}^{ls} = (X^\top X)^{-1} X^\top y$$

Challenge with high dimension:

- if $p > n$ we lose the uniqueness,
- $X^\top X$ may be ill conditioned ($\kappa = \frac{\text{largest singular value}}{\text{smallest singular value}} \gg 1$) due to multicolinearity amongst features
- too many active features is not interpretable (genomics dataset)

We denote $X = [x_1, \dots, x_p] \in \mathbb{R}^{n\times p}$, $y \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^p$ such that $y \simeq X\beta$

Ordinary least squares:

$$\hat{\beta}^{ls} = \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\|y - X\beta\|_2^2 \iff \hat{\beta}^{ls} = (X^\top X)^{-1}X^\top y$$

Challenge with high dimension:

- if $p > n$ we lose the uniqueness,
  - make the problem strictly convex.
- $X^\top X$ may be ill conditioned ($\kappa = \frac{\text{largest singular value}}{\text{smallest singular value}} \gg 1$) due to multicolinearity amongst features
  - Shift spectrum by a small quantity using ℓ_2 penalty
- too many active features is not interpretable (genomics dataset)
  - Feature selection using ℓ_1 penalty

Elastic-Net[1] $=$ combination of LASSO[2] and Ridge[3]:

Elastic-Net

Considering tuning parameters $\lambda_1, \lambda_2 > 0$:

$$\hat{\beta}^{enet} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|_2^2$$

(1) H. Zou and T. Hastie (2005). "Regularization and variable selection via the elastic net". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67.2, pp. 301–320.

(2) R. Tibshirani (1996). "Regression Shrinkage and Selection via the Lasso". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1, pp. 267–288.

(3) A. Tikhonov (1943). "On the stability of inverse problems". In: *Dokl. Akad. Nauk SSSR* 39, pp. 176–179.

Elastic-Net[1] = combination of LASSO[2] and Ridge[3]:

Elastic-Net

Considering tuning parameters $\lambda_1, \lambda_2 > 0$:

$$\hat{\beta}^{enet} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|_2^2$$

And if we add the first order interactions to the model:

Elastic-Net with interactions

The interactions matrix is $Z \in \mathbb{R}^{n \times q}$, and coefficients are $\Theta \in \mathbb{R}^q$.

$$\hat{\beta}^{inter} \in \underset{\substack{\beta \in \mathbb{R}^p \\ \Theta \in \mathbb{R}^q}}{\arg\min} \frac{1}{2n} \|y - X\beta - Z\Theta\|_2^2 + \lambda_{\beta,\ell_1} \|\beta\|_1 + \frac{\lambda_{\beta,\ell_2}}{2} \|\beta\|_2^2 + \lambda_{\Theta,\ell_1} \|\Theta\|_1 + \frac{\lambda_{\Theta,\ell_2}}{2} \|\Theta\|_2^2$$

(1) H. Zou and T. Hastie (2005). "Regularization and variable selection via the elastic net". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67.2, pp. 301–320.

(2) R. Tibshirani (1996). "Regression Shrinkage and Selection via the Lasso". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1, pp. 267–288.

(3) A. Tikhonov (1943). "On the stability of inverse problems". In: *Dokl. Akad. Nauk SSSR* 39, pp. 176–179.

Optimization problem

Our goal: Solve the Elastic-Net problem

- for high dimensional genomics data,
- using graphics card parallelization,
- at least as fast as currently used algorithms like Coordinate Descent with interactions[4].

[4] F. Bascou, S. Lèbre, and J. Salmon (2020). "Debiasing the Elastic Net for models with interactions". In: *Journées de Statistique*.

Optimization problem

Our goal: Solve the Elastic-Net problem

- for high dimensional genomics data,
- using graphics card parallelization,
- at least as fast as currently used algorithms like Coordinate Descent with interactions[4].

Problems:

- the interactions matrix is not storable in high dimensions,
- graphics cards need a lot of data to parallelize operations efficiently,
- we use solvers, but when do we stop them ?

(4) F. Bascou, S. Lèbre, and J. Salmon (2020). "Debiasing the Elastic Net for models with interactions". In: *Journées de Statistique*.

Minimize $F(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2$: with step $\eta > 0$ at epoch $k \in \mathbb{N}$,

Gradient Descent: 1 problem of dimension p

$$\beta^{k+1} \longleftarrow \beta^k - \eta \underbrace{\frac{1}{n} X^\top (X\beta^k - y)}_{\frac{\partial F}{\partial \beta}(\beta^k)}$$

Minimize $F(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2$: with step $\eta > 0$ at epoch $k \in \mathbb{N}$,

Gradient Descent: 1 problem of dimension p

$$\beta^{k+1} \longleftarrow \beta^k - \eta \underbrace{\frac{1}{n} X^\top (X\beta^k - y)}_{\frac{\partial F}{\partial \beta}(\beta^k)}$$

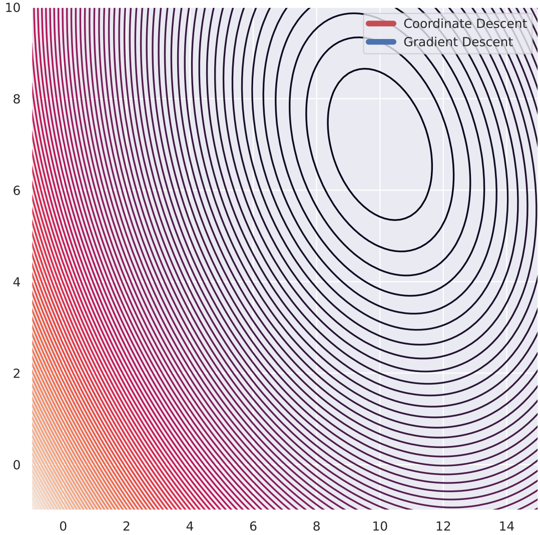
Coordinate Descent: p problems of dimension 1

For $j = 1, \dots, p$,

$$\beta_j^{k+1} \longleftarrow \beta_j^k - \eta \underbrace{\frac{1}{n} x_j^\top (X\beta^k - y)}_{\frac{\partial F}{\partial \beta_j}(\beta^k)}$$

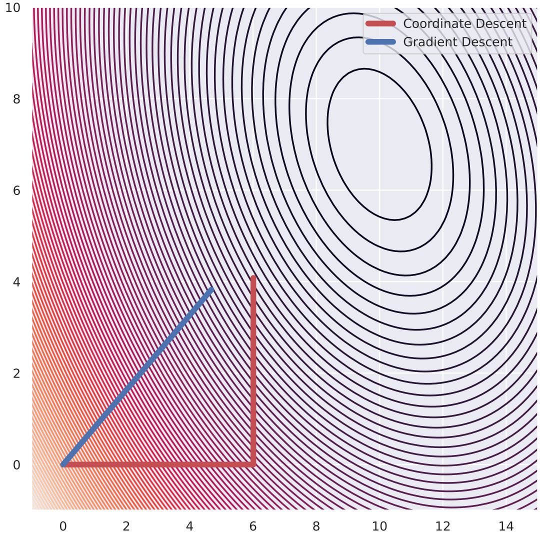
Visualization of the behavior: OLS

Fixed-step size



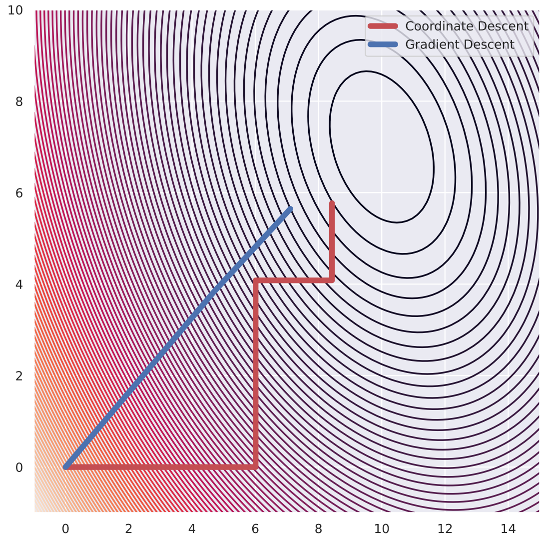
Visualization of the behavior: OLS

Fixed-step size



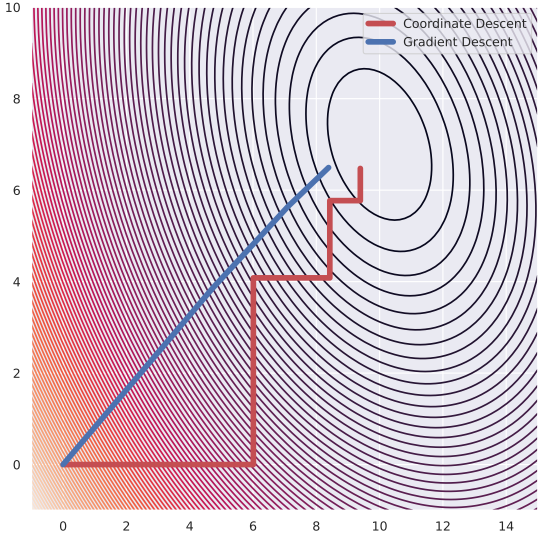
Visualization of the behavior: OLS

Fixed-step size



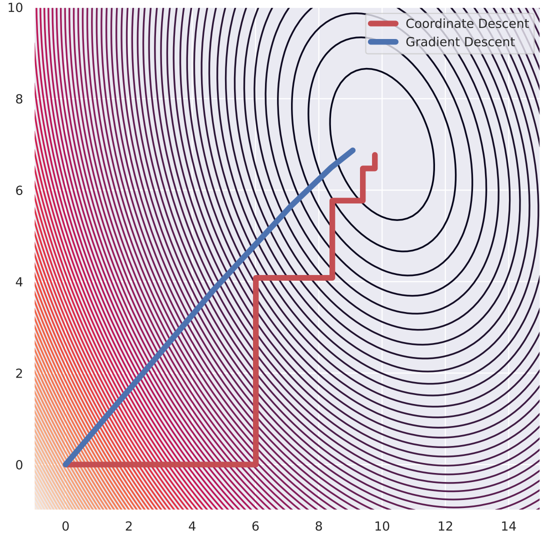
Visualization of the behavior: OLS

Fixed-step size



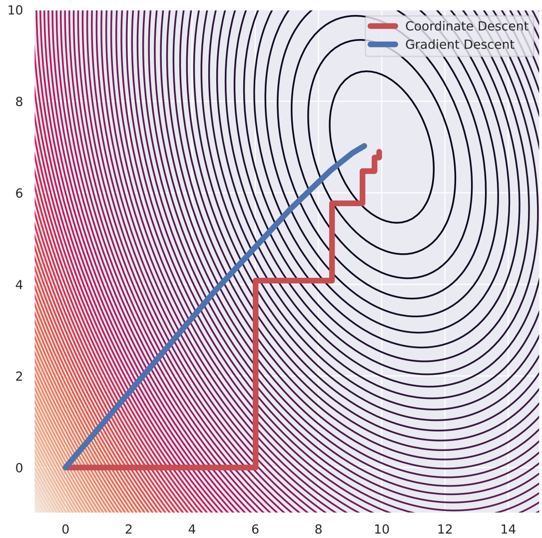
Visualization of the behavior: OLS

Fixed-step size



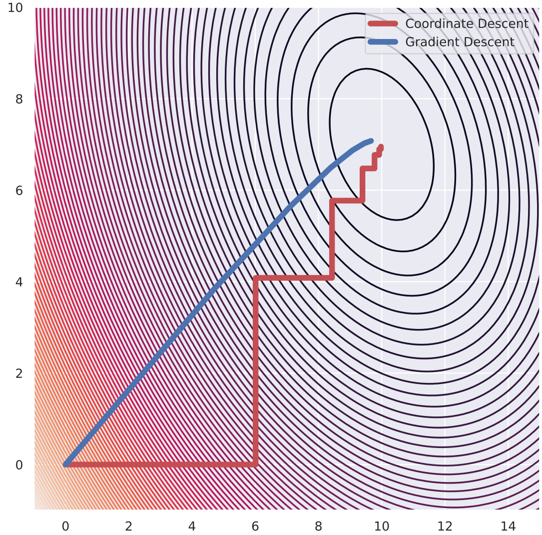
Visualization of the behavior: OLS

Fixed-step size



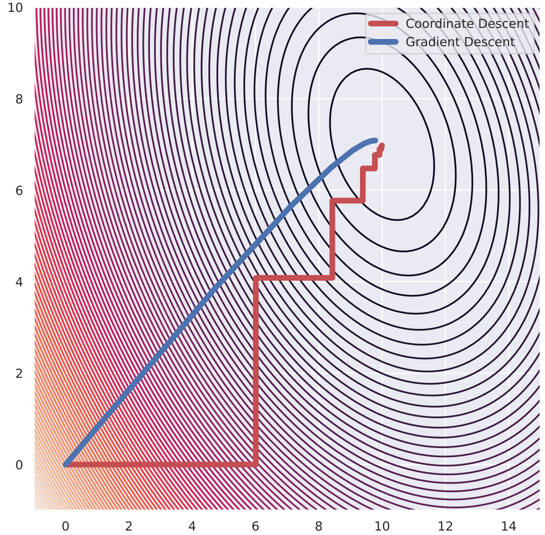
Visualization of the behavior: OLS

Fixed-step size



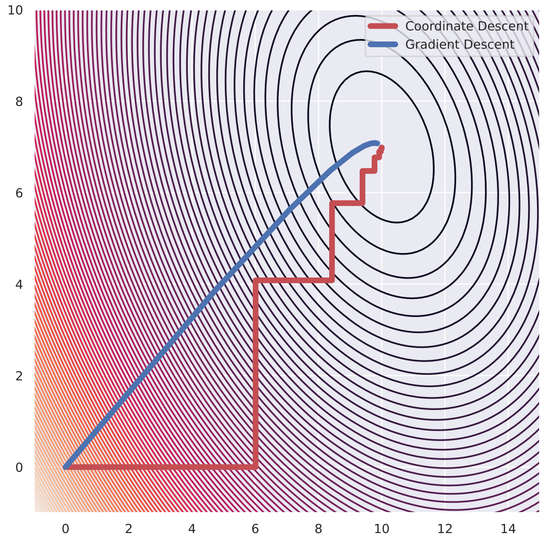
Visualization of the behavior: OLS

Fixed-step size



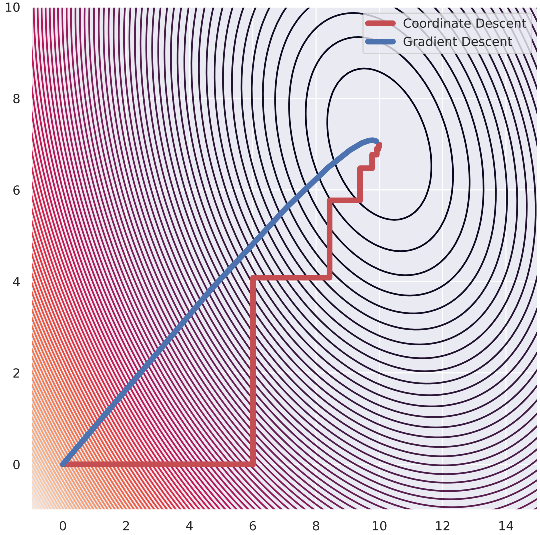
Visualization of the behavior: OLS

Fixed-step size



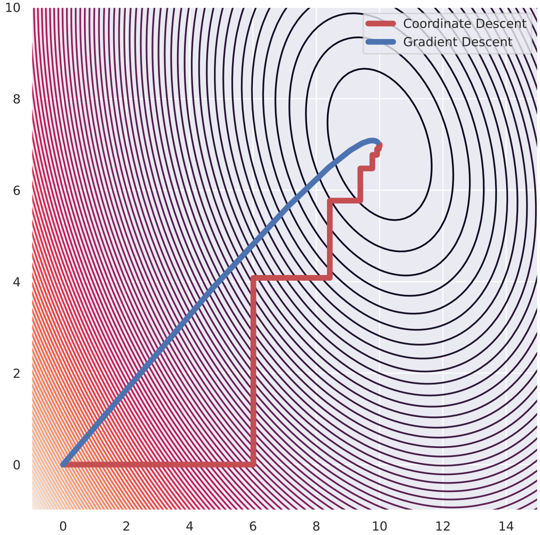
Visualization of the behavior: OLS

Fixed-step size



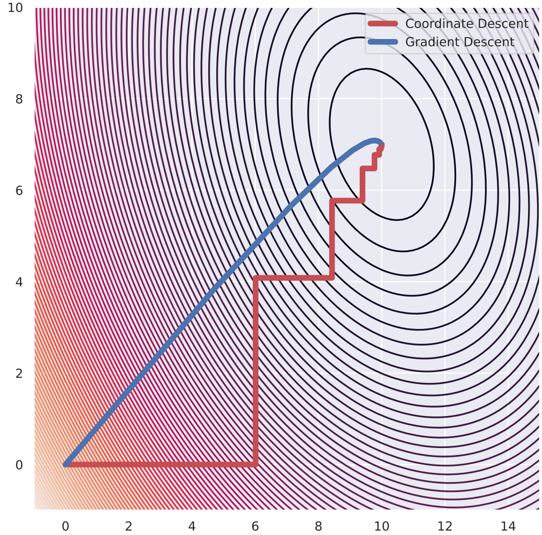
Visualization of the behavior: OLS

Fixed-step size



Visualization of the behavior: OLS

Fixed-step size



$$\underset{\beta\in\mathbb{R}^p}{\arg\min}\ \underbrace{\frac{1}{2n}\|y - X\beta\|_2^2}_{\text{smooth } F(\beta)} + \underbrace{\lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2}_{\text{non-smooth separable } g(\beta)}$$

$$\underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2n} \|y - X\beta\|_2^2}_{\text{smooth } F(\beta)} + \underbrace{\lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|_2^2}_{\text{non-smooth separable } g(\beta)}$$

Gradient descent on F with step $\eta > 0$:

$$\beta^{k+1} \longleftarrow \beta^k - \eta \frac{1}{n} X^\top (X\beta - y)$$

$$\underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2n} \|y - X\beta\|_2^2}_{\text{smooth } F(\beta)} + \underbrace{\lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|_2^2}_{\text{non-smooth separable } g(\beta)}$$

Proximal Gradient descent on $F + g$ with step $\eta > 0$:

$$\beta^{k+1} \longleftarrow \text{prox}_{\eta g} \left(\beta^k - \eta \frac{1}{n} X^\top (X\beta^k - y)\right)$$

Proximal operator

Let f a convex proper closed function, for $\mu > 0$:

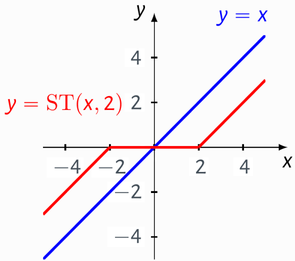
$$\text{prox}_{\mu f}(u) = \underset{x \in \text{dom } f}{\arg\min} \left\{ f(x) + \frac{1}{2\mu} \|x - u\|_2^2 \right\} \quad .$$

Elastic-Net penalty proximal operator

Let $h(x) = \|x\|_1 + \frac{\gamma}{2}\|x\|_2^2$, $\gamma > 0$, we know [5] that for $\mu > 0$:

$$\operatorname{prox}_{\mu h}(x) = \frac{1}{1+\mu\gamma}\operatorname{prox}_{\mu\|\cdot\|_1}(x) = \frac{\operatorname{sign}(x)}{1+\mu\gamma}(|x| - \mu)_+$$

where $\operatorname{sign}(x)(|x| - \mu)_+$ is the soft thresholding operator $\mathrm{ST}(x, \mu)$.



[5] N. Parikh and S. Boyd (2014). "Proximal Algorithms". In: *Found. Trends Optim.* 1.3, pp. 127–239, p. 189

Accelerate the algorithms

Possibilities to use accelerations:

- Theoretical:
  - inertial: heavy ball-like[6],
  - structure of the iterates: Anderson[7],
  - stochastic directions:[8],
  - structure of the problem: use block updates[9].
- Computational:
  - Numba library:[10],
  - GPU acceleration with CUDA.

(6) Y. Nesterov (1983). "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$". In: *Sov. Math. Dokl.* Vol. 27. 2.

(7) Q. Bertrand and M. Massias (2021). *Anderson acceleration of coordinate descent*.

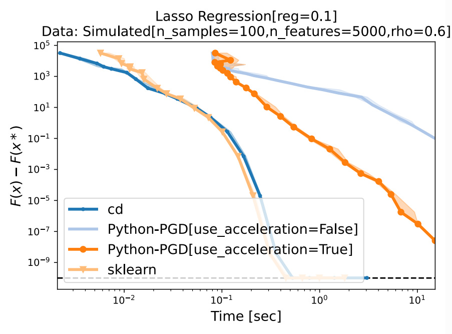
(8) Y. Nesterov (2012). "Efficiency of coordinate descent methods on huge-scale optimization problems". In: *SIAM Journal on Optimization* 22.2, pp. 341–362.

(9) A. Beck (2017). *First-Order Methods in Optimization*. Vol. 25. SIAM.

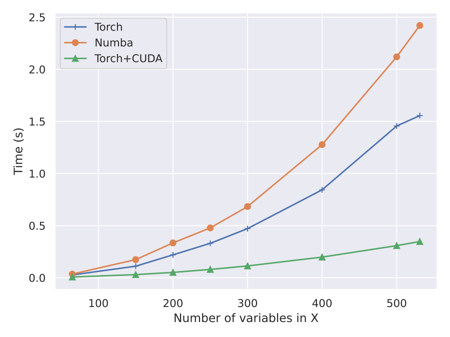
(10) S. Lam, A. Pitrou, and S. Seibert (2015). "Numba: A llvm-based python jit compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6.

Why use a GPU ?

Accelerated PGD on CPU is not very competitive



Benchmark LASSSO problem on CPU
(Figure made with `Bench0pt` library [11])

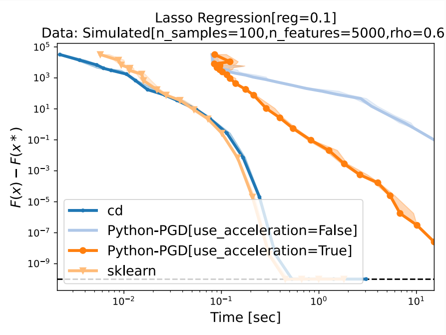


Benchmark product $Z\Theta$

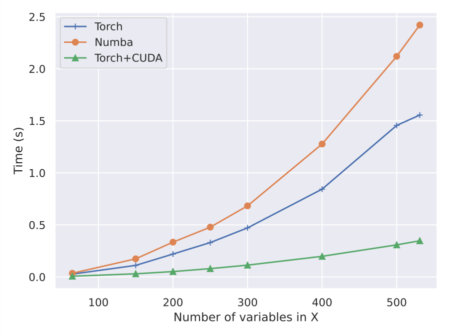
[11] `https://benchopt.github.io`

Why use a GPU ?

Accelerated PGD on CPU is not very competitive



Benchmark LASSSO problem on CPU (Figure made with BenchOpt library [11])



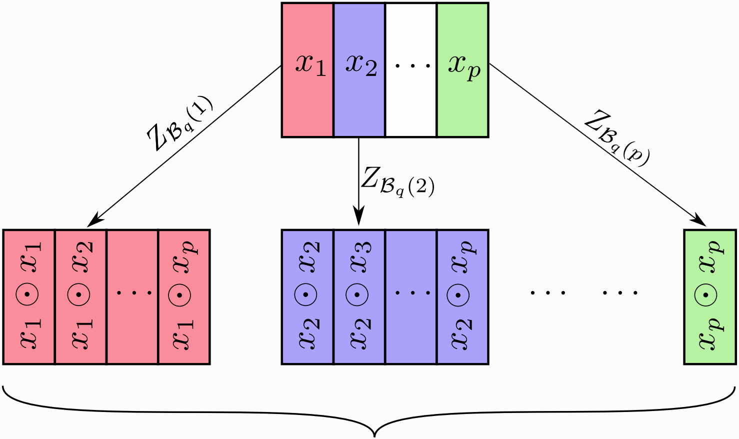
Benchmark product $Z\Theta$

And it is **easy** with PyTorch:

```
A = torch.tensor([1., 2.], device="cuda")
B = torch.tensor([1., 2.]).to("cuda")
```

[11] https://benchopt.github.io

See the interactions as blocks generated from $X = [x_1 | \dots | x_p]$:



$$Z = \left[Z_{\mathcal{B}_q(j)} \right]_{j=1}^{p}$$

$\Longrightarrow$ exploit the blocks in Z for the updates on Θ[12].

CBPG update on Θ

For $j = 1, \ldots, p$:

$$\Theta^{k+1}_{\mathcal{B}_q(j)} \longleftarrow \frac{1}{1 + \frac{1}{L_j}\lambda_{\Theta,\ell_2}} \mathrm{ST}\left(\Theta^{k}_{\mathcal{B}_q(j)} - \frac{1}{L_j n} Z^\top_{\mathcal{B}_q(j)}(X\beta^{k+1} + Z\Theta^k - y), \frac{1}{L_j}\lambda_{\Theta,\ell_1}\right)$$

- steps $L_j = \frac{\|Z^\top_{\mathcal{B}_q(j)} Z\|_2}{n}, j = 1, \ldots, p$ Lipschitz constants for each block
- computed with iterative method (Lánczos algorithm[13]).

[12] M. Massias (2019). "Sparse high dimensional regression in the presence of colored heteroscedastic noise: application to M/EEG source imaging". PhD thesis. Telecom Paristech; A. Beck (2017). *First-Order Methods in Optimization*. Vol. 25. SIAM.

[13] C. Lánczos (1952). "Solution of systems of linear equations by minimized iterations". In: *J. Res. Nat. Bur. Standards* 49.1, pp. 33–53.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real convex function.

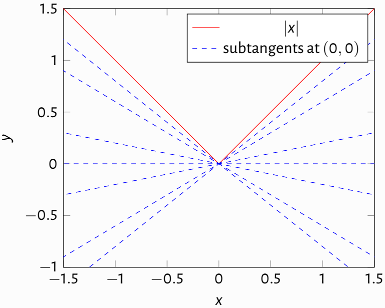
Subdifferential ∂f

At $x_0 \in \mathbb{R}^n$:

$$\partial f(x_0) = \{u \in \mathbb{R}^n,\ f(x) \geqslant f(x_0) + \langle u, x - x_0 \rangle\ \forall x \in \mathbb{R}^n\}$$

Example: The absolute value at the origin

$$\partial\,|\cdot|_{|x} = \begin{cases} \{-1\}, & \text{if } x < 0 \\ \{1\}, & \text{if } x > 0 \\ [-1, 1], & \text{if } x = 0 \end{cases}$$



Elastic-Net: $\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2 = \arg\min F_{enet}(\beta)$.

KKT violation

Our criterion: how much do we violate the KKT conditions:

$$d_{\|\cdot\|_\infty}(0, \partial F_{enet}(\beta)) \leqslant \epsilon \iff \inf_{g \in \partial F_{enet}(\beta)} \|g\|_\infty \leqslant \epsilon$$

Splitting along the coordinates, denoting $r = y - X\beta$:

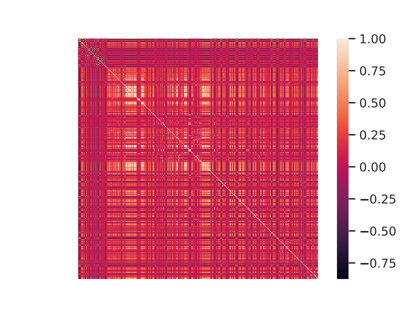
$$d\left(0, \frac{1}{n}X_j^\top(X\beta - y) + \lambda_1 \partial_{|\cdot|}(\beta_j) + \lambda_2\beta_j\right) = \frac{1}{n}\left|\mathrm{ST}\left(X_j^\top r - n\lambda_2\beta_j, n\lambda_1\right)\right|$$

X from Gaussian distribution, $n = 20000$, $p = 500$ (train/test = 75%/25%), $SNR = 10$, 1% of non-zero values in β^* and Θ^*.

- ℓ_1 penalty is $\frac{\lambda_{max}}{\ell_1 \ factor}$,
- ℓ_2 penalty is $\frac{\lambda_{max}}{10}$,
- $\epsilon = 10^{-3}$ (PGD did not converge for ℓ_1 factor > 250).

- CD faster at the beginning,
- CBPG faster after,
- convergence issues with PGD.

- $n = 19393$ samples (genes) and 531 features (141246 interactions *i.e.*, way too much!) [14]
- y is the gene expression in one patient (the first)

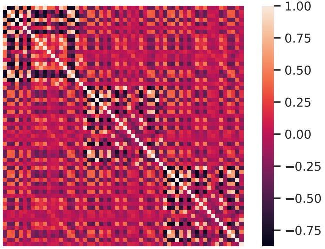


Correlation matrix of X

- 20 features for (di)nucleotides in Core region
- 20 Distal Upstream region promoter
- 20 in Distal Downstream region promoter
- 471 for motif scores in the Core region (in $[0, 1]$ close to 1)

[14] C. Bessière et al. (2018). "Probing instructions for expression regulation in gene nucleotide compositions". In: *PLOS Computational Biology* 14.1, pp. 1–28.

- $n = 19393$ samples (genes) and 531 features (141246 interactions *i.e.*, way too much!) [14]
- y is the gene expression in one patient (the first)



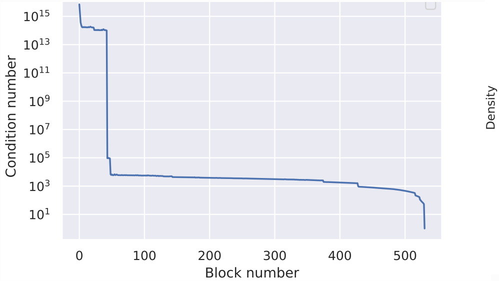
Correlation matrix of the first 60 features

- 20 features for (di)nucleotides in Core region
- 20 Distal Upstream region promoter
- 20 in Distal Downstream region promoter
- 471 for motif scores in the Core region (in $[0, 1]$ close to 1)

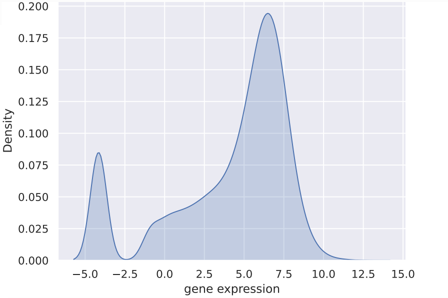
[14] C. Bessière et al. (2018). "Probing instructions for expression regulation in gene nucleotide compositions". In: *PLOS Computational Biology* 14.1, pp. 1–28.

Genomics dataset

What we need to know (numerically)



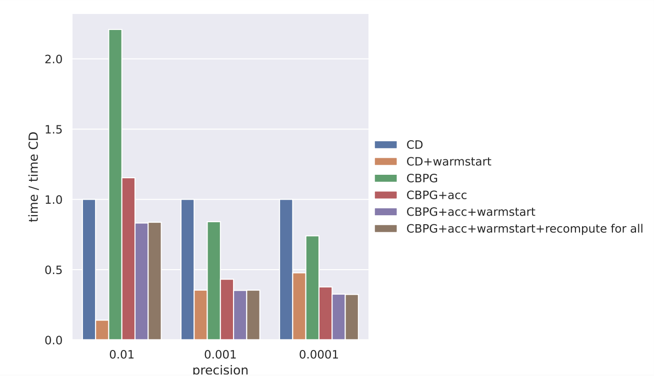
Very ill conditioned data (by block)



Preprocessing on y: log-transformed for unimodaltiy (shifted by $\epsilon > 0$):

Running our solvers **considering warmstarts**:

- ℓ_1 penalty: 10 log-spaced values on a grid from $\lambda_{\max}$ to $\lambda_{\max}/100$
- ℓ_2 penalty: set to $20\lambda_{\ell_1,\max}$



- All resulting in the same active features.

In short:

- **It is possible** to be faster using GPU and inertial acceleration …

Conclusion

In short:

- **It is possible** to be faster using GPU and inertial acceleration ...
- **but** there is a tradeoff with the precision.

In short:

- **It is possible** to be faster using GPU and inertial acceleration …
- **but** there is a tradeoff with the precision.
- CBPG algorithms needs more epochs than CD but compute them faster.

In short:

- **It is possible** to be faster using GPU and inertial acceleration …
- **but** there is a tradeoff with the precision.
- CBPG algorithms needs more epochs than CD but compute them faster.

Possible leads:

- Consider other types of accelerations,
- Compute more precise convergence rates for the KKT violation criterion.
- Keep working on the `BenchOpt` library.

- Bascou, F., S. Lèbre, and J. Salmon (2020). “Debiasing the Elastic Net for models with interactions”. In: *Journées de Statistique*.
- Beck, A. (2017). *First-Order Methods in Optimization*. Vol. 25. SIAM.
- Bertrand, Q. and M. Massias (2021). *Anderson acceleration of coordinate descent*.
- Bessière, C. et al. (Jan. 2018). “Probing instructions for expression regulation in gene nucleotide compositions”. In: *PLOS Computational Biology* 14.1, pp. 1–28.
- Lam, S., A. Pitrou, and S. Seibert (2015). “Numba: A llvm-based python jit compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6.
- Lánczos, C. (1952). “Solution of systems of linear equations by minimized iterations”. In: *J. Res. Nat. Bur. Standards* 49.1, pp. 33–53.
- Massias, M. (2019). “Sparse high dimensional regression in the presence of colored heteroscedastic noise: application to M/EEG source imaging”. PhD thesis. Telecom Paristech.

References II

- Nesterov, Y. (1983). “A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$”. In: *Sov. Math. Dokl.* Vol. 27. 2.
- — (2012). “Efficiency of coordinate descent methods on huge-scale optimization problems”. In: *SIAM Journal on Optimization* 22.2, pp. 341–362.
- Parikh, N. and S. Boyd (2014). “Proximal Algorithms”. In: *Found. Trends Optim.* 1.3, pp. 127–239.
- Tibshirani, R. (1996). “Regression Shrinkage and Selection via the Lasso”. In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1, pp. 267–288.
- Tikhonov, A. (1943). “On the stability of inverse problems”. In: *Dokl. Akad. Nauk SSSR* 39, pp. 176–179.
- Zou, H. and T. Hastie (2005). “Regularization and variable selection via the elastic net”. In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67.2, pp. 301–320.

Thank you for your attention!

https://benchopt.github.io/results

BenchOpt benchmark results

Last updated: 2021-06-08 15:02

8 benchmarks in total.

Available Benchmarks

HUBER L2

LASSO

3 FILES AVAILABLE

LOGREG L1

LOGREG L2

MCP

NNLS

OLS

QUANTILE REGRESSION

cpu
Any

ram (GB)
Any

cuda
Any

BenchOpt results: logreg l2

Last updated: 2021-06-08 15:01 with **3** benchmark results in total.

☰ Filter Informations

Show 10 entries

Search:

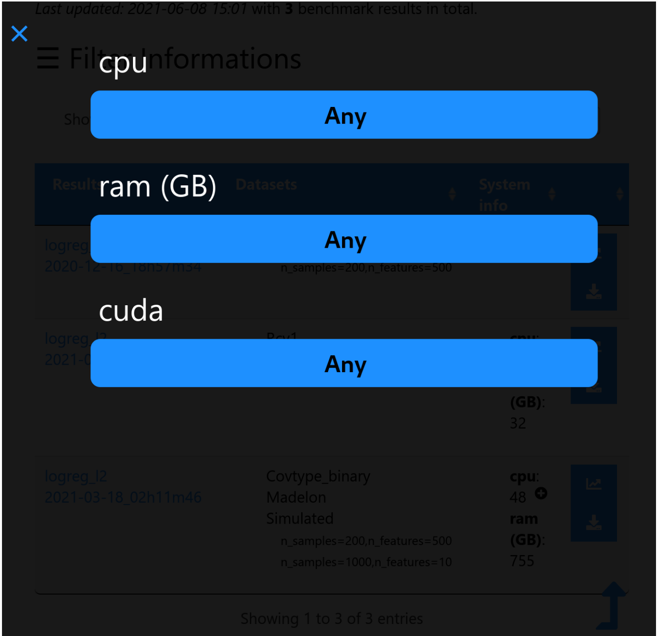
Results	Datasets	System info	
logreg_l2 2020-12-16_18h57m34	Simulated n_samples=200,n_features=500		
logreg_l2 2021-03-06_13h48m00	Rcv1	**cpu**: 16 **ram (GB)**: 32 **platform**: Darwin19.0.0-x86_64 **processor**: Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	
logreg_l2 2021-03-18_02h11m46	Covtype_binary Madelon Simulated n_samples=200,n_features=500 n_samples=1000,n_features=10	**cpu**: 48 **ram (GB)**: 755	

Showing 1 to 3 of 3 entries

Previous 1 Next

The BenchOpt library

Filter for mobile devices



Result on benchmark_quantile_regression benchmark

benchmark_quantile_regression_benchopt_run_2021-04-01_13h28m43.csv

System informations: **cpu**: 16 **ram (GB)**: 32

- **platform**: Darwin19.0.0-x86_64
- **processor**: Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz
- **numpy**: 1.19.4 blas=NO_ATLAS_INFO lapack=lapack
- **scipy**: 1.6.2

Dataset Simulated[n_samples=100,n_features=50] Objective L1-regularized Quantile Regression[reg=0.05,quantile=0.2] Kind suboptimality_curve
Log-scale semilog-y

The BenchOpt library
Interactive results