Theory and practice of coordinate descent: tutorial and actual acceleration

Mathurin MASSIAS (Inria Lyon)
https://mathurinm.github.io

Joint work with Quentin BERTRAND (MILA)

Statistical motivation: generalized linear models

- supervised learning framework:

$$\text{i.i.d dataset } (a_i, y_i)_{i\in[n]} \in \mathbb{R}^p \times \mathcal{Y}$$

- generalized linear model: parameters of the target distribution depend linearly on a_i:

$$y_i | a_i \sim \phi(a_i^\top x)$$

- inference through negative log-likelihood minimization

$$\boxed{x^* \in \arg\min_{x\in\mathbb{R}^p} \sum_{i=1}^{n} f_i(a_i^\top x) = F(x)}$$

Ex: least squares, $f_i = \frac{1}{2}(y_i - \cdot)^2$, $F(x) = \frac{1}{2}\|Ax - y\|_2^2$

First order methods: gradient descent

$$\boxed{x^* \in \arg\min_{x \in \mathbb{R}^p} \sum_{i=1}^{n} f_i(a_i^\top x) = F(x)}$$

Ass: F is L-smooth: ∇F L-Lipschitz

Gradient descent is an iterative algorithm:

$$x^{(k+1)} = x^{(k)} - \frac{1}{L} \nabla F(x^{(k)})$$

Interpretation: F is upper bounded by an isotropic parabola:

$$\text{upper}(x) = F(x^{(k)}) + \langle \nabla F(x^{(k)}), x - x^{(k)} \rangle + \frac{L}{2} \left\| x - x^{(k)} \right\|^2 \geq F(x)$$

GD iteratively computes and minimizes this upper bound

Gradient descent: cost and rate

$$\nabla F(x) = A^\top \begin{pmatrix} f_1'(a_1^\top x) \\ \vdots \\ f_n'(a_n^\top x) \end{pmatrix}$$

$\hookrightarrow$ cost: 1 multiplication by $A \in \mathbb{R}^{n \times p}$, $\mathcal{O}(np)$

Thm: F convex L-smooth, rate of GD:

$$F(x^{(k)}) - F(x^*) \leq \frac{2L\|x^{(0)} - x^*\|^2}{k} = \mathcal{O}(1/k)$$

Like all first order methods (no Hessian, no n^3/p^3)), GD is good to solve large problems up to moderate accuracy

Strongly convex case

Def: F μ-strongly convex:

$$F(\lambda x + (1-\lambda)y) \leq \lambda F(x) + (1-\lambda)F(y) - \mu\frac{\lambda(1-\lambda)}{2}\|x-y\|^2$$

if F C^2, means $\nabla^2 F(x) \succeq \mu \,\mathrm{Id} \quad \forall x$

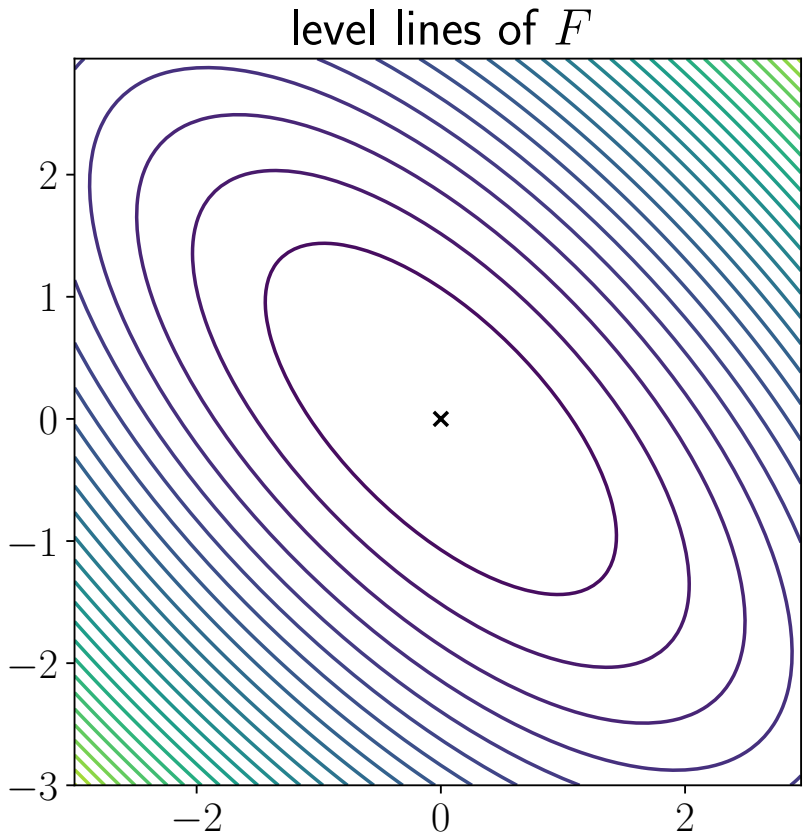
Thm: F convex L-smooth, μ-strongly convex, rate of GD:

$$\left\|x^{(k)} - x^*\right\|^2 \leq \exp\left(-\frac{\mu}{L}k\right)\left\|x^{(0)} - x^*\right\|^2$$

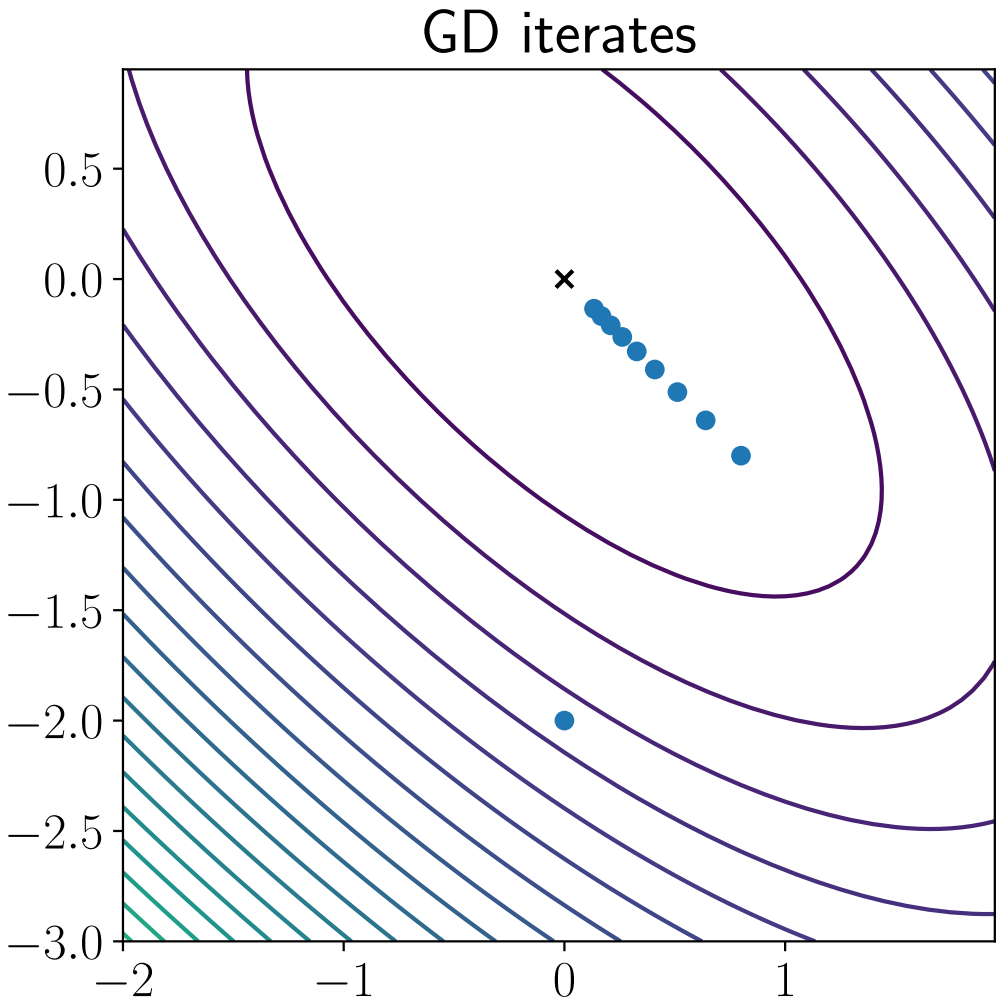
Much better than $1/k$, but we can have $\mu/L \sim 10^{-6}$

Isotropic parabola may be coarse

$$F(x_1, x_2) = (x_1 - x_2)^2 + 5(x_1 + x_2)^2$$

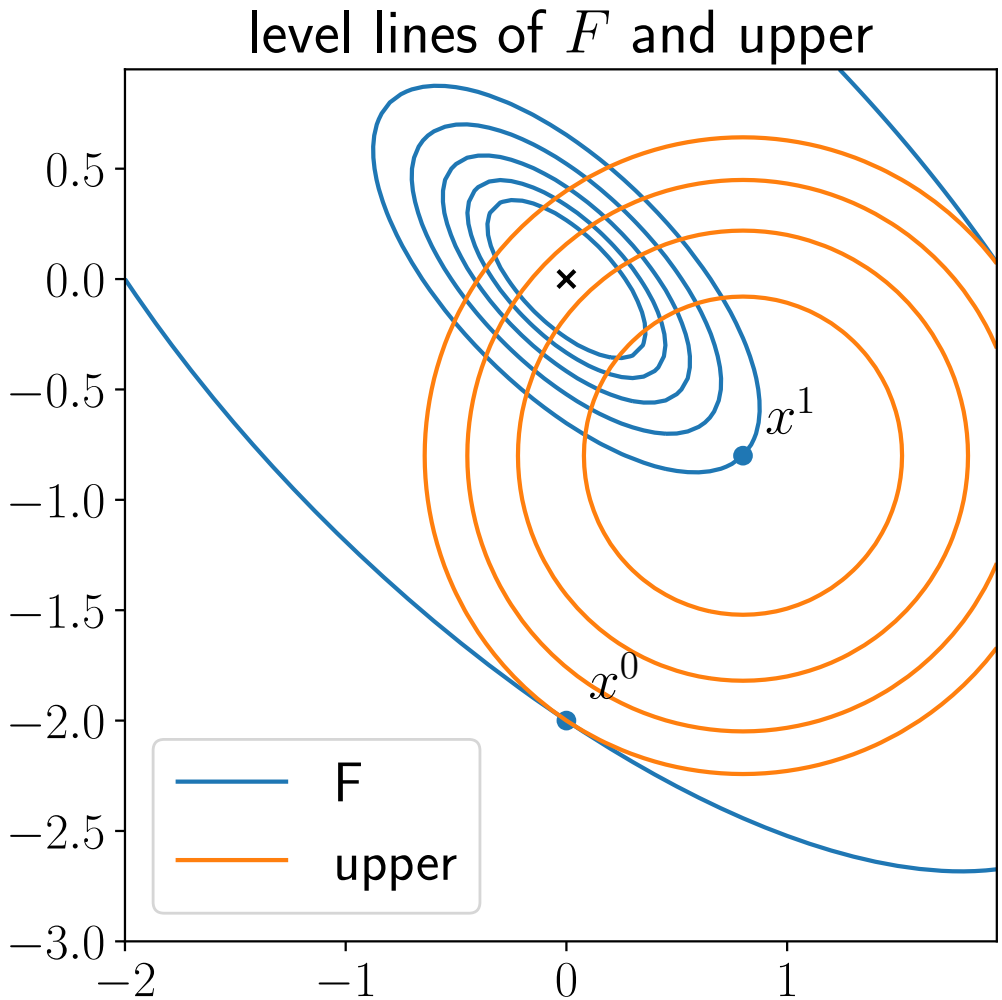


Iterates of gradient descent



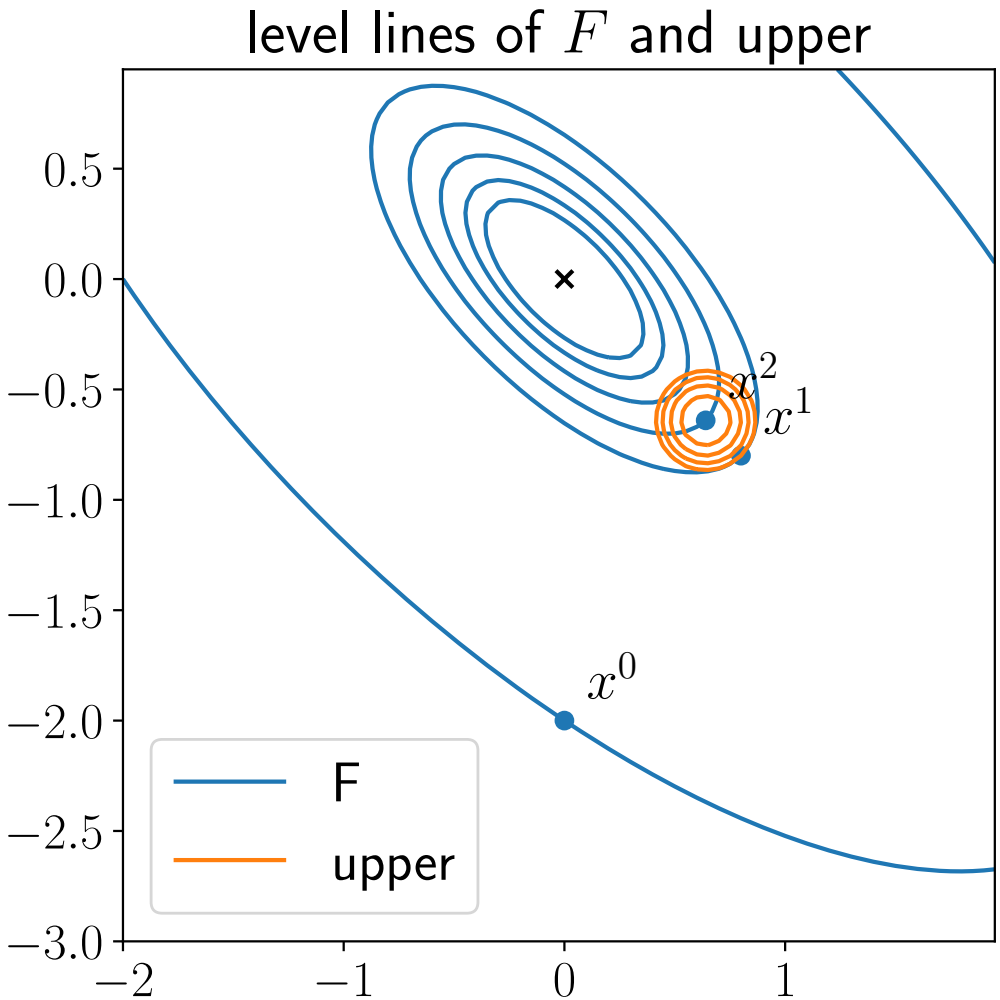
GD makes very small steps in the second direction because its upper bound is crude

Iterates of gradient descent



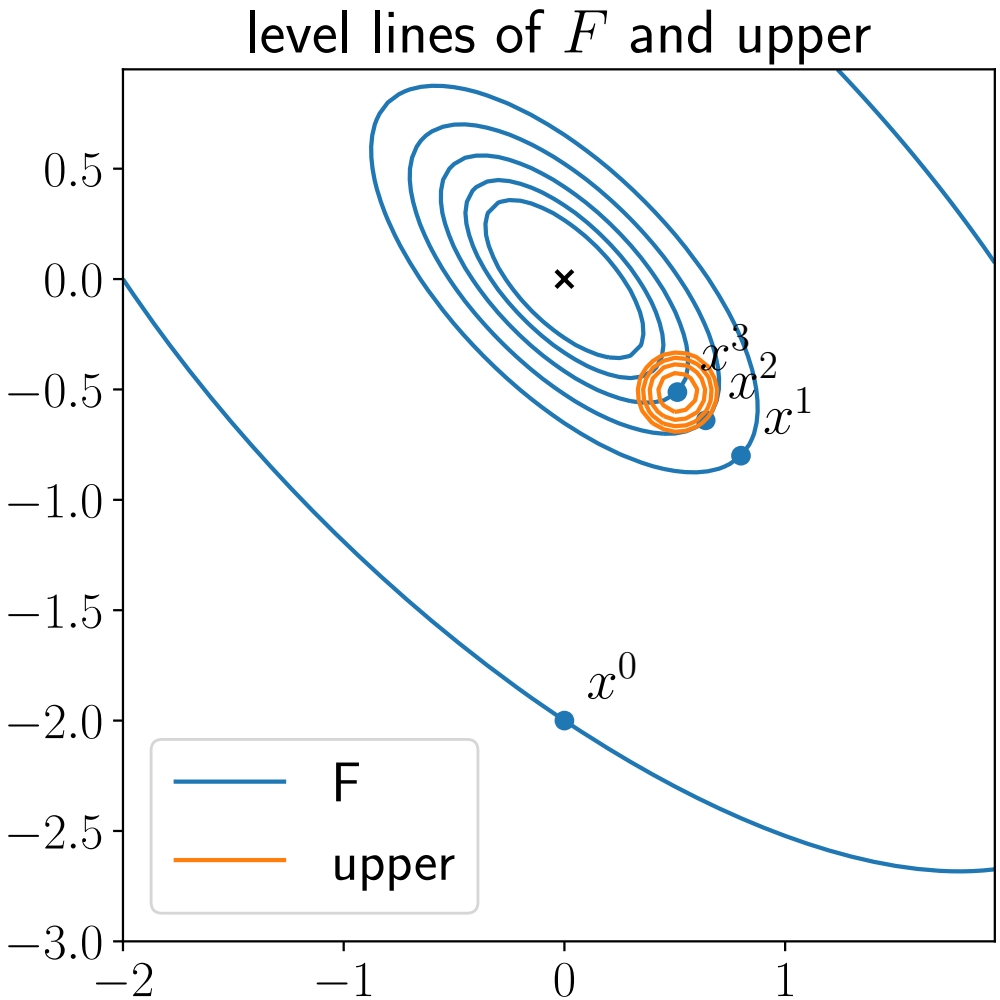
GD makes very small steps in the second direction because its upper bound is crude

Iterates of gradient descent



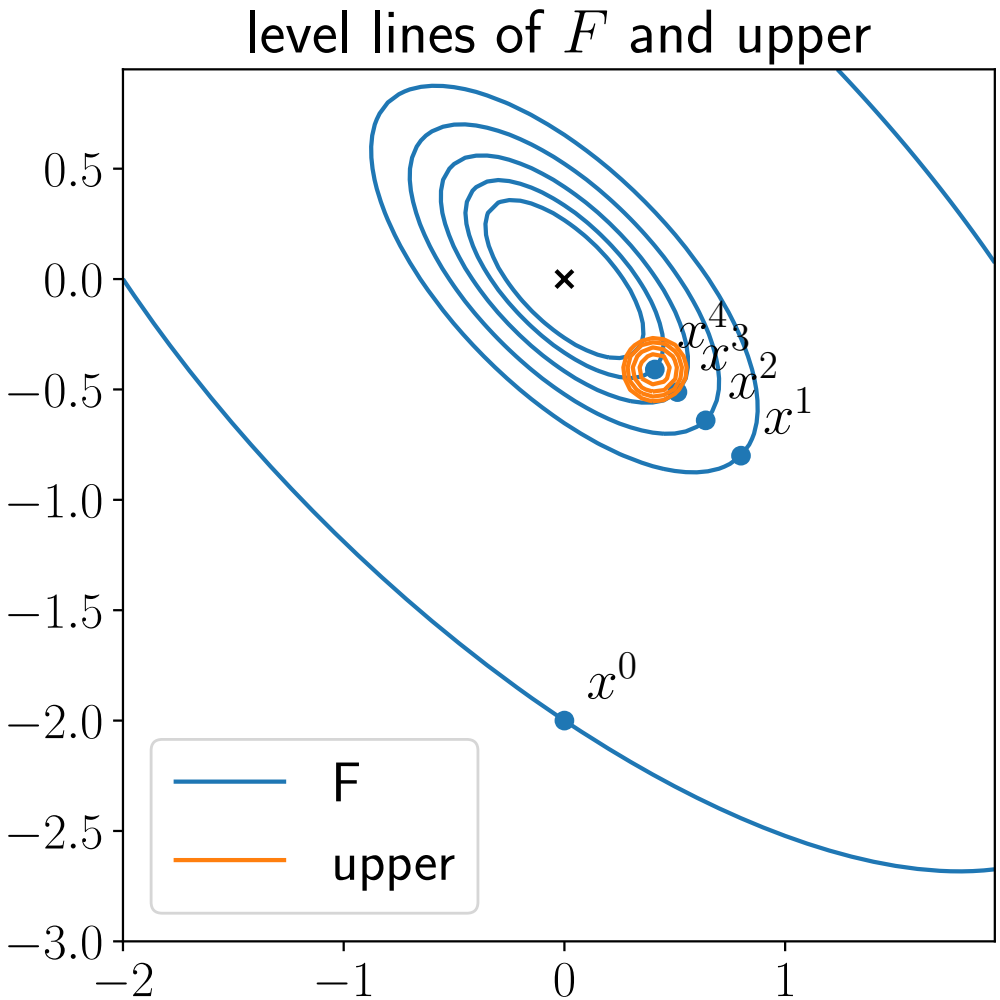
GD makes very small steps in the second direction because its upper bound is crude

Iterates of gradient descent



GD makes very small steps in the second direction because its upper bound is crude

Iterates of gradient descent



GD makes very small steps in the second direction because its upper bound is crude

Coordinate (gradient) descent

Simple idea:

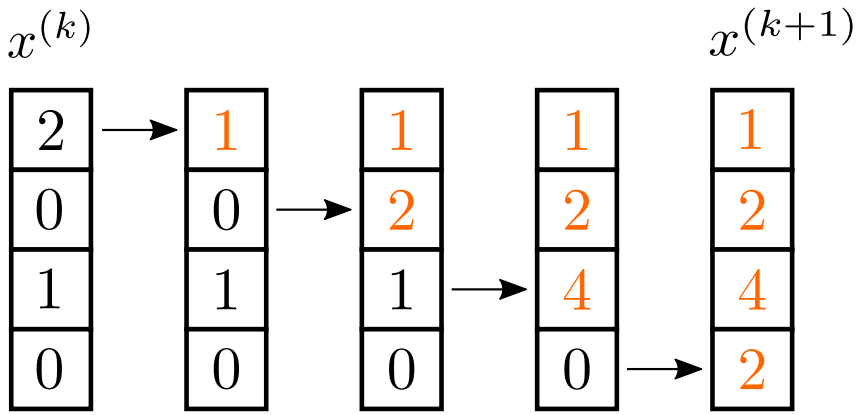
- fix all variables but 1 ($j \in [p]$)
- perform 1D gradient step on this variable only (tighter parabola)

Cyclic coordinate descent:

$$x_j^{(k+1)} = \begin{cases} x_j^{(k)} - \frac{1}{L_j} \nabla_j F(x^{(k)}) & \text{if } j = k \,(\text{mod}\, p) \\ x_j^{(k)} & \text{otherwise} \end{cases}$$

where $\quad |\nabla_j F(x + h e_j) - \nabla_j F(x)| \leq L_j |h| \qquad \forall x$

Principle of coordinate descent



CD on least squares

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

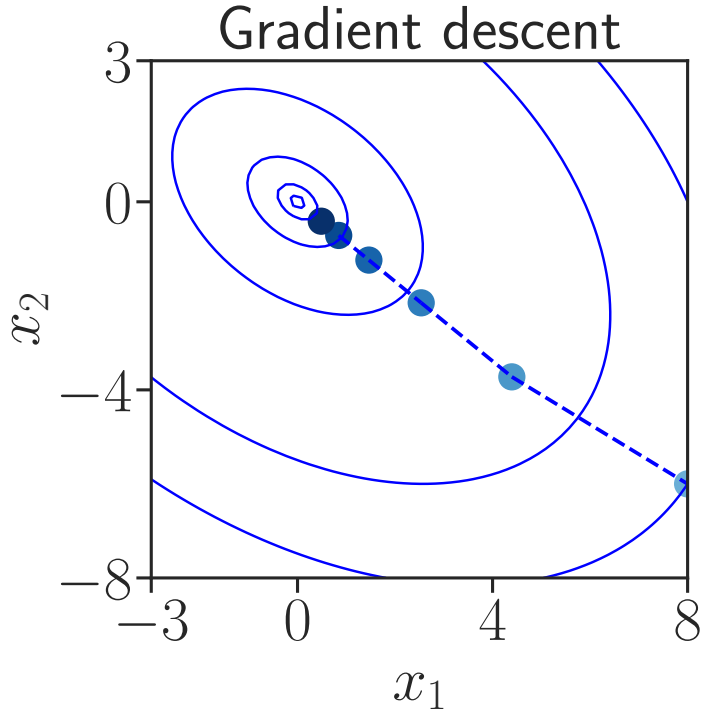
Algorithm: Gradient descent

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

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

$\quad x \leftarrow x - \frac{A^\top (Ax - y)}{\|A\|_2^2}$

return x



Algorithm: CD

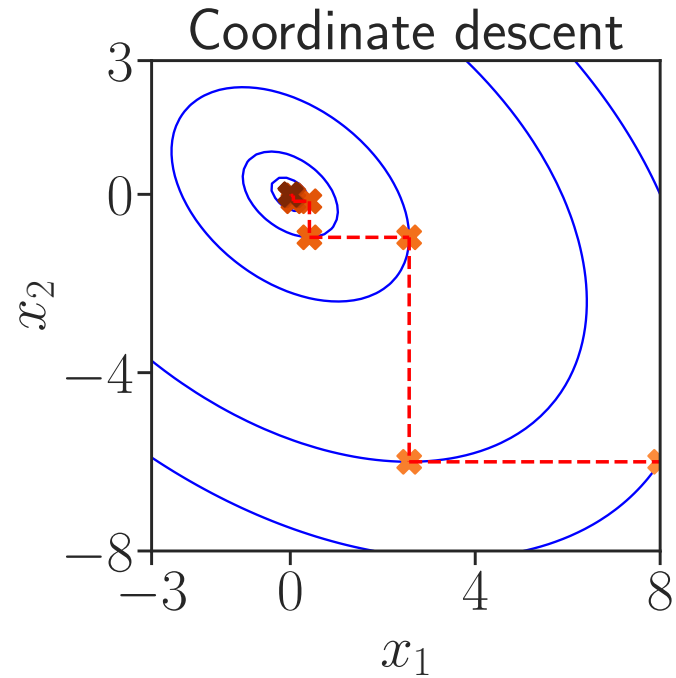
init : $x \in \mathbb{R}^p$

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

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

$\quad x_j \leftarrow x_j - \frac{A_{:j}^\top (Ax - y)}{\|A_j\|^2}$

return x



When can CD be practically used?

In principle computing $\nabla_j F$ as expensive as ∇F

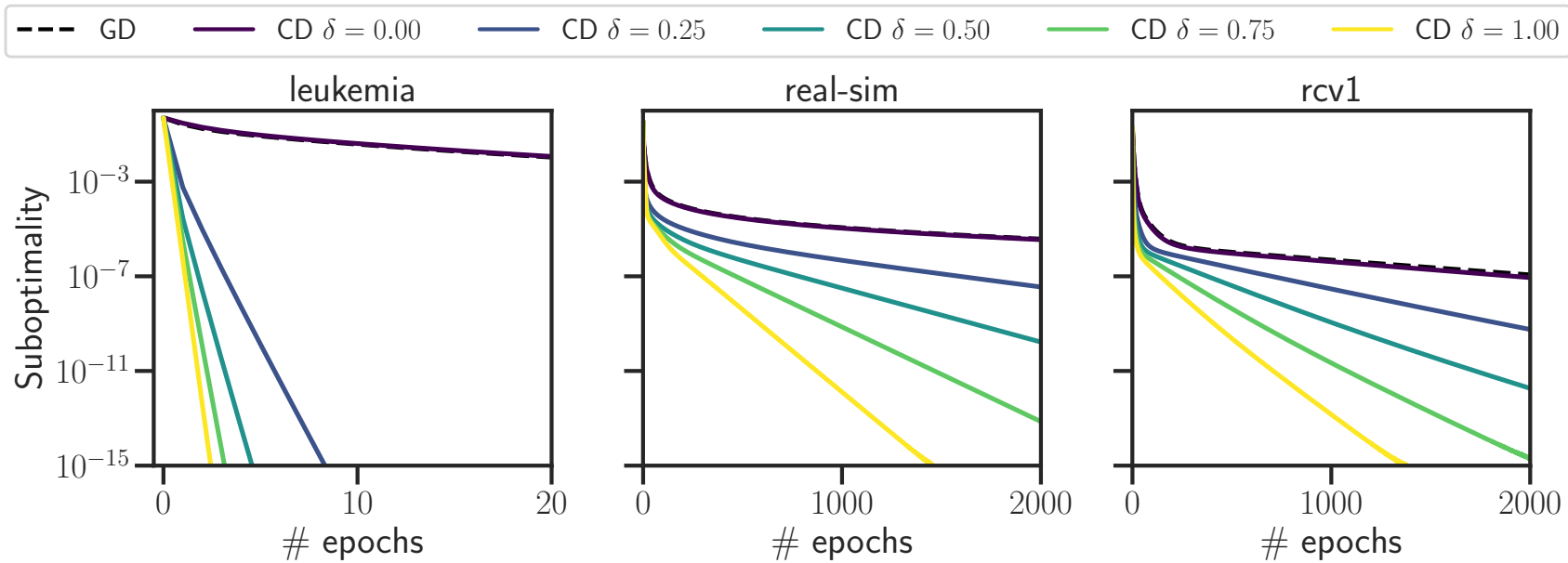
But not in our case! $F(x) = f(Ax)$ and f separable + simple:

$$\nabla_j F(x) = A_{:j}^\top \nabla f(Ax) = A_{:j}^\top \begin{pmatrix} f_1'(a_1^\top x) \\ \dots \\ f_n'(a_n^\top x) \end{pmatrix}$$

Combine with keeping Ax up-to-date, cost is $\mathcal{O}(p)$

$\hookrightarrow$ one CD update of each entry in x costs np, as for GD

Insights on the success of CD



Stepsize influence for coordinate descent, OLS. Gradient descent against CD with step sizes $\gamma_j = \delta/L_j + (1-\delta)/L$

$\delta = 0$: CD with stepsize of GD, behaves as GD

improvements as stepsize increases ($\delta \to 1$)

Worst case analysis on quadratics

Algorithm	GD	Random CD	Cyclic CD
Worst-case factor	$\frac{L}{\mu}$	$\frac{\sum_1^p L_j}{p\mu}$	$\frac{p\sum_1^p L_j}{\mu}$

Worst case convergence factors on quadratics $x \mapsto \frac{1}{2}x^\top Hx$, $\mu \stackrel{\Delta}{=} \lambda_{\min}(H), L \stackrel{\Delta}{=} \lambda_{\max}(H), L_j = H_{jj}$. **Lower is better**.

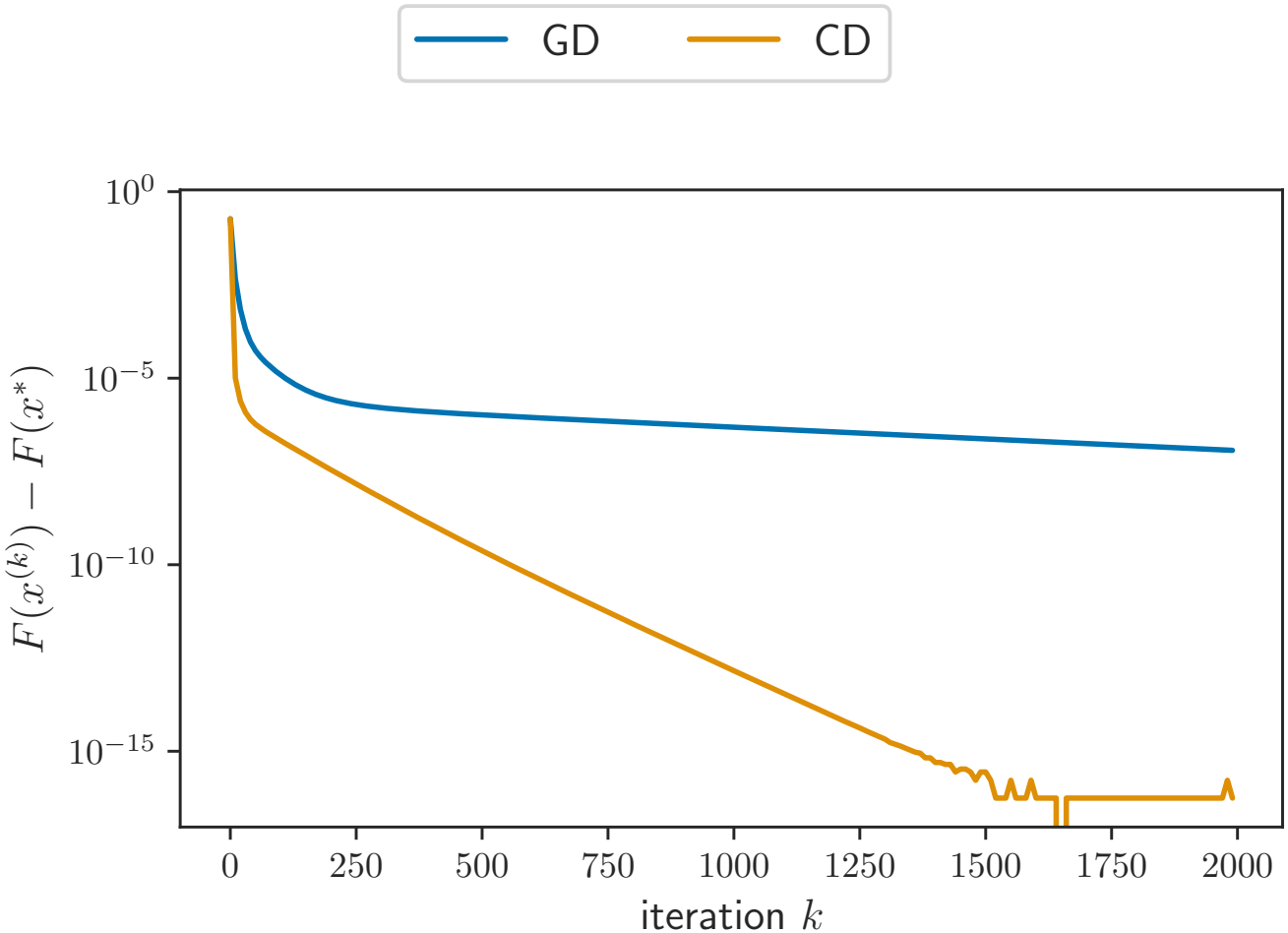
RCD is better than GD[1]:

$$\frac{1}{p}\sum L_j = \frac{1}{p}\left\|\operatorname{spec} H\right\|_1 \leq L = \left\|\operatorname{spec} H\right\|_\infty$$

and it may even happen that $\frac{1}{p}\sum L_j \lesssim \frac{1}{p}L$

[1] in practice cyclic is at least as good as random

CD outperforms GD



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

Can we fix gradient descent?

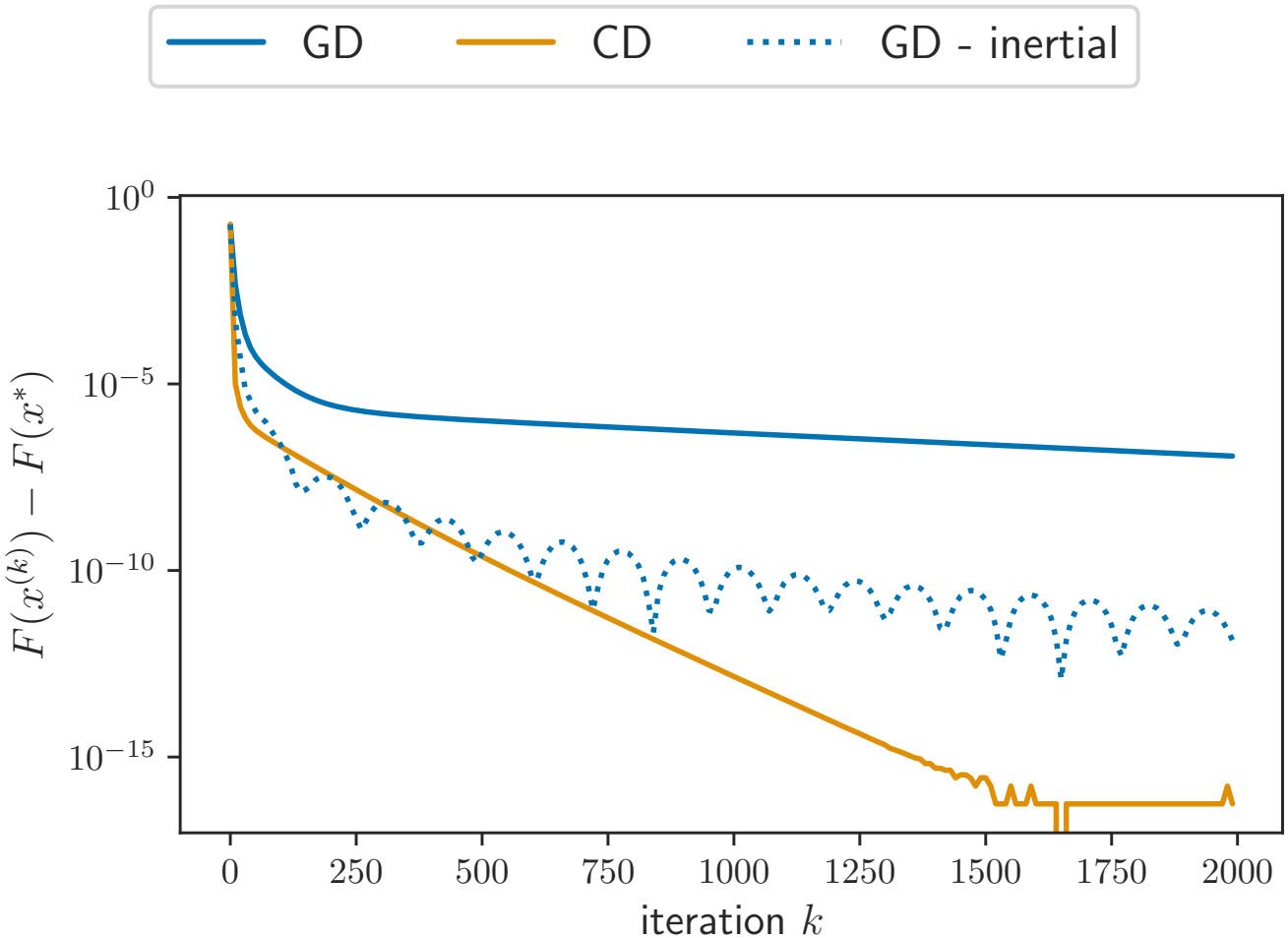
Nesterov acceleration/heavy ball/momentum (various ρ_k)

$$x^{(k+1)} = y^{(k)} - \frac{1}{L}\nabla F(y^{(k)})$$
$$y^{(k+1)} = x^{(k+1)} + \rho_k(x^{(k+1)} - x^{(k)})$$

Rate improvement :

- $\mathcal{O}(1/k) \to \mathcal{O}(1/k^2)$ (optimal)
- $\mathcal{O}(\exp(-\frac{\mu}{L}k)) \to \mathcal{O}(\exp(-\sqrt{\frac{\mu}{L}}k))$

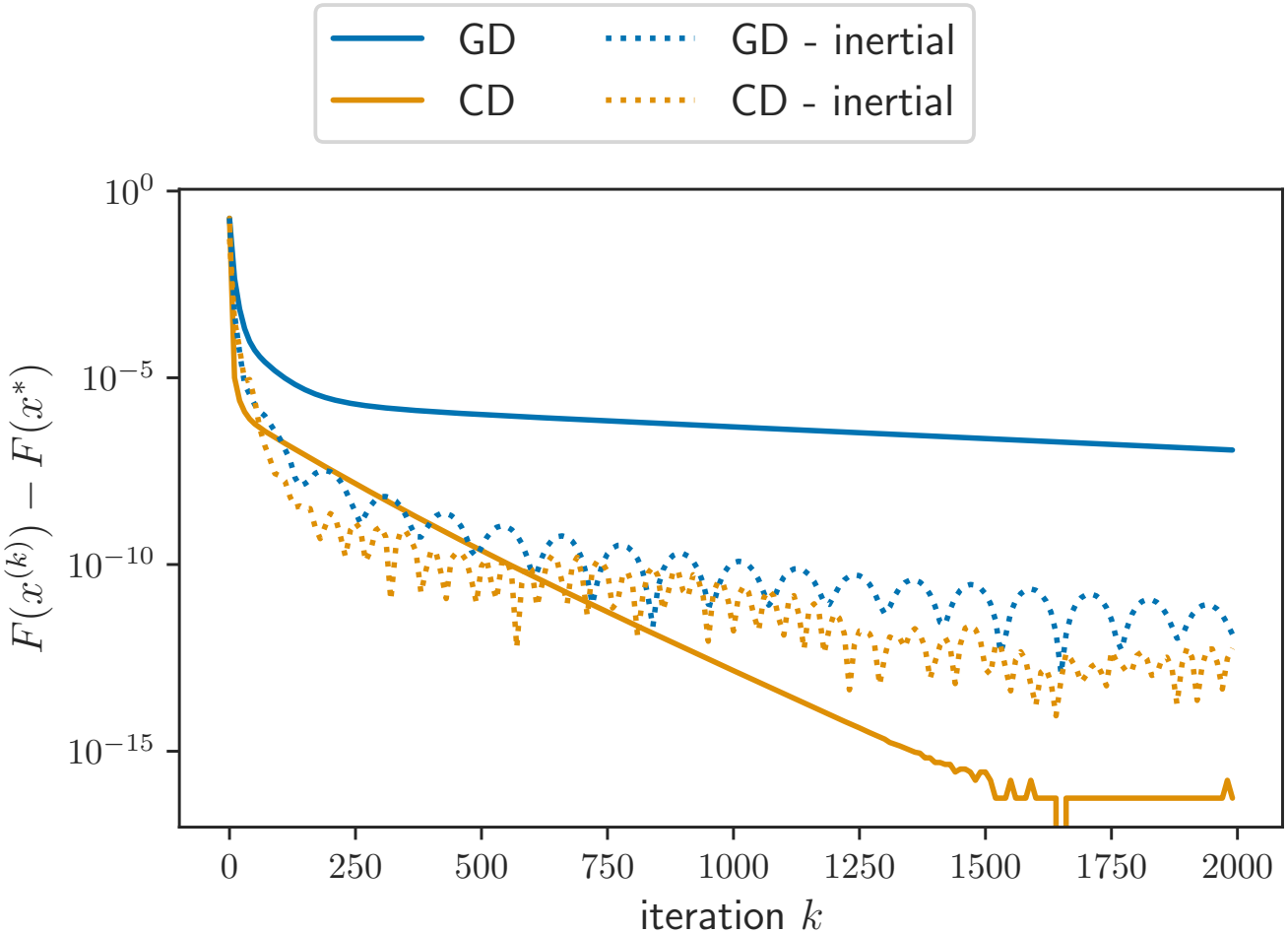
Nesterov acceleration of GD



Nesterov/momentum largely improves gradient descent[2]

[2] even more with *restart*, but out of scope today

Nesterov acceleration of CD?

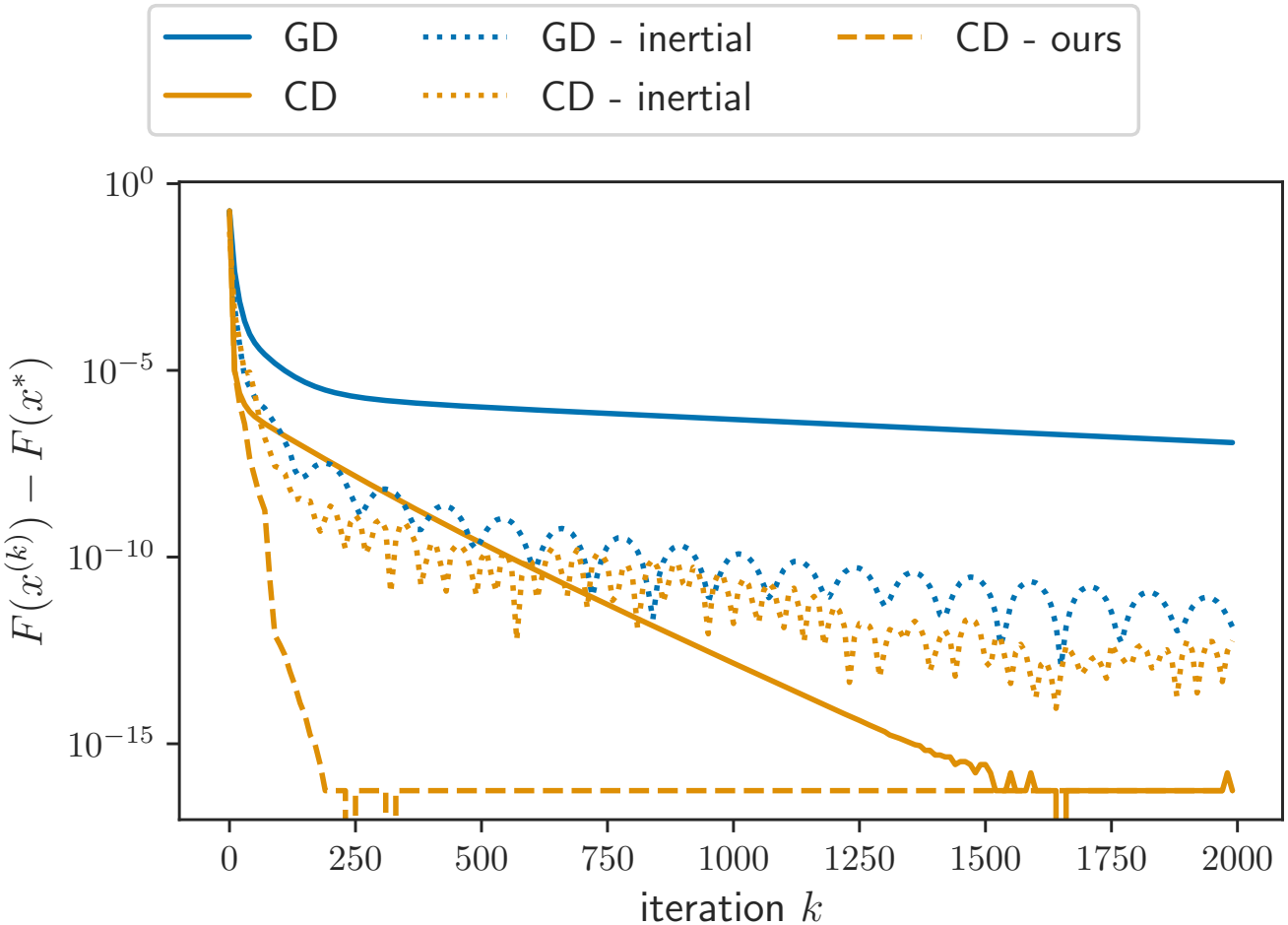


Nesterov **accelerated CD**[3,4] eventually **slows down** convergence

[3] Q. Lin, Z. Lu, and L. Xiao. "An Accelerated Proximal Coordinate Gradient Method". In: *NeurIPS*. 2014, pp. 3059–3067.

[4] O. Fercoq and P. Richtárik. "Accelerated, parallel, and proximal coordinate descent". In: *SIAM Journal on Optimization* 25.4 (2015), pp. 1997–2023.

A successful, practical acceleration of CD



Our version **accelerated CD**[5] results in **practical gains**

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

Anderson acceleration[6]: intuition

How to accelerate fixed point algorithms

$$x^{(k+1)} = Tx^{(k)} + b \ ? \tag{1}$$

Idea: search a fixed point of the form

$$x^* = \sum_{i=1}^{k} c_i x^{(k-1)}$$

[6] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560.

Anderson acceleration[6]: intuition

How to accelerate fixed point algorithms

$$x^{(k+1)} = Tx^{(k)} + b \ ? \tag{1}$$

Idea: search a fixed point of the form

$$x^* = \sum_{i=1}^{k} c_i x^{(k-1)}$$

From (1), one should have:

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

[6] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560.

Anderson acceleration[6]: intuition

How to accelerate fixed point algorithms

$$x^{(k+1)} = Tx^{(k)} + b \ ? \tag{1}$$

Idea: search a fixed point of the form

$$x^* = \sum_{i=1}^{k} c_i x^{(k-1)}$$

From (1), one should have:

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

Choose c_i such that

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

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

[6] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560.

Anderson acceleration[6]: intuition

How to accelerate fixed point algorithms

$$x^{(k+1)} = Tx^{(k)} + b \ ? \tag{1}$$

Idea: search a fixed point of the form

$$x^* = \sum_{i=1}^{k} c_i x^{(k-1)}$$

From (1), one should have:

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

Choose c_i such that

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

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

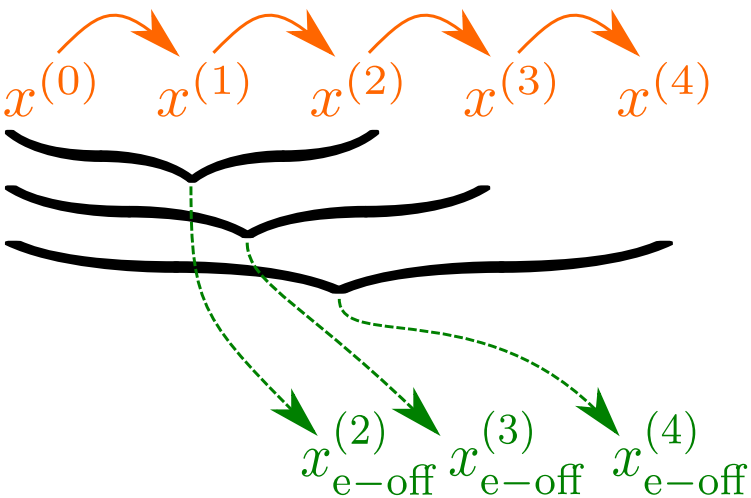
[6] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560.

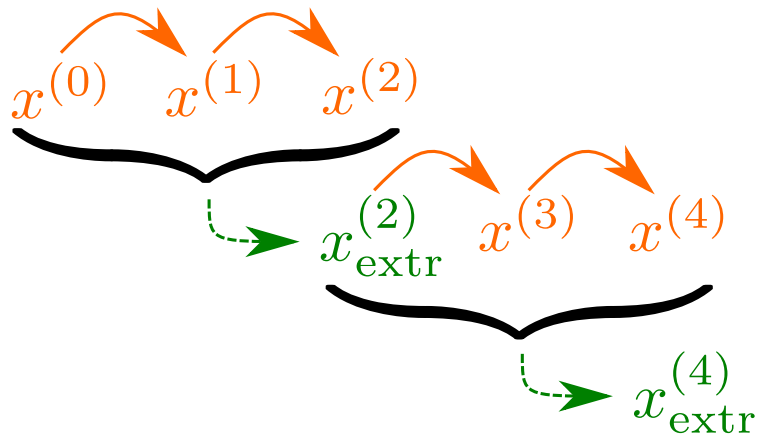
Offline vs online extrapolation

Candidate point:

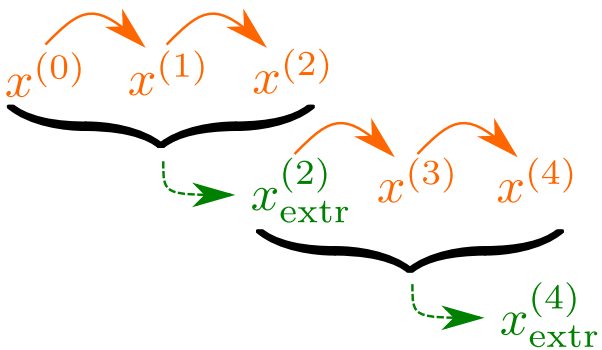
$$x^* \approx \sum_{i=1}^{k} c_i x^{(k-1)}$$

Finding $(c_i)_{i\in[k]}$ requires solving a $k \times k$ linear system





Online Anderson acceleration



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

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

$x^{(k)} = Tx^{(k-1)} + b$ // regular iter.

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

$U = [x^{(k-K+1)} - x^{(k-K)}, \ldots,]$

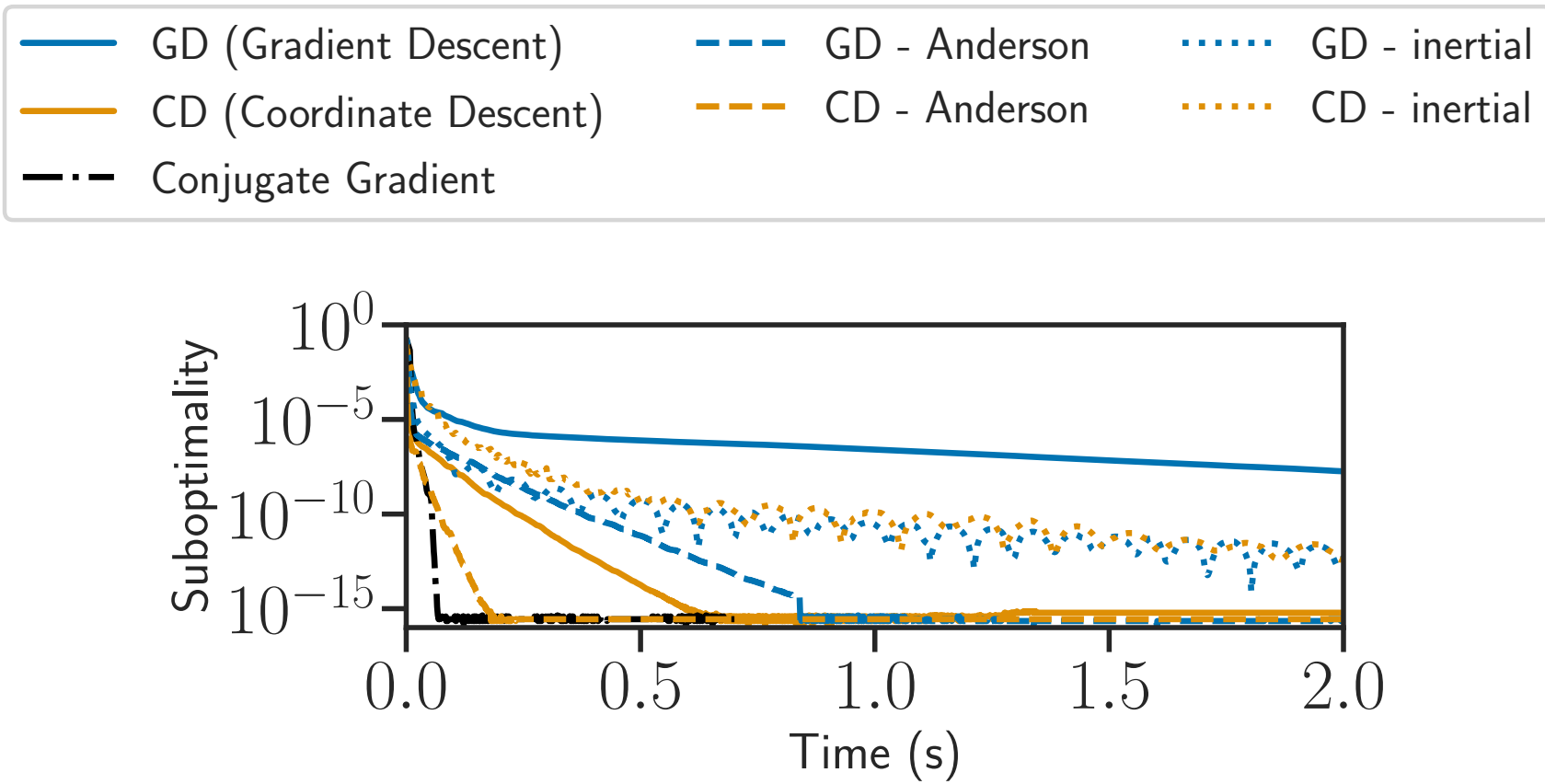
$c = (U^\top U)^{-1} \mathbf{1}_K$

$x^{(k)}_{\text{extr}} = \sum_i^K c_i x^{(k-K+i)} / \sum_i c_i$

$x^{(k)} = x^{(k)}_{\text{extr}}$ // base sequence changes

return $x^{(k)}$

Time acceleration of CD



Least squares on subsampled *news20*

- Anderson acceleration provides *time* speedups for CD

Theoretical properties

Proposition (Symmetric T)

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

$$\|x_{extr}^{(k)} - x^*\|_B \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|x^{(0)} - x^*\|_B \ .$$

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

Symmetric T: gradient descent $T = \mathrm{Id} - \frac{1}{L} X^\top X$ ✓

Coordinate descent?

Theoretical properties

Proposition (Symmetric T)

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

$$\|x_{extr}^{(k)} - x^*\|_B \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|x^{(0)} - x^*\|_B \ .$$

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

Symmetric T: gradient descent $T = \mathrm{Id} - \frac{1}{L} X^\top X$ ✓

Coordinate descent?

Coordinate descent (CD)

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

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

- The updates of coordinate descent write, for all $j \in 1, \dots, p$:

$$x_j \leftarrow x_j - (H_{j:} x + b_j)/H_{jj}$$

- One pass on all the coordinates gives a **fixed point iteration**:

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

$$T = \left(\mathrm{Id}_p - e_p e_p^\top H / H_{pp}\right) \dots \left(\mathrm{Id}_p - e_1 e_1^\top H / H_{11}\right)$$

nonsymmetric ✗

Theoretical properties

Weak theoretical properties for AA with non-symmetric T[7]

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

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

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

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

Extension to the composite model

$$x^* \in \arg\min_{x \in \mathbb{R}^p} f(Ax) + g(x)$$

g regularizer, often not smooth (to enforce some structure[8] on x^*)

For CD to remain applicable, we need g *separable* and to know its proximal operator

$$g(x) = \sum_1^p g_j(x_j)$$

$$\operatorname{prox}_{\gamma g_j}(u) = \arg\min_{v \in \mathbb{R}} \tfrac{1}{2} \|u - v\|^2 + \gamma g_j(v)$$

[8] F. Iutzeler and J. Malick. "Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications". In: *Set-Valued and Variational Analysis* 28.4 (2020), pp. 661–678.

Successful applications of coordinate descent

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

In ML CD is state-of-the-art[9, 10] for such problems:

- Lasso $\arg\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - Ax\|^2 + \lambda\|x\|_1$
- Elastic net $\arg\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - Ax\|^2 + \lambda\|x\|_1 + \frac{\rho}{2}\|x\|_2^2$
- sparse logistic regression
- (dual) SVM

[9] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *JMLR* 12 (2011), pp. 2825–2830.

[10] J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (2007), pp. 302–332.

Non smooth case

Key idea: even if g is non-smooth, it is well-behaved: CD identifies the correct structure in x^* and non-smoothness vanishes[11]
CD update:

$$x^{(k+1)} = \psi(x^{(k)}) \ . \tag{2}$$

Proposition (Proximal CD)

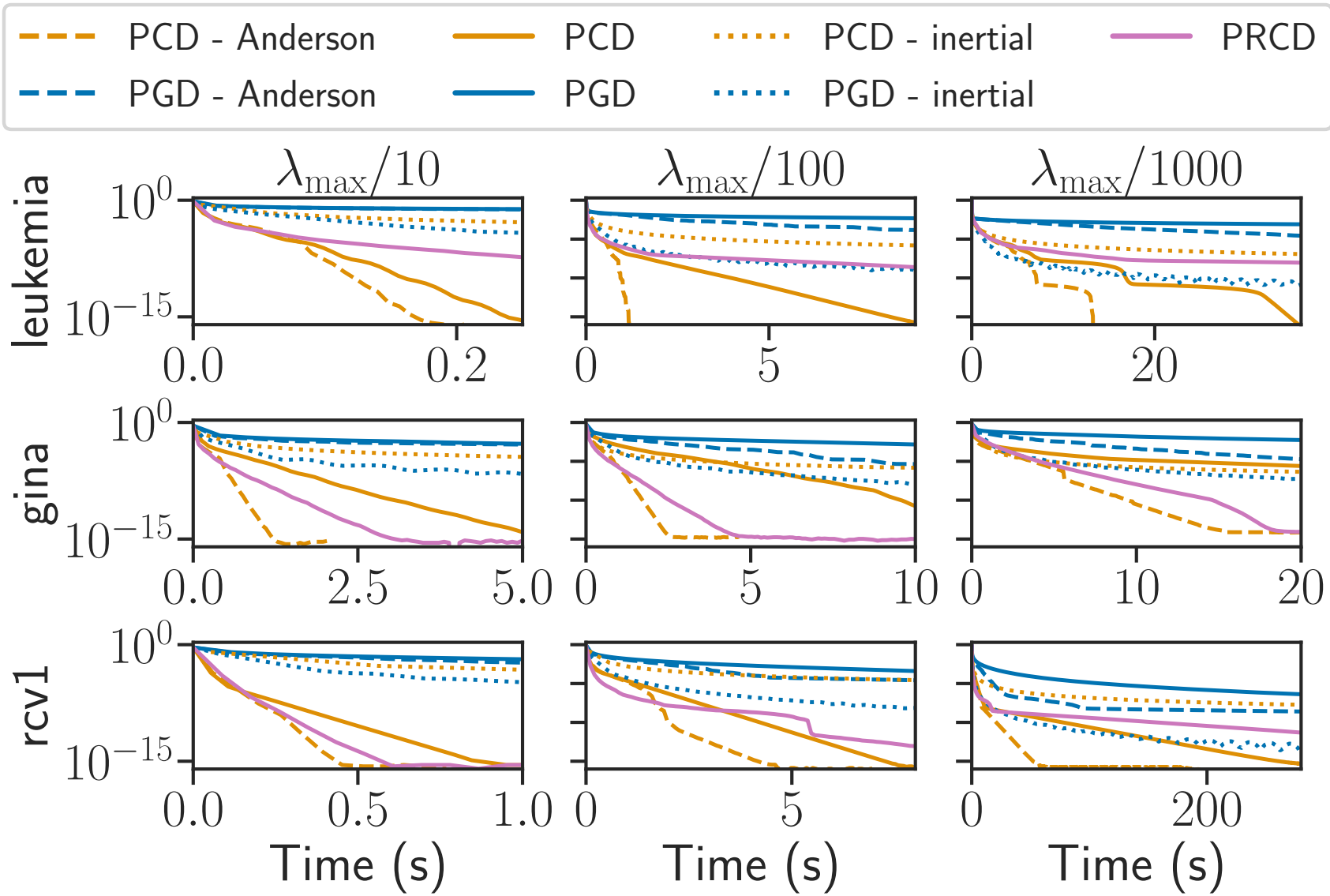
If f is convex and smooth and $\mathcal{C}^2$, g_j are convex smooth and $\mathcal{C}^2$, then eventually ψ is differentiable and[a]

$$x^{(k+1)} = D\psi(x^*)(x^{(k)} - x^*) + x^* + o(\|x^{(k)} - x^*\|) \ .$$

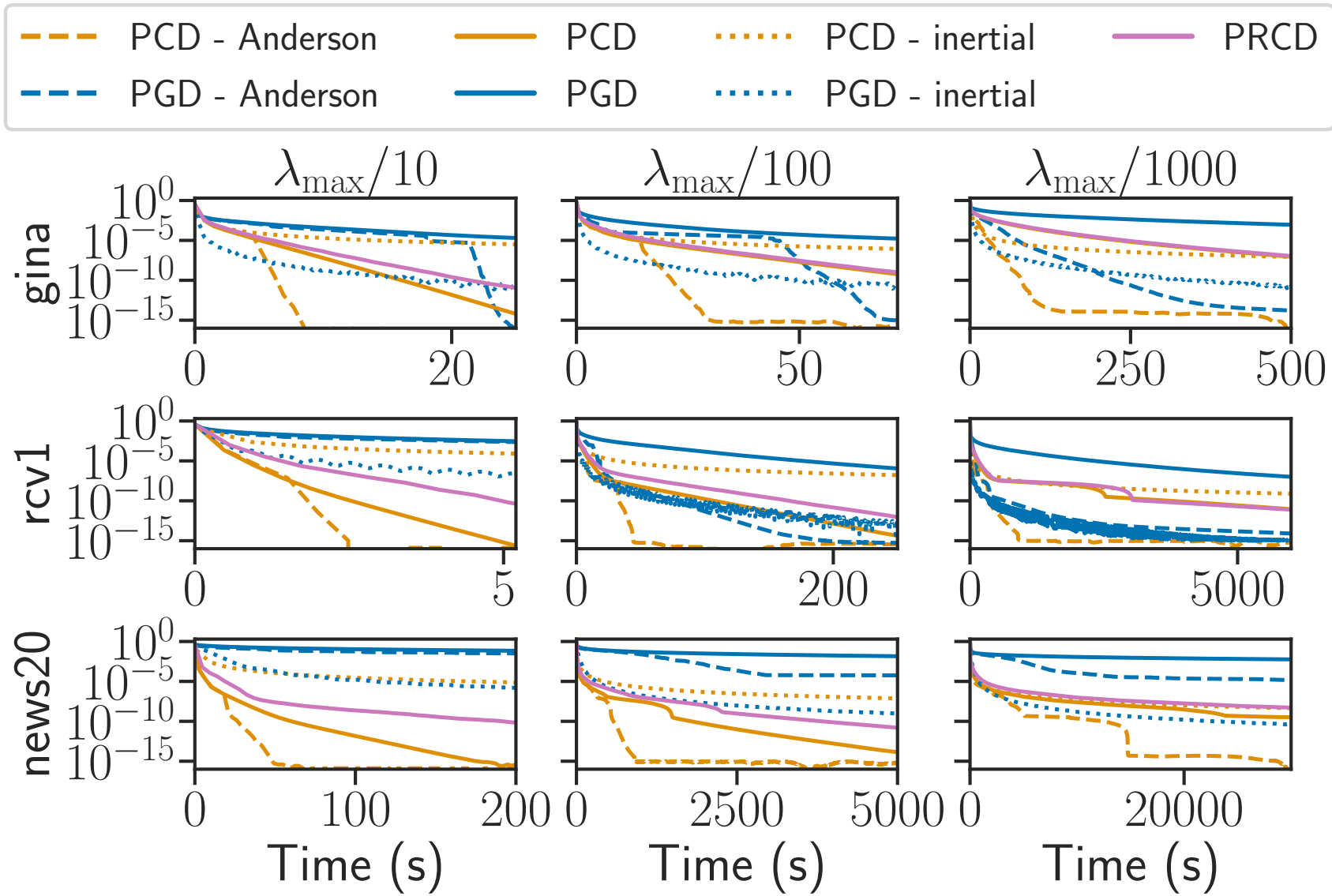
[a] Q. Bertrand and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.

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

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}
0.0
0.2
0
5
0
20
0.0
2.5
5.0
0
5
10
0
10
20
0.0
0.5
1.0
0
5
0
200
Time (s)
Time (s)
Time (s)

Sparse logistic regression



Conclusion and latest work

- First practical accelerated coordinate descent
- Extension to proximal case by model identification
- AISTATS paper: http://proceedings.mlr.press/v130/bertrand21a/bertrand21a.pdf
- Open code: https://github.com/mathurinm/andersoncd

Latest work:

- Combination with working sets for more acceleration
- Handling of non-convex penalties

Bibliography I

- Anderson, D. G. "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560.
- Bertrand, Q. and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.
- Bollapragada, R., D. Scieur, and A. d'Aspremont. "Nonlinear acceleration of momentum and primal-dual algorithms". In: *arXiv preprint arXiv:1810.04539* (2018).
- Fercoq, O. and P. Richtárik. "Accelerated, parallel, and proximal coordinate descent". In: *SIAM Journal on Optimization* 25.4 (2015), pp. 1997–2023.
- Friedman, J. et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (2007), pp. 302–332.
- Iutzeler, F. and J. Malick. "Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications". In: *Set-Valued and Variational Analysis* 28.4 (2020), pp. 661–678.

Bibliography II

- Klopfenstein, Q. et al. "Model identification and local linear convergence of coordinate descent". In: *arXiv preprint arXiv:2010.11825* (2020).
- Lin, Q., Z. Lu, and L. Xiao. "An Accelerated Proximal Coordinate Gradient Method". In: *NeurIPS*. 2014, pp. 3059–3067.
- Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python". In: *JMLR* 12 (2011), pp. 2825–2830.
- Scieur, D. "Generalized Framework for Nonlinear Acceleration". In: *arXiv preprint arXiv:1903.08764* (2019).