Dual Extrapolation for Sparse Generalized Linear Models

joint work with

Alexandre GRAMFORT

Mathurin MASSIAS

Joseph SALMON

Samuel VAITER, CNRS & IMB

July 8 2020 @ (virtual) SIAM IS

Dual Extrapolation for Sparse Generalized Linear Models

observations $\rightarrow y \in \mathbb{R}^n$

design matrix $\rightarrow X = \left[\ X_1 \mid \cdots \mid X_p\ \right] \in \mathbb{R}^{n \times p}$

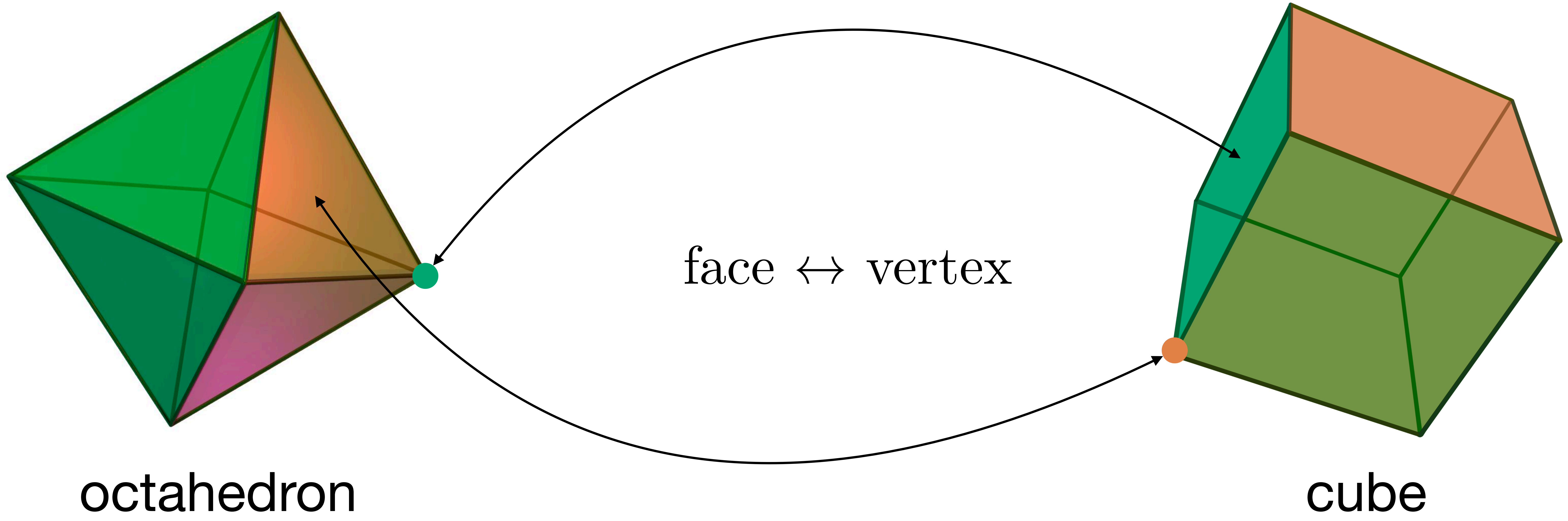
Lasso (Donoho '95, Tibshirani '96)

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$

Sparse logistic regression (Koh et al. '07)

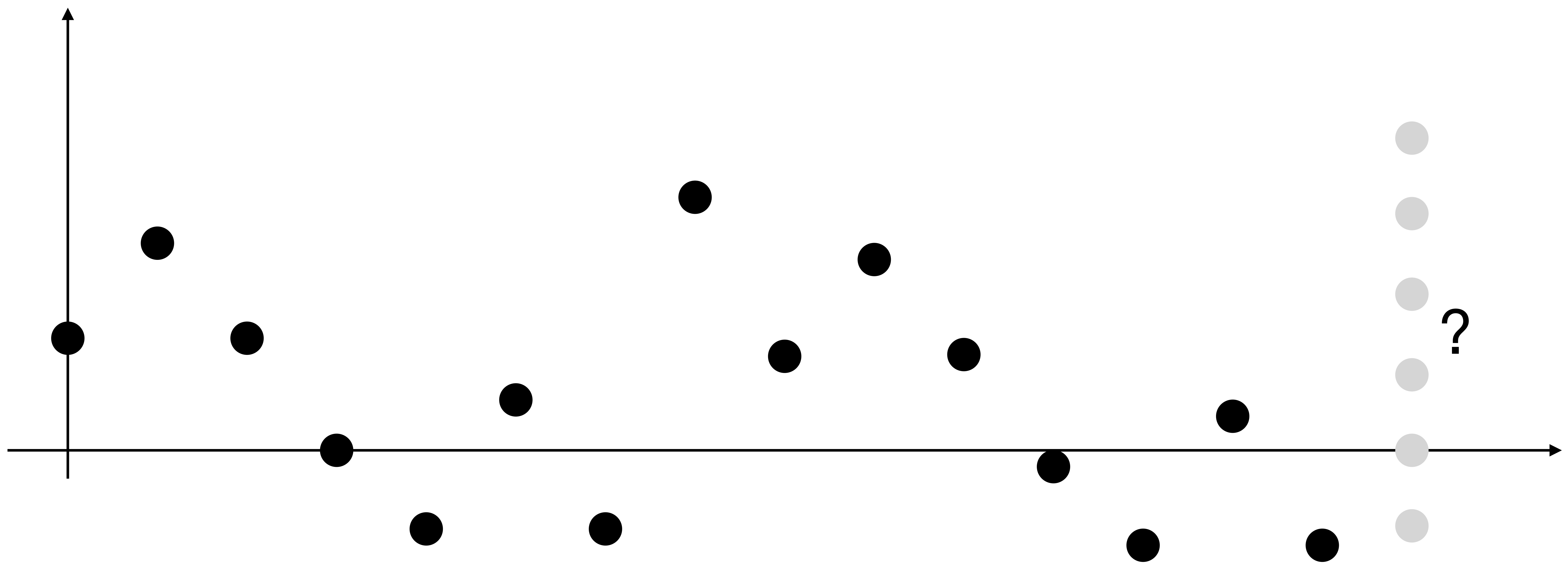
$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \log(1 + \exp(-y_i \langle \beta, X_i \rangle)) + \lambda \|\beta\|_1$$

Dual Extrapolation for Sparse Generalized Linear Models



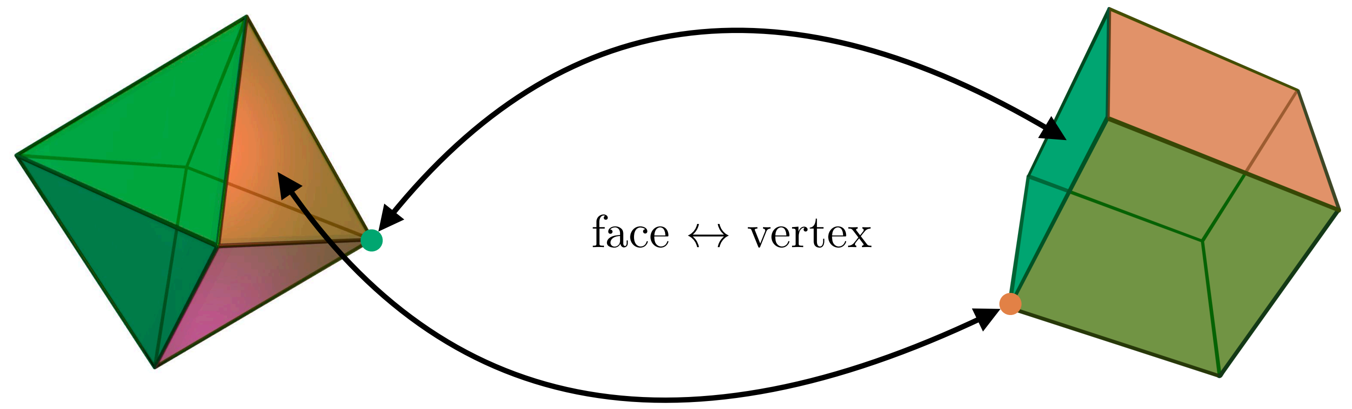
(src: wikipedia)

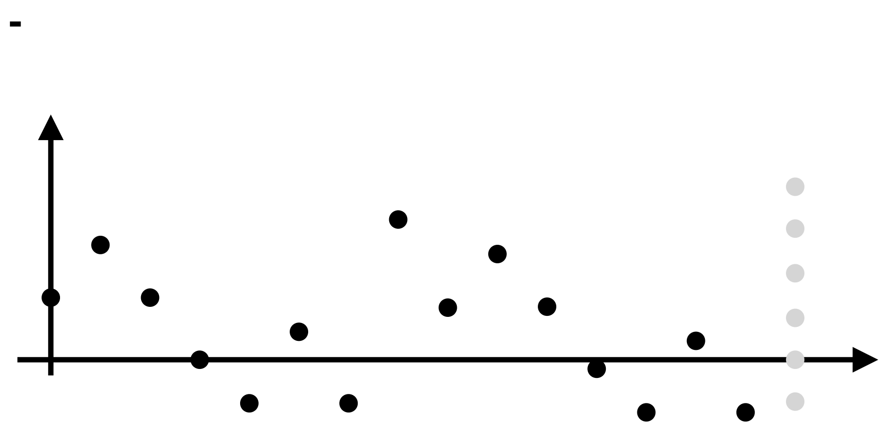
Dual Extrapolation for Sparse Generalized Linear Models



Dual Extrapolation for Sparse Generalized Linear Models

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$

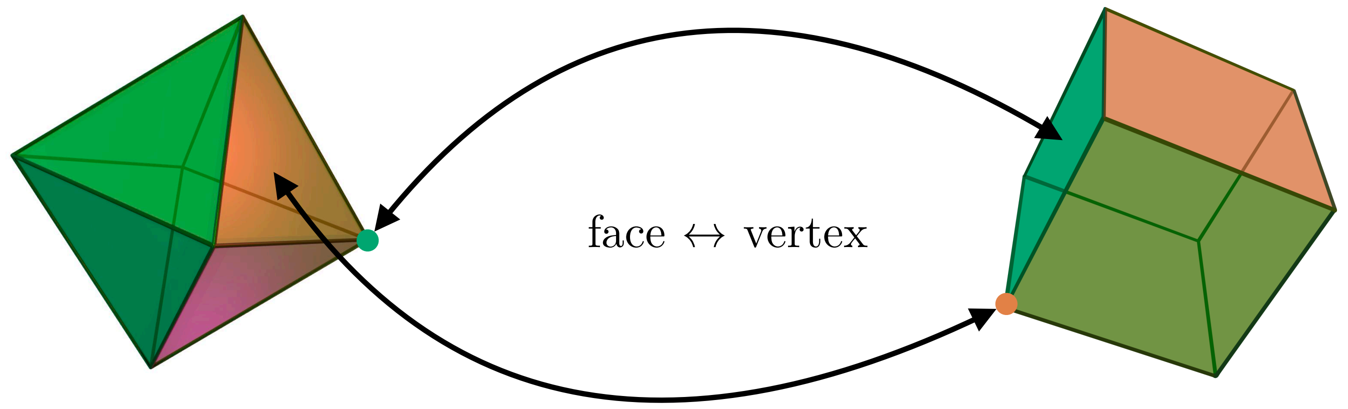


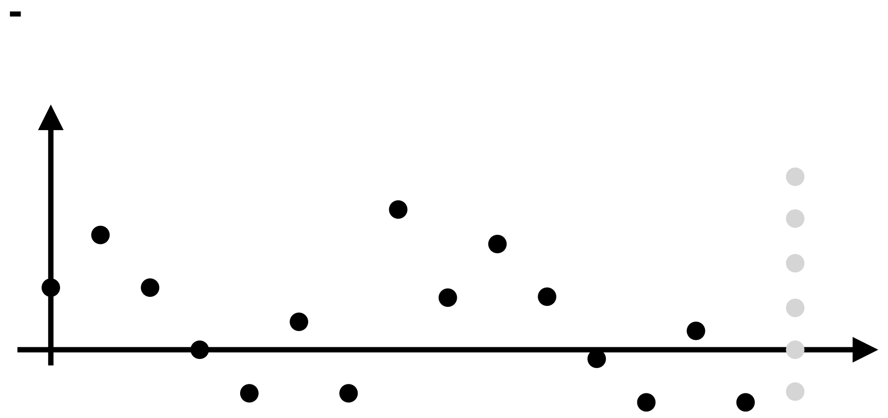


1. Why?
2. How?
3. Performance?

Dual Extrapolation for Sparse Generalized Linear Models

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$





1. Why?
2. How?
3. Performance?

A typical Lasso solver

Iterative Shrinkage-Thresholding Algorithm

```
for _ in range(n_epoch):
    primal = soft_thresholding(primal - 1/L * grad(primal))
```

A typical Lasso solver

Iterative Shrinkage-Thresholding Algorithm

```
for _ in range(n_epoch):
    primal = soft_thresholding(primal - 1/L * grad(primal))
```

Goal: Choose n_epoch such that

- primal close to the solution $\hat{\beta}$
- does not take too much time (how to select n_epoch?)

hard to have guarantees!

Duality for the Lasso

Primal problem

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1}_{\overset{\text{def.}}{=} \mathcal{P}(\beta)}$$

Dual problem

$$\hat{\theta} = \underset{\theta \in \Delta_X}{\operatorname{argmax}} \underbrace{\frac{1}{2}\|y\|_2^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|_2^2}_{\overset{\text{def.}}{=} \mathcal{D}(\theta)}$$

dual feasible set

$$\Delta_X = \left\{\theta \in \mathbb{R}^n \ : \ \forall 1 \leqslant j \leqslant p, |X_j^\top \theta| \leqslant 1\right\}$$

link equation

$$\hat{\theta} = \lambda^{-1}(y - X\hat{\beta})$$

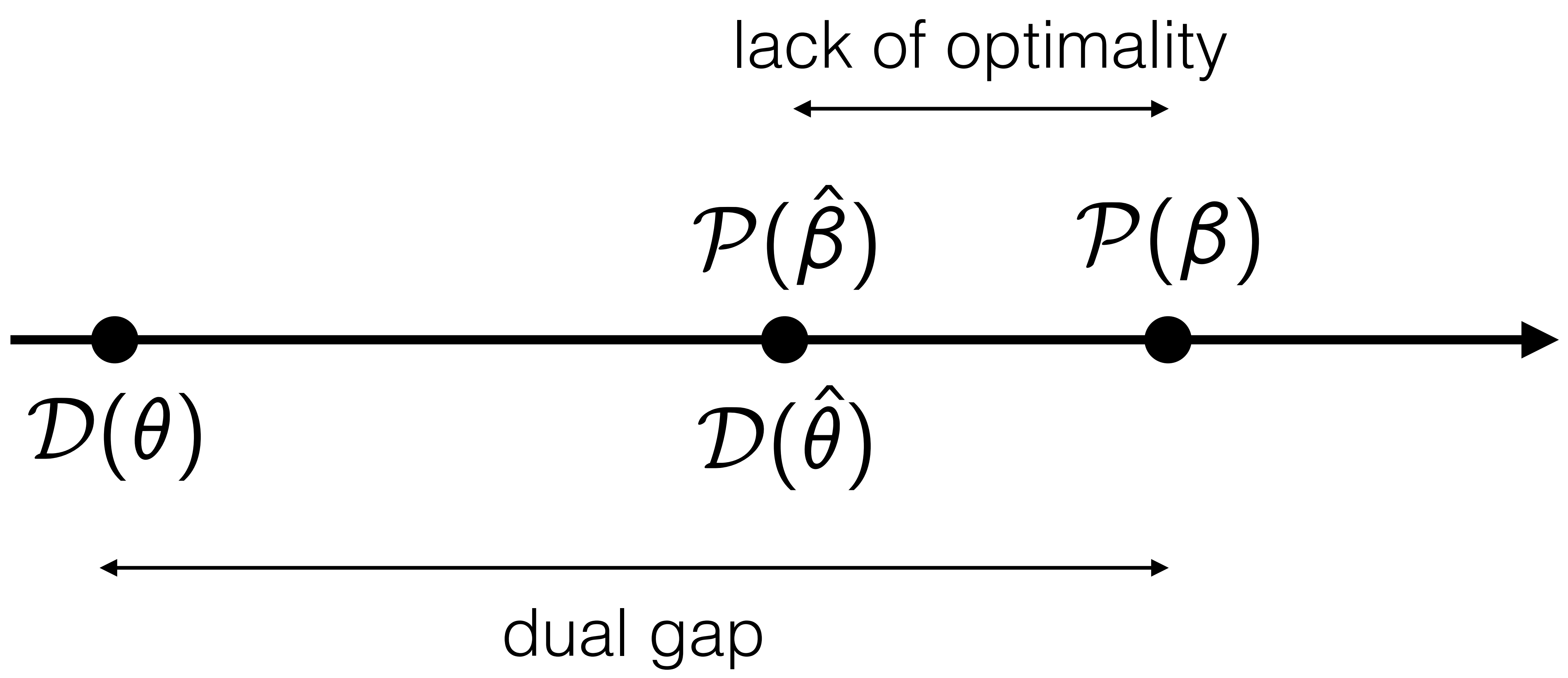
Consequence of strong duality

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1}_{\stackrel{\text{def.}}{=} \mathcal{P}(\beta)}$$

$$\mathcal{P}(\hat{\beta}) = \mathcal{D}(\hat{\theta})$$

$$\hat{\theta} = \lambda^{-1}(y - X\hat{\beta})$$

$$\hat{\theta} = \underset{\theta \in \Delta_X}{\operatorname{argmax}} \underbrace{\frac{1}{2}\|y\|_2^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|_2^2}_{\stackrel{\text{def.}}{=} \mathcal{D}(\theta)}$$



A typical Lasso solver — slightly modified

Iterative Shrinkage-Thresholding Algorithm

```
while dual_gap(primal, dual) > tol:
    primal = ST(primal - 1/L * grad(primal))

    dual = ????
```

$$\hat{\theta} = \lambda^{-1}(y - X\hat{\beta}) \longrightarrow \theta^{(t)} = \lambda^{-1}(y - X\beta^{(t)}) \in \Delta_X \, ?$$

A typical Lasso solver — slightly modified

Iterative Shrinkage-Thresholding Algorithm

```
while dual_gap(primal, dual) > tol:
    primal = ST(primal - 1/L * grad(primal))
    residual = y - X @ primal
    dual = residual / max(lam, norm(X.T @ residual, Inf)
```

$$\hat{\theta} = \lambda^{-1}(y - X\hat{\beta}) \longrightarrow \theta^{(t)} = \lambda^{-1}(y - X\beta^{(t)}) \in \Delta_X \; ?$$

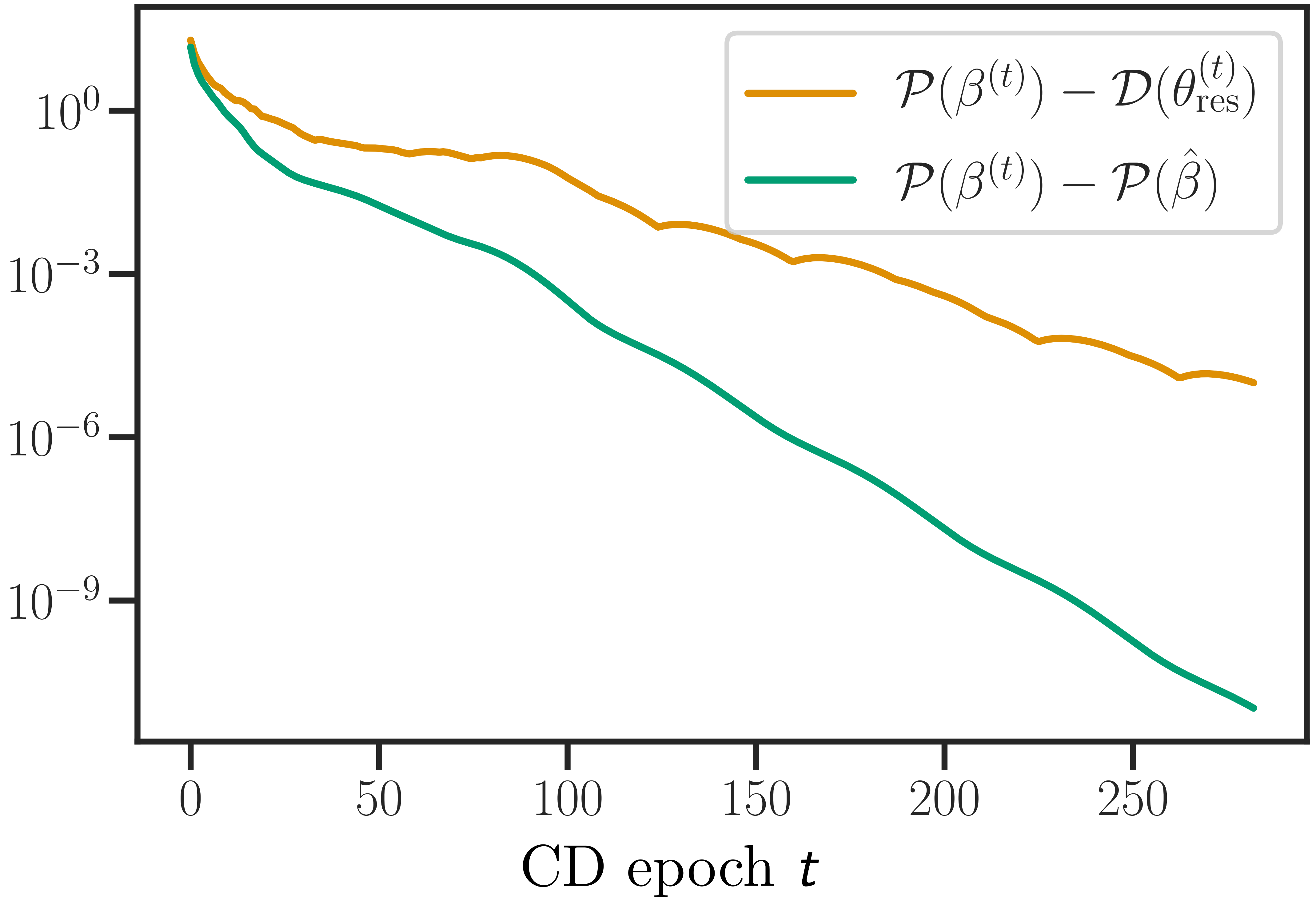
$\downarrow$ (Mairal, 2010)

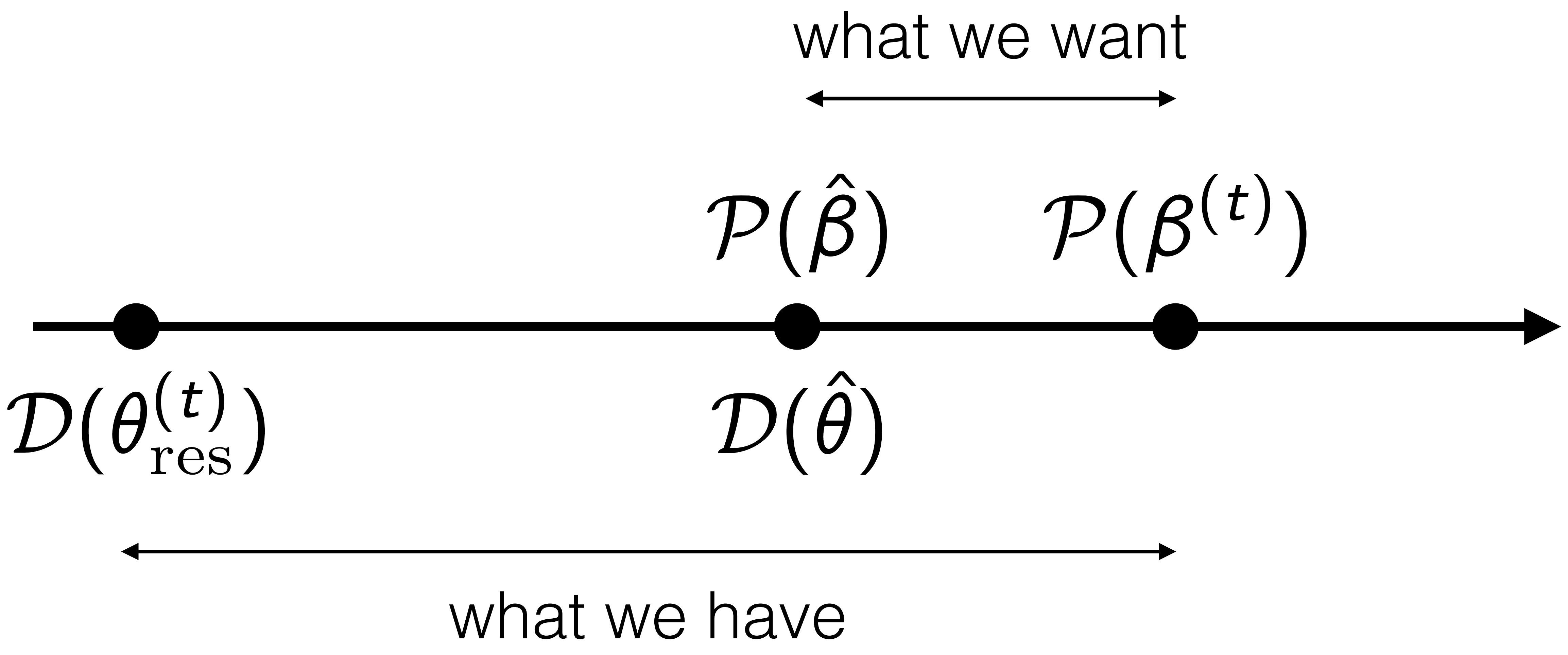
$$\theta^{(t)}_{\mathrm{res}} = r^{(t)} / \max(\lambda, \|X^\top r^{(t)}\|_\infty)$$

residual

$$r^{(t)} = y - X\beta^{(t)}$$

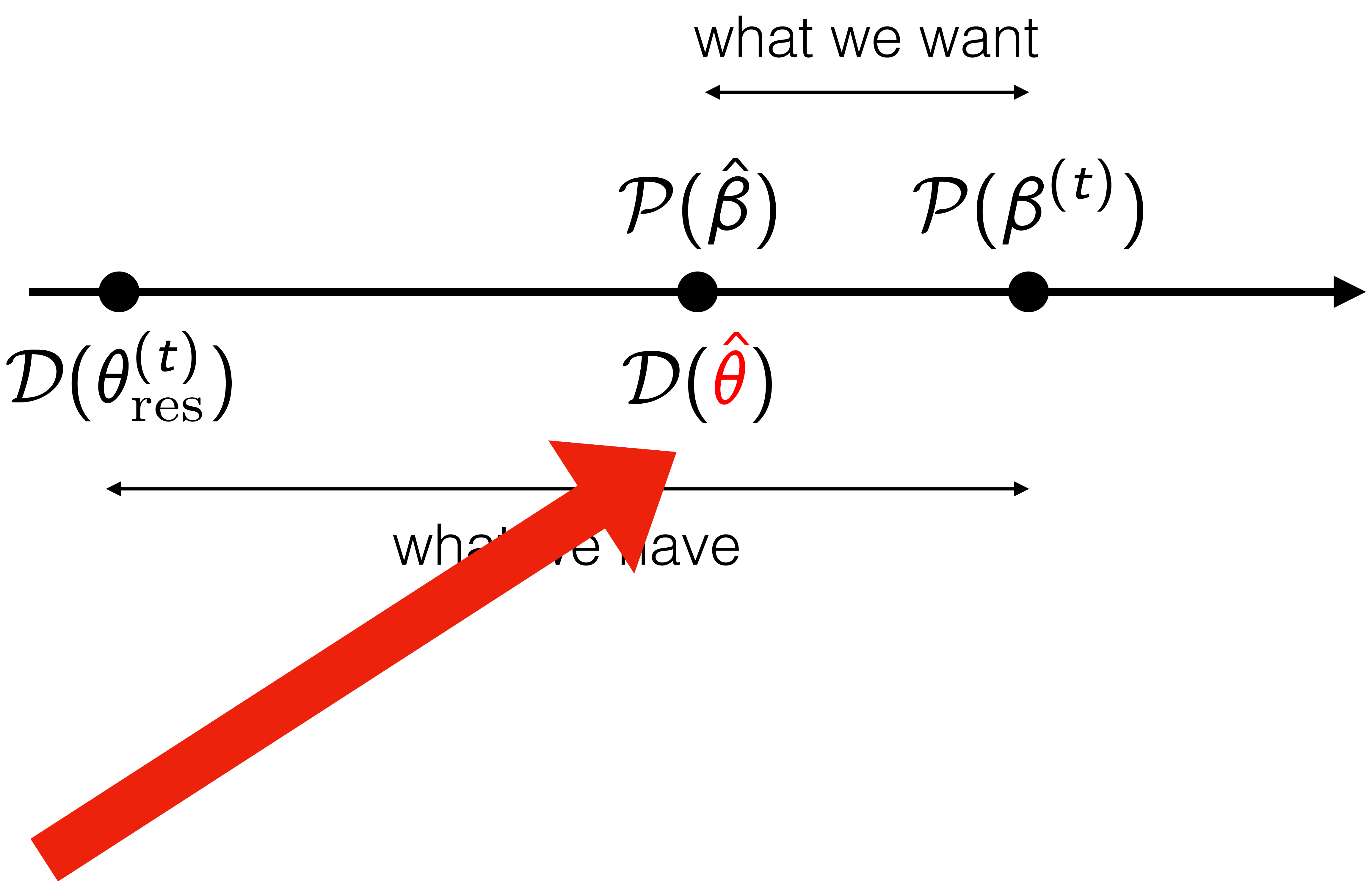
Dual gap is (way) slower than lack of optim.





Leukemia dataset: $p = 7129$, $n = 72$, $\lambda = \lambda_{\max}/10$

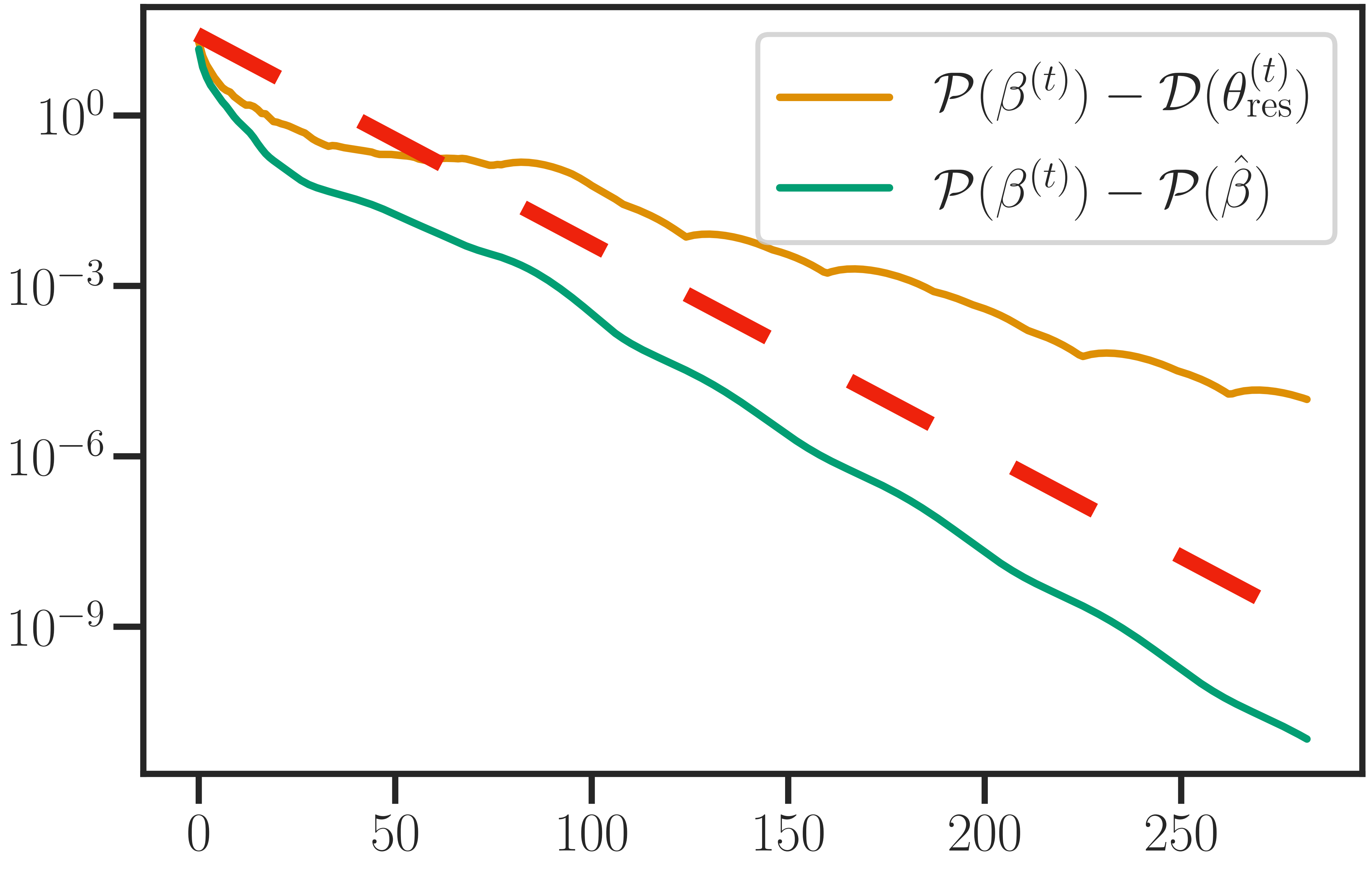
Our goal



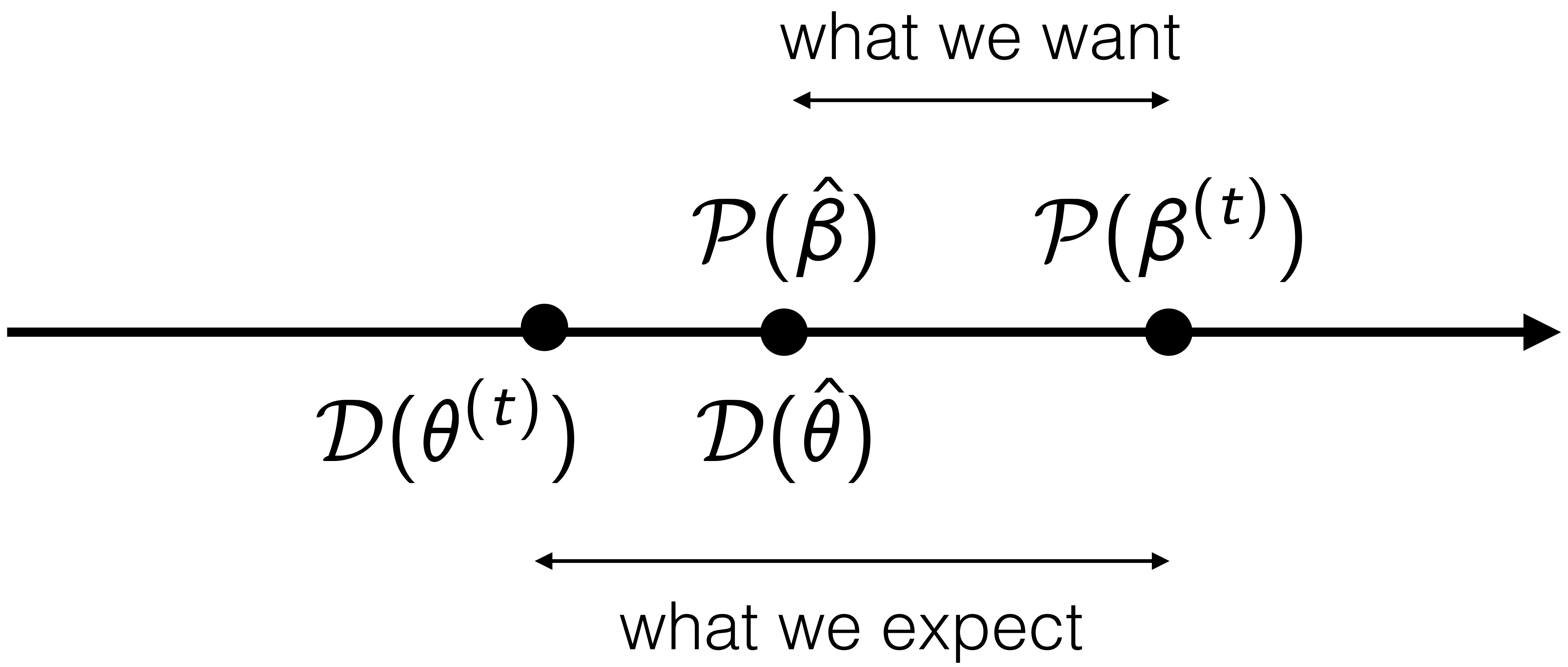
Find a good dual candidate

$$\theta \approx \hat{\theta}$$

Our goal



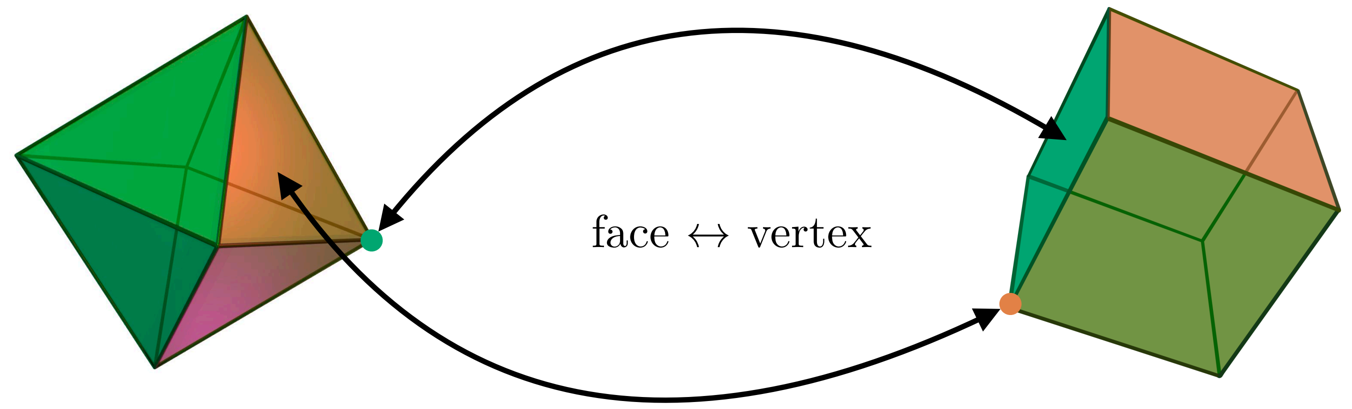
$\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta_{\text{res}}^{(t)})$
$\mathcal{P}(\beta^{(t)}) - \mathcal{P}(\hat{\beta})$
10^{0}
10^{-3}
10^{-6}
10^{-9}
0
50
100
150
200
250

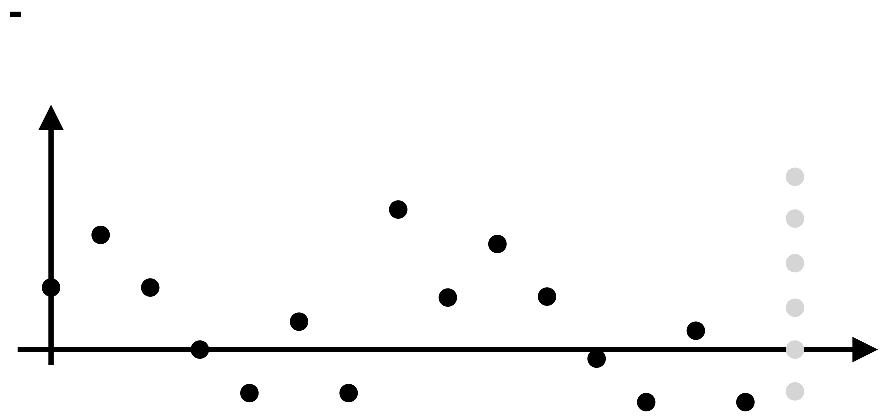


what we want
$\mathcal{P}(\hat{\beta})$
$\mathcal{P}(\beta^{(t)})$
$\mathcal{D}(\theta^{(t)})$
$\mathcal{D}(\hat{\theta})$
what we expect

Dual Extrapolation for Sparse Generalized Linear Models

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$



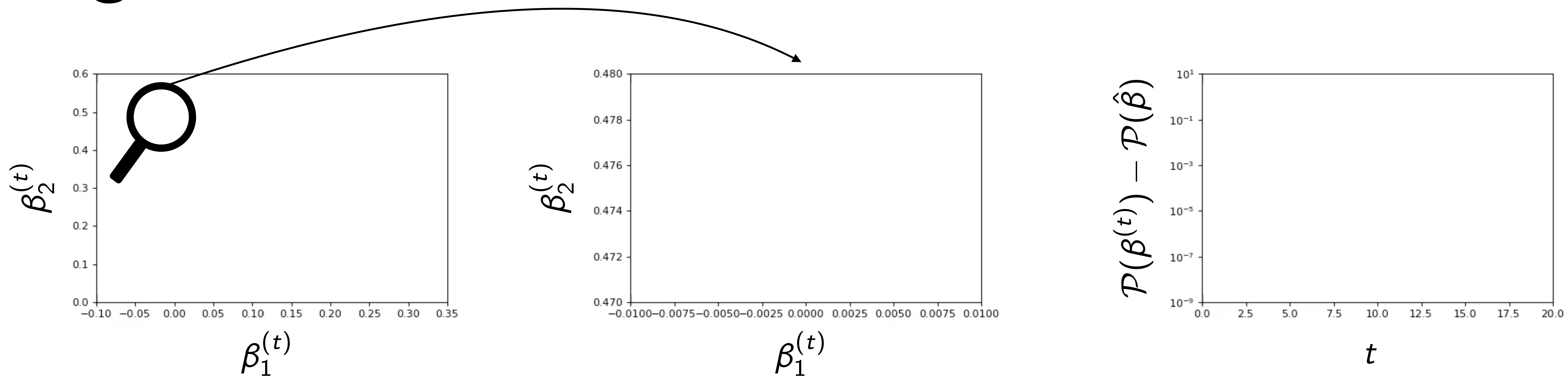


1. Why?
2. How?
3. Performance?

iterate: $\beta^{(t)}$

residual: $r^{(t)} = y - X\beta^{(t)}$

Sign identification



Vector AutoRegressive sequence (VAR)

$$x^{(t)} = Ax^{(t-1)} + b \in \mathbb{R}^n$$

Theorem

$\exists T :$ $\forall t \geqslant T : \mathrm{sign}(\beta^{(t)}) = \mathrm{sign}(\hat{\beta})$
$(r^{(t)})_{t \geqslant T}$ is a VAR

🧐 Fit a VAR to infer $\lim r^{(t)} = \lambda\hat{\theta}$

😢 *When* sign identified?

😭 high dimensional fit

see https://francisbach.com/acceleration-without-pain/

Extrapolation in 1D: Aitken Δ^2 method

$$x^{(t)} = a x^{(t-1)} + b \xrightarrow{t \to \infty} \hat{x}$$

XXV.—On Bernoulli's Numerical Solution of Algebraic Equations.
By A. C. Aitken, D.Sc.

(MS. received April 21, 1926. Read May 24, 1926.)

§ 1. Introductory.

The aim of the present paper is to extend Daniel Bernoulli's method* of approximating to the numerically greatest root of an algebraic equation. On the basis of the extension here given it now becomes possible to make Bernoulli's method a means of evaluating not merely the greatest root, but all the roots of an equation, whether real, complex, or repeated, by an

(Aitken, Proc. R. Soc. Edinb., 1927)

Aitken Δ^2 method

2 equations w/ 2 unknowns:

$$x^{(t)} - \hat{x} = a\left(x^{(t-1)} - \hat{x}\right)$$

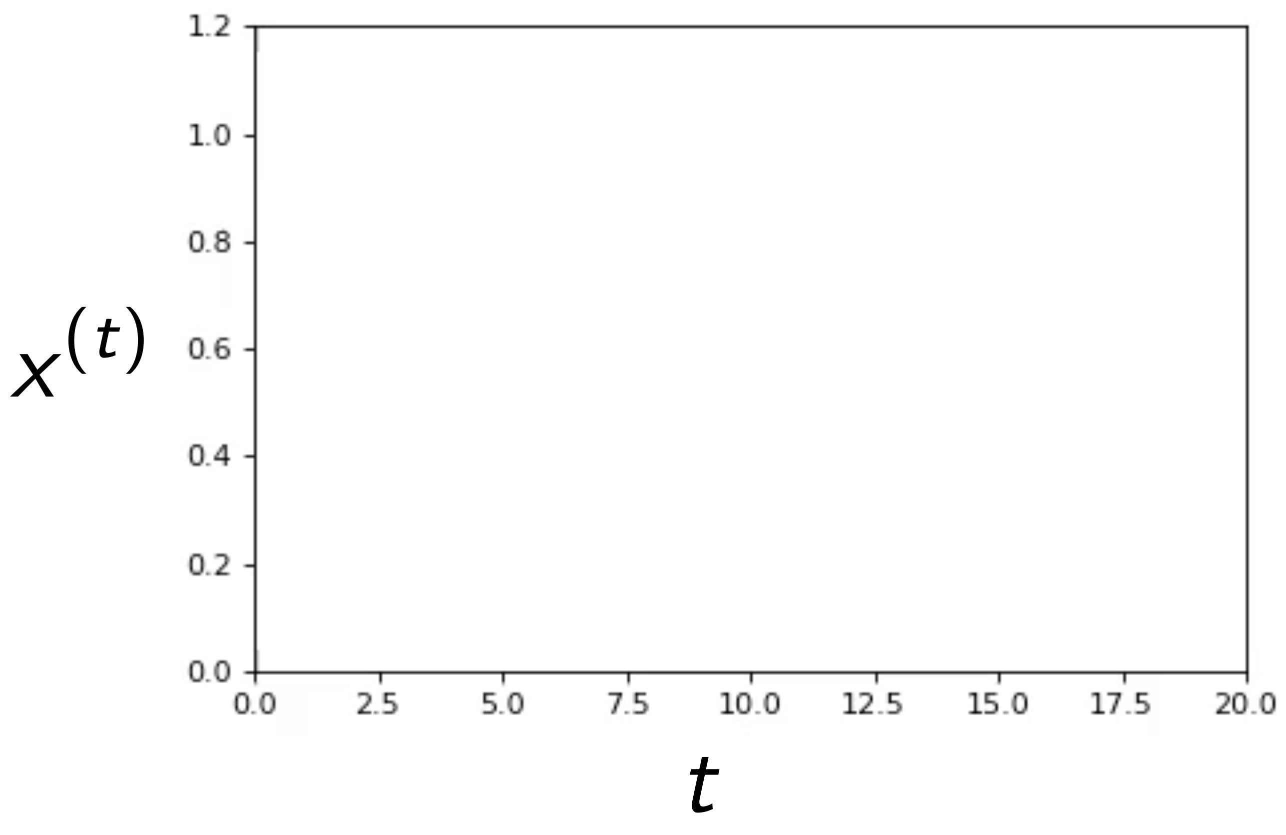
$$x^{(t-1)} - \hat{x} = a\left(x^{(t-2)} - \hat{x}\right)$$

need 3 iterates to extrapolate

work also when $x^{(t)}$ asymptotic VAR

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

k	$x^{(t)}$	$\hat{x}^{(t)}$
1	4.0000	×
2	2.6667	×
3	**3**.4667	**3.1**667
4	2.8952	**3.1**333
5	**3**.3397	**3.14**52
6	2.9760	**3.1**397
7	**3**.2837	**3.14**27
8	**3**.0171	**3.14**09
9	**3**.2524	**3.14**21
10	**3**.0418	**3.141**3



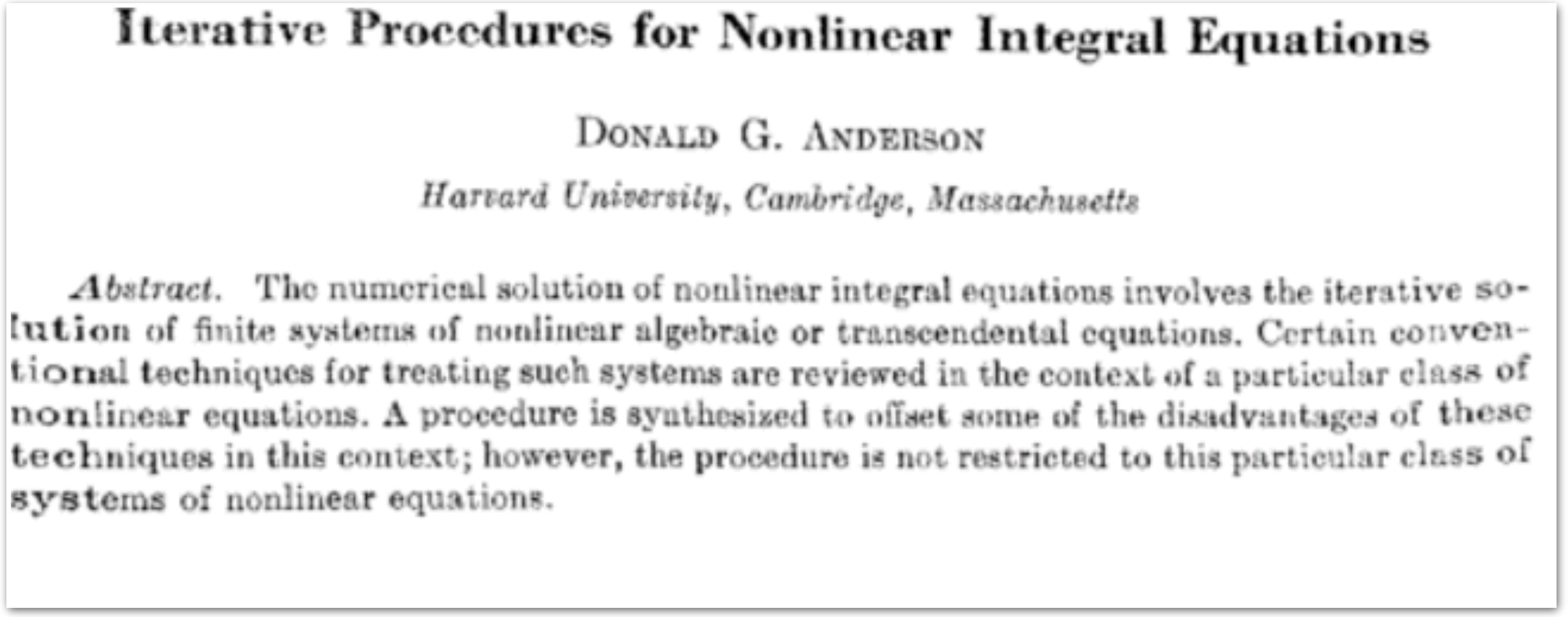
Extrapolation in higher dimension

$$x^{(t)} = Ax^{(t-1)} + b \xrightarrow{t\to\infty} \hat{x}$$

$\rightarrow$ needs $n+1$ equations

$\rightarrow$ needs $n+2$ iterates

😢 But n is large



Iterative Procedures for Nonlinear Integral Equations

Donald G. Anderson

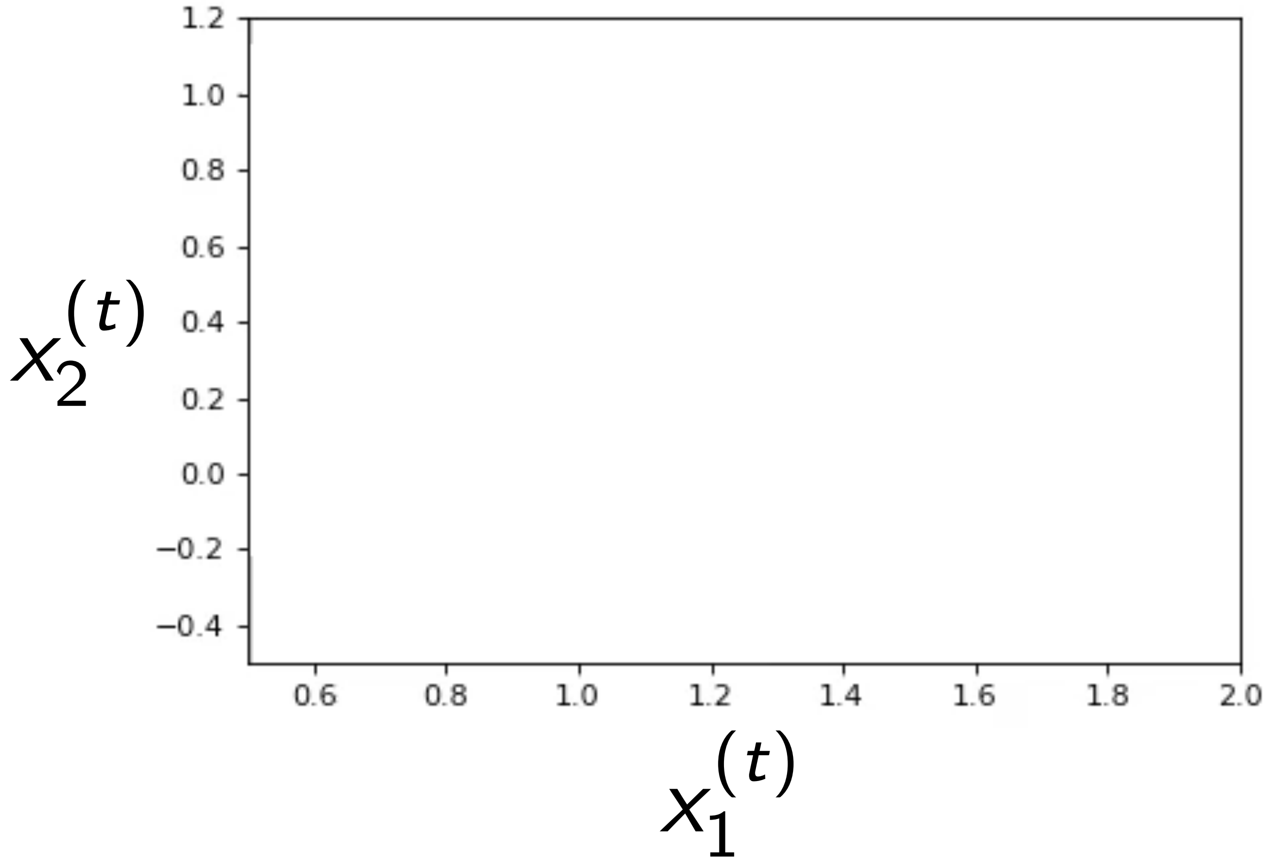
Harvard University, Cambridge, Massachusetts

Abstract. The numerical solution of nonlinear integral equations involves the iterative solution of finite systems of nonlinear algebraic or transcendental equations. Certain conventional techniques for treating such systems are reviewed in the context of a particular class of nonlinear equations. A procedure is synthesized to offset some of the disadvantages of these techniques in this context; however, the procedure is not restricted to this particular class of systems of nonlinear equations.

(Anderson, J. ACM., 1965)

Anderson idea: choose a budget K of previous iterates

$$\hat{x}^{(t)} = c_0 x^{(t)} + c_1 x^{(t-1)} + \cdots + c_{K-1}^{(t+1-K)}$$



see https://francisbach.com/acceleration-without-pain/

Extrapolation in higher dimension

$$x^{(t)} = Ax^{(t-1)} + b \quad \xrightarrow{t\to\infty} \quad \hat{x}$$

$\rightarrow$ needs $n+1$ equations

$\rightarrow$ needs $n+2$ iterates

😢 But n is large

Anderson idea: choose a budget K of previous iterates

$$\hat{x}^{(t)} = c_0 x^{(t)} + c_1 x^{(t-1)} + \cdots + c_{K-1}^{(t+1-K)}$$

with

$$c = \operatorname*{argmin}_{\sum c_i = 1} \| c_0 \Delta x^{(t)} + \cdots + c_{K-1} \Delta x^{(t-K+1)} \|_2^2$$

$$\Delta x^{(t)} = x^{(t)} - x^{(t-1)}$$

"consecutive iterates lead to close extrapolation"

not a convexity constraint

Iterative Procedures for Nonlinear Integral Equations

Donald G. Anderson

Harvard University, Cambridge, Massachusetts

Abstract. The numerical solution of nonlinear integral equations involves the iterative solution of finite systems of nonlinear algebraic or transcendental equations. Certain conventional techniques for treating such systems are reviewed in the context of a particular class of nonlinear equations. A procedure is synthesized to offset some of the disadvantages of these techniques in this context; however, the procedure is not restricted to this particular class of systems of nonlinear equations.

(Anderson, J. ACM., 1965)

Regularized Nonlinear Acceleration

Damien Scieur
INRIA & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
damien.scieur@inria.fr

Alexandre d'Aspremont
CNRS & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
aspremon@di.ens.fr

Francis Bach
INRIA & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
francis.bach@inria.fr

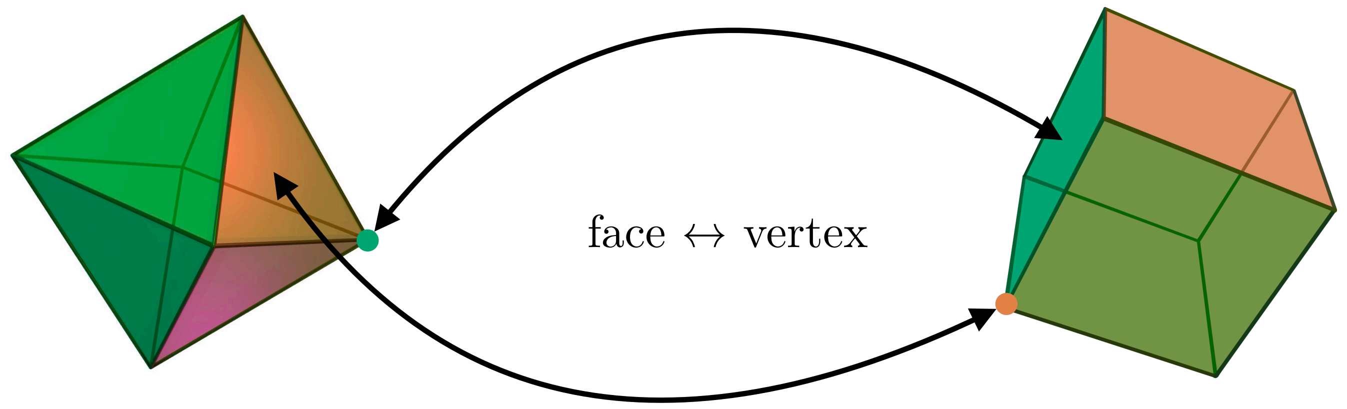
Abstract

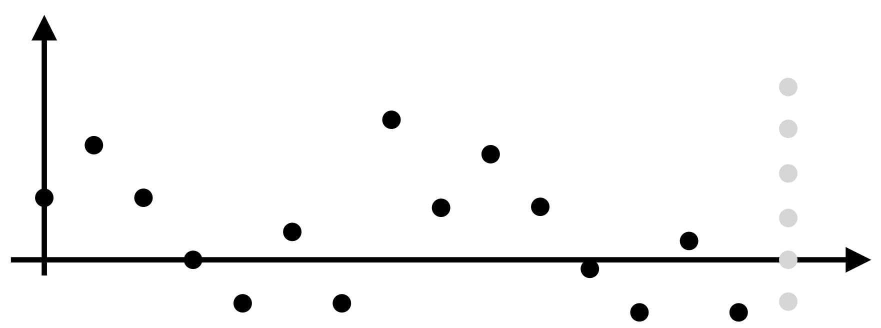
We describe a convergence acceleration technique for generic optimization problems. Our scheme computes estimates of the optimum from a nonlinear average of the iterates produced by any optimization method. The weights in this average are computed via a simple and small linear system, whose solution can be updated online. This acceleration scheme runs in parallel to the base algorithm, providing improved estimates of the solution on the fly, while the original optimization method is running. Numerical experiments are detailed on classical classification problems.

(Scieur, d'Aspremont, Bach, NeurIPS, 2016)

Dual Extrapolation for Sparse Generalized Linear Models

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n}\|y - X\beta\|^2 + \lambda\|\beta\|_1$$





1. Why?
2. How?
3. Performance?

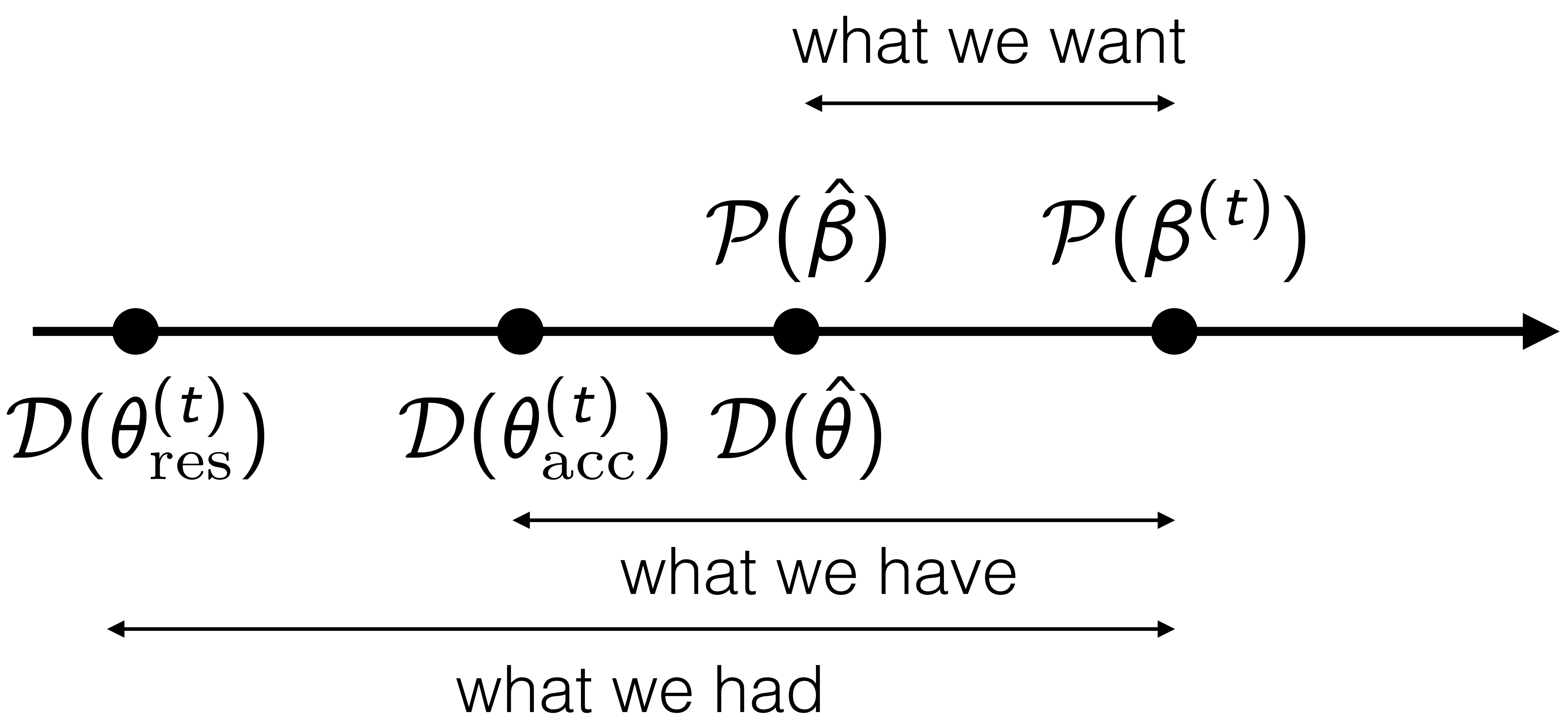
Extrapolation for the Lasso

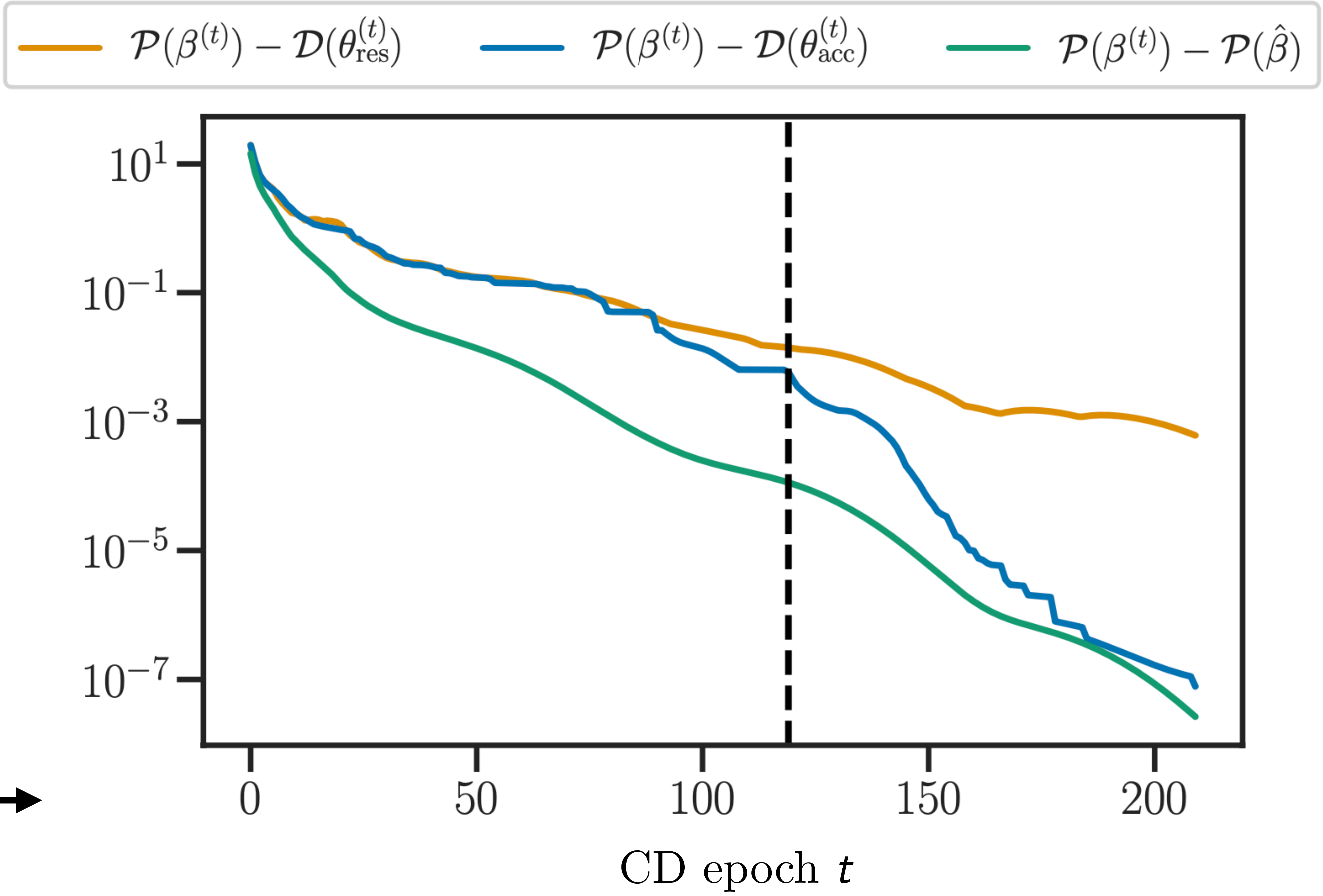
Extrapolated residuals

$$r_{\mathrm{acc}}^{(t)} = c_0 r^{(t)} + c_1 r^{(t-1)} + \cdots + r_{K-1}^{(t+1-K)}$$

Extrapolated dual point

$$\theta_{\mathrm{acc}}^{(t)} = r_{\mathrm{acc}}^{(t)} / \max(\lambda, \|X^\top r_{\mathrm{acc}}^{(t)}\|_\infty)$$

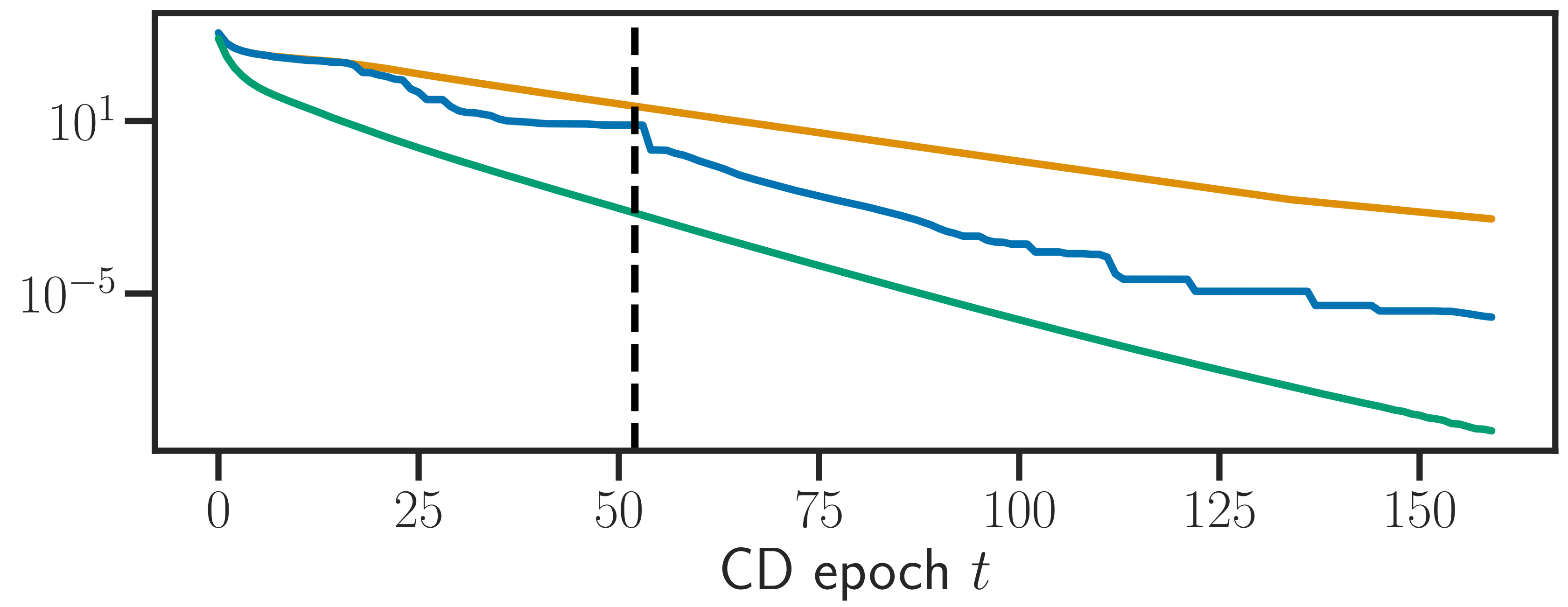




Leukemia dataset: $p = 7129$, $n = 72$, $\lambda = \lambda_{\max}/10$

Extrapolation for other models

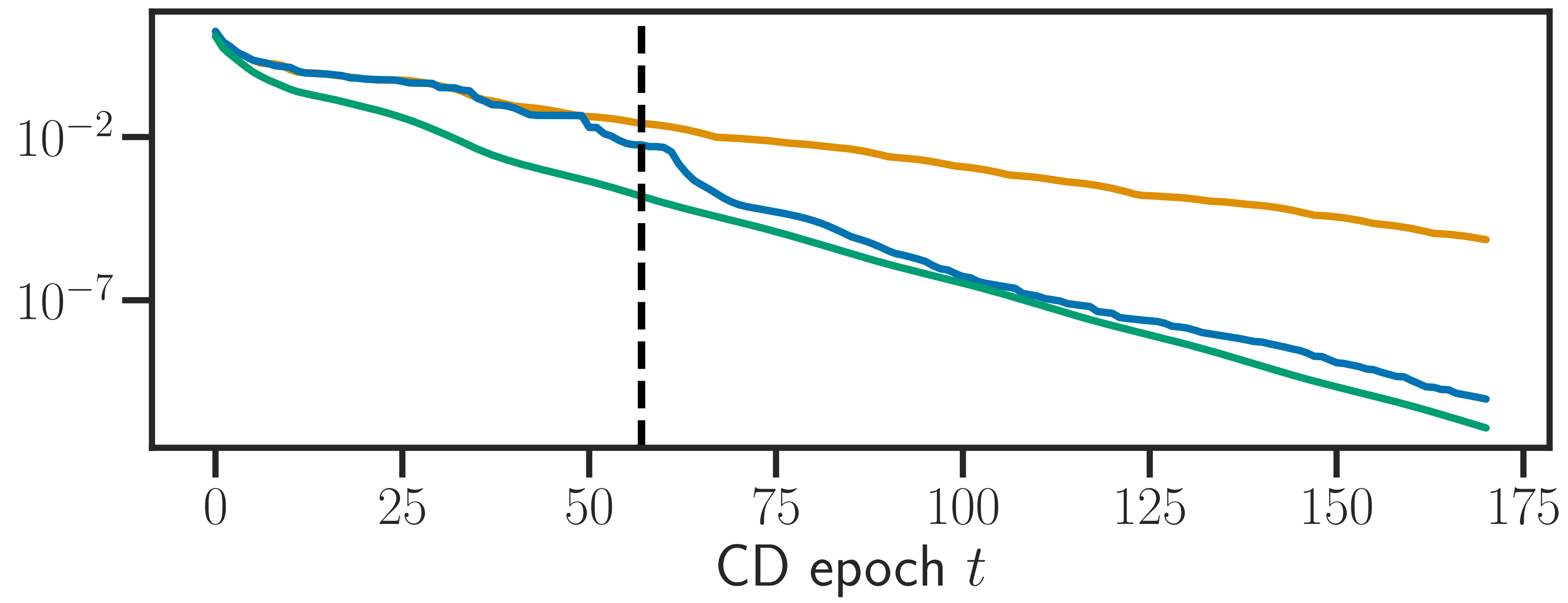
$\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta_{\mathrm{res}}^{(t)})$ $\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta_{\mathrm{acc}}^{(t)})$ $\mathcal{P}(\beta^{(t)}) - \mathcal{P}(\hat{\beta})$



sparse logistic regression, *rcv1* dataset

$p = 20k, \quad n = 20k$

$\lambda = \lambda_{\max}/20$



Multitask Lasso, MEG data

$p = 7498, \quad n = 305$

$\lambda = \lambda_{\max}/10$

celer: a dropin Lasso class for scikit-learn

~~**from** sklearn.linear_model **import** Lasso, LassoCV~~
from celer **import** Lasso, LassoCV

Implements **dual extrapolation** (this talk), gap safe screen and working sets strategy

Performance & implementation on imaging settings is an open question!

mathurinm/celer

arxiv:1907.05830

mathurinm.github.io/celer/

Thanks for your attention!