




RCD SCD RCDM Shotgun UCDC RCDC PCDM SDCA mSDCA ICD ASDCA RBCD ACDM Acc-Prox-SDCA SPCDM Hydra Nsync AsySCD RCM APPROX DisDCA I-Prox-SDCA Asy-SPCD DBCD Hydra^2 DBCD APCG SPDC CoCoA Quartz S2CD ALPHA SDNA CoCoA+ AdaSDCA dfSDCA

Optimization in Machine Learning

(A Basic Course)

Peter Richtárik

University of Edinburgh

Machine Learning Summer School, Toulouse, September 14-18, 2015

Outline

Lecture 1

1. ERM
2. Linear Systems

Lecture 2

3. Arbitrary Sampling
4. Acceleration

Lecture 3

5. ERM & An Efficient Dual Method
6. ERM & An Efficient Primal Method
7. Parallelization / Minibatching

Extra

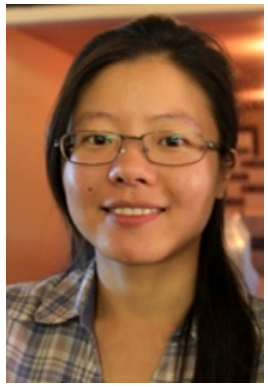
8. Distributed Optimization
9. Curvature

Coauthors

Martin Takáč
(Lehigh)

Jakub Mareček
(IBM)

Jakub Konečný
(Edinburgh)



Zheng Qu
(University of Hong Kong)

Olivier Fercoq
(Telecom ParisTech)

Rachael Tappenden
(Johns Hopkins)

Martin Jaggi
(ETH Zurich)

Tong Zhang
(Rutgers & Baidu)

Virginia Smith
(Berkeley)

Nati Srebro
(TTI Chicago)

Jie Liu
(Lehigh)

Michael Jordan
(Berkeley)

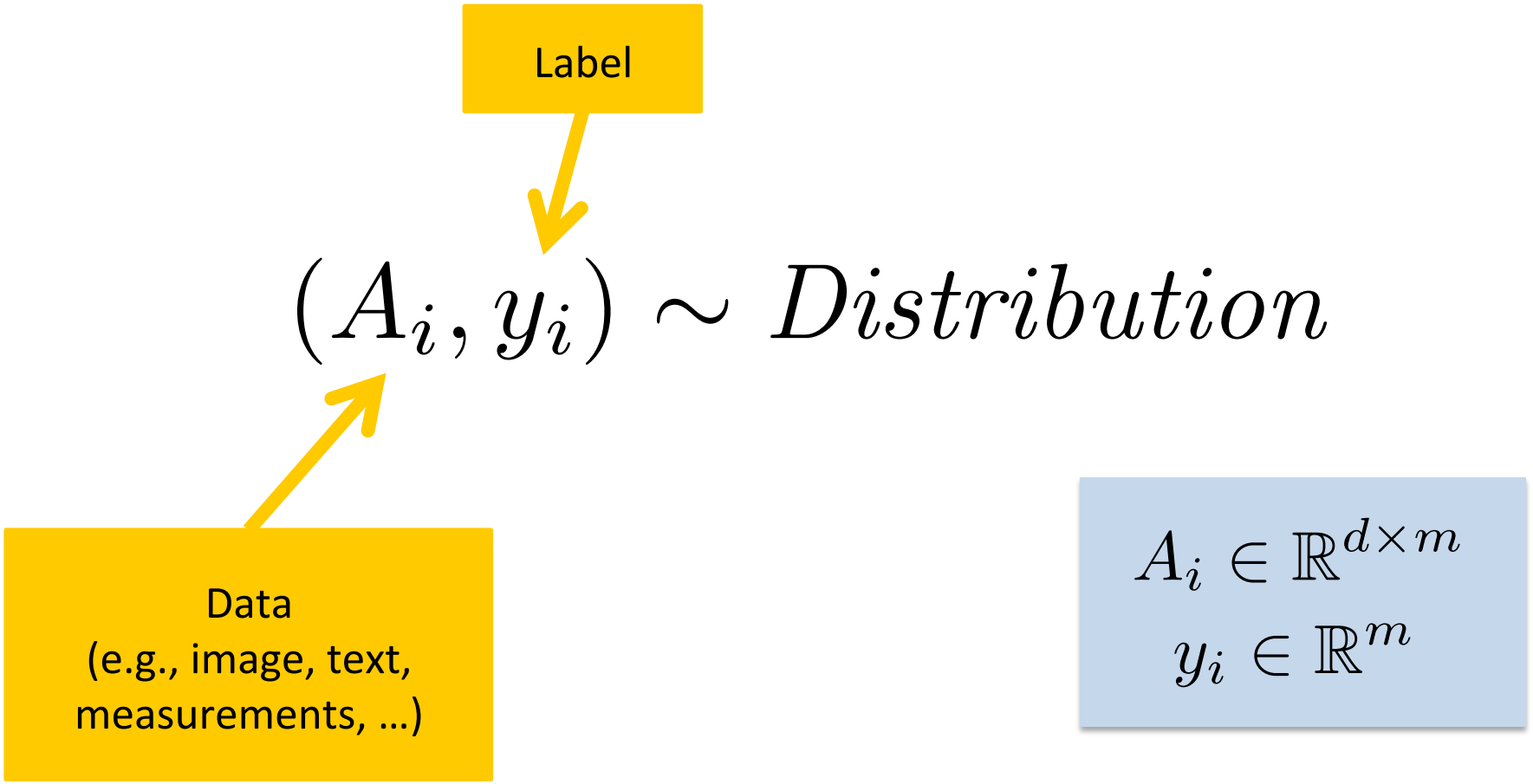
Dominik Csba
(Edinburgh)

Robert M Gower
(Edinburgh)

1. Empirical Risk Minimization

1.1
Training Linear Predictors

Statistical Nature of Data



Prediction of Labels from Data

Find $w \in \mathbb{R}^d$ ← Linear predictor

Such that when (data, label) pair is drawn from the distribution

$$(A_i, y_i) \sim Distribution$$

Then

$$A_i^\top w \approx y_i$$

Predicted label → $A_i^\top w$ ← True label

Measure of Success

$$loss(a, b)$$

We want the expected loss (=risk) to be small:

$$\mathbf{E}\left[loss(A_i^\top w, y_i)\right]$$

$$(A_i, y_i) \sim Distribution$$

Finding a Linear Predictor via Empirical Risk Minimization (ERM)

Draw i.i.d. data (samples) from the distribution

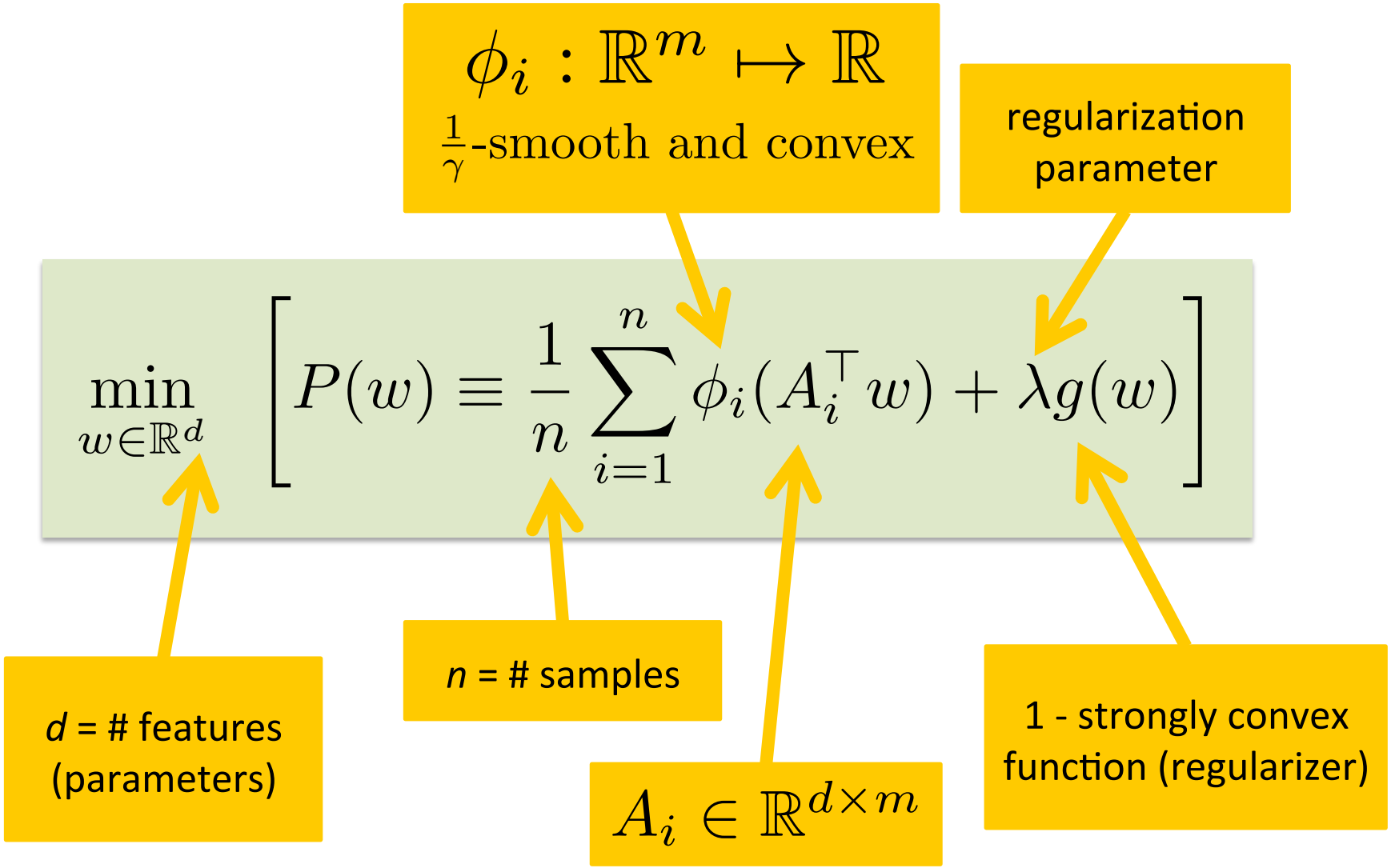
$$(A_1, y_1), (A_2, y_2), \ldots, (A_n, y_n) \sim Distribution$$

Output predictor which minimizes the empirical risk:

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} loss(A_i^\top w, y_i)$$

1.2
Primal and Dual Problems

Primal Problem: ERM



Is the difficulty in *n* or *d*?

- **Big n**
  - Work in the **primal**
  - Process one loss function (= one example) at a time
  - Type of methods: stochastic gradient descent (modern variants: SAG, SVRG, S2GD, mS2GD, SAGA, S2CD, MISO, FINITO, …)
- **Big d**
  - Work in the **primal**
  - Process one primal variable at a time
  - Type of methods: randomized coordinate descent (e.g., Hydra, Hydra2)
- **Big n**
  - Work in the **dual**
  - Process one dual variable (=one example) at a time
  - Type of methods: randomized coordinate descent (modern variants: RCDM, PCDM, Shotgun, SDCA, APPROX, Quartz, ALPHA, SDNA, SPDC, ASDCA, …)
  - E.g. SDCA = run coordinate descent on the dual problem

Dual Problem

$$D(\alpha) \equiv -\lambda g^* \left(\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i \right) - \frac{1}{n} \sum_{i=1}^{n} \phi_i^*(-\alpha_i)$$

$\in \mathbb{R}^m$ (over $-\alpha_i$)

$\in \mathbb{R}^d$ (over $\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i$)

1 – smooth & convex

γ - strongly convex

$$g^*(w') = \max_{w \in \mathbb{R}^d} \left\{ (w')^\top w - g(w) \right\}$$

$$\phi_i^*(a') = \max_{a \in \mathbb{R}^m} \left\{ (a')^\top a - \phi_i(a) \right\}$$

$$\max_{\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^N = \mathbb{R}^{nm}} D(\alpha)$$

$\alpha_1 \in \mathbb{R}^m$, $\alpha_n \in \mathbb{R}^m$

Warmup

2. Linear Systems

Robert Gower and P.R.
Randomized Iterative Methods for Linear Systems
arXiv:1506.03296, 2015
(submitted to SIAM Journal of Matrix Analysis and Applications)

2.1
The Problem

The Problem

$$\underbrace{\quad}_{n}$$

$$\in \mathbb{R}^n$$

$$\underbrace{\left[\begin{array}{c} \\ \\ \end{array}\right.}_{m} \quad Ax = b \quad \underbrace{\left.\begin{array}{c} \\ \\ \end{array}\right]}_{m}$$

Assumption: The system is consistent (i.e., has a solution)

We can also think of this as *m* linear equations, where the *i*th equation looks as follows:

$$\sum_{j=1}^{n} A_{ij} x_j = b_i$$

$$A_{i:} x = b_i$$

Minimizing Convex Quadratics

$$\min_{x \in \mathbb{R}^n} \left[f(x) = \frac{1}{2} \|Ax - b\|^2 \right] \quad \Rightarrow \quad \nabla f(x) = 0 \quad \Rightarrow \quad A^T A x = A^T b$$

This system is consistent

$$\min_{x \in \mathbb{R}^n} \left[f(x) = \tfrac{1}{2} x^T A x + b^T x + c \right] \quad \Rightarrow \quad \nabla f(x) = 0 \quad \Rightarrow \quad Ax = b$$

A = positive definite

This system is consistent

2.2
The Solution
(6 Ways to Skin the Cat)

TOP DEFINITION

skin the cat

Term refers to a task which has several ways by which it can be completed. Often used in the expression "there are many ways to skin the cat" or by using "skin this cat" in place of "skin the cat."

My friends and I are going to start a business, but we don't even know where to begin because there are so many ways to skin the cat.

by CRubio April 15, 2007

157 56

1. Relaxation Viewpoint
"Sketch and Project"

$$\langle x, y \rangle_B := x^T B y, \qquad \|x\|_B := \sqrt{\langle x, x \rangle_B}$$

B: Symmetric and positive definite

$$x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^t\|_B^2$$

$$\text{subject to} \quad S^T A x = S^T b$$

One Step Method: $S = m \times m$ invertible (with probability 1)

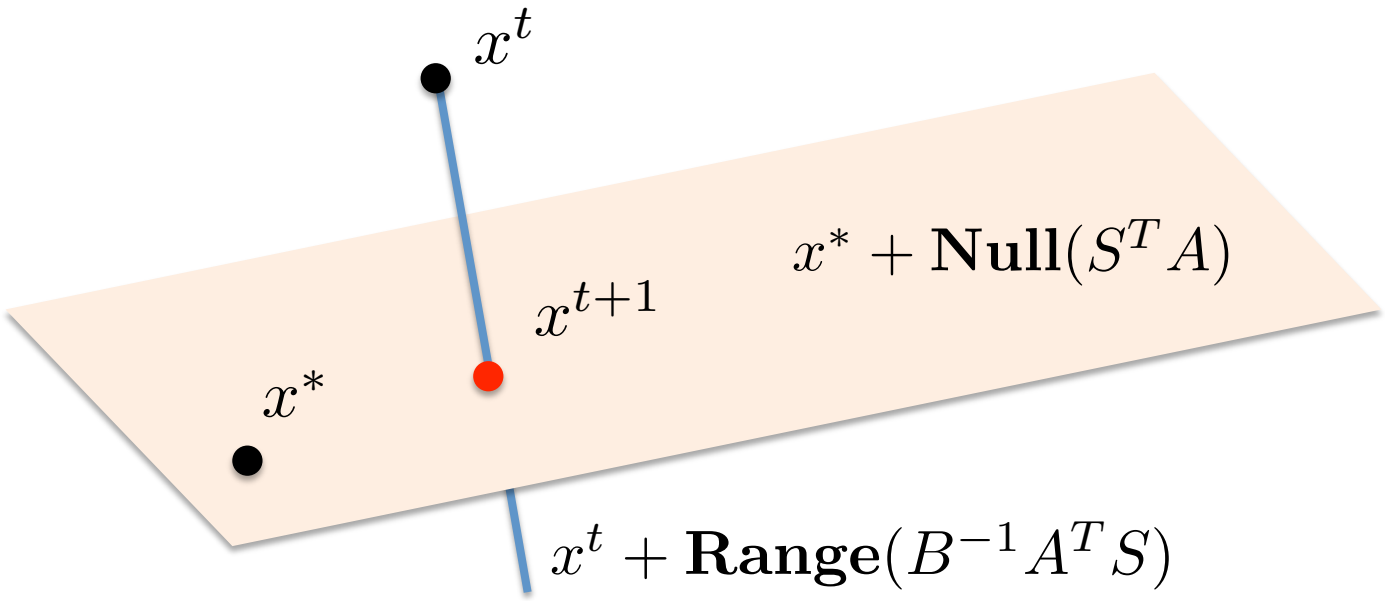
2. Optimization Viewpoint "Constrain and Approximate"

$$x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|_B^2$$

$$\text{subject to} \quad x = x^t + B^{-1}A^T S y$$

$$y \quad \text{is free}$$

3. Geometric Viewpoint
"Random Intersect"



Lemma $\mathbf{Null}(S^T A)$ and $\mathbf{Range}(B^{-1}A^T S)$ are B-orthogonal complements

Proof $h \in \mathbf{Null}(S^T A) \quad \Rightarrow \quad \langle B^{-1}A^T Sy, h\rangle_B = (y^T S^T AB^{-1})Bh = y^T S^T Ah = 0$

$$\{x^{t+1}\} \quad = \quad \left(x^* + \mathbf{Null}(S^T A)\right) \quad \bigcap \quad \left(x^t + \mathbf{Range}(B^{-1}A^T S)\right)$$

4. Algebraic Viewpoint
"Random Linear Solve"

x^{t+1} = solution in x of the linear system

$$S^T A x = S^T b$$

$$x = x^t + B^{-1} A^T S y$$

Unknown: *x*

Unknown: *y*

5. Algebraic Viewpoint "Random Update"

Random Update Vector

$$x^{t+1} = x^t - B^{-1}A^T S(S^T AB^{-1}A^T S)^\dagger S^T(Ax^t - b)$$

Moore-Penrose pseudo-inverse

Fact: Every (not necessarily square) real matrix M has a real pseudo-inverse $M^\dagger$.

Some properties:

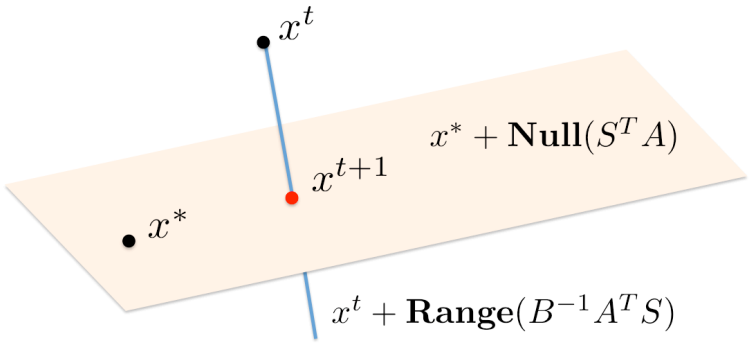
1. $MM^\dagger M = M$
2. $M^\dagger MM^\dagger = M^\dagger$
3. $(M^\top M)^\dagger M^\top = M^\dagger$
4. $(M^\top)^\dagger = (M^\dagger)^\top$
5. $(MM^\top)^\dagger = (M^\dagger)^\top M^\dagger$

6. Analytic Viewpoint
"Random Fixed Point"

$$Z := A^T S (S^T A B^{-1} A^T S)^\dagger S^T A$$

$$x^{t+1} - x^* = (I - B^{-1} Z)(x^t - x^*)$$

Random Iteration Matrix



$$(B^{-1}Z)^2 = B^{-1}Z$$
$$(I - B^{-1}Z)^2 = I - B^{-1}Z$$

$B^{-1}Z$ projects orthogonally onto $\mathbf{Range}(B^{-1}A^T S)$
$I - B^{-1}Z$ projects orthogonally onto $\mathbf{Null}(S^T A)$

Verifying that $B^{-1}Z$ is a Projection

$$M^\dagger \qquad M \qquad M^\dagger$$

$$\begin{aligned}(B^{-1}Z)^2 &= B^{-1}A^TS(S^TAB^{-1}A^TS)^\dagger S^TAB^{-1}A^TS(S^TAB^{-1}A^TS)^\dagger S^TA \\ &= B^{-1}A^TS(S^TAB^{-1}A^TS)^\dagger S^TA \\ &= B^{-1}Z\end{aligned}$$

$$Z := A^TS(S^TAB^{-1}A^TS)^\dagger S^TA$$

$$M^\dagger M M^\dagger = M^\dagger$$

Eigenvalues of $B^{-1}Z$ are in {0,1}

2.3
Complexity

Complexity / Convergence

Theorem [RG'15] For every solution x^* of $Ax = b$ we have

$$\mathbf{E}\left[x^{t+1} - x^*\right] = \left(I - B^{-1}\mathbf{E}[Z]\right)\mathbf{E}\left[x^t - x^*\right]$$

Moreover,

1. $$\left\|\mathbf{E}\left[x^t - x^*\right]\right\|_B \leq \rho^t \|x^0 - x^*\|_B$$

2. $$\mathbf{E}[Z] \succ 0$$

$$\rho := \|I - B^{-1}\mathbf{E}[Z]\|_B$$

$$\|M\|_B := \max_{\|x\|_B = 1} \|Mx\|_B$$

$$\mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \rho^t \|x^0 - x^*\|_B^2$$

Proof of 1

$$x^{t+1} - x^* \quad = \quad (I - B^{-1}Z)(x^t - x^*)$$

Taking expectations conditioned on x^t, we get

$$\mathbf{E}[x^{t+1} - x^* \mid x^t] = (I - B^{-1}\mathbf{E}[Z])(x^t - x^*).$$

Taking expectation again gives

$$\begin{aligned} \mathbf{E}[x^{t+1} - x^*] &= \mathbf{E}\left[\mathbf{E}\left[x^{t+1} - x^* \mid x^t\right]\right] \\ &= \mathbf{E}\left[(I - B^{-1}\mathbf{E}[Z])(x^t - x^*)\right] \\ &= (I - B^{-1}\mathbf{E}[Z])\mathbf{E}[x^t - x^*]. \end{aligned}$$

Applying the norms to both sides we obtain the estimate

$$\left\|\mathbf{E}[x^{t+1} - x^*]\right\|_B \leq \underbrace{\left\|I - B^{-1}\mathbf{E}[Z]\right\|_B}_{\rho} \left\|\mathbf{E}[x^t - x^*]\right\|_B .$$

The Rate: Lower and Upper Bounds

$$d := \mathbf{Rank}(S^T A) = dim(\mathbf{Range}(B^{-1}A^T S)) = \mathbf{Tr}(B^{-1}Z)$$

Theorem [RG'15]

$$0 \leq 1 - \frac{\mathbf{E}[d]}{n} \leq \rho \leq 1$$

Insight: The method is a *contraction* (without any assumptions on *S* whatsoever). That is, things can not get worse.

Insight: The lower bound on the rate improves as the dimension of the search space in the "constrain and approximate" viewpoint grows.

Proof

$$\begin{aligned}
\rho &= \|I - B^{-1}\mathbf{E}[Z]\|_B \\
&= \lambda_{\max}(I - B^{-1/2}\mathbf{E}[Z]B^{-1/2}) \\
&= 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}[Z]B^{-1/2}) \\
&= 1 - \lambda_{\min}(\mathbf{E}[B^{-1/2}ZB^{-1/2}]) \\
&= 1 - \lambda_{\min}(\mathbf{E}[B^{-1}Z]) \\
&\geq 1 - \frac{\mathbf{Tr}\left(\mathbf{E}\left[B^{-1}Z\right]\right)}{n} \\
&= 1 - \frac{\mathbf{E}\left[\mathbf{Tr}(B^{-1}Z)\right]}{n}
\end{aligned}$$

Direct calculation

$\|M\|_B := \max_{\|x\|_B=1} \|Mx\|_B$

XY and *YX* have the same spectrum

Upper bound

Smallest eigenvalue is smaller then the average of all eigenvalues

The Rate: Sufficient Condition for Convergence

$$\rho = 1 - \lambda_{\min}(B^{-1}\mathbf{E}[Z])$$

Lemma

If $\mathbf{E}[Z]$ is invertible, then

(i) $\rho < 1$,

(ii) A has full column rank, and

(iii) x^* is unique

2.4
Special Case: Randomized Kaczmarz Method

Randomized Kaczmarz (RK) Method

M. S. Kaczmarz. **Angenaherte Auflosung von Systemen linearer Gleichungen,** *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* 35, pp. 355–357, 1937

Kaczmarz method (1937)

T. Strohmer and R. Vershynin. **A Randomized Kaczmarz Algorithm with Exponential Convergence**. *Journal of Fourier Analysis and Applications* 15(2), pp. 262–278, 2009

Randomized Kaczmarz method (2009)

RK arises as a special case for parameters *B*, *S set as follows*:

$$B = I \qquad S = e^i = (0, \ldots, 0, 1, 0, \ldots, 0) \text{ with probability } p_i$$

$$x^{t+1} = x^t - \frac{A_{i:}x^t - b_i}{\|A_{i:}\|_2^2}(A_{i:})^T$$

RK was analyzed for $p_i = \frac{\|A_{i:}\|^2}{\|A\|_F^2}$

RK: Derivation and Rate

General Method

$$x^{t+1} = x^t - B^{-1}A^T S (S^T A B^{-1} A^T S)^\dagger S^T (Ax^t - b)$$

Special Choice of Parameters

$$B = I$$

$$\mathbf{P}(S = e^i) = p_i \rightarrow S = e^i$$

$$x^{t+1} = x^t - \frac{A_{i:}x^t - b_i}{\|A_{i:}\|_2^2}(A_{i:})^T$$

Complexity Rate

$$p_i = \frac{\|A_{i:}\|^2}{\|A\|_F^2}$$

$$\mathbf{E}\left[\|x^t - x^*\|_2^2\right] \leq \left(1 - \frac{\lambda_{\min}\left(A^T A\right)}{\|A\|_F^2}\right)^t \|x^0 - x^*\|_2^2$$

RK = SGD with a "smart" stepsize

$$Ax = b \qquad \text{vs} \qquad \min_x \frac{1}{2}\|Ax - b\|^2$$

Apply RK

$$f(x) = \sum_{i=1}^{m} p_i f_i(x) = \mathbf{E}_i\left[f_i(x)\right]$$
$$f_i(x) = \frac{1}{2p_i}(A_{i:}x - b_i)^2$$

Apply SGD

$$x^{t+1} = x^t - \frac{A_{i:}x^t - b_i}{\|A_{i:}\|_2^2}(A_{i:})^T$$

$$x^{t+1} = x^t - h^t \nabla f_i(x^t)$$
$$= x^t - \frac{h^t}{p_i}(A_{i:}x^t - b_i)(A_{i:})^T$$

RK is equivalent to applying SGD with a specific (smart!) constant stepsize!

$$x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|_2^2 \quad \text{s.t.} \quad x = x^t + y(A_{i:})^T, \quad y \in \mathbb{R}$$

RK: Further Reading

D. Needell. **Randomized Kaczmarz solver for noisy linear systems.** *BIT* 50 (2), pp. 395-403, 2010

D. Needell and J. Tropp. **Paved with good intentions: analyzis of a randomized block Kaczmarz method.** *Linear Algebra and its Applications* 441, pp. 199-221, 2012

D. Needell, N. Srebro and R. Ward. **Stochastic gradient descent, weighted sampling and the randomized Kaczmarz algorithm.** *Mathematical Programming*, 2015 (arXiv:1310.5715)

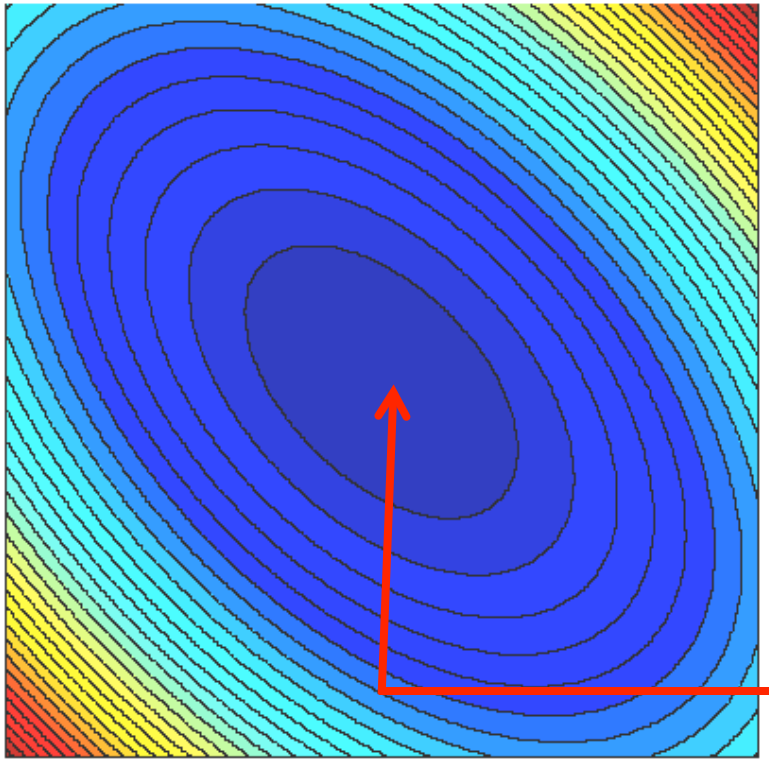
A. Ramdas. **Rows vs Columns for Linear Systems of Equations – Randomized Kaczmarz or Coordinate Descent?** *arXiv:1406.5295*, 2014

2.5
Special Case: Randomized Coordinate Descent

Coordinate Descent in 2D

Contours of a function

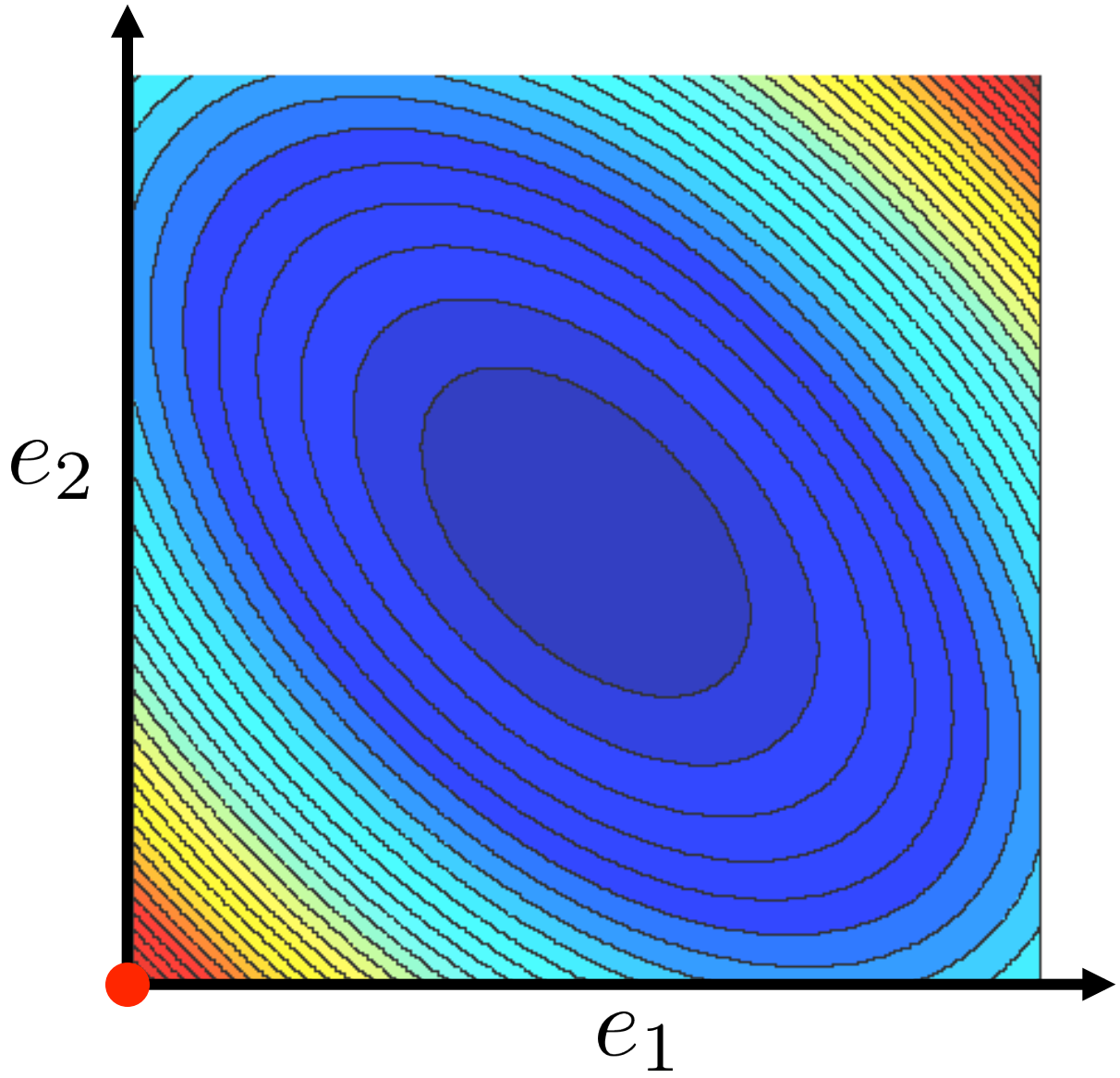
$$f : \mathbb{R}^2 \mapsto \mathbb{R}$$

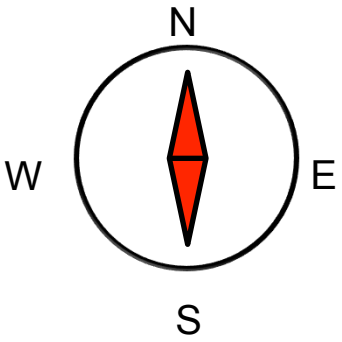


Goal:

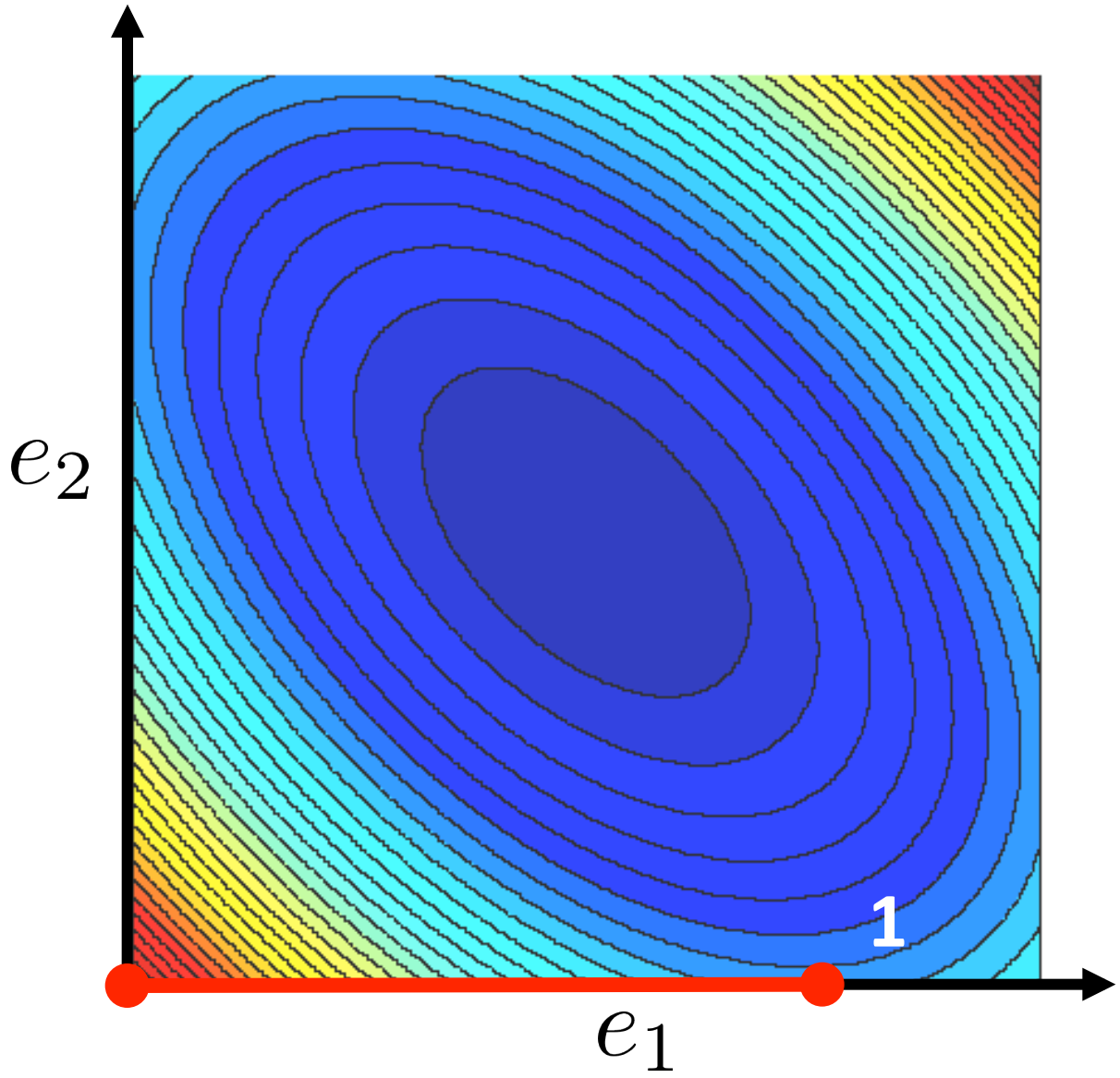
Find the minimizer

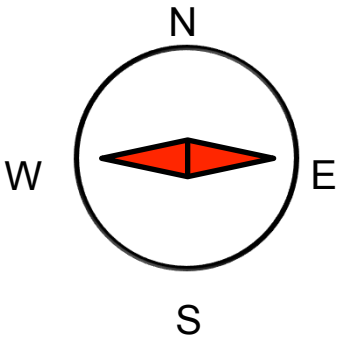
Randomized Coordinate Descent in 2D



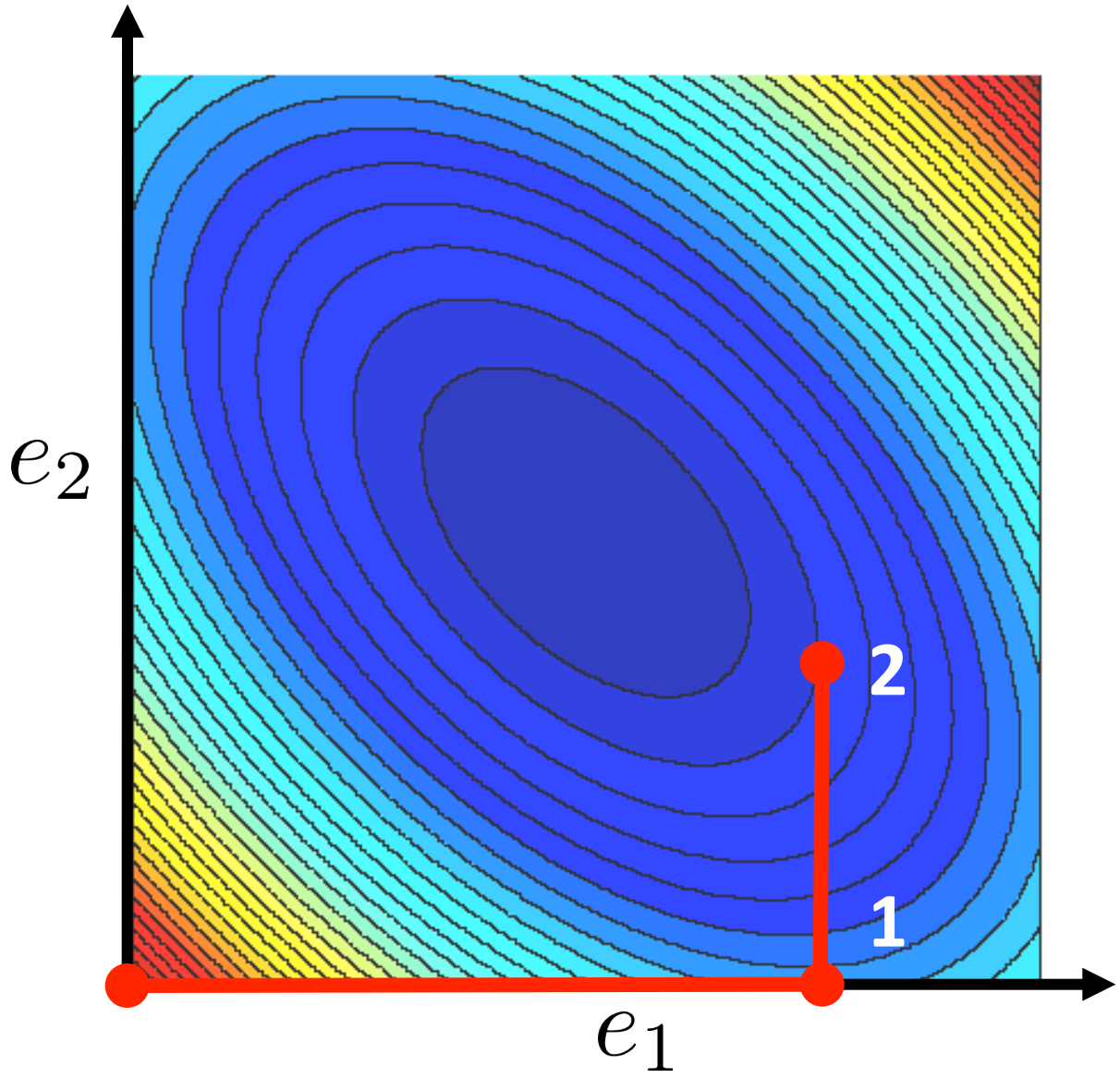


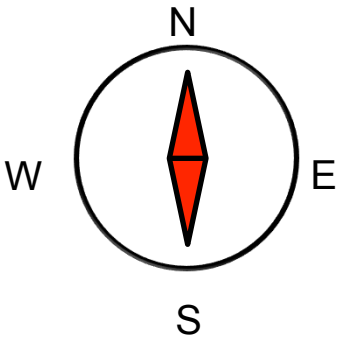
Randomized Coordinate Descent in 2D



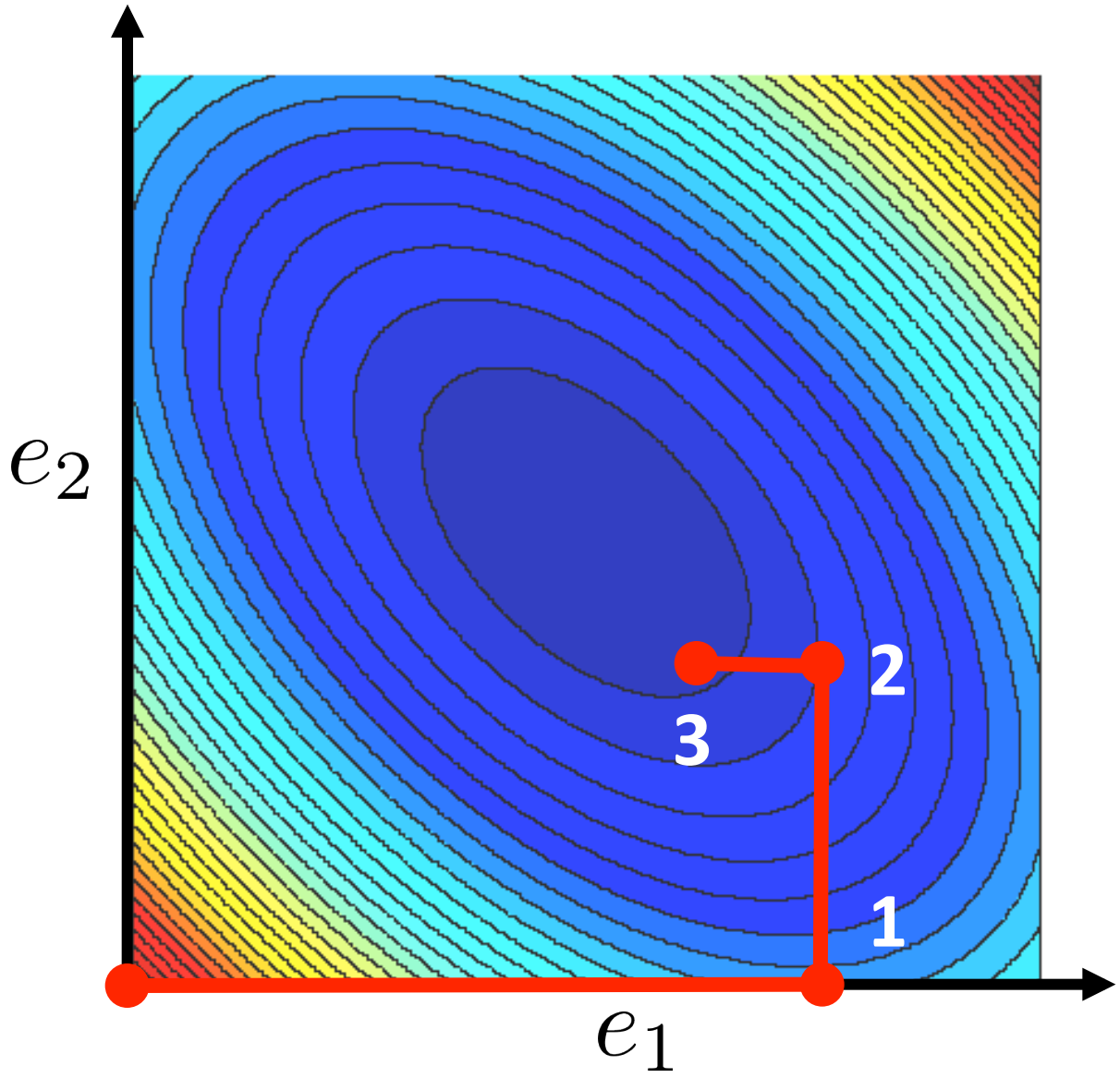


Randomized Coordinate Descent in 2D

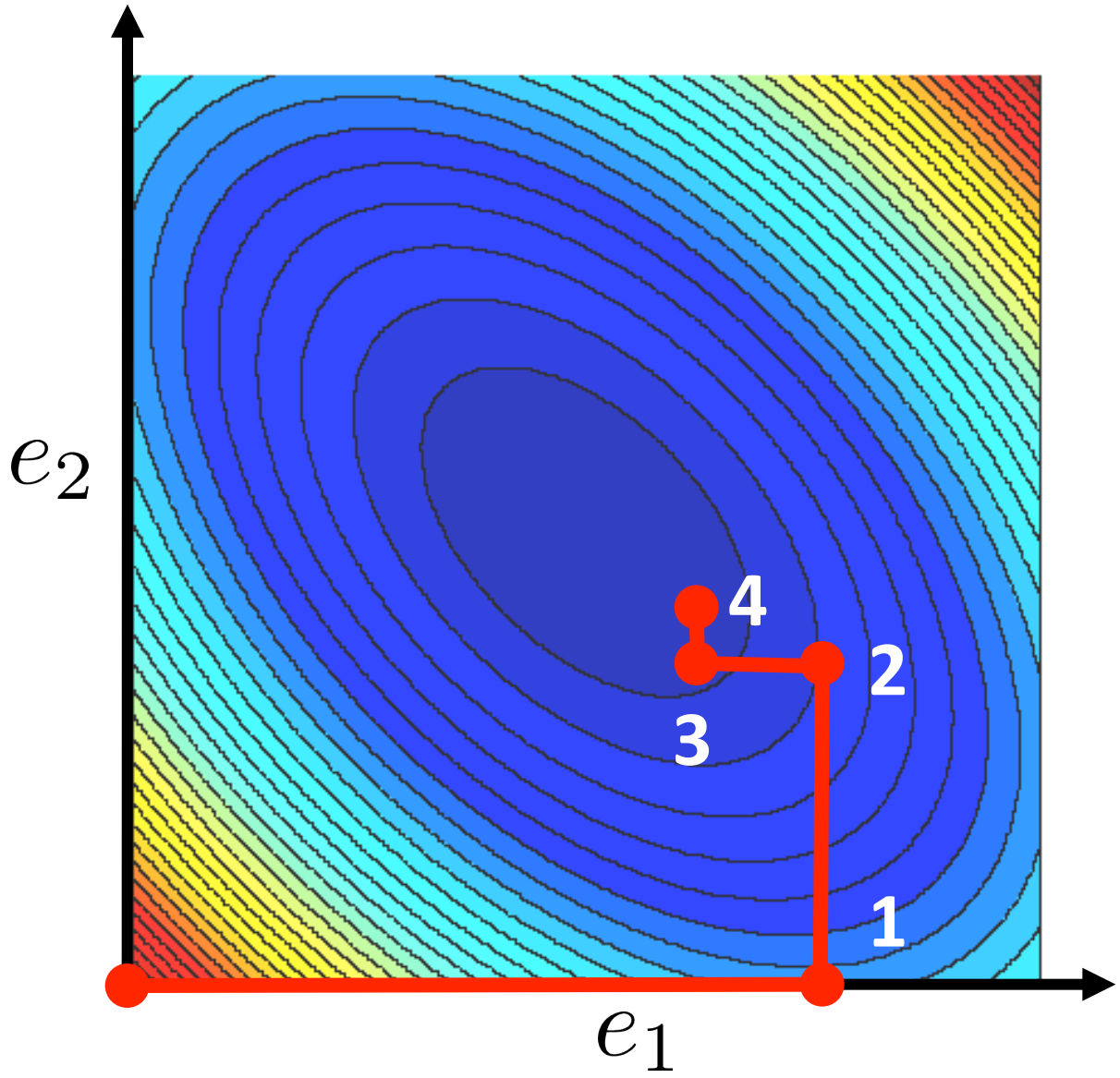


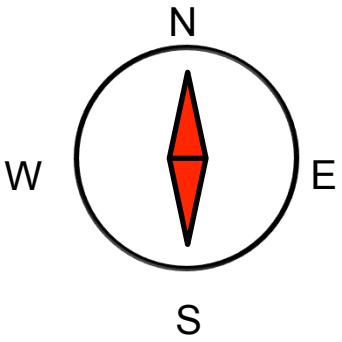


Randomized Coordinate Descent in 2D

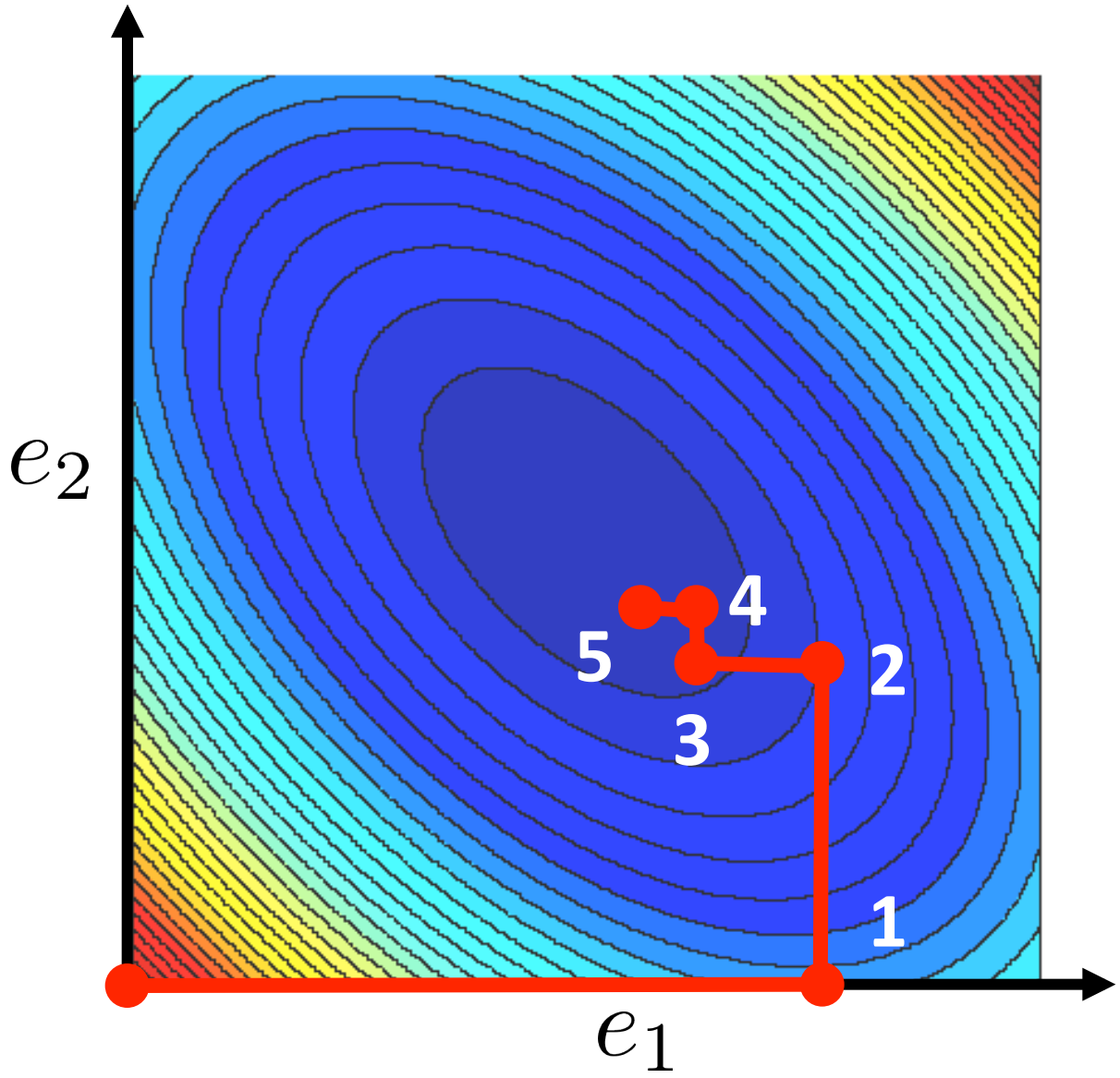


Randomized Coordinate Descent in 2D

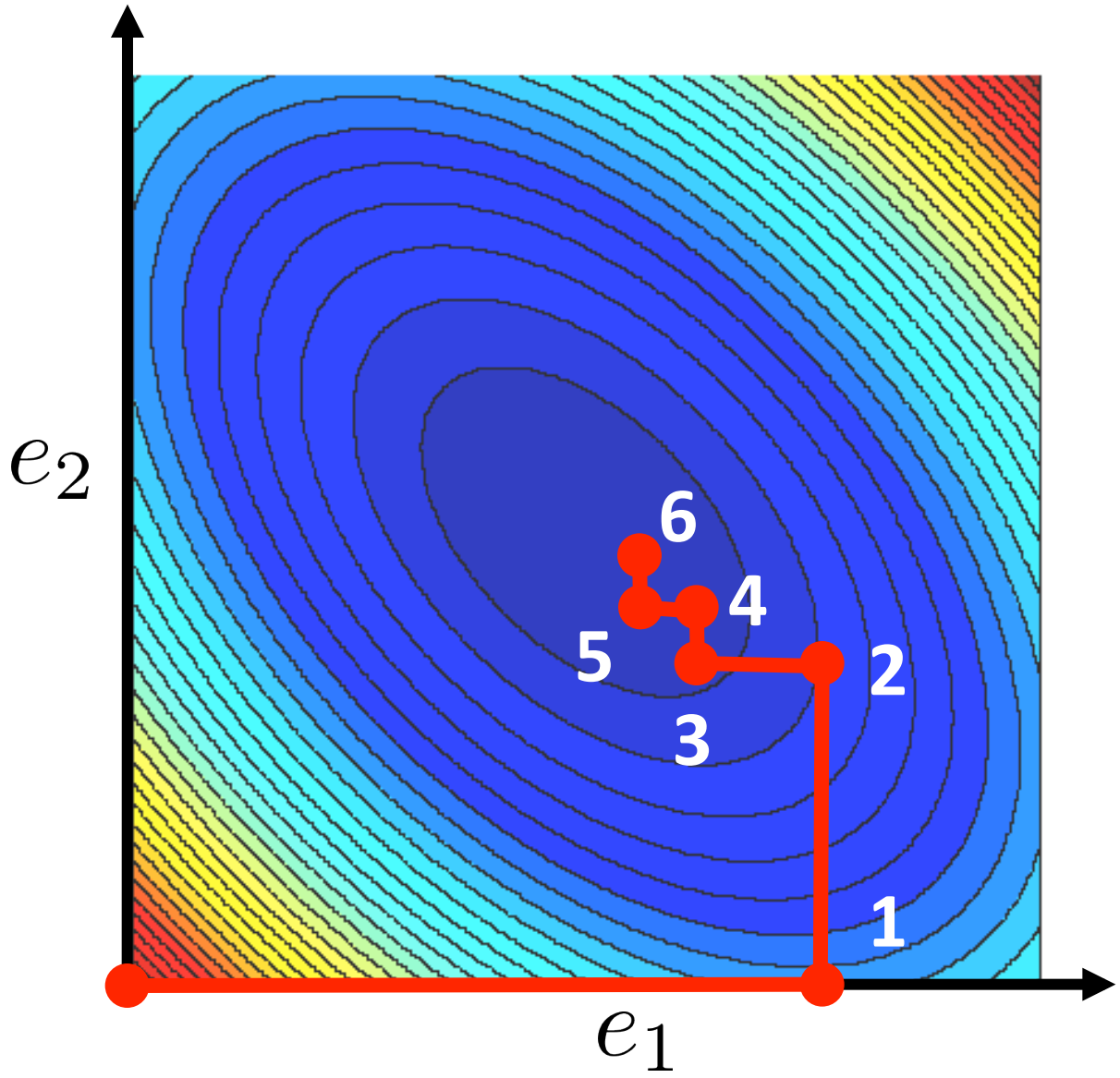


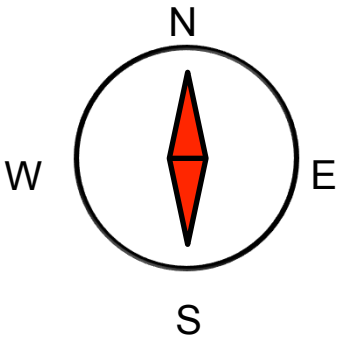


Randomized Coordinate Descent in 2D

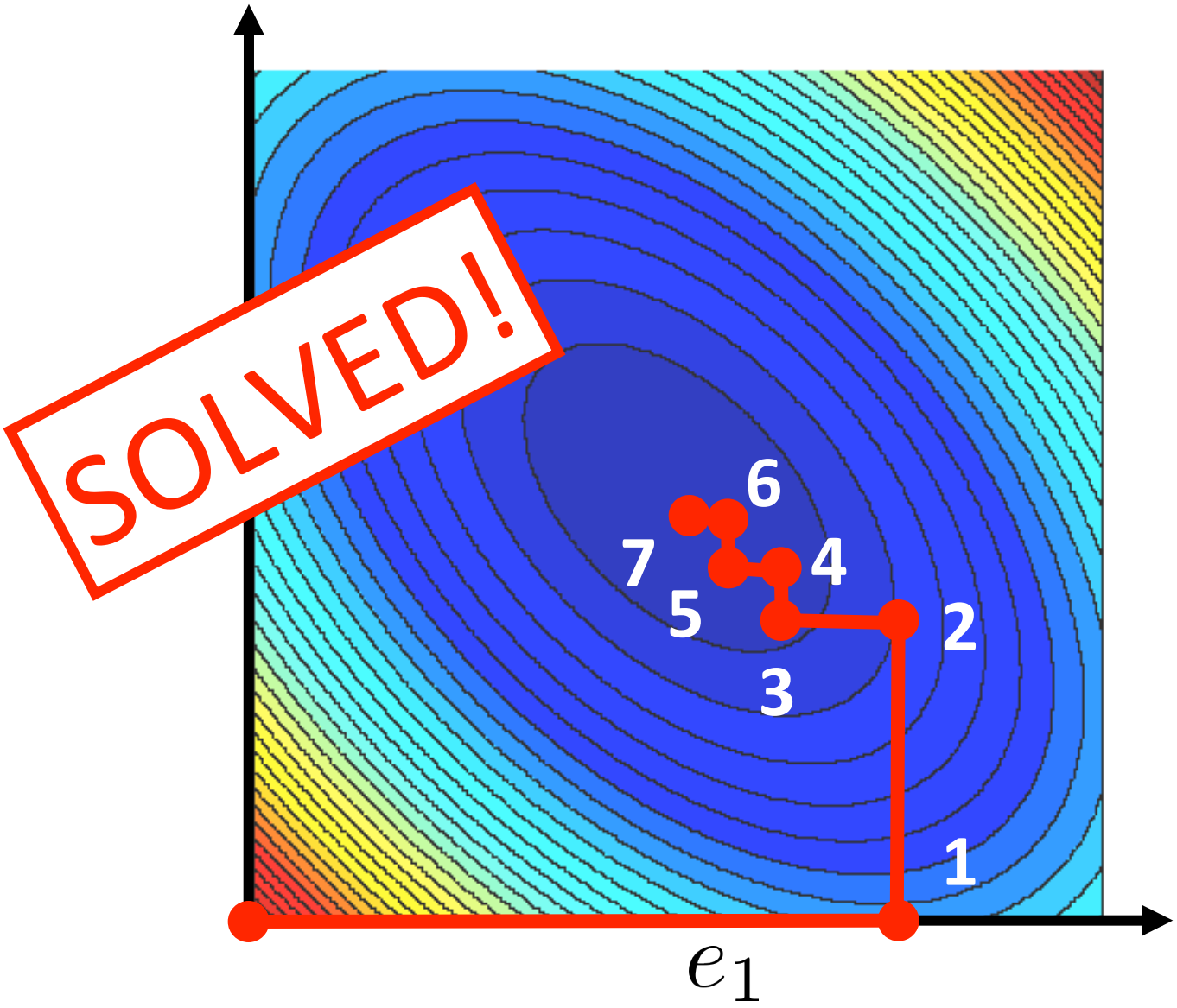


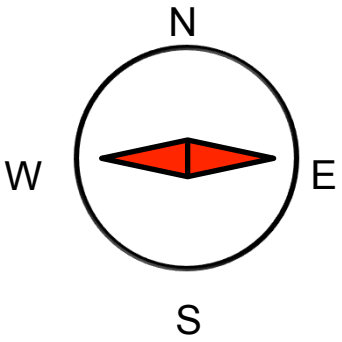
Randomized Coordinate Descent in 2D





Randomized Coordinate Descent in 2D





Randomized Coordinate Descent (RCD)

A. S. Lewis and D. Leventhal. **Randomized methods for linear constraints: convergence rates and conditioning.** *Mathematics of OR* 35(3), 641-654, 2010 (arXiv:0806.3015)

RCD (2008)

$$\min_{x \in \mathbb{R}^n} \left[f(x) = \frac{1}{2} x^T A x - b^T x \right]$$

$$x^* = A^{-1} b$$

Assume: Positive definite

RCD arises as a special case for parameters *B*, *S* *set as follows*:

$$B = A \qquad S = e^i = (0, \ldots, 0, 1, 0, \ldots, 0) \text{ with probability } p_i$$

Recall: In RK we had $B = I$

$$x^{t+1} = x^t - \frac{(A_{i:})^T x^t - b_i}{A_{ii}} e^i$$

RCD was analyzed for $p_i = \frac{A_{ii}}{\mathbf{Tr}(A)}$

RCD: Derivation and Rate

General Method

$$x^{t+1} = x^t - B^{-1}A^T S (S^T A B^{-1} A^T S)^\dagger S^T (Ax^t - b)$$

Special Choice of Parameters

$$B = A$$

$$\mathbf{P}(S = e^i) = p_i \quad \Rightarrow \quad S = e^i$$

$$x^{t+1} = x^t - \frac{(A_{i:})^T x^t - b_i}{A_{ii}} e^i$$

Complexity Rate

$$p_i = \frac{A_{ii}}{\mathbf{Tr}(A)}$$

$$\mathbf{E}\left[\|x^t - x^*\|_A^2\right] \leq \left(1 - \frac{\lambda_{\min}(A)}{\mathbf{Tr}(A)}\right)^t \|x^0 - x^*\|_A^2$$

RCD uses "Exact Line Search"

Recall Viewpoint 2 ("Constrain and Approximate"):

$$\begin{aligned} x^{t+1} \quad = \quad & \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|_B^2 \\ & \text{subject to} \quad x = x^t + B^{-1}A^T S y \\ & \qquad\qquad\quad y \quad \text{is free} \end{aligned}$$

In RCD we have:

$$B = A \quad S = e^i$$

Observation: $\|x - x^*\|_A^2 = (x - x^*)^T A(x - x^*)$

$$\begin{aligned} &= x^T Ax - 2(x^*)^T Ax + (x^*)^T Ax^* \\ &= x^T Ax - 2b^T x + b^T x^* \\ &= 2f(x) + b^T x^* \end{aligned}$$

$x^* = A^{-1}b$

$$\begin{aligned} x^{t+1} \quad = \quad & \arg\min_{x \in \mathbb{R}^n} f(x) \\ & \text{subject to} \quad x = x^t + ye^i \\ & \qquad\qquad\quad y \in \mathbb{R} \end{aligned}$$

Insight:

RCD exactly minimizes *f* along a random coordinate direction!

RCD: "Standard" Optimization Form

Yurii Nesterov. **Efficiency of coordinate descent methods on huge-scale optimization problems.** *SIAM J. on Optimization*, 22(2):341–362, 2012 (CORE Discussion Paper 2010/2)

Nesterov considered the problem:

$$\min_{x \in \mathbb{R}^n} f(x)$$

Convex and smooth

Nesterov assumed that the following inequality holds for all *x*, *h* and *i*:

$$f(x + he^i) \leq f(x) + \nabla_i f(x) h + \frac{L_i}{2} h^2$$

$$f(x) = \tfrac{1}{2} x^T A x - b^T x \quad \Rightarrow$$
$$L_i = A_{ii} \qquad \nabla_i f(x) = (A_{i:})^T x - b_i$$

Given a current iterate *x*, choosing *h* by minimizing the RHS gives:

Nesterov's RCD method:

$$x^{t+1} = x^t - \frac{1}{L_i} \nabla_i f(x^t) e^i$$

We recover RCD as we have seen it:

$$x^{t+1} = x^t - \frac{(A_{i:})^T x^t - b_i}{A_{ii}} e^i$$

2.6
Special Case: Randomized Newton Method

Randomized Newton (RN)

Z. Qu, PR, M. Takáč and O. Fercoq. **Stochastic Dual Newton Ascent for Empirical Risk Minimization.** *arXiv:1502.02268*, 2015

SDNA

$$\min_{x\in\mathbb{R}^n} \left[f(x) = \frac{1}{2}x^T Ax - b^T x \right]$$

$$x^* = A^{-1}b$$

Assume: Positive definite

RN arises as a special case for parameters *B, S set as follows*:

$$B = A \qquad S = I_{:C} \text{ with probability } p_C$$

$$p_C \geq 0 \quad \forall C \subseteq \{1,\dots,n\} \qquad \sum_{C\subseteq\{1,\dots,n\}} p_C = 1$$

RCD is special case with $p_C = 0$ whenever $|C| \neq 1$

RN: Derivation

General Method

$$x^{t+1} = x^t - B^{-1}A^T S (S^T A B^{-1} A^T S)^\dagger S^T (Ax^t - b)$$

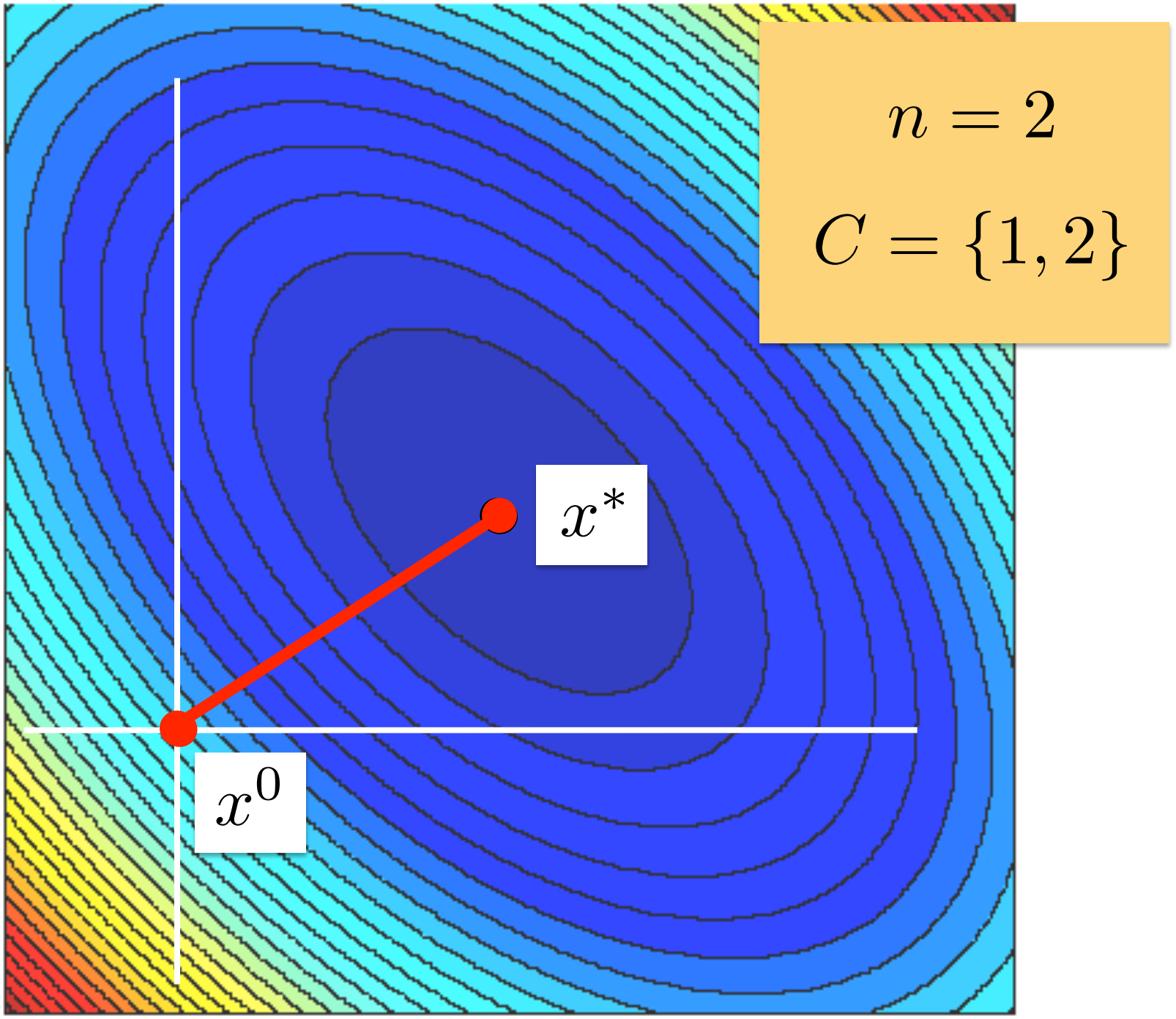
Special Choice of Parameters $B = A$

$S = I_{:C}$ with probability p_C

$$x^{t+1} = x^t - I_{:C}((I_{:C})^T A I_{:C})^{-1}(I_{:C})^T(Ax^t - b)$$

This method minimizes *f* exactly in a random subspace spanned by the coordinates belonging to *C*

Complexity Rate Will talk about this more later in the "curvature" part


$n = 2$
$C = \{1, 2\}$
x^*
x^0

2.7
Special Case:
Gaussian Descent

Gaussian Descent

General Method

$$x^{t+1} = x^t - B^{-1}A^T S (S^T A B^{-1} A^T S)^\dagger S^T (Ax^t - b)$$

Special Choice of Parameters

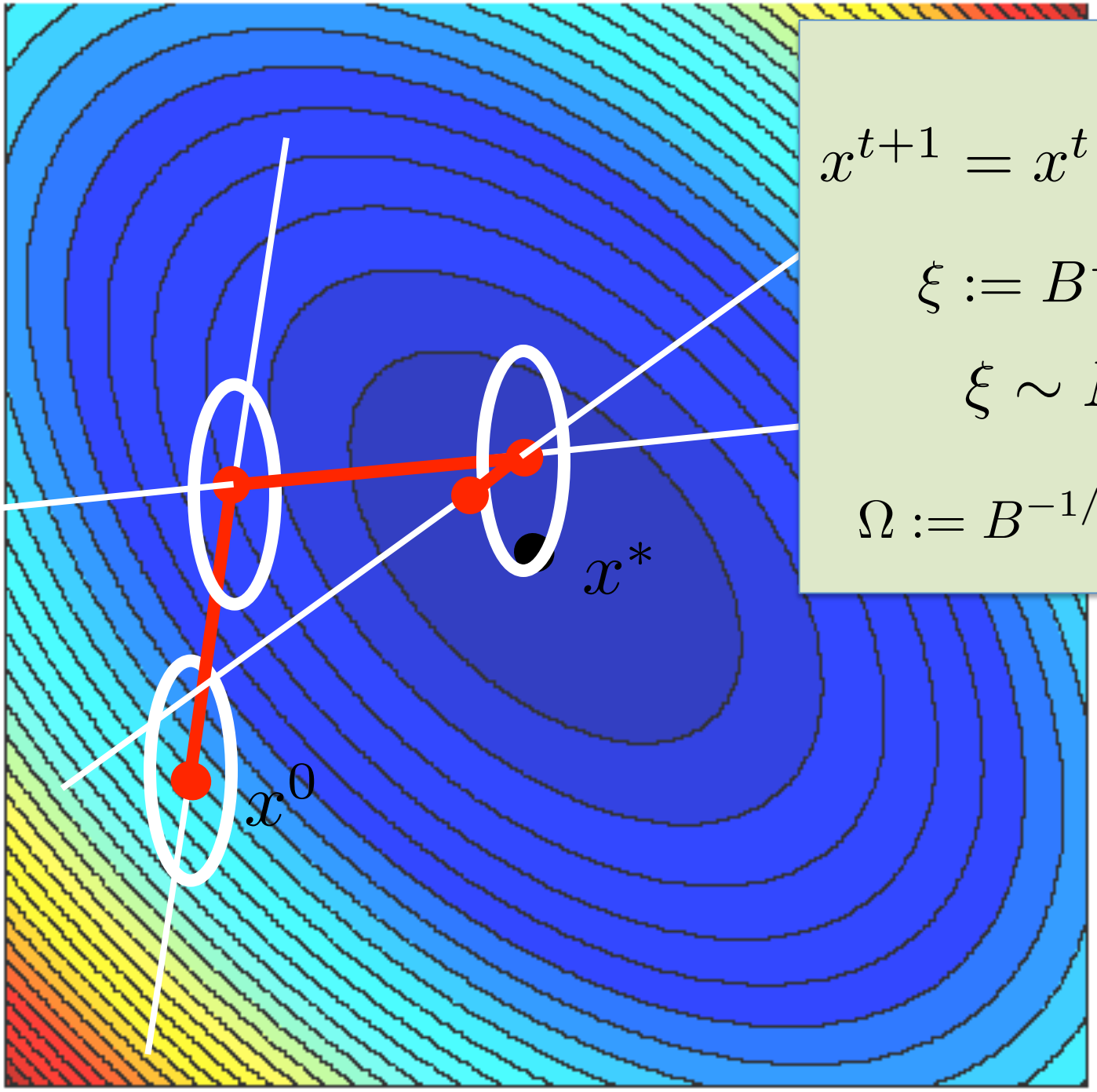
$$S \sim N(0, \Sigma)$$

Positive definite covariance matrix

$$x^{t+1} = x^t - \frac{S^T(Ax^t - b)}{S^T A B^{-1} A^T S} B^{-1} A^T S$$

Complexity Rate

$$\mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \rho^t \|x^0 - x^*\|_B^2$$



$$x^{t+1} = x^t - h^t B^{-1/2} \xi$$

$$\xi := B^{-1/2} A^T S$$

$$\xi \sim N(0, \Omega)$$

$$\Omega := B^{-1/2} A^T \Sigma A B^{-1/2}$$

Gaussian Descent: The Rate

XY and *YX* have the same spectrum

$$
\begin{aligned}
\rho &= 1 - \lambda_{\min}(B^{-1}\mathbf{E}[Z]) \\
&= 1 - \lambda_{\min}\left(B^{-1/2}\mathbf{E}[Z]B^{-1/2}\right) \\
&= 1 - \lambda_{\min}\left(B^{-1/2}\mathbf{E}\left[A^T S(S^T AB^{-1}A^T S)^\dagger S^T A\right]B^{-1/2}\right) \\
&= 1 - \lambda_{\min}\left(\mathbf{E}\left[B^{-1/2}A^T S(S^T AB^{-1}A^T S)^\dagger S^T AB^{-1/2}\right]\right) \\
&= 1 - \lambda_{\min}\left(\mathbf{E}\left[\frac{\xi\xi^T}{\|\xi\|_2^2}\right]\right)
\end{aligned}
$$

Z

$$\xi := B^{-1/2}A^T S$$

$$\xi \sim N(0, \Omega)$$

$$\Omega := B^{-1/2}A^T \Sigma A B^{-1/2}$$

Gaussian Descent: The Rate

Lemma [GR'15]

$$\mathbf{E}\left[\frac{\xi\xi^T}{\|\xi\|_2^2}\right] \succeq \frac{2}{\pi}\frac{\Omega}{\mathbf{Tr}(\Omega)}$$

$$\rho \leq 1 - \frac{2}{\pi}\frac{\lambda_{\min}(\Omega)}{\mathbf{Tr}(\Omega)}$$

This follows from the general lower bound $1 - \frac{\mathbf{E}[d]}{n} \leq \rho$ since $d = 1$

Gaussian Descent: Further Reading

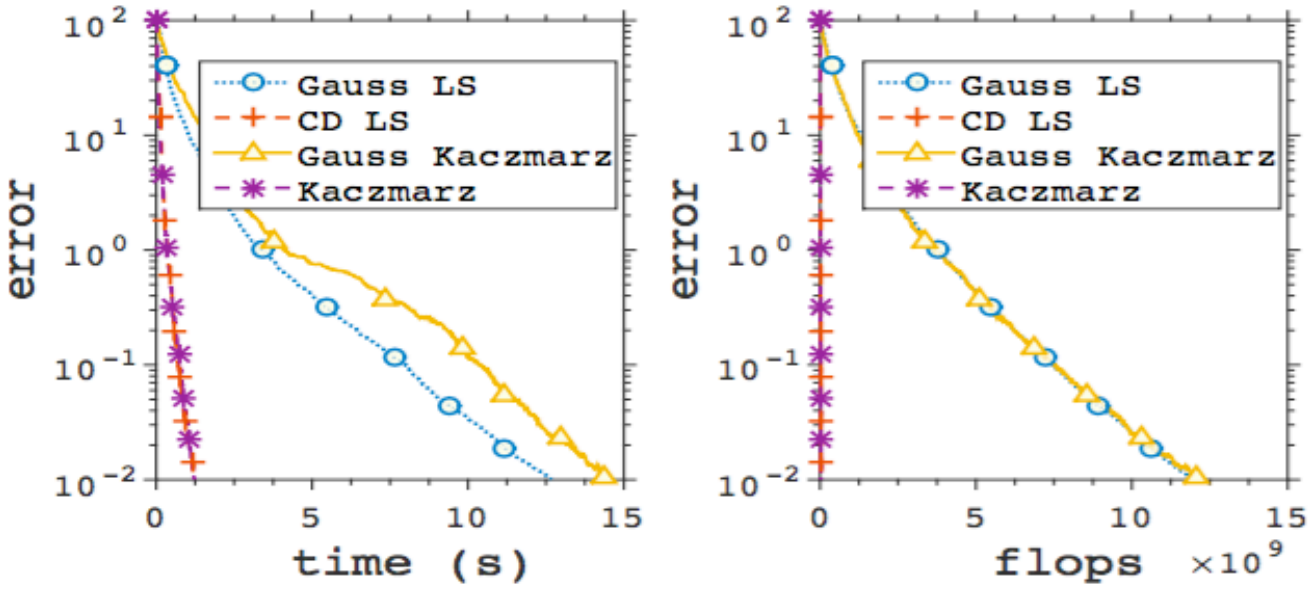
Yurii Nesterov. **Random gradient-free minimization of convex functions.** CORE Discussion Paper # 2011/1, 2011

S. U. Stitch, C. L. Muller and G. Gartner. **Optimization of convex functions with random pursuit.** SIAM Journal on Optimization 23 (2), pp. 1284-1309, 2014

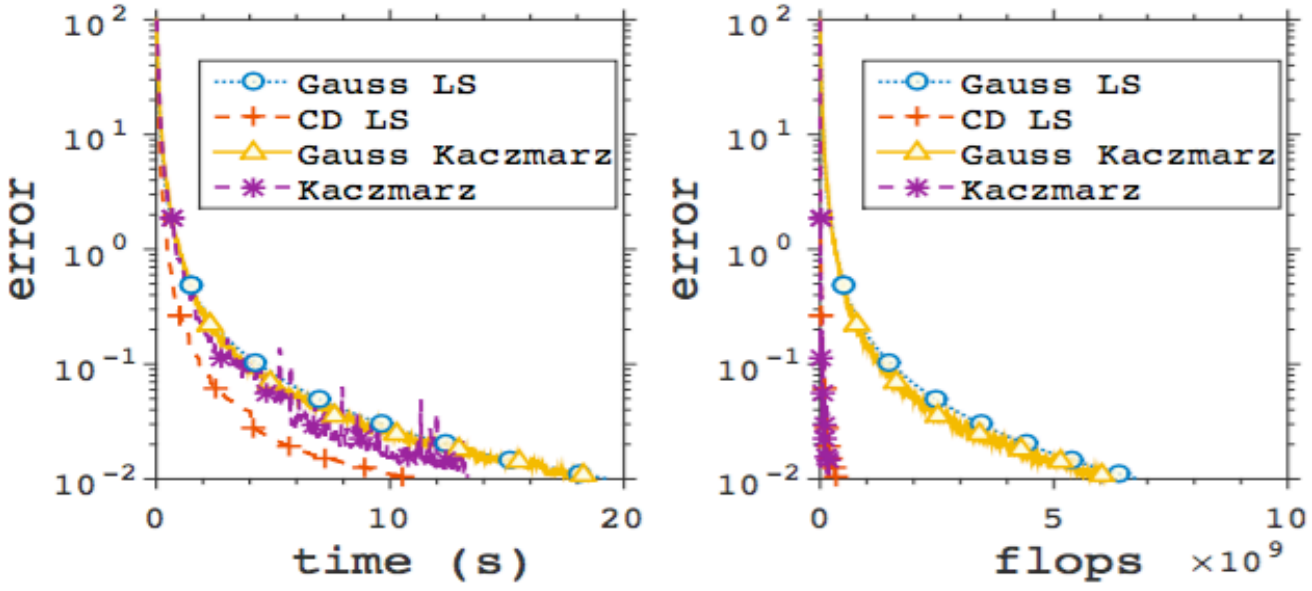
S. U. Stitch. **Convex optimization with random pursuit.** PhD Thesis, ETH Zurich, 2014

2.8
Experiments

Synthetic data

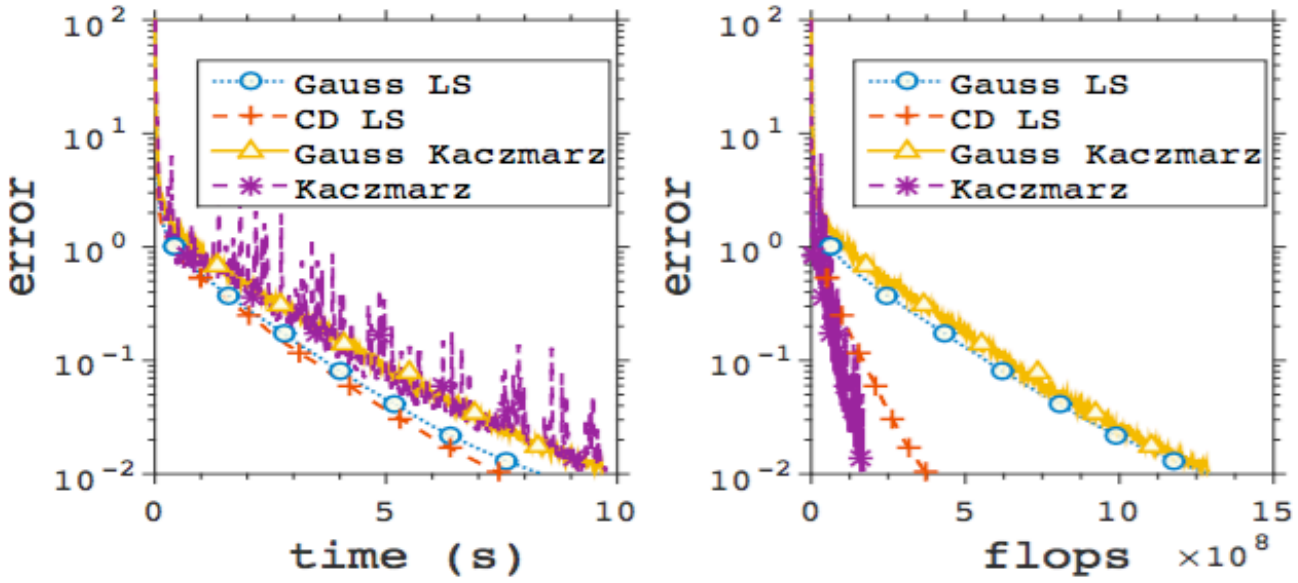


(a) **rand** (m = 1,000; n = 500)

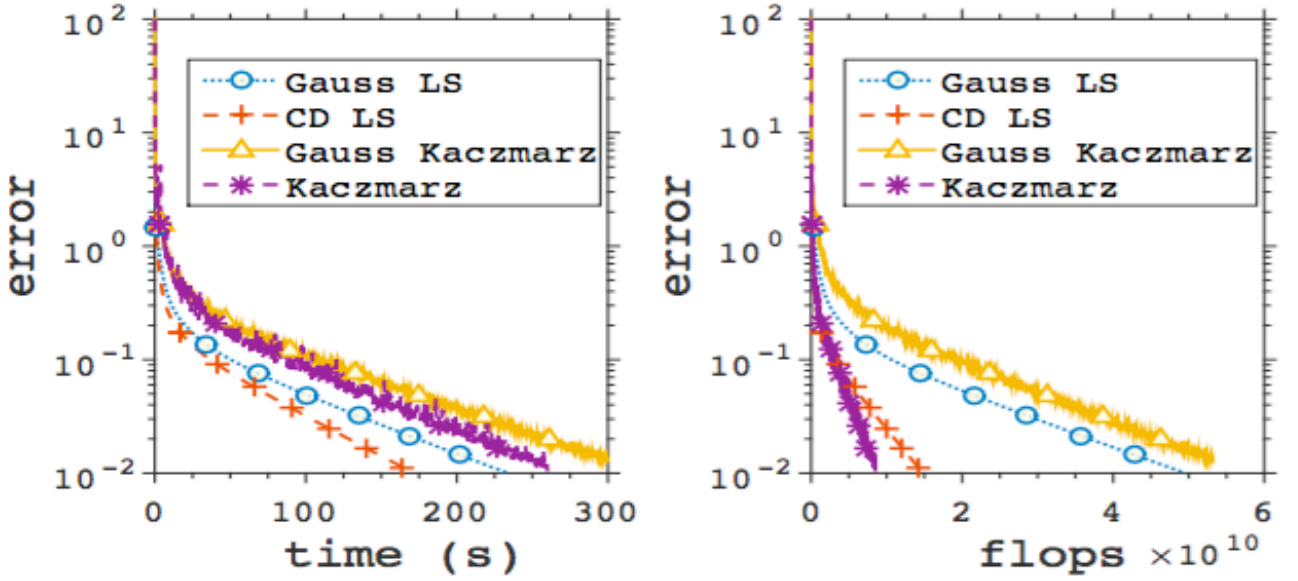


(b) **sprandn** (m = 1,000; n = 500)

Real data (Matrix Market)



(a) illc1033 (m = 1,850; n = 750)



(b) well1033 (m = 1,033; n = 320)

2.9
Importance Sampling

Importance Sampling

Assume that S is discrete:

$$S = S_i \quad \text{with probability} \quad p_i \quad (i = 1, \dots, r)$$

Question

Consider $S_1, \dots, S_r$ fixed. How to choose the probabilities $p_1, \dots, p_r$ which optimize the convergence rate $\rho = 1 - \lambda_{\min}(B^{-1}\mathbf{E}[Z])$?

$$\max_p \left\{ \lambda_{\min}(B^{-1}\mathbf{E}[Z]) \quad \text{subject to} \quad \sum_{i=1}^r p_i = 1,\ p \geq 0 \right\}$$

- Can be reformulated as an SDP (Semidefinite Program)
- Leads to different probabilities than those proposed for RK and RCD!

$$\begin{aligned} \max_{p,t} \quad & t \\ \text{subject to} \quad & \sum_{i=1}^r p_i \left(V_i (V_i^T V_i)^\dagger V_i^T \right) \succeq t \cdot I, \\ & p \geq 0, \quad \sum_{i=1}^r p_i = 1 \end{aligned}$$

$$V_i = B^{-1/2} A^T S_i$$

RCD: Optimal Probabilities Can Lead to a Remarkable Improvement

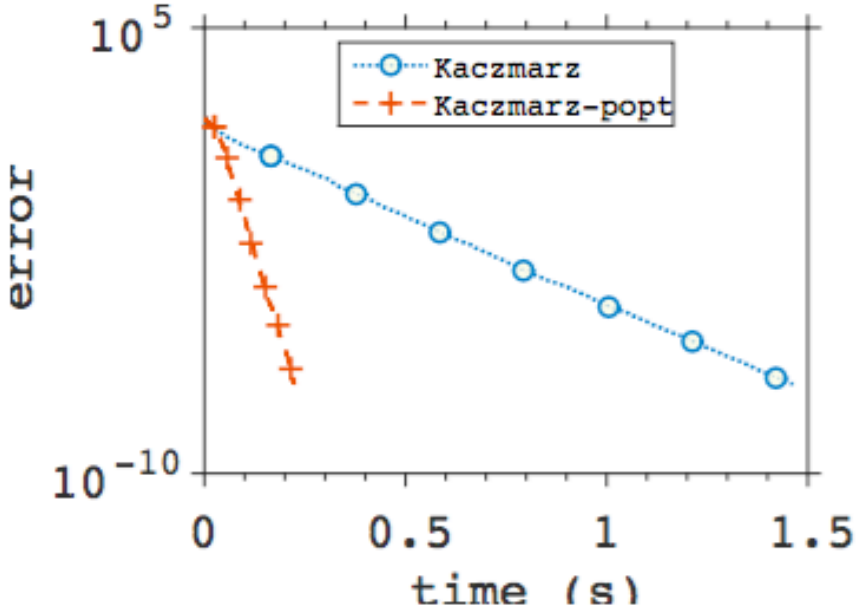
Rate for **convenient** (standard) **probabilities**

Rate for **optimal probabilities** (solving SDP)

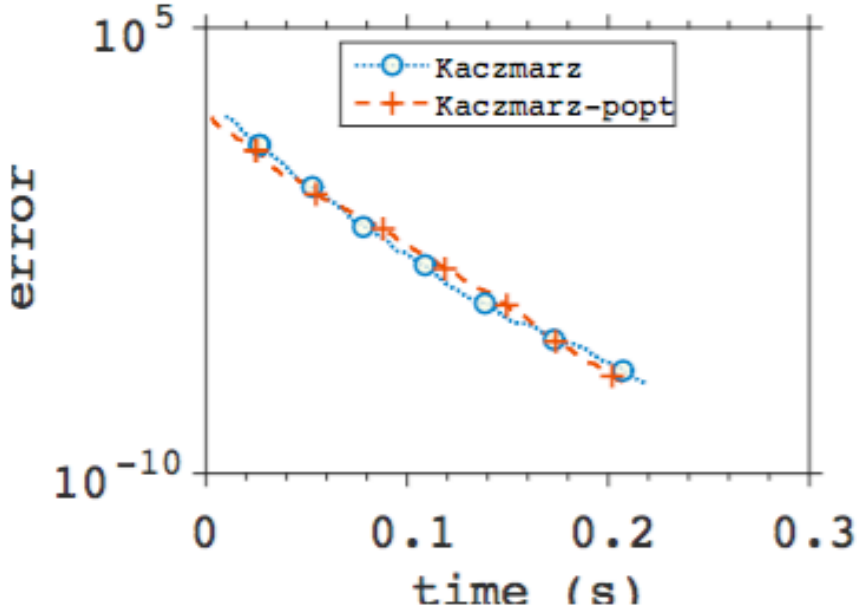
Lower bound on the rate

data set	ρ_c	ρ^*	$1-1/n$
rand(50,50)	$1-2\cdot 10^{-6}$	$1-3.05\cdot 10^{-6}$	$1-2.10^{-2}$
mushrooms-ridge	$1-5.86\cdot 10^{-6}$	$1-7.15\cdot 10^{-6}$	$1-8.93\cdot 10^{-3}$
aloi-ridge	$1-2.17\cdot 10^{-7}$	$1-1.26\cdot 10^{-4}$	$1-7.81\cdot 10^{-3}$
liver-disorders-ridge	$1-5.16\cdot 10^{-4}$	$1-8.25\cdot 10^{-3}$	$1-1.67\cdot 10^{-1}$
covtype.binary-ridge	$1-7.57\cdot 10^{-14}$	$1-1.48\cdot 10^{-6}$	$1-1.85\cdot 10^{-2}$

RK: Convenient vs Optimal

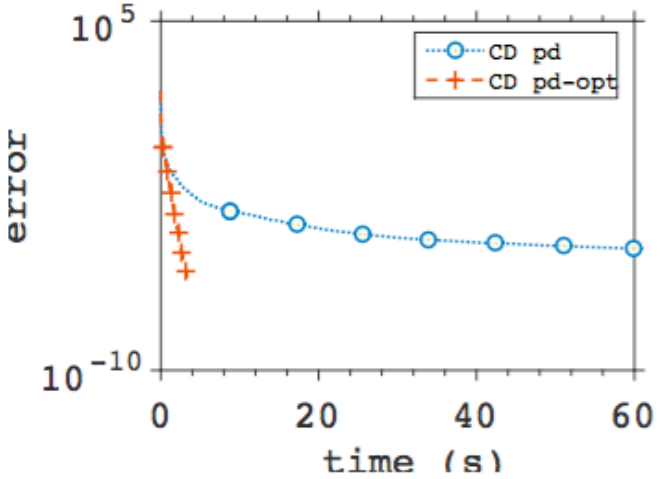


(a) liver-disorders-popt-k

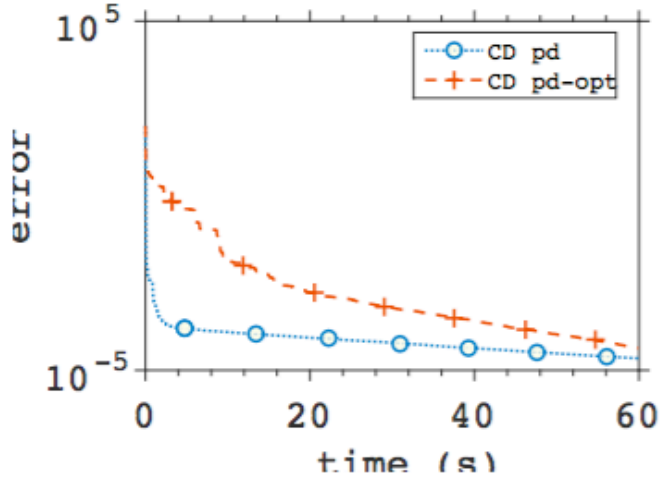


(b) rand(500,100)

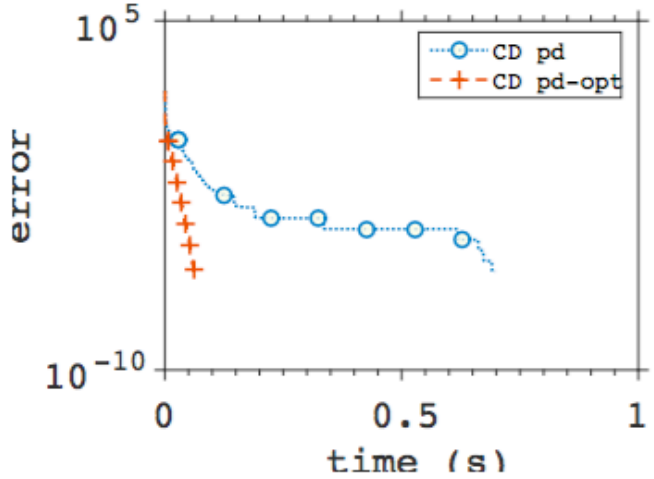
RCD: Convenient vs Optimal



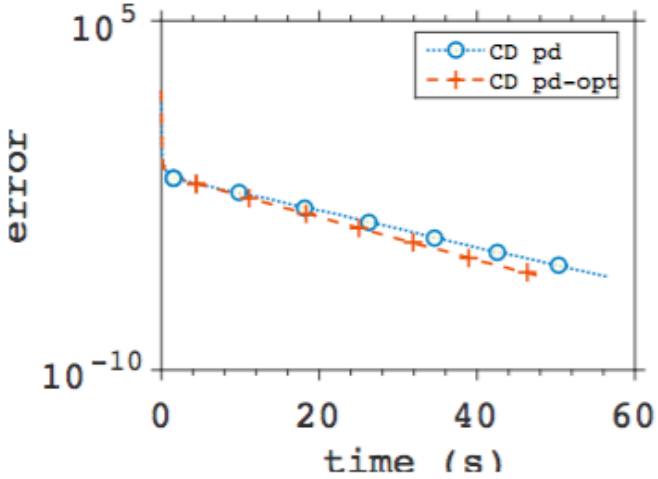
(a) aloi



(b) covtype.libsvm.binary



(c) liver-disorders-ridge



(d) mushrooms-ridge-opt

3. Arbitrary Sampling

P.R. and Martin Takáč

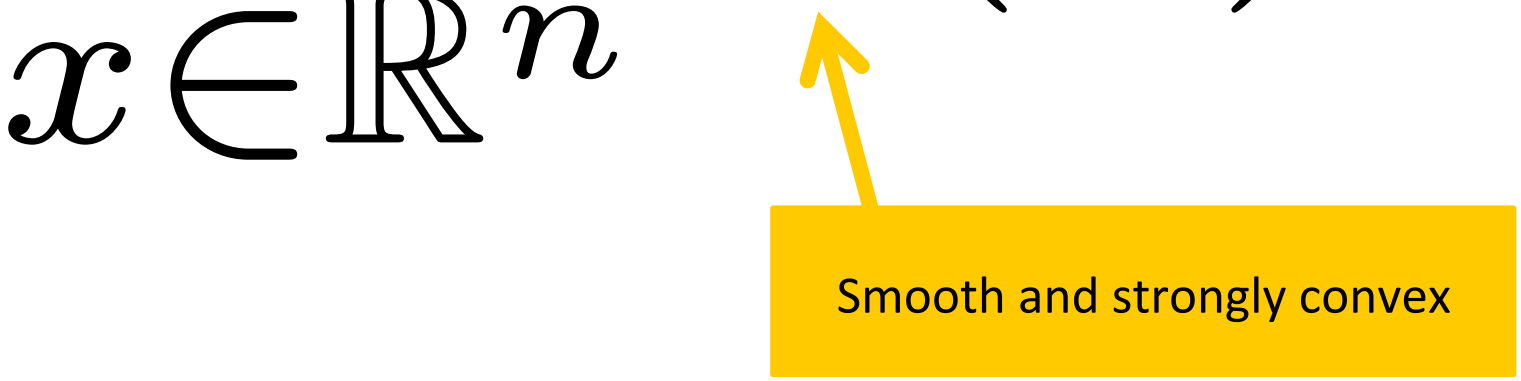
On optimal probabilities in stochastic coordinate descent methods

Optimization Letters, 2015 (arXiv:1412.8060)

3.1
The Problem

The Problem

$$\min_{x \in \mathbb{R}^n} f(x)$$



3.2
The Algorithm

"Coordinate Descent" with Arbitrary Sampling

i.i.d. subsets of [n] = {1, 2, …, n}
(arbitrary distribution is allowed!)

Choose a random set S_t of coordinates

For $i \in S_t$ do

$$x_i^{t+1} \leftarrow x_i^t - \frac{1}{v_i}(\nabla f(x^t))^\top e_i$$

For $i \notin S_t$ do

$$x_i^{t+1} \leftarrow x_i^t$$

[R & Takáč 13b]

3.3
Complexity

Complexity Result

Theorem [R & Takáč 13b]

$$t \quad \geq \quad \left(\max_i \frac{v_i}{p_i \lambda} \right) \log \left(\frac{f(x^0) - f(x^*)}{\epsilon \rho} \right)$$

$p_i = \mathbf{P}(i \in S_t)$

strong convexity constant

$$\mathbf{P}\left(f(x^t) - f(x^*) \leq \epsilon\right) \geq 1 - \rho$$

Key Assumption: Expected Separable Overapproximation (ESO)

Parameters $v_1, \ldots, v_n$ satisfy:

$$\mathbf{E}\left[f\left(x + \sum_{i \in S_t} h_i e_i\right)\right] \leq f(x) + \sum_{i=1}^{n} p_i \nabla_i f(x) h_i + \sum_{i=1}^{n} p_i v_i h_i^2$$

Inequality must hold for all

$$x, h \in \mathbb{R}^n$$

$$p_i = \mathbf{P}(i \in S_t)$$

Proof

Theorem 3. *Let Assumptions 1 and 2 be satisfied. Choose* $x^0 \in \mathbf{R}^n$, $0 < \epsilon < \phi(x^0) - \phi^*$ *and* $0 < \rho < 1$, *where* $\phi^* := \min_x \phi(x)$. *Let*

$$\Lambda := \max_i \frac{w_i}{p_i v_i}. \tag{4}$$

If $\{x^k\}$ *are the random iterates generated by 'NSync, then*

$$K \geq \frac{\Lambda}{\gamma} \log\left(\frac{\phi(x^0)-\phi^*}{\epsilon\rho}\right) \quad \Rightarrow \quad \mathbf{Prob}(\phi(x^K) - \phi^* \leq \epsilon) \geq 1 - \rho. \tag{5}$$

Moreover, we have the lower bound $\Lambda \geq (\sum_i \frac{w_i}{v_i})/\mathbf{E}[|\hat{S}|]$.

Proof. We first claim that ϕ is μ-strongly convex with respect to the norm $\|\cdot\|_{w \bullet p^{-1}}$, i.e.,

$$\phi(x+h) \geq \phi(x) + \langle \nabla\phi(x), h\rangle + \tfrac{\mu}{2}\|h\|^2_{w\bullet p^{-1}}, \tag{6}$$

where $\mu := \gamma/\Lambda$. Indeed, this follows by comparing (3) and (6) in the light of (4). Let x^* be such that $\phi(x^*) = \phi^*$. Using (6) with $h = x^* - x$,

$$\phi^* - \phi(x) \overset{(6)}{\geq} \min_{h' \in \mathbf{R}^n} \langle \nabla\phi(x), h'\rangle + \tfrac{\mu}{2}\|h'\|^2_{w\bullet p^{-1}} = -\tfrac{1}{2\mu}\|\nabla\phi(x)\|^2_{p\bullet w^{-1}}. \tag{7}$$

Let $h^k := -(\mathrm{Diag}(w))^{-1}\nabla\phi(x^k)$. Then $x^{k+1} = x^k + (h^k)_{[\hat{S}]}$, and utilizing Assumption 1, we get

$$\mathbf{E}[\phi(x^{k+1}) \mid x^k] = \mathbf{E}[\phi(x^k + (h^k)_{[\hat{S}]})] \overset{(2)}{\leq} \phi(x^k) + \langle \nabla\phi(x^k), h^k\rangle_p + \tfrac{1}{2}\|h^k\|^2_{p\bullet w} \tag{8}$$

$$= \phi(x^k) - \tfrac{1}{2}\|\nabla\phi(x^k)\|^2_{p\bullet w^{-1}} \overset{(7)}{\leq} \phi(x^k) - \mu(\phi(x^k) - \phi^*). \tag{9}$$

Taking expectations in the last inequality and rearranging the terms, we obtain $\mathbf{E}[\phi(x^{k+1}) - \phi^*] \leq (1-\mu)\mathbf{E}[\phi(x^k) - \phi^*] \leq (1-\mu)^{k+1}(\phi(x^0) - \phi^*)$. Using this, Markov inequality, and the definition of K, we finally get $\mathbf{Prob}(\phi(x^K) - \phi^* \geq \epsilon) \leq \mathbf{E}[\phi(x^K) - \phi^*]/\epsilon \leq (1-\mu)^K(\phi(x^0) - \phi^*)/\epsilon \leq \rho$.

Let us now establish the last claim. First, note that (see [16, Sec 3.2] for more results of this type),

$$\textstyle\sum_i p_i = \sum_i \sum_{S: i \in S} p_S = \sum_S \sum_{i: i\in S} p_S = \sum_S p_S |S| = \mathbf{E}[|\hat{S}|]. \tag{10}$$

Letting $\Delta := \{p' \in \mathbf{R}^n : p' \geq 0, \sum_i p'_i = \mathbf{E}[|\hat{S}|]\}$, we have

$$\Lambda \overset{(4)+(10)}{\geq} \min_{p' \in \Delta} \max_i \frac{w_i}{p'_i v_i} = \frac{1}{\mathbf{E}[|\hat{S}|]} \sum_i \frac{v_i}{w_i},$$

where the last equality follows since optimal p'_i is proportional to v_i/w_i. □

Copy-paste from the paper

3.4 Stepsizes

Zheng Qu and P.R.
Coordinate Descent with Arbitrary Sampling II: Expected Separable Overapproximation
arXiv:1412.8063, 2014

How to compute the stepsize parameters *v*?

Theorem [Qu & R 14a]

$M_j : \mathbb{R}^n \to \mathbb{R}^m$

γ_j-smooth

$$f(x) = \sum_j \phi_j(M_j x)$$

$$A^\top A = \sum_j \gamma_j M_j^\top M_j$$

Theorem [Qu & R 14a]

The assumption holds if for some matrix *A*, *f* satisfies

$$f(x+h) \le f(x) + \nabla f(x)^\top h + \frac{1}{2} h^\top A^\top A h$$

and *v* satisfies

$$P \circ A^\top A \preceq \mathrm{Diag}(p \circ v)$$

$P_{ij} = \mathbf{P}(\{i,j\} \subseteq S_t)$

Hadamard (element-wise) product

[Qu & R 14a] give formulas for *v* as a function of the data matrix *A* and sampling S_t

A Conservative Formula for v

"Normalized" largest eigenvalue of M:

$$\lambda'(M) := \max_{x \in \mathbb{R}^n} \frac{x^T M x}{x^T \mathbf{Diag}(M) x}$$

Theorem [Qu & R 14b]

For any sampling S_t, ESO holds with

$$v_i = \min\left\{\lambda'(P), \lambda'(A^T A)\right\} \|A_{:i}\|_2^2$$

$$\mathbf{P}(|S_t| = \tau) = 1 \quad \Rightarrow \quad \lambda'(P) = \tau$$

$$1 \leq \lambda'(A^\top A) \leq \underbrace{\max_j \|A_{j:}\|_0}_{\omega} \leq n$$

What Does This Mean?

- **Computation of stepsizes.** The previous result says that the ESO "stepsize" parameters $v_1, \ldots, v_n$
  - can be efficiently computed (and hence the method can be implemented)
  - are small if the minibatch size $|S_t|$ is small
  - are small if the data A has
    - good spectral properties, or
    - is sparse
- **Smaller v, faster algorithm.** Other things equal, small v means faster convergence!
- **Better stepsizes possible for special samplings:**
  - The formula is conservative as it holds for *all* samplings
  - Better bounds on v can be obtained for particular samplings (e.g., if S_t is a subset of $[n]$ chosen uniformly at random) [Qu & R 14b]

What Does This Mean?

- **Speedup.** Complexity improves with the size of the mini-batch $|S_t|$, but less than linearly
  - The amount of speedup depends on
    - data sparsity [R & Takáč 12], [Fercoq & R 13b], [Qu & R 14b]
    - spectral properties of the data [Bradley et al 11], [Takáč et al 13], [R & Takáč 13a], [Fercoq et al 14], [Qu & R 14b]
  - Hence mini-batching helps if there are gains from parallelism or reduction of memory transfers
- **Flexibility.** Sometimes we may be forced to sample in a certain way (e.g., distributed implementation)
  - Results with arbitrary sampling say it's OK to sample as we like

3.5
Importance Sampling

Importance Sampling Helps

$$\mathbf{P}(|S_t| = 1) = 1 \quad \Rightarrow \quad v = \mathrm{Diag}(A^\top A)$$

- If we update a single coordinate in each iteration, *P* is diagonal, and we get a simple formula for *v* (independent of the probability vector *p*)
- In particular, we can choose *p* which optimizes the complexity, which leads to importance sampling:

Importance sampling: $p_i = \dfrac{v_i}{\sum_i v_i} \quad \Rightarrow \quad \max_i \dfrac{v_i}{p_i \lambda} = \dfrac{\sum_i v_i}{\lambda}$

Uniform sampling: $p_i = \dfrac{1}{n} \quad \Rightarrow \quad \max_i \dfrac{v_i}{p_i \lambda} = \dfrac{n \max_i v_i}{\lambda}$

Average can be much smaller than max !

Optimal 2-Level Sampling

Definition of a parametric family of random subsets {1,2, … , *n*} of fixed cardinality:

STEP 0: Choose m subsets of $\{1, 2, \ldots, n\}$ satisfying

$$\cup_{j=1}^{m} C_j = \{1, 2, \ldots, n\} \qquad |C_j| \geq \tau \quad \forall j$$

STEP 1: Pick set C_j with probability q_j

STEP 2: Output a random subset of C_j of size τ

Finding optimal $q_1, \ldots, q_m$: Linear Program

Bibliographic Remarks I

- [Leventhal & Lewis 08] were first to study randomized CD methods (for linear systems & least squares). Moreover, they proposed nonuniform probabilities.
  - Convenient; not optimal
  - Optimal probabilities for linear systems can be computed via SDP: [Gower & R 15]
- [Nesterov 10] considered probabilities proportional to powers of coordinate-wise Lipschitz constants (for smooth convex minimization)
  - Not interpreted as optimal
- [R & Takáč 11b] gave complexity results for an arbitrary probability vector p
- [R & Takáč 13b] introduced arbitrary sampling (NSync)
  - Importance sampling as a corollary
  - Also studied importance sampling over subsets of coordinates (leads to LP)
- [Zhao & Zhang 14] studied stochastic optimization (I-Prox SGD and I-Prox SDCA) with importance sampling

Bibliographic Remarks II

- [Qu, R & Zhang 14] were first to study ERM with arbitrary sampling (Quartz)
- [Qu & R 14a] studied standard and accelerated methods for convex composite problems with arbitrary sampling (ALPHA)
- [Csiba & R 15] extended the **dual-free** analysis of SDCA [S-Shwartz 15] to arbitrary sampling (dfSDCA)
  - analysis works also for non-convex loss functions as long as the average loss is convex
- [Konečný, Qu & R 14] studied a semi-stochastic coordinate descent method (S2CD) utilizing importance sampling

3.6
RCD or
Gradient Descent ?

RCD is faster than GD

$$S_t \equiv [n] \qquad \Rightarrow \qquad v_i = \lambda_{\max}(A^\top A)$$

$$p_i = 1$$

Standard condition number

Gradient Descent =
RCD with deterministic sampling:

$$\frac{\lambda_{\max}(A^\top A)}{\lambda}$$

RCD with importance sampling:

$$\frac{\mathrm{Tr}(A^\top A)}{\lambda}$$

1 iteration of CD is often *n* times cheaper than 1 iteration of GD. However, complexity of CD can be as good as complexity of GD, and is always at most *n* times as bad. So, CD is better.

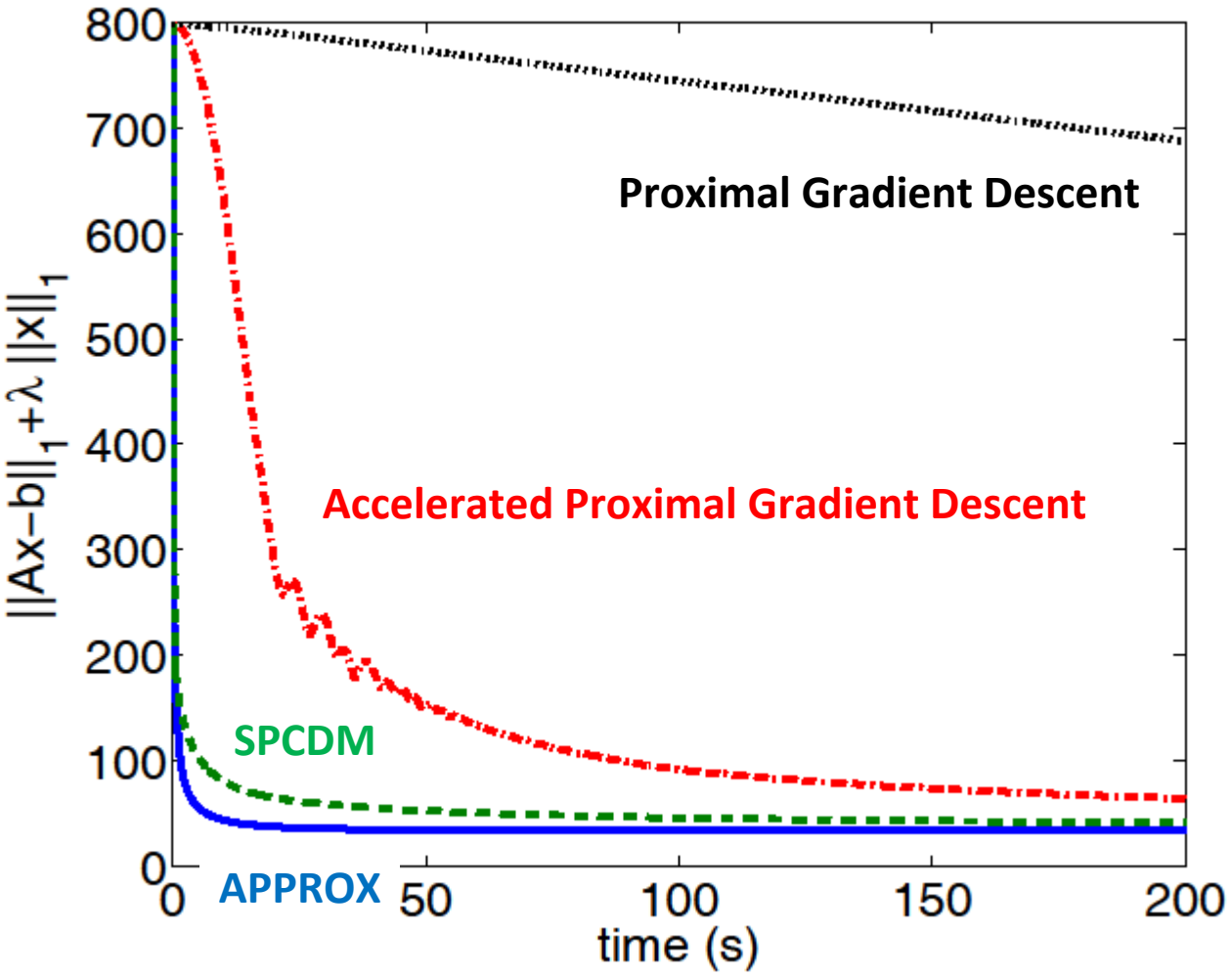
4. Acceleration

APPROX

α

Zheng Qu and P.R.
Coordinate descent with arbitrary sampling I: algorithms and complexity *arXiv:1412.8060,* 2014

L1 Regularized L1 Regression



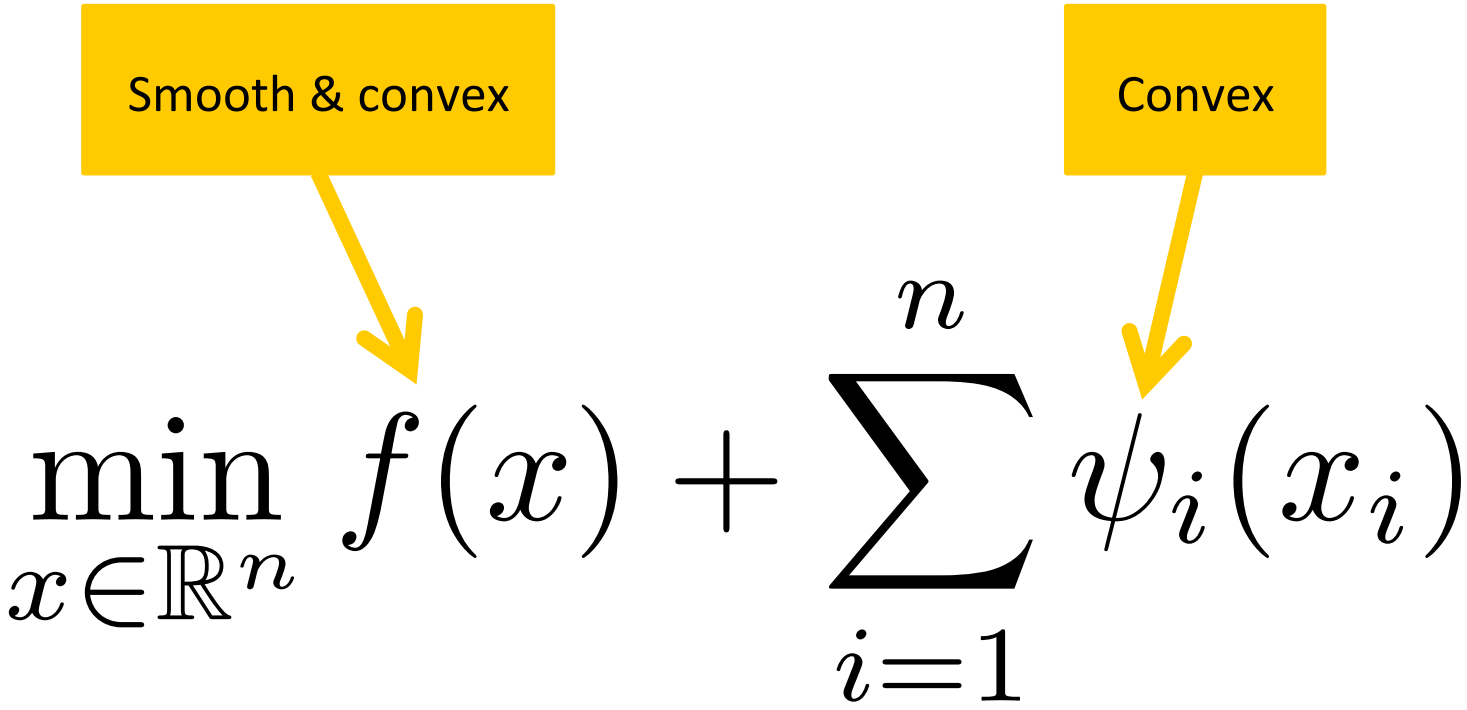
APPROX

[Fercoq & R 13b]

Dorothea dataset: $N = 100,000$ $m = 800$ $\omega = 6,061$

4.1
The Problem

The Problem



ALPHA (for Smooth Minimization)

STEP 0: $z^0 = x^0$

STEP 1: $y^t \leftarrow (1-\theta_t)x^t + \theta_t z^t$

STEP 2: For $i \in S_t$

$$z_i^{t+1} \leftarrow z_i^t - \frac{p_i}{v_i \theta_t} \nabla_i f(y^t)$$

For $i \notin S_t$

$$z_i^{t+1} \leftarrow z_i^t$$

i.i.d. random subsets of coordinates (any distribution allowed)

Same as in NSync

STEP 3: $x^{t+1} \leftarrow y^t + \theta_t Diag^{-1}(p)(z^{t+1} - z^t)$

4.2
Complexity

Complexity Result

Theorem [Qu & R 14a]

$$\theta_0 = 1, \quad \theta_{t+1} = \frac{\sqrt{\theta_t^4 + 4\theta_t^2} - \theta_t^2}{2}$$

$$\mathbf{E}[f(x^t)] - f(y) \leq \frac{2\sum_{i=1}^n (x_i^0 - y_i)^2 \frac{v_i}{p_i^2}}{(t+1)^2}$$

Same as in NSync

Arbitrary point

$p_i = \mathbf{P}(i \in \hat{S})$

4.3
Wrap-Up

Insights

- **The result makes sense:** If a coordinate is optimal – do not update it!
- **Unification:**
  - Stochastic (CD, ACD) and deterministic (GD, AGD) methods
  - Single analysis recovers the best bounds

Bibliographic Remarks

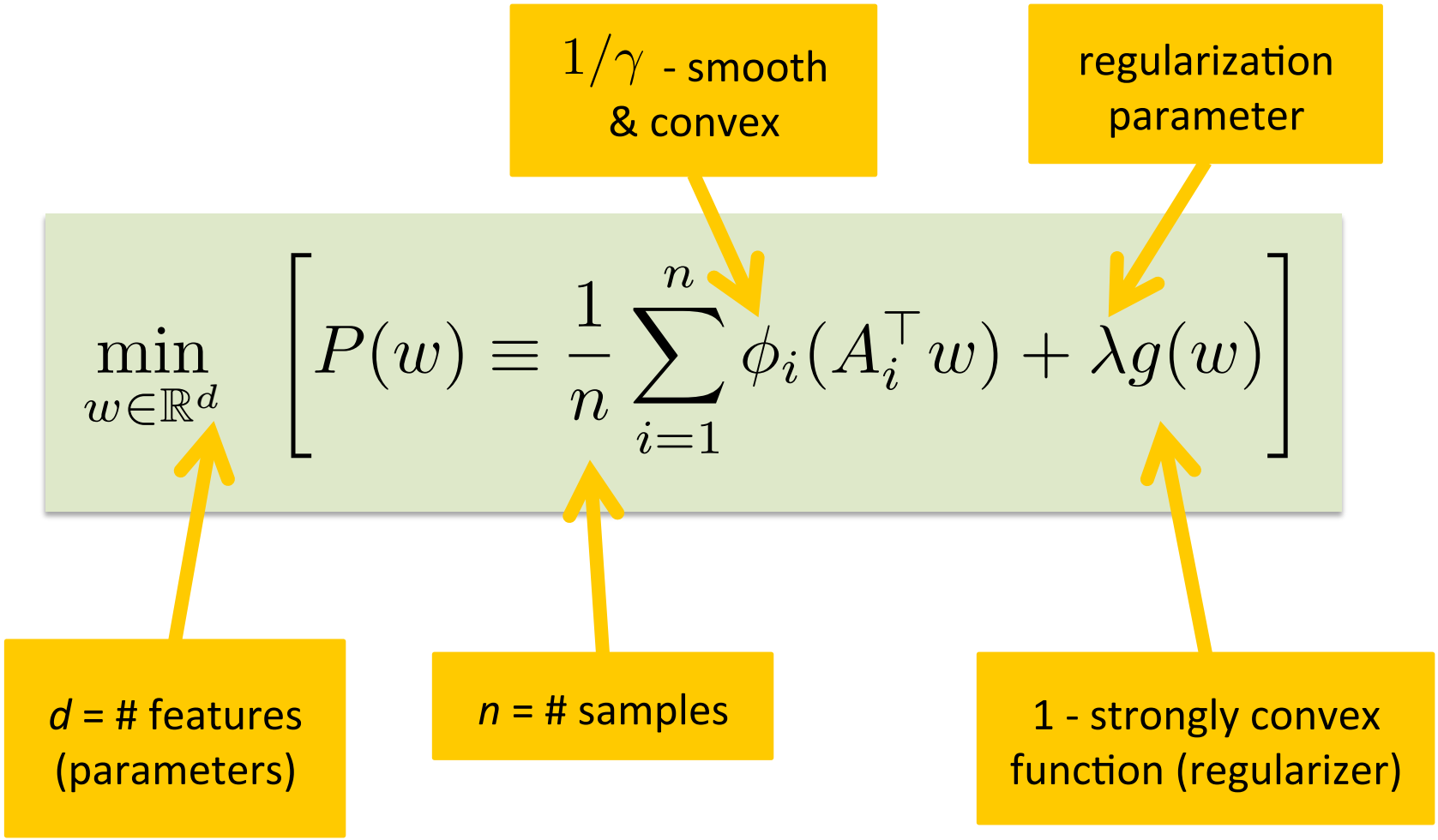
- UCDM, RCDM, ACDM [Nesterov 10]
  - First combination of acceleration & randomized coordinate descent
  - Inefficient in both theory and practice
- ASDCA [S-Shwartz & Zhang 13a]
  - Interpolates between SDCA and Accelerated Gradient Descent
- Acc Prox-SDCA [S-Shwartz & Zhang 13b]
- APPROX [Fercoq & R 13b]
  - Efficient version of accelerated coordinate descent
  - Arbitrary uniform sampling
  - Incorporates accelerated coordinate descent & accelerated gradient descent as special cases
- APCG [Lin, Lu & Xiao 14]
  - Extension of APPROX to strongly convex functions & application to ERM
- SPDC [Zhang & Xiao 14]
  - Mini-batching, importance sampling, designed for ERM
- ALPHA [Qu & R 14a]
  - Extension of APPROX to arbitrary samplings
  - Unified analysis of non-accelerated and accelerated methods

5. Duality

Zheng Qu, P.R. and Tong Zhang
Randomized dual coordinate ascent with arbitrary sampling
In NIPS 2015 (arXiv:1411.5873)

5.1
Empirical Risk Minimization

Primal Problem: ERM



Assumption 1

The loss functions $\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$ are $\frac{1}{\gamma}$-smooth:

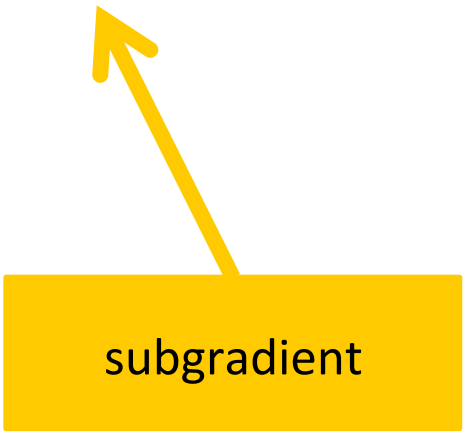
$$\|\nabla\phi_i(a) - \nabla\phi_i(a')\| \leq \frac{1}{\gamma}\|a - a'\|, \quad a, a' \in \mathbb{R}^m$$

Lipschitz constant of the gradient of the function

Assumption 2

Regularizer is 1-strongly convex

$$g(w) \geq g(w') + \langle \nabla g(w'), w - w' \rangle + \frac{1}{2}\|w - w'\|^2, \quad w, w' \in \mathbb{R}^d$$



Dual Problem

$$D(\alpha) \equiv -\lambda g^* \left(\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i \right) - \frac{1}{n} \sum_{i=1}^{n} \phi_i^*(-\alpha_i)$$

$\in \mathbb{R}^m$ (over $-\alpha_i$)

$\in \mathbb{R}^d$ (over $\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i$)

1 – smooth & convex

γ - strongly convex

$$g^*(w') = \max_{w \in \mathbb{R}^d} \left\{ (w')^\top w - g(w) \right\}$$

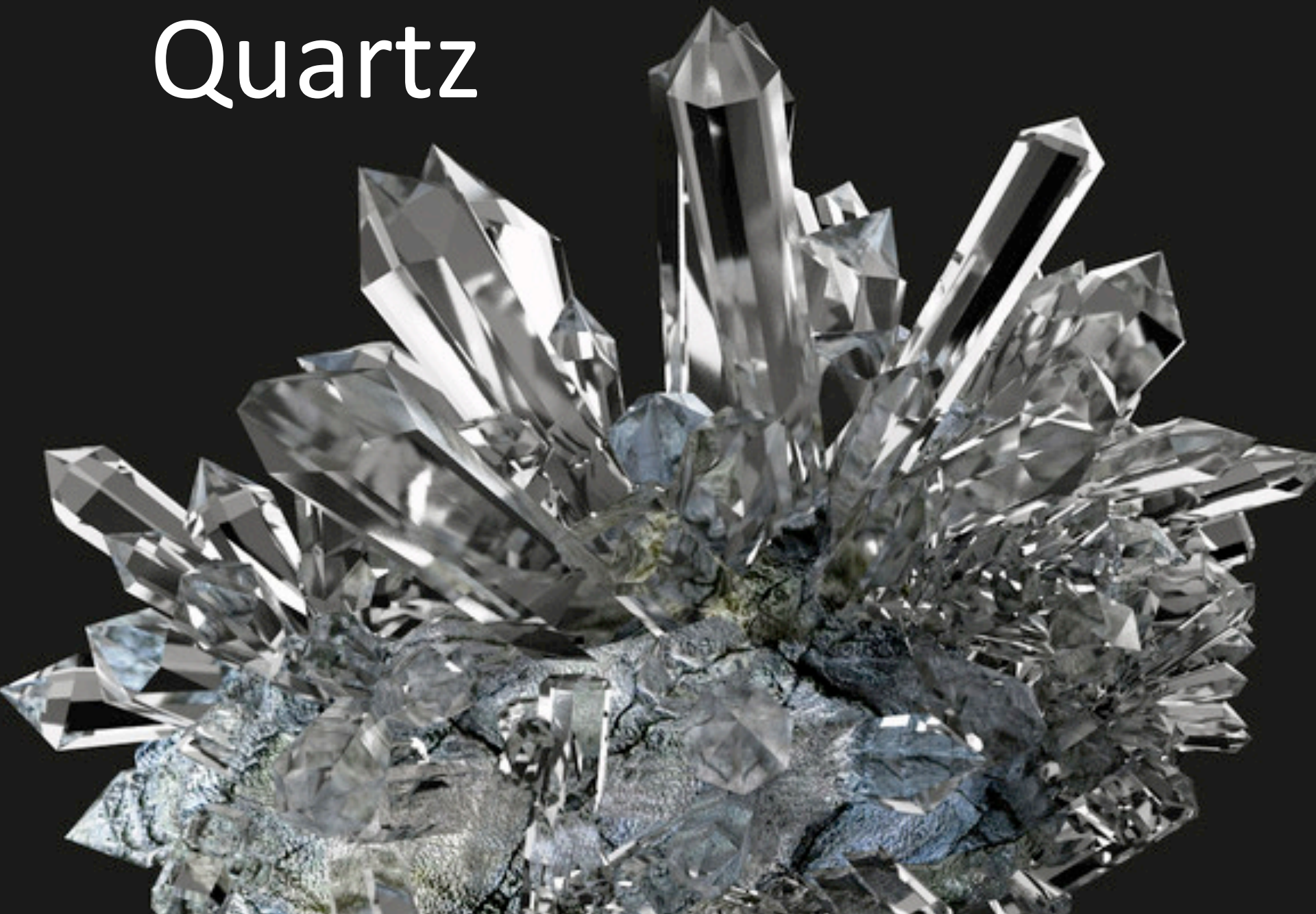
$$\phi_i^*(a') = \max_{a \in \mathbb{R}^m} \left\{ (a')^\top a - \phi_i(a) \right\}$$

$$\max_{\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^N = \mathbb{R}^{nm}} D(\alpha)$$

$\alpha_1 \in \mathbb{R}^m$, $\alpha_n \in \mathbb{R}^m$

5.2
The Algorithm

Quartz



Fenchel Duality

$$\bar{\alpha} = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i$$

$$P(w) - D(\alpha) = \lambda \left(g(w) + g^* \left(\bar{\alpha} \right) \right) + \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \phi_i^*(-\alpha_i) =$$

$$\lambda(g(w) + g^* \left(\bar{\alpha} \right) - \langle w, \bar{\alpha} \rangle) + \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \phi_i^*(-\alpha_i) + \left\langle A_i^\top w, \alpha_i \right\rangle$$

≥ 0 ⟵ **Weak duality** ⟶ ≥ 0

Optimality conditions

$$w = \nabla g^*(\bar{\alpha}) \qquad \alpha_i = -\nabla \phi_i(A_i^\top w)$$

The Algorithm

$$(\alpha^t, w^t) \quad \Rightarrow \quad (\alpha^{t+1}, w^{t+1})$$

Quartz: Bird's Eye View

STEP 1: PRIMAL UPDATE

$$w^{t+1} \leftarrow (1-\theta)w^t + \theta \nabla g^*(\bar{\alpha}^t)$$

STEP 2: DUAL UPDATE

Choose a random set S_t of dual variables

For $i \in S_t$ do

$$p_i = \mathbf{P}(i \in S_t)$$

$$\alpha_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right)\alpha_i^t + \frac{\theta}{p_i}\left(-\nabla \phi_i(A_i^\top w^{t+1})\right)$$

Algorithm 1 Quartz

Parameters: proper random sampling $\hat{S}$ and a positive vector $v \in \mathbb{R}^n$

Initialization: Choose $\alpha^0 \in \mathbb{R}^N$ and $w^0 \in \mathbb{R}^d$

Set $p_i = \mathbb{P}(i \in \hat{S})$, $\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$ and $\bar{\alpha}^0 = \frac{1}{\lambda n} \sum_{i=1}^n A_i \alpha_i^0$

Convex combination constant

for $t \geq 1$ **do**

$w^t = (1-\theta) w^{t-1} + \theta \nabla g^*(\bar{\alpha}^{t-1})$ **STEP 1**

$\alpha^t = \alpha^{t-1}$

Generate a random set $S_t \subseteq [n]$, following the distribution of $\hat{S}$

for $i \in S_t$ **do**

Calculate $\Delta\alpha_i^t$ using one of the following options:

~~**Option I** :~~

~~$\Delta\alpha_i^t = \arg\max_{\Delta \in \mathbb{R}^m} \left[-\phi_i^*(-(\alpha_i^{t-1} + \Delta)) - \nabla g^*(\bar{\alpha}^{t-1})^\top A_i \Delta - \frac{v_i \|\Delta\|^2}{2\lambda n} \right]$~~

Option II :

$\Delta\alpha_i^t = -\theta p_i^{-1} \alpha_i^{t-1} - \theta p_i^{-1} \nabla \phi_i(A_i^\top w^t)$

$\alpha_i^t = \alpha_i^{t-1} + \Delta\alpha_i^t$ **STEP 2**

end for

$\bar{\alpha}^t = \bar{\alpha}^{t-1} + (\lambda n)^{-1} \sum_{i \in S_t} A_i \Delta\alpha_i^t$

Just maintaining $\bar{\alpha}$

end for

Output: w^t, α^t

5.3
Other Stochastic Dual Methods for ERM

Randomized Dual Coordinate Ascent Methods for ERM

Algorithm	1-nice	1-optimal	τ-nice	arbitrary	additional speedup	direct p-d analysis	acceleration
SDCA	●						
mSDCA	●		●		●		
ASDCA	●		●				●
AccProx-SDCA	●						●
DisDCA	●		●				
Iprox-SDCA	●	●					
APCG	●						●
SPDC	●	●	●			●	●
Quartz	●	●	●	●	●	●	

SDCA: SS Shwartz & T Zhang, 09/2012
mSDCA M Takac, A Bijral, P R & N Srebro, 03/2013
ASDCA: SS Shwartz & T Zhang, 05/2013
AccProx-SDCA: SS Shwartz & T Zhang, 10/2013
DisDCA: T Yang, 2013
Iprox-SDCA: P Zhao & T Zhang, 01/2014
APCG: Q Lin, Z Lu & L Xiao, 07/2014
SPDC: Y Zhang & L Xiao, 09/2014
Quartz: Z Qu, P R & T Zhang, 11/2014

5.4
Complexity

Assumption 3
(Expected Separable Overapproximation)

Parameters $v_1, \ldots, v_n$ satisfy:

$$\mathbf{E}\left\|\sum_{i\in S_t} A_i\alpha_i\right\|^2 \leq \sum_{i=1}^{n} p_i v_i \|\alpha_i\|^2$$

inequality must hold for all

$$\alpha_1, \ldots, \alpha_n \in \mathbb{R}^m$$

$$p_i = \mathbf{P}(i \in S_t)$$

Complexity

$$\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$$

Theorem [Qu, R & Zhang 14]

$$\mathbf{E}[P(w^t) - D(\alpha^t)] \leq (1-\theta)^t (P(w^0) - D(\alpha^0))$$

$$t \geq \max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i \lambda \gamma n} \right) \log \left(\frac{P(w^0) - D(\alpha^0)}{\epsilon} \right)$$

$$\Rightarrow \quad \mathbf{E}\left[P(w^t) - D(\alpha^t)\right] \leq \epsilon$$

Example

Data: $n = 7 \times 10^5$

$\gamma = \frac{1}{4} \quad v_i \equiv \lambda_{\max}(A_i^\top A_i) \leq 1$

Method: $|S_t| \equiv 1 \quad p_i = \frac{1}{n} \quad \lambda = \frac{1}{n}$

$$(1-\theta)^n = 0.8187$$

$$(1-\theta)^{12n} = 0.0907 < \tfrac{1}{10}$$

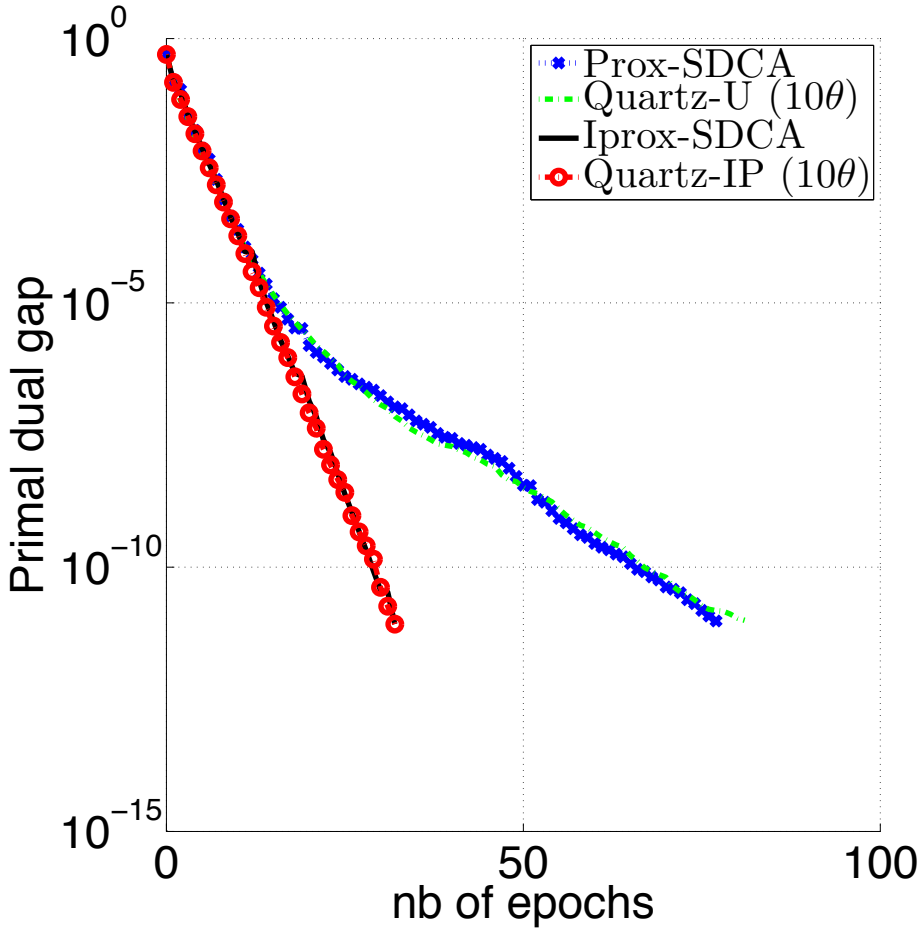
5.5
Updating One Dual Variable at a Time

Complexity of Quartz specialized to serial sampling

Optimal sampling	$n + \frac{\frac{1}{n}\sum_{i=1}^{n} L_i}{\lambda\gamma}$
Uniform sampling	$n + \frac{\max_i L_i}{\lambda\gamma}$

$$L_i \equiv \lambda_{\max}\left(A_i^\top A_i\right)$$

Experiment: Quartz vs SDCA, uniform vs optimal sampling



$$\text{Data} = \texttt{cov1}, \quad n = 522,911, \quad \lambda = 10^{-6}$$

6. An Efficient Primal Method for ERM

S. Shalev-Shwartz
SDCA without Duality, *NIPS 2015* (arXiv:1502.06177)

Dominik Csiba and P.R.
Primal method for ERM with flexible mini-batching schemes and non-convex losses, *arXiv:1506.02227,* 2015

6.1
Empirical Risk Minimization

Primal Problem: ERM

$$\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$$

$\frac{1}{\gamma}$-smooth and convex

regularization parameter

$$\min_{w \in \mathbb{R}^d} \left[P(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \frac{\lambda}{2} \|w\|_2^2 \right]$$

d = # features (parameters)

n = # samples

$A_i \in \mathbb{R}^{d \times m}$

We had a general 1-strongly convex function *g* here before

Assumption

The loss functions $\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$ are $\frac{1}{\gamma}$-smooth:

$$\|\nabla\phi_i(a) - \nabla\phi_i(a')\| \leq \frac{1}{\gamma}\|a - a'\|, \quad a, a' \in \mathbb{R}^m$$

Lipschitz constant of the gradient of the function

Dual Problem

$$D(\alpha) \equiv -\lambda$$

$$g^*(w') = \max_{w \in \mathbb{R}^d} \{(w')^\top w$$

1 – smooth & convex

$$\alpha = (\alpha_1, \quad \in \mathbb{R}^m \quad \in \mathbb{R}^m$$

Goal: An efficient algorithm which naturally operates in the primal space (i.e., on the primal problem) only

The method will have the "same" theoretical guarantee as Quartz

The computer lab will be based on this

6.2
The Algorithm

Motivation I

w^* is optimal

$$0 = \nabla P(w^*) = \left(\frac{1}{n} \sum_{i=1}^{n} A_i \nabla \phi_i(A_i^\top w^*) \right) + \lambda w^*$$

$$w^* = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^*$$

$$\alpha_i^* := -\nabla \phi_i(A_i^\top w^*)$$

Motivation II

Algorithmic Ideas:

1. Simultaneously search for both w^* and $\alpha_1^*, \ldots, \alpha_n^*$

2. Try to do "something like"

$$\alpha_i^{t+1} \leftarrow -\nabla \phi_i(A_i^\top w^t)$$

3. Maintain the relationship

$$w^t = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^t$$

Does not quite work: too "greedy"

The Algorithm: dfSDCA

STEP 0: INITIALIZE

Choose $\alpha_1^0, \ldots, \alpha_n^0 \in \mathbb{R}^m$

Initialize the relationship

$$w^0 = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^0$$

STEP 1: "DUAL" UPDATE

Choose a random set S_t of "dual variables"

For $i \in S_t$ do

Controlling "greed" by taking a convex combination

$$\alpha_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right) \alpha_i^t + \frac{\theta}{p_i} \left(-\nabla \phi_i(A_i^\top w^t)\right)$$

$$\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$$

$$p_i = \mathbf{P}(i \in S_t)$$

STEP 2: PRIMAL UPDATE

$$w^{t+1} \leftarrow w^t + \sum_{i \in S_t} \frac{\theta}{n \lambda p_i} A_i \left(-\nabla \phi_i(A_i^\top w^t) + \alpha_i^t\right)$$

This is just maintaining the relationship

6.3
Complexity

ESO Assumption (same as before!)

Parameters $v_1, \ldots, v_n$ satisfy:

$$\mathbf{E}\left\|\sum_{i\in S_t} A_i\alpha_i\right\|^2 \leq \sum_{i=1}^{n} p_i v_i \|\alpha_i\|^2$$

inequality must hold for all

$$\alpha_1, \ldots, \alpha_n \in \mathbb{R}^m$$

$$p_i = \mathbf{P}(i \in S_t)$$

Complexity

Theorem [Csiba & R '15]

A constant depending on $P, w^0, \alpha_i^0, w^*, \alpha_i^*$

$$t \geq \max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i \lambda \gamma n} \right) \log \left(\frac{C}{\epsilon} \right)$$

$p_i = \mathbf{P}(i \in S_t)$

$$\mathbf{E}\left[P(w^t) - P(w^*)\right] \leq \epsilon$$

6.4
Experiments

Some More Efficient Primal Methods for ERM: SAG, SVRG and S2GD

SAG: Stochastic Average Gradient

N. Le Roux, M. Schmidt, and F. Bach. **A stochastic gradient method with an exponential convergence rate for finite training sets**. *NIPS*, 2012

SVRG: Stochastic Variance Reduced Gradient

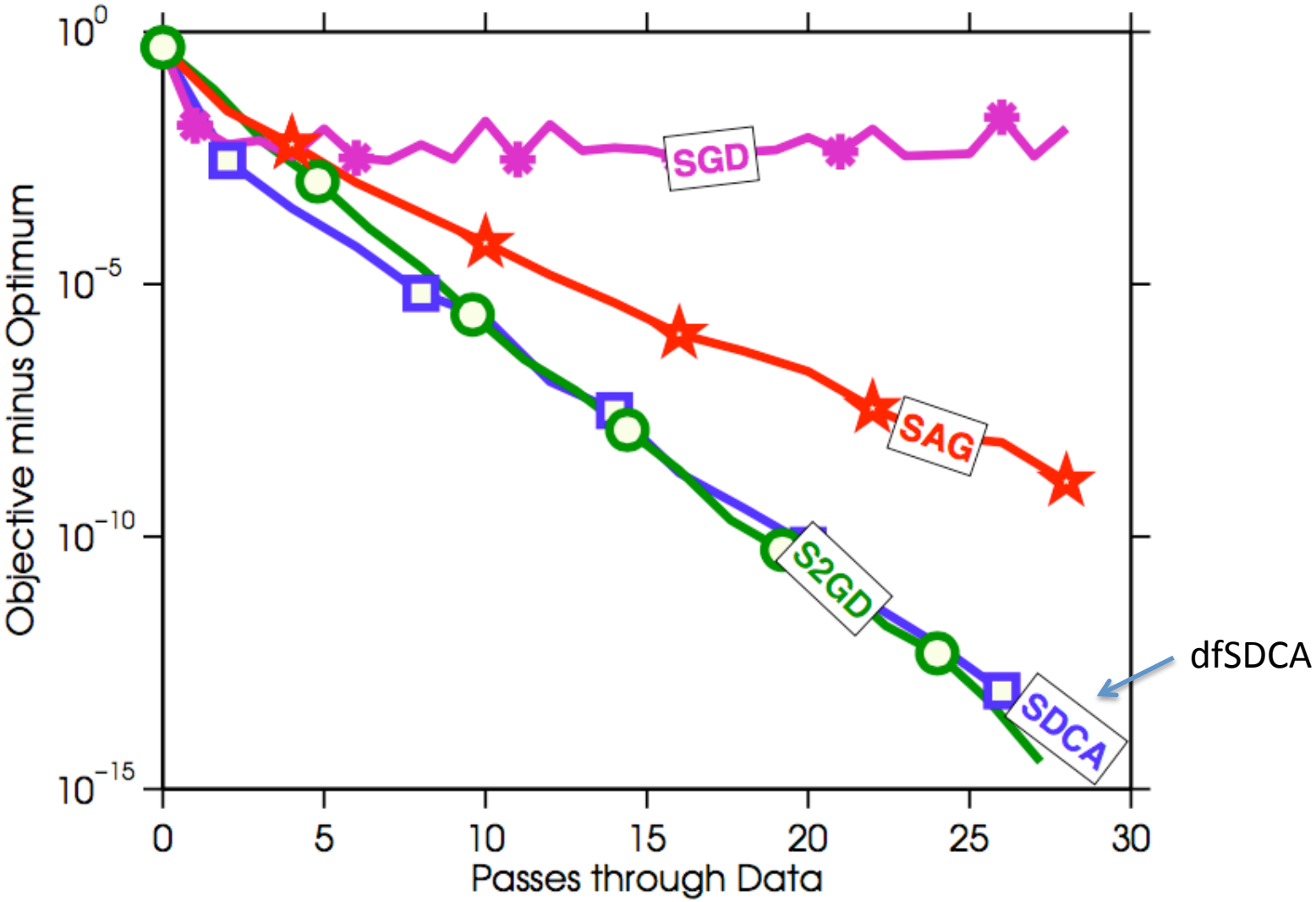
Rie Johnson and Tong Zhang. **Accelerating stochastic gradient descent using predictive variance reduction.** *NIPS*, 2013.

S2GD: Semi-Stochastic Gradient Descent

J. Konečný and P. R. **Semi-stochastic gradient descent methods.** *arXiv:1312.1666*, 2013

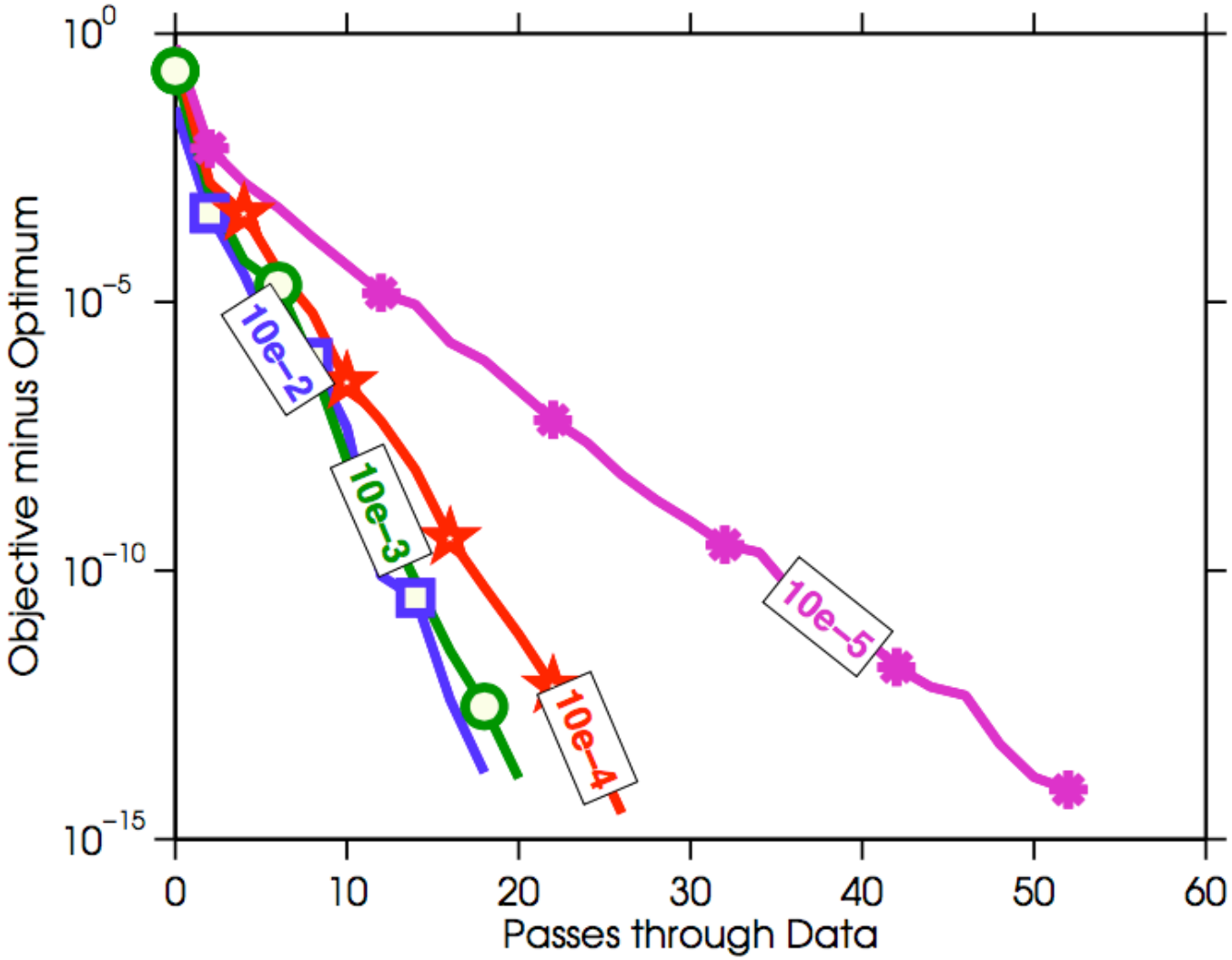
Modern Methods for ERM vs SGD

Dataset: rcv1 (n = 20,241 ; d = 47,232)



Behavior of dfSDCA for various λ

Dataset: rcv1 (n = 20,241 ; d = 47,232)

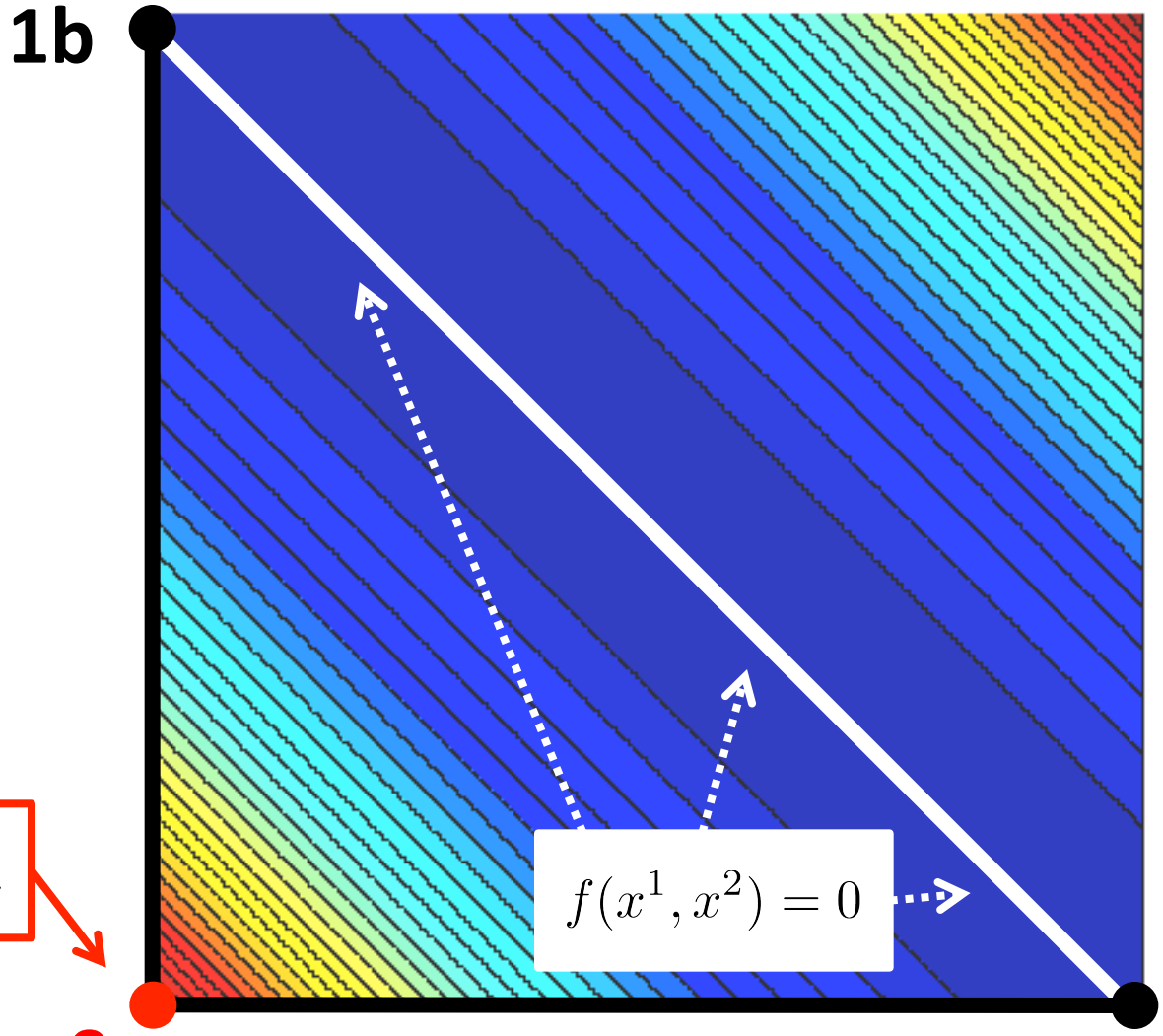


7. Parallelization (Minibatching)

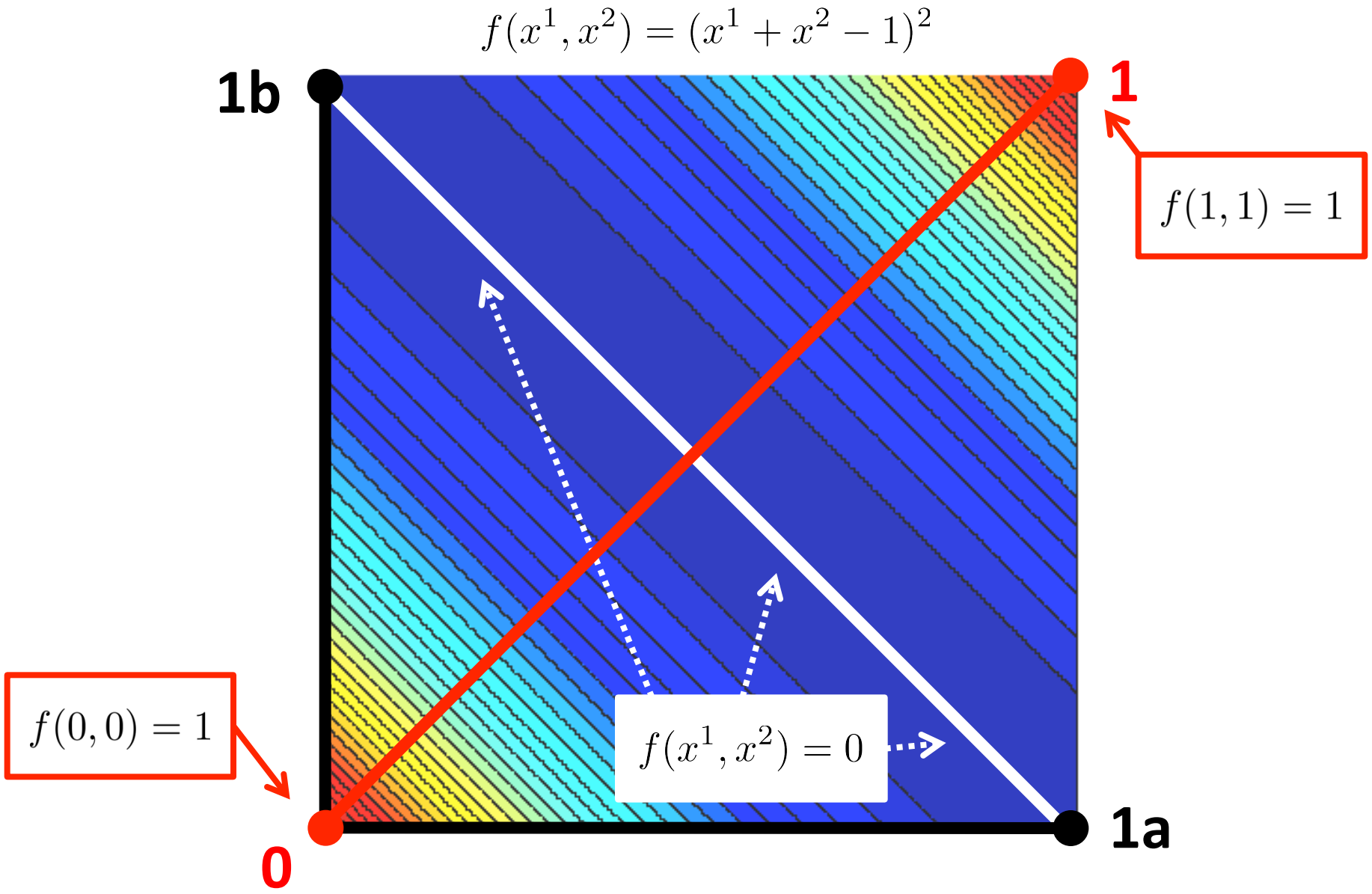
7.1 NAIVE APPROACH

Failure of naive parallelization

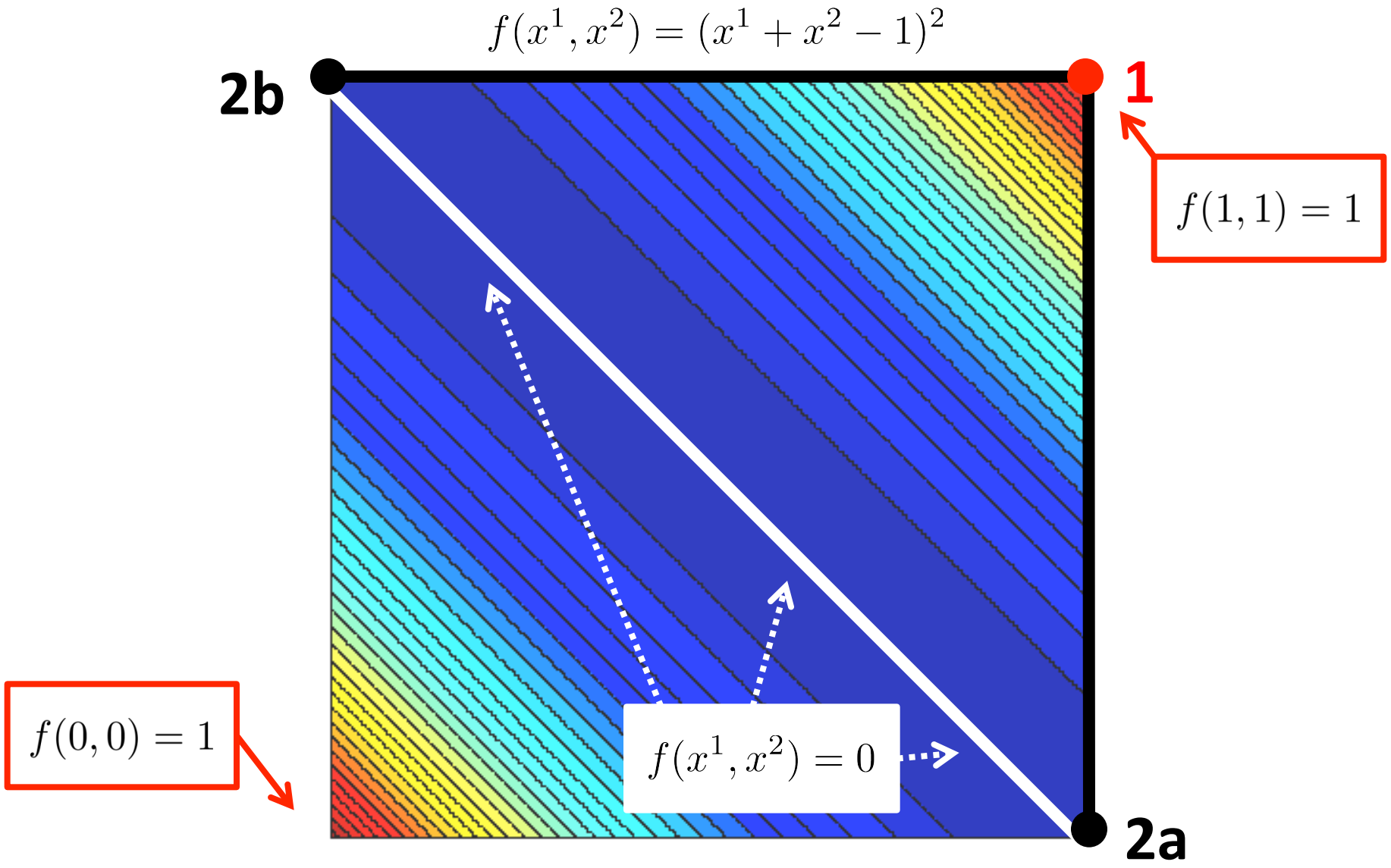
$$f(x^1, x^2) = (x^1 + x^2 - 1)^2$$



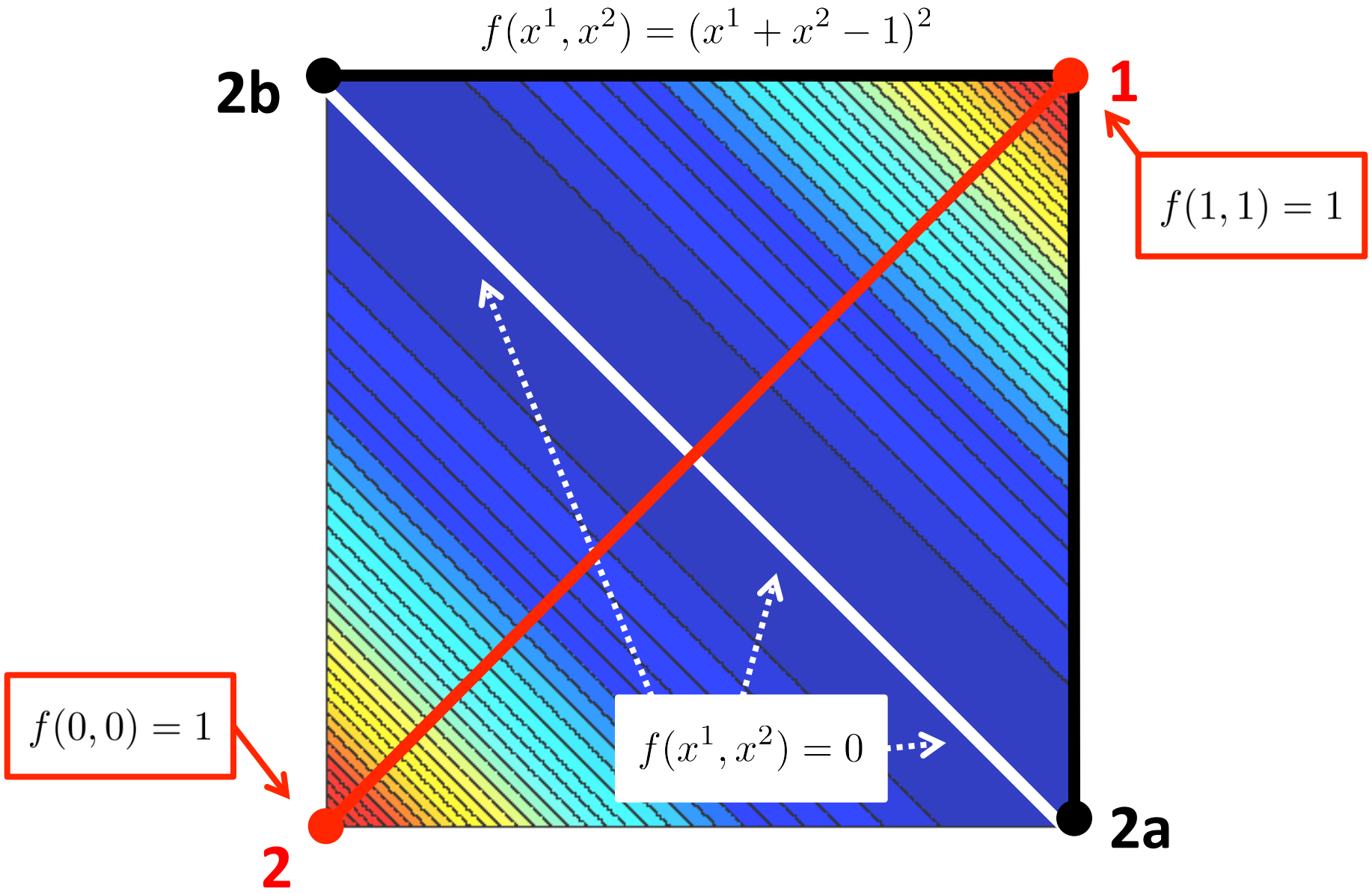
Failure of naive parallelization



Failure of naive parallelization

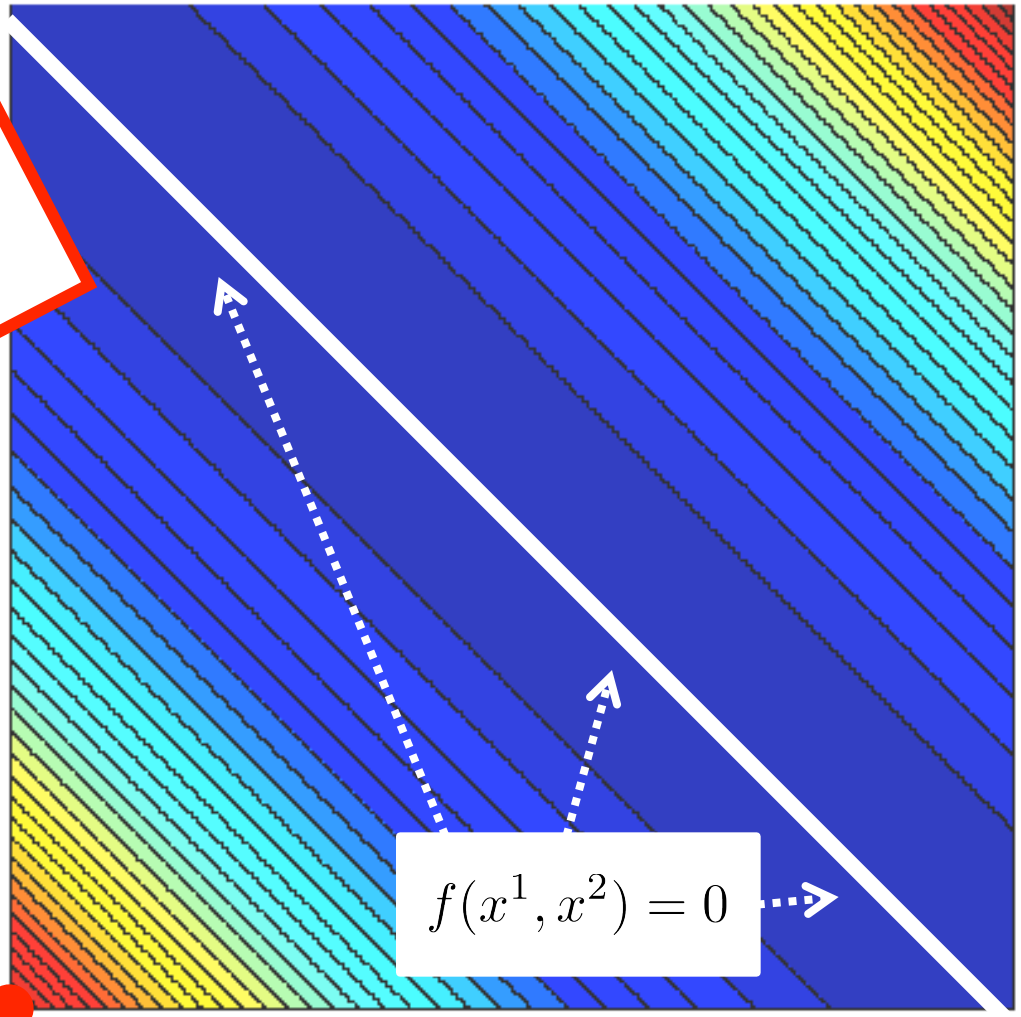


Failure of naive parallelization



Failure of naive parallelization

$$f(x^1, x^2) = (x^1 + x^2 - 1)^2$$



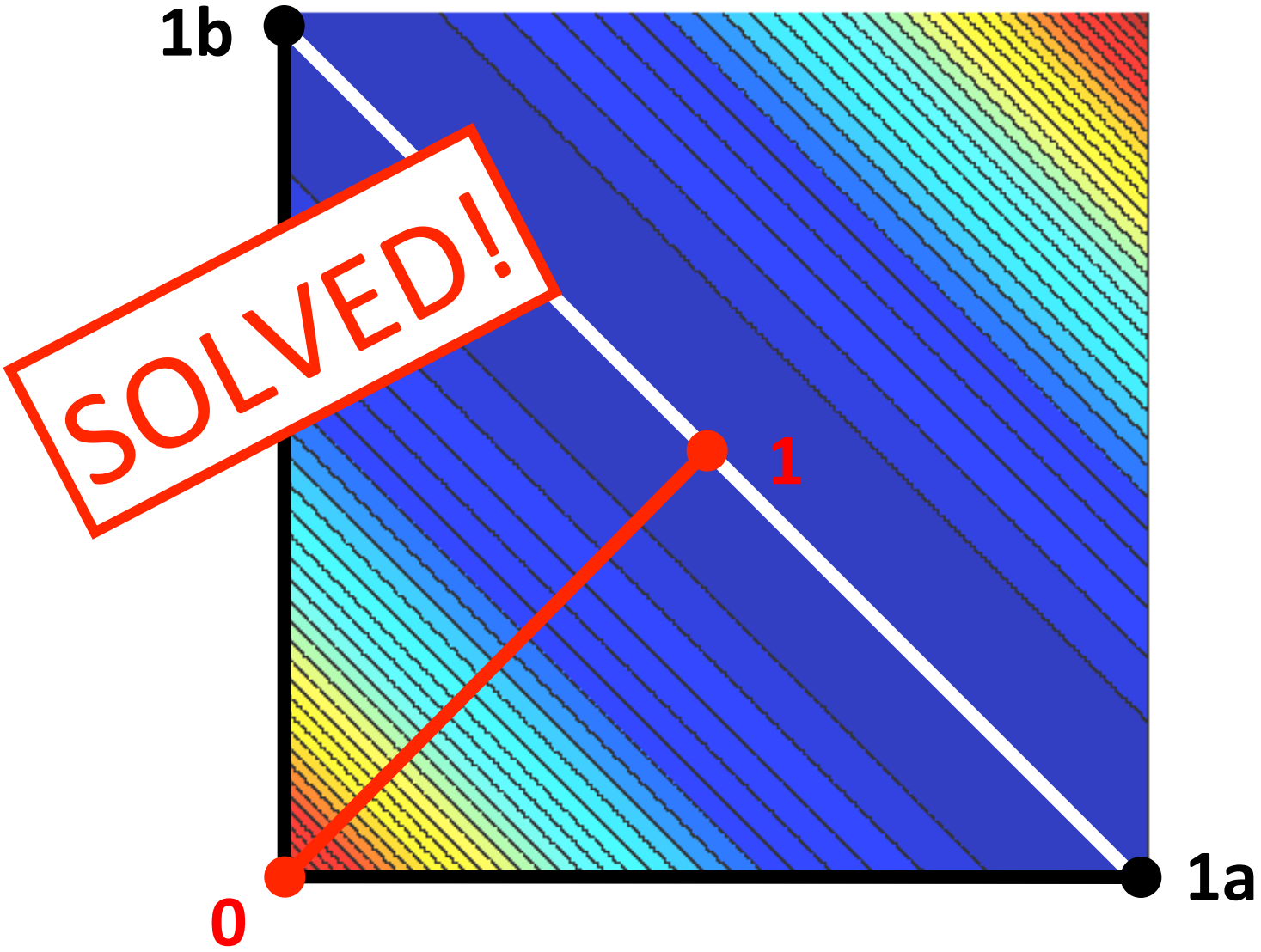
OOPS!

$f(1,1) = 1$

$f(0,0) = 1$

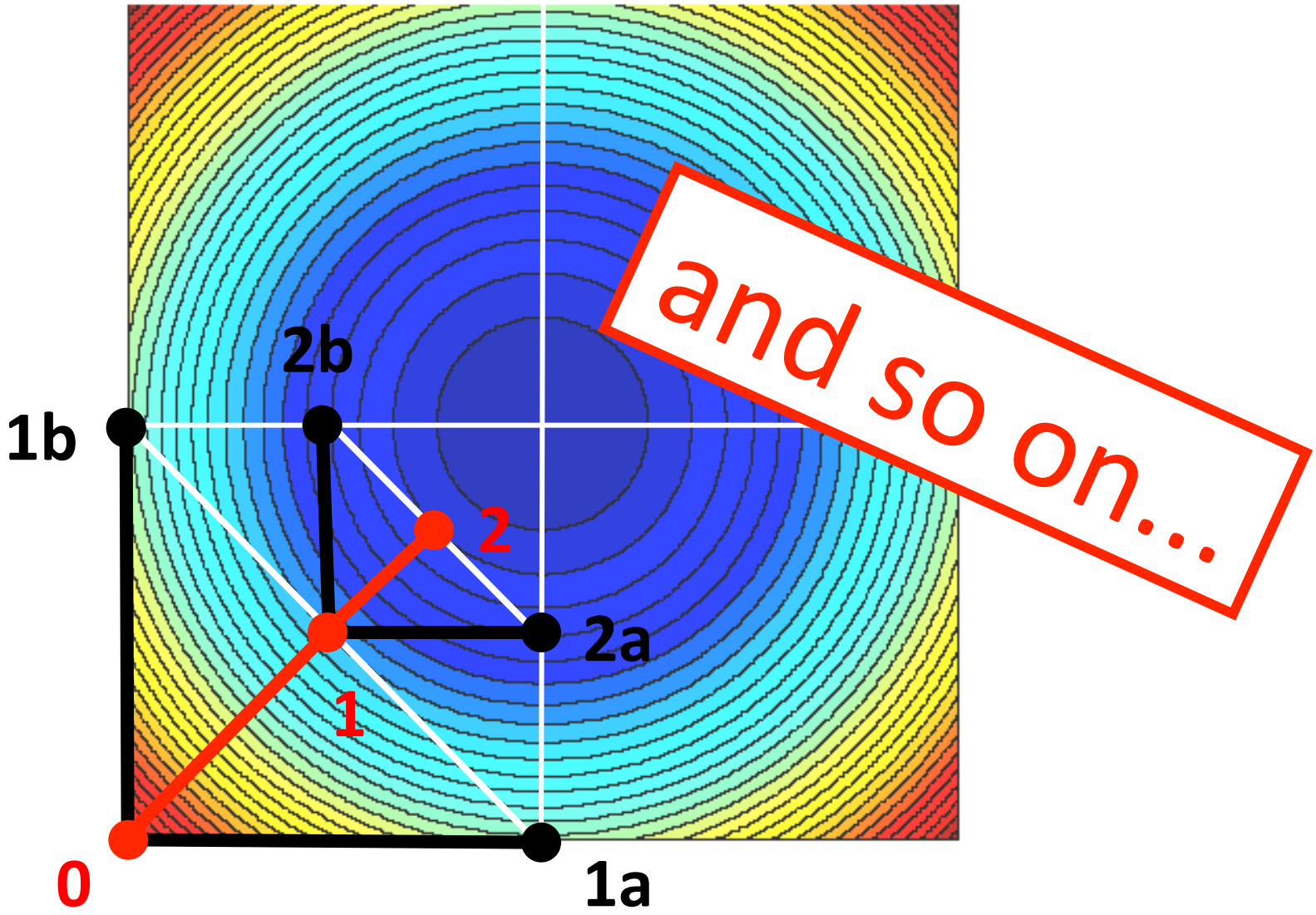
2

Idea: averaging updates may help



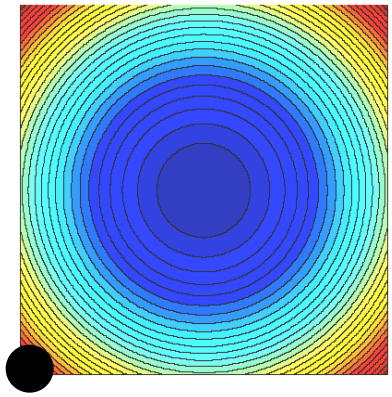
Averaging can be too conservative

$$f(x^1, x^2) = (x^1 - 1)^2 + (x^2 - 1)^2$$



Averaging can be too conservative

$$f(x) = (x^1 - 1)^2 + (x^2 - 1)^2 + \cdots + (x^n - 1)^2$$



$x_0 = 0 \quad f(x_0) = n$

BAD!!!

$$k \geq \frac{n}{2} \log\left(\frac{n}{\epsilon}\right)$$

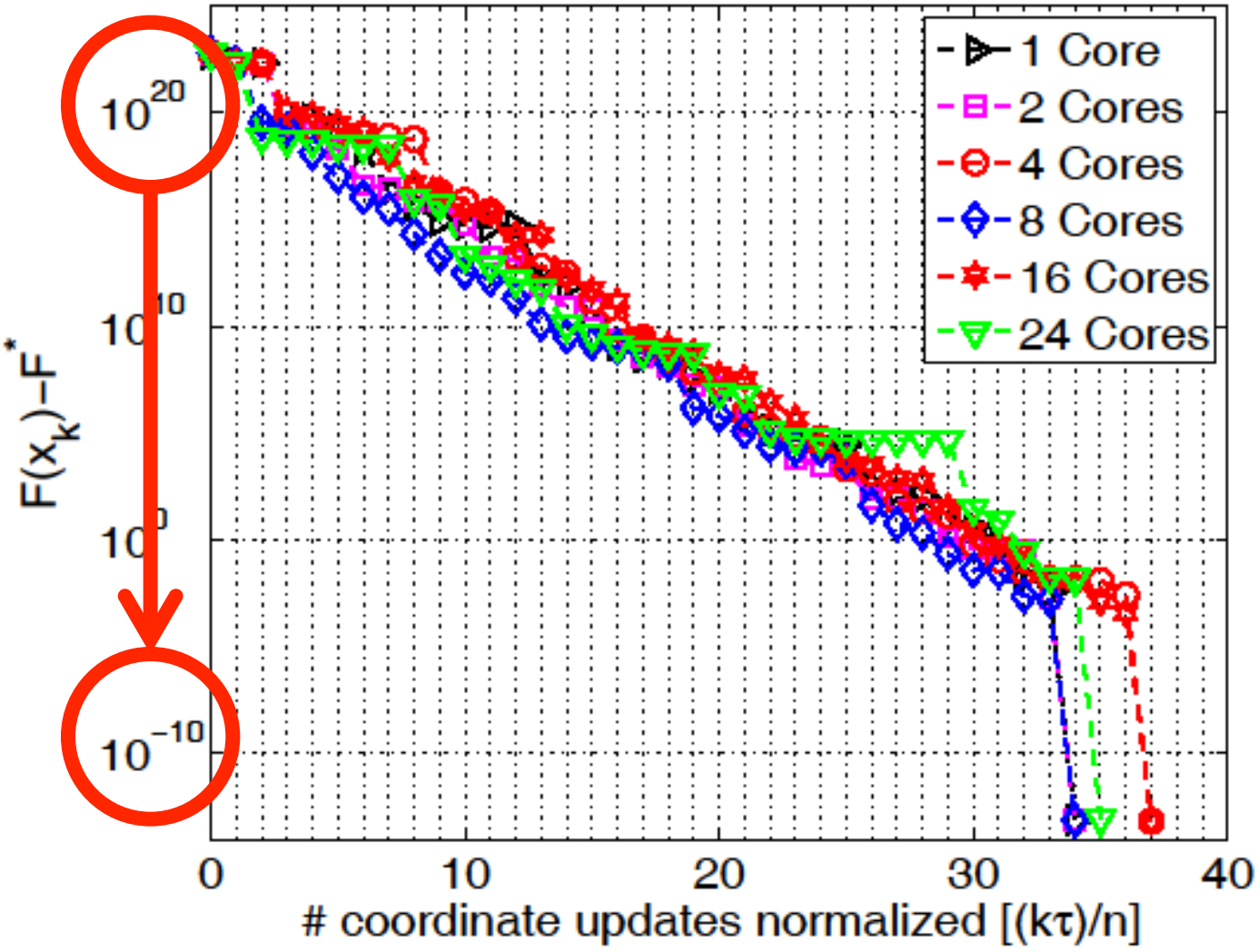
$$f(x_k) = n\left(1 - \frac{1}{n}\right)^{2k} \leq \epsilon$$

WANT

7.2
Experiment with a 1 billion-by-2 billion LASSO problem

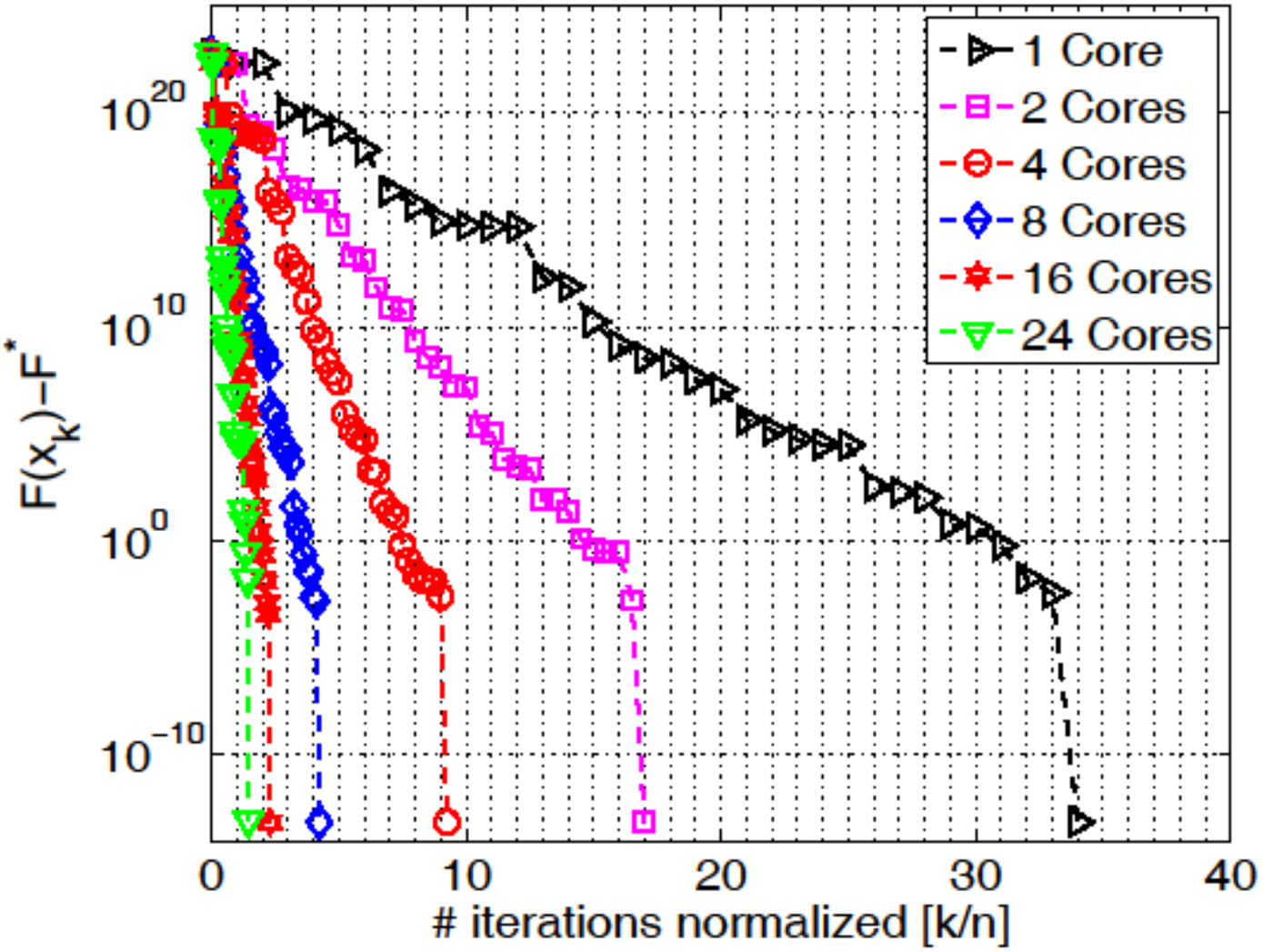
P.R. and Martin Takáč
Parallel coordinate descent methods for big data optimization
Mathematical Programming, 2015 (arXiv:1212.0873)

Coordinate Updates



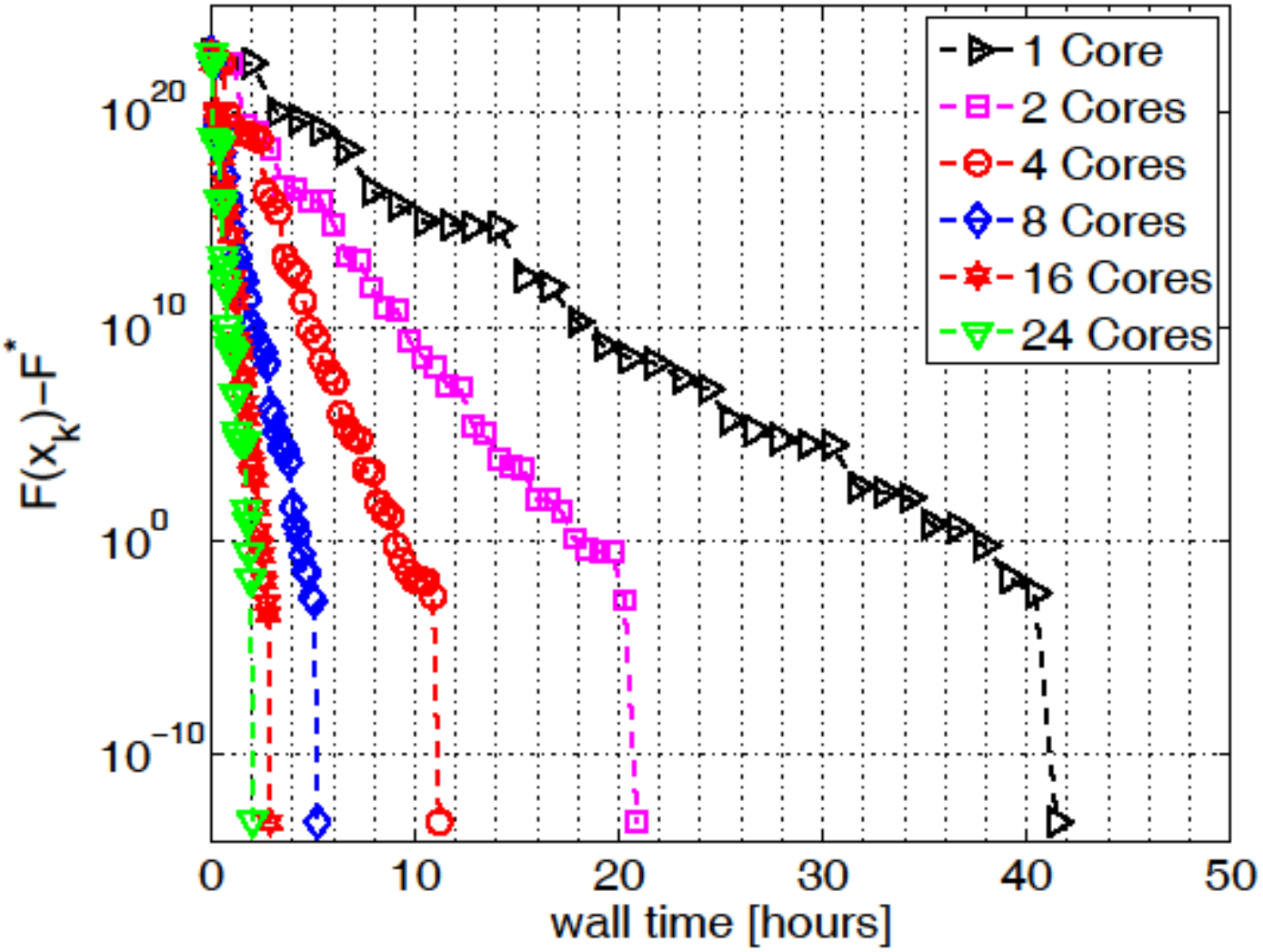
LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $n = 10^9$ and $m = 2 \times 10^9$

Iterations



LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $n = 10^9$ and $m = 2 \times 10^9$

Wall Time



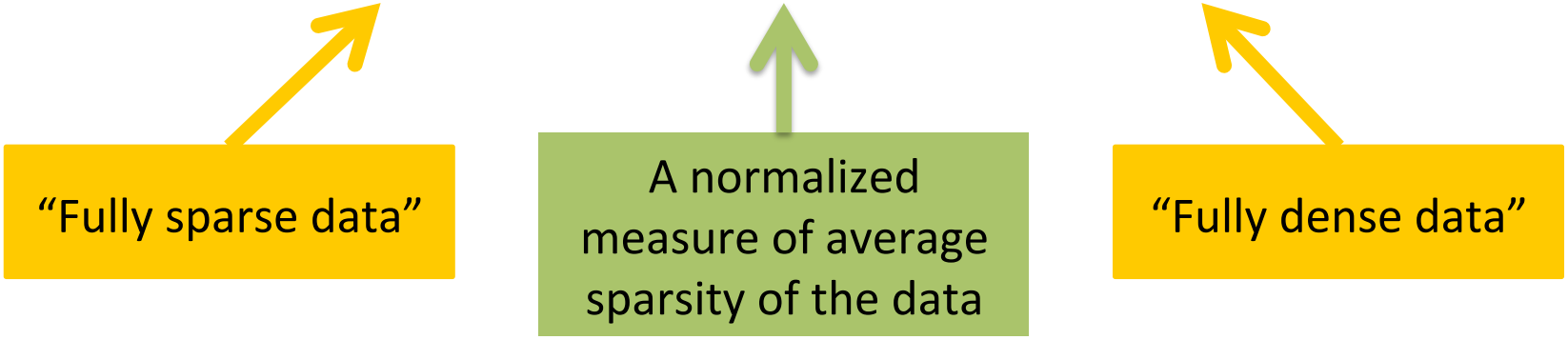
LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $n = 10^9$ and $m = 2 \times 10^9$

7.3
Minibatching & Quartz

[Qu, R & Zhang 14]

Data Sparsity

$$1 \leq \tilde{\omega} \leq n$$



Quartz [Qu, R & Zhang 14]

Complexity of Quartz

Fully sparse data ($\tilde{\omega} = 1$)	$\frac{n}{\tau} + \frac{\max_i L_i}{\lambda\gamma\tau}$
Fully dense data ($\tilde{\omega} = n$)	$\frac{n}{\tau} + \frac{\max_i L_i}{\lambda\gamma}$
Any data ($1 \le \tilde{\omega} \le n$)	$\frac{n}{\tau} + \frac{\left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{n-1}\right) \max_i L_i}{\lambda\gamma\tau} \equiv T(\tau)$

Speedup

Assume the data is normalized: $L_i \equiv \lambda_{\max}(A_i^\top A_i) \leq 1$

Then:

$$T(\tau) \quad = \quad \frac{\left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{(n-1)(1+\lambda\gamma n)}\right)}{\tau} \times T(1)$$

Linear speedup up to a certain data-independent minibatch size:

$$\tau \leq 2 + \lambda\gamma n \quad \Rightarrow \quad T(\tau) \leq \frac{2}{\tau} \times T(1)$$

Further data-dependent speedup, up to the extreme case:

$$\tilde{\omega} = \mathcal{O}(\lambda\gamma n) \quad \Rightarrow \quad T(\tau) = \mathcal{O}\left(\frac{T(1)}{\tau}\right)$$

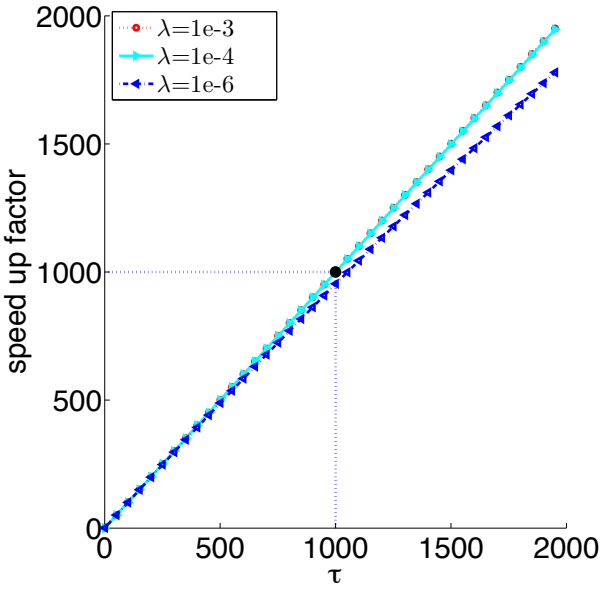
Quartz: Parallelization Speedup

examples: $n = 10^6$

Smoothness of loss functions: $\gamma = 1$

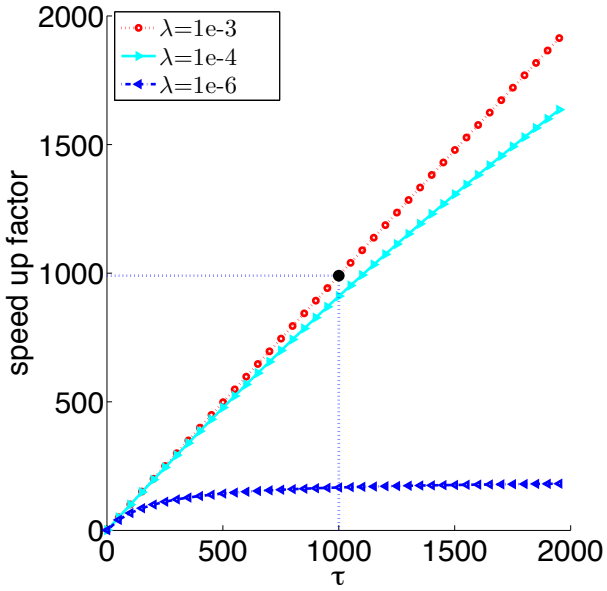
Low regularization: $\lambda = 1/n$

High regularization: $\lambda = 1/\sqrt{n}$



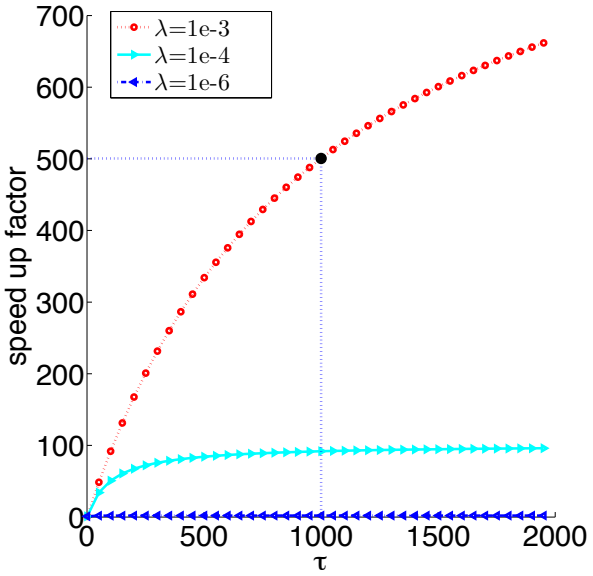
Sparse Data

$\tilde{\omega} = 10^2$



Denser Data

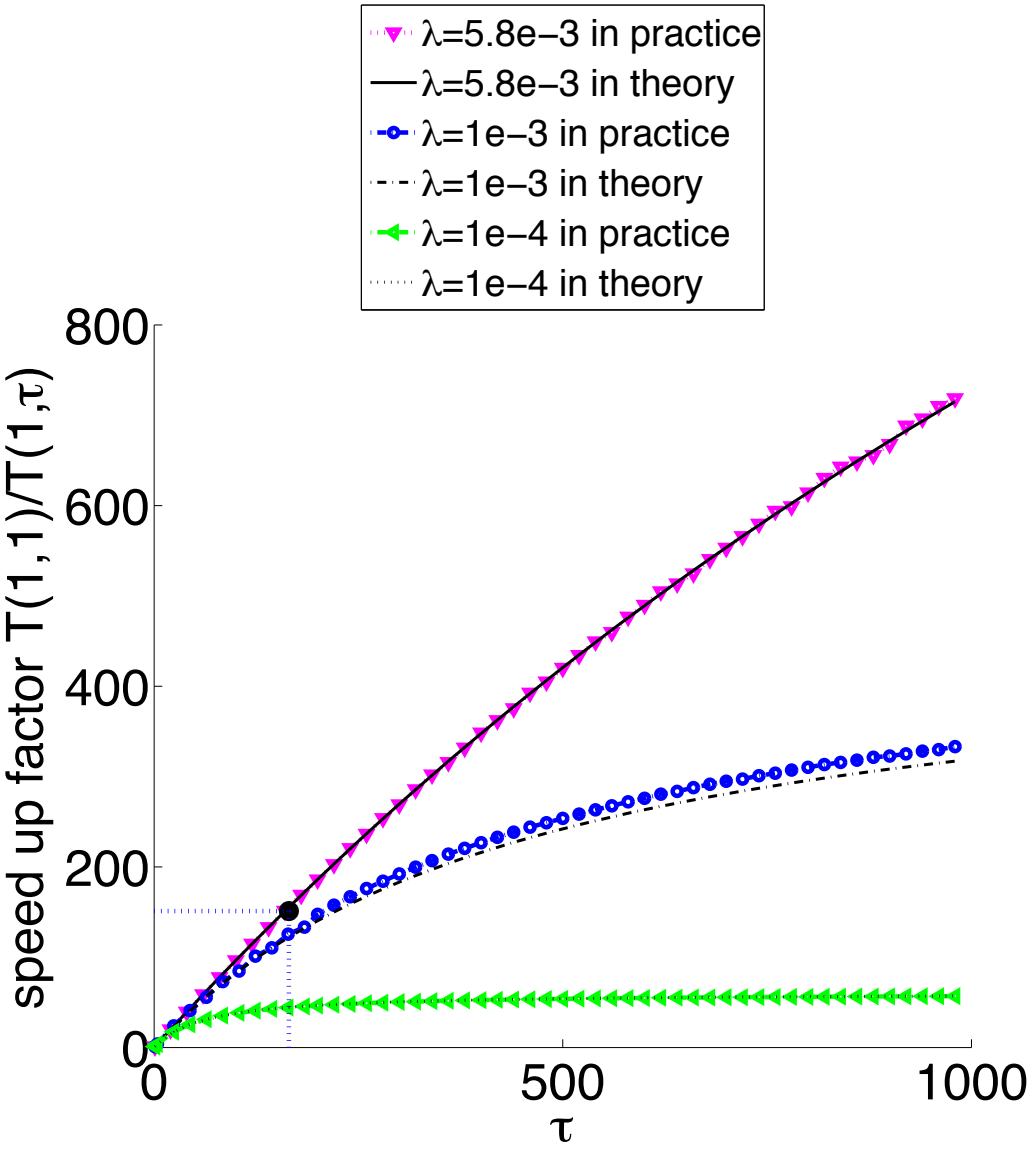
$\tilde{\omega} = 10^4$



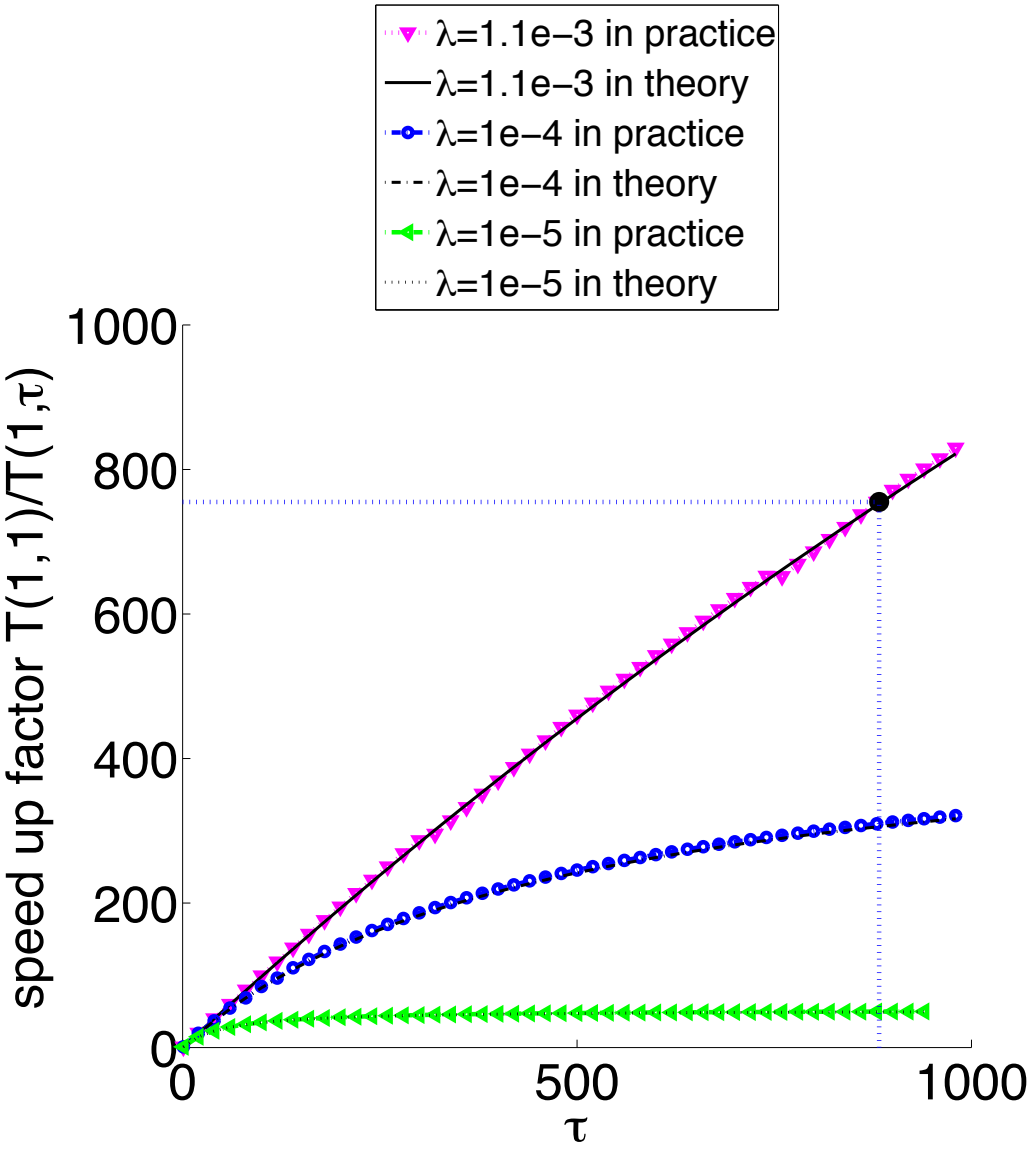
Fully Dense Data

$\tilde{\omega} = 10^6$

astro_ph: n = 29,882 density = 0.08%



CCAT: n = 781,265 density = 0.16%



Primal-dual methods with tau-nice sampling

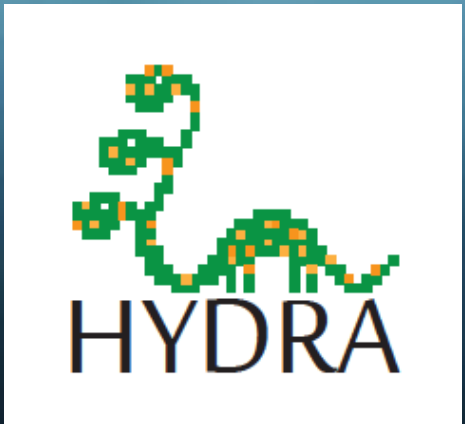
$L_i = 1$

Algorithm	Iteration complexity	g
SDCA [S-Shwartz & Zhang 12]	$n + \frac{1}{\lambda\gamma}$	$\frac{1}{2}\|\cdot\|^2$
ASDCA [S-Shwartz & Zhang 13a]	$4 \times \max\left\{\frac{n}{\tau}, \sqrt{\frac{n}{\lambda\gamma\tau}}, \frac{1}{\lambda\gamma\tau}, \frac{n^{\frac{1}{3}}}{(\lambda\gamma\tau)^{\frac{2}{3}}}\right\}$	$\frac{1}{2}\|\cdot\|^2$
SPDC [Zhang & Xiao 14]	$\frac{n}{\tau} + \sqrt{\frac{n}{\lambda\gamma\tau}}$	general
Quartz	$\frac{n}{\tau} + \left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{n-1}\right)\frac{1}{\lambda\gamma\tau}$	general

For sufficiently sparse data, Quartz wins even when compared against accelerated methods

Algorithm	$\gamma\lambda n = \Theta(\frac{1}{\tau})$	$\gamma\lambda n = \Theta(1)$	$\gamma\lambda n = \Theta(\tau)$	$\gamma\lambda n = \Theta(\sqrt{n})$
	$\kappa = n\tau$	$\kappa = n$	$\kappa = n/\tau$	$\kappa = \sqrt{n}$
SDCA	$n\tau$	n	n	n
Accelerated: ASDCA	n	$\frac{n}{\sqrt{\tau}}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{n^{3/4}}{\sqrt{\tau}}$
Accelerated: SPDC	n	$\frac{n}{\sqrt{\tau}}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{n^{3/4}}{\sqrt{\tau}}$
Quartz	$n + \tilde{\omega}\tau$	$\frac{n}{\tau} + \tilde{\omega}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{\tilde{\omega}}{\sqrt{n}}$

8. Distributed Optimization



References

P.R. and Martin Takáč. **Distributed coordinate descent for learning with big data.** *arXiv:1310.2059*, 2013

Olivier Fercoq, Zheng Qu, P.R. and Martin Takáč. **Fast distributed coordinate descent for minimizing non-strongly convex losses.** *In 2014 IEEE International Workshop on Machine Learning for Signal Processing*, 2014

Zheng Qu, P.R. and Tong Zhang. **Randomized dual coordinate ascent with arbitrary sampling.** *In NIPS 2015 (arXiv:1411.5873)*

8.1
Distributed Quartz

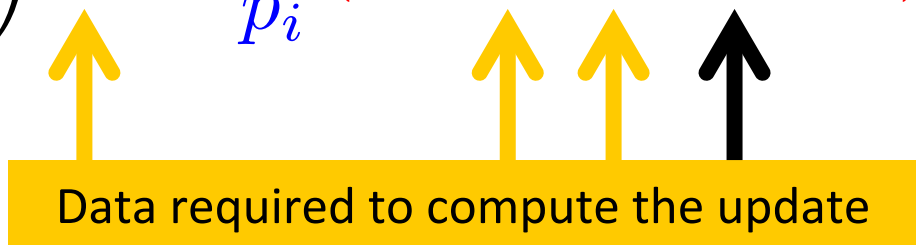
Distributed Quartz: Perform the Dual Updates in a Distributed Manner

Quartz STEP 2: DUAL UPDATE

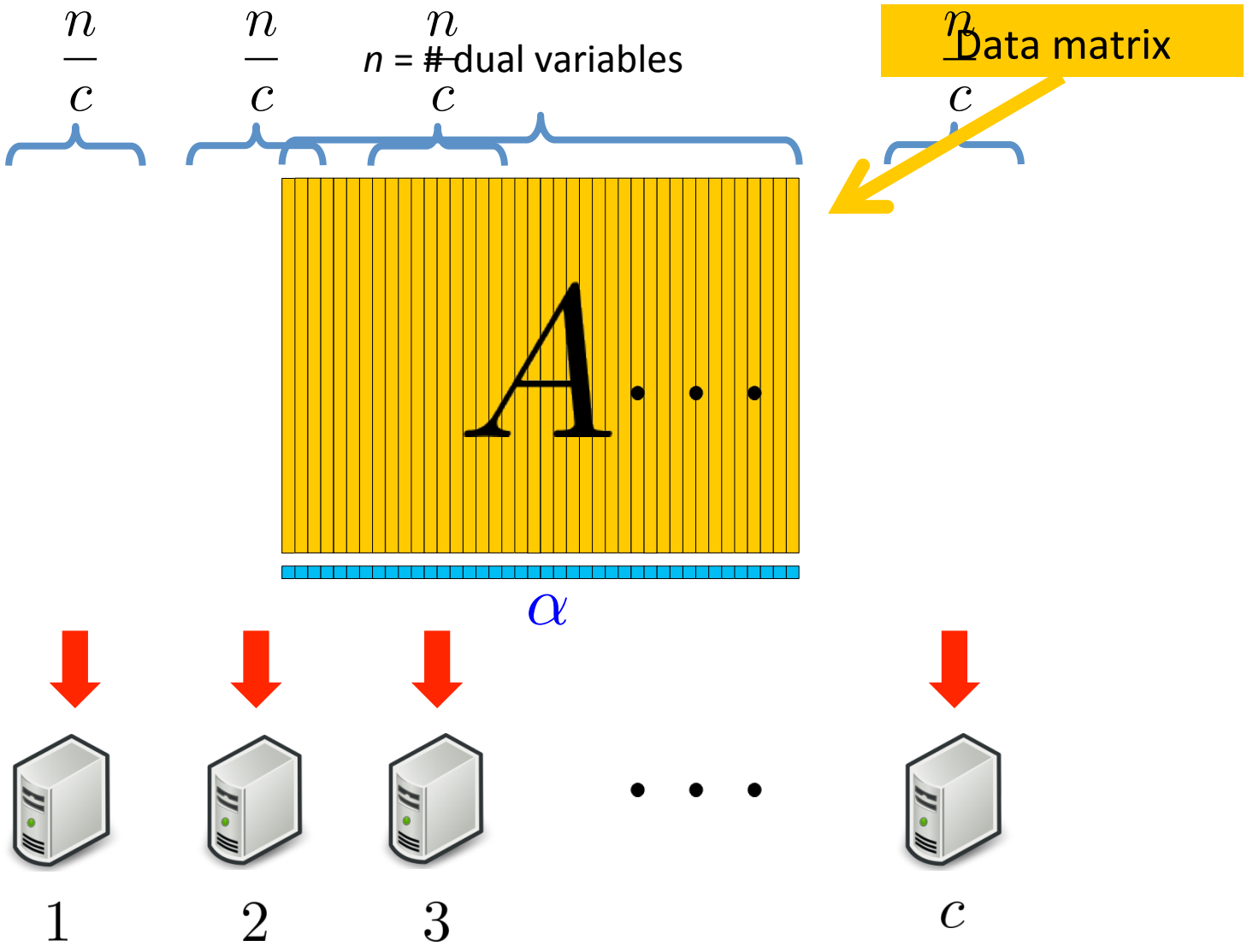
Choose a random set S_t of dual variables

For $i \in S_t$ do

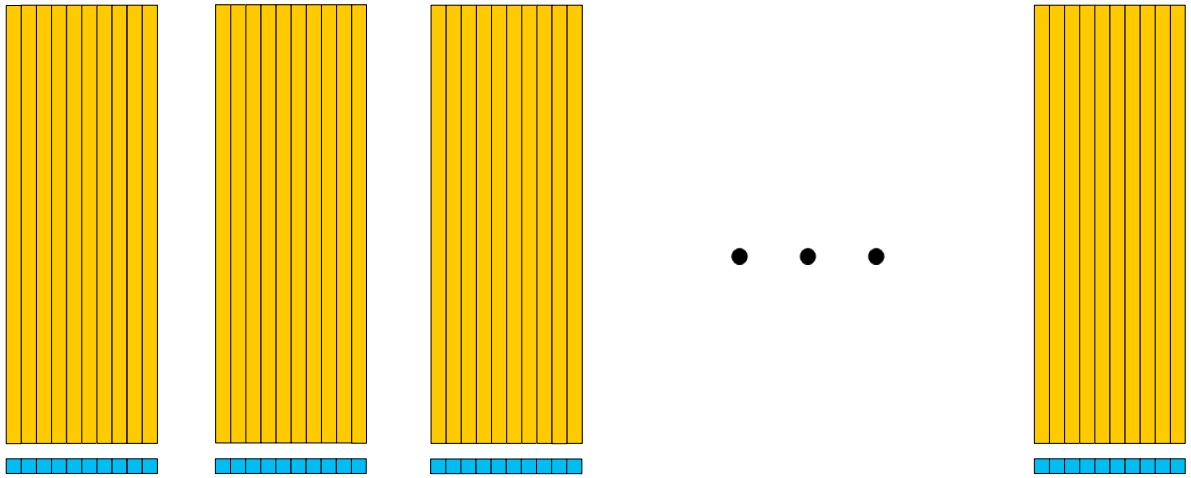
$$\alpha_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right)\alpha_i^t + \frac{\theta}{p_i}\left(-\nabla\phi_i(A_i^\top w^{t+1})\right)$$



Distribution of Data



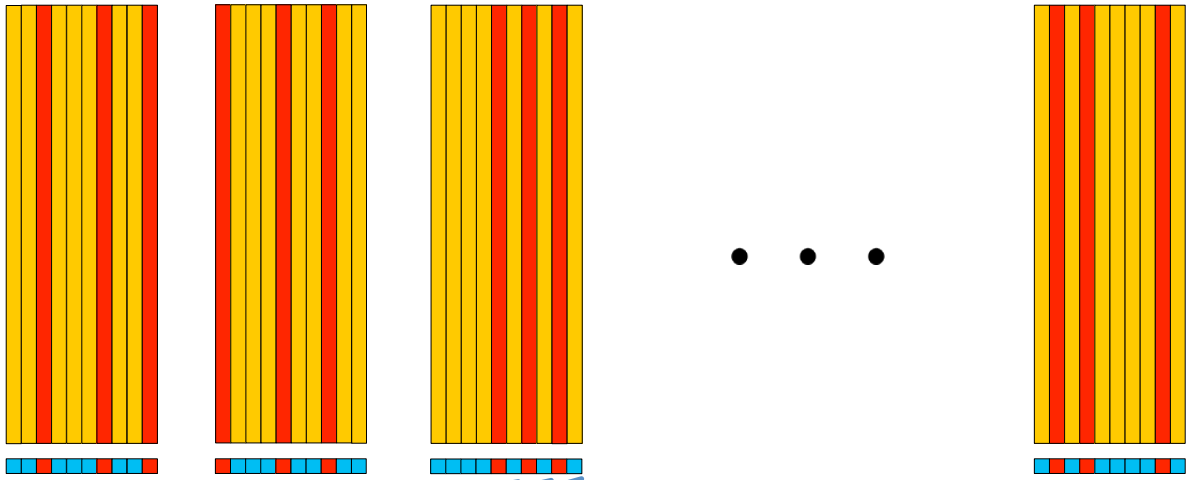
Distributed sampling



Distributed sampling

Each node independently picks τ dual variables from those it owns, uniformly at random

$\tau = 3$



$$\hat{S} = \{3, 7, 10\} \cup \{11, 15, 18\} \cup \{25, 27, 29\} \cup \cdots$$

Random set of dual variables

Also see: CoCoA+ [Ma, Smith, Jaggi et al 15]

8.2
Complexity

Complexity of Distributed Quartz

Key: Get the right stepsize parameters v (so that the ESO inequality holds)

The leading term in the complexity bound then is:

$$\max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i \lambda \gamma n} \right)$$

$$||$$

$$\frac{n}{c\tau} + \frac{\text{Something that looks complicated}}{\lambda \gamma c \tau}$$

$$||$$

$$\frac{n}{c\tau} + \max_i \frac{\lambda_{\max}\left(\sum_{j=1}^{d} \left(1 + \frac{(\tau-1)(\omega_j - 1)}{\max\{n/c-1,1\}} + \left(\frac{\tau c}{n} - \frac{\tau - 1}{\max\{n/c-1,1\}} \right) \frac{\omega_j' - 1}{\omega_j'} \omega_j \right) A_{ji}^\top A_{ji} \right)}{\lambda \gamma c \tau}$$

8.3
Experiments

Experiment

Machine: 128 nodes of Hector Supercomputer (4096 cores)

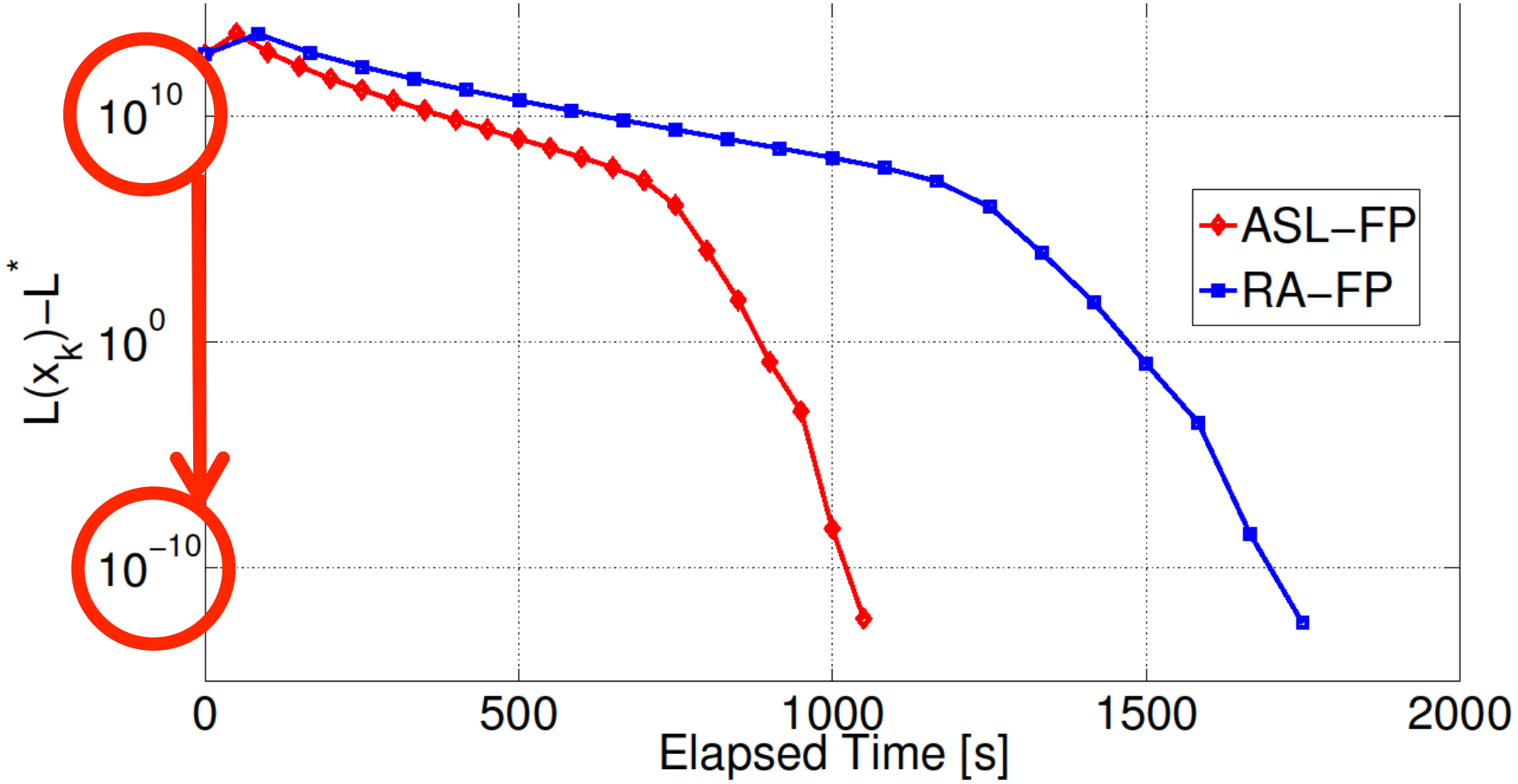
Problem: LASSO, n = 1 billion, d = 0.5 billion, 3 TB

Algorithm: 

with c = 512

P.R. and Martin Takáč. **Distributed coordinate descent for learning with big data.** *arXiv:1310.2059*, 2013

LASSO: 3TB data + 128 nodes



Experiment

Machine: 128 nodes of Archer Supercomputer

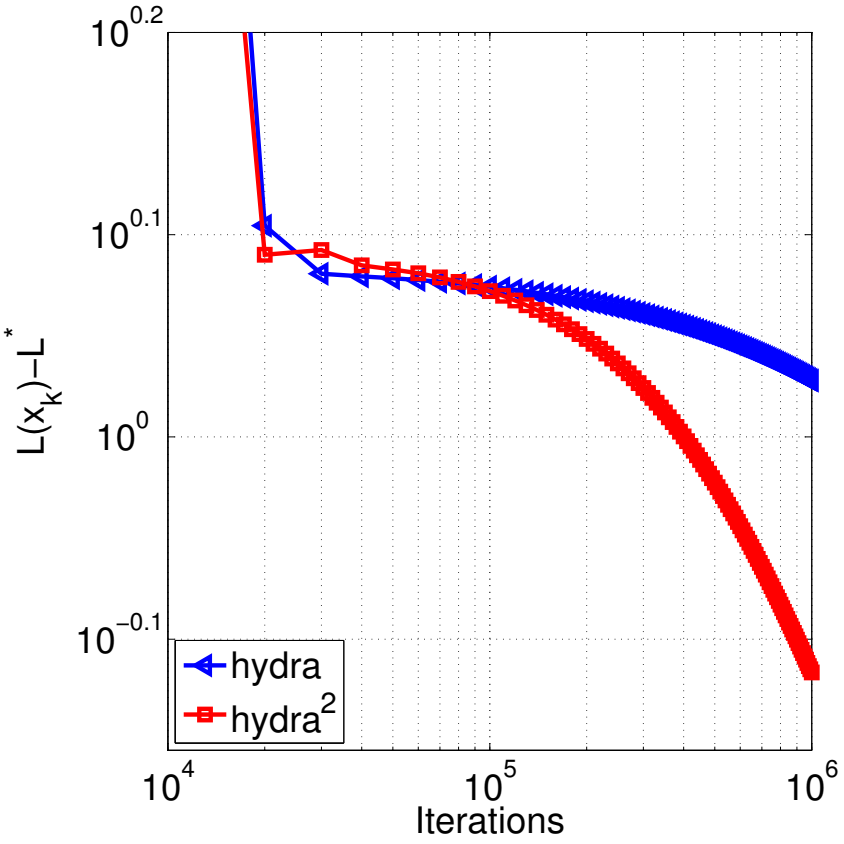
Problem: LASSO, n = 5 million, d = 50 billion, 5 TB
(60,000 nnz per row of A)

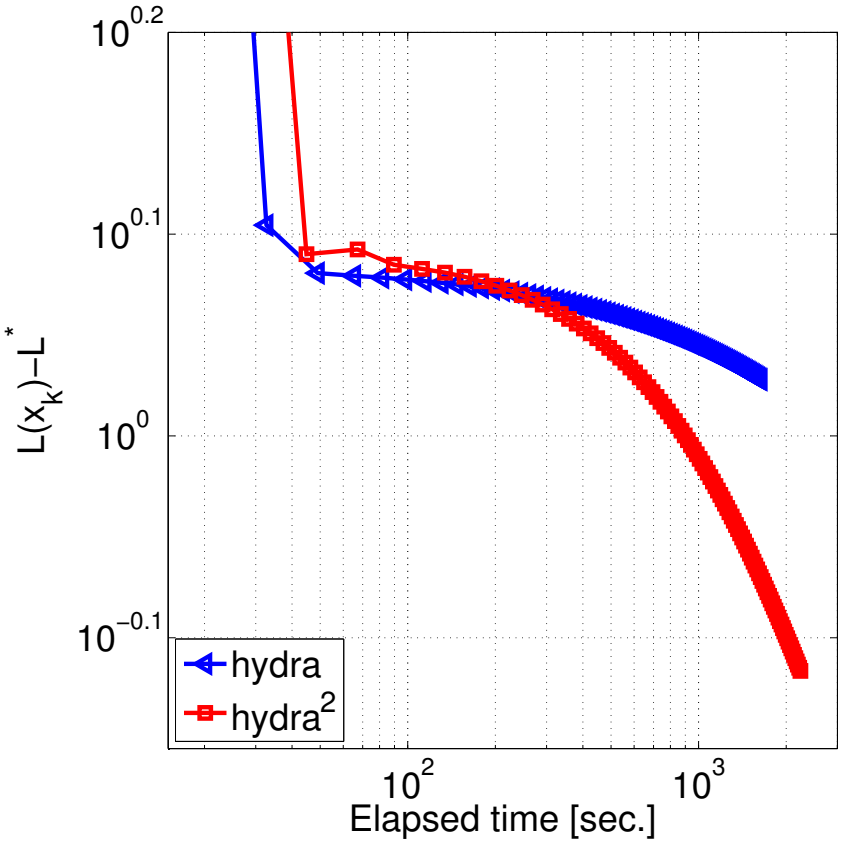
Algorithm 

with c = 256

Olivier Fercoq, Zheng Qu, P.R. and Martin Takáč. **Fast distributed coordinate descent for minimizing non-strongly convex losses.** *In 2014 IEEE International Workshop on Machine Learning for Signal Processing*, 2014

LASSO: 5TB data (d = 50 billion)
128 nodes





9. Curvature

SDNA

Zheng Qu, P.R., Martin Takáč and Olivier Fercoq
SDNA: Stochastic Dual Newton Ascent for empirical risk minimization
arXiv:1502.02268, 2015

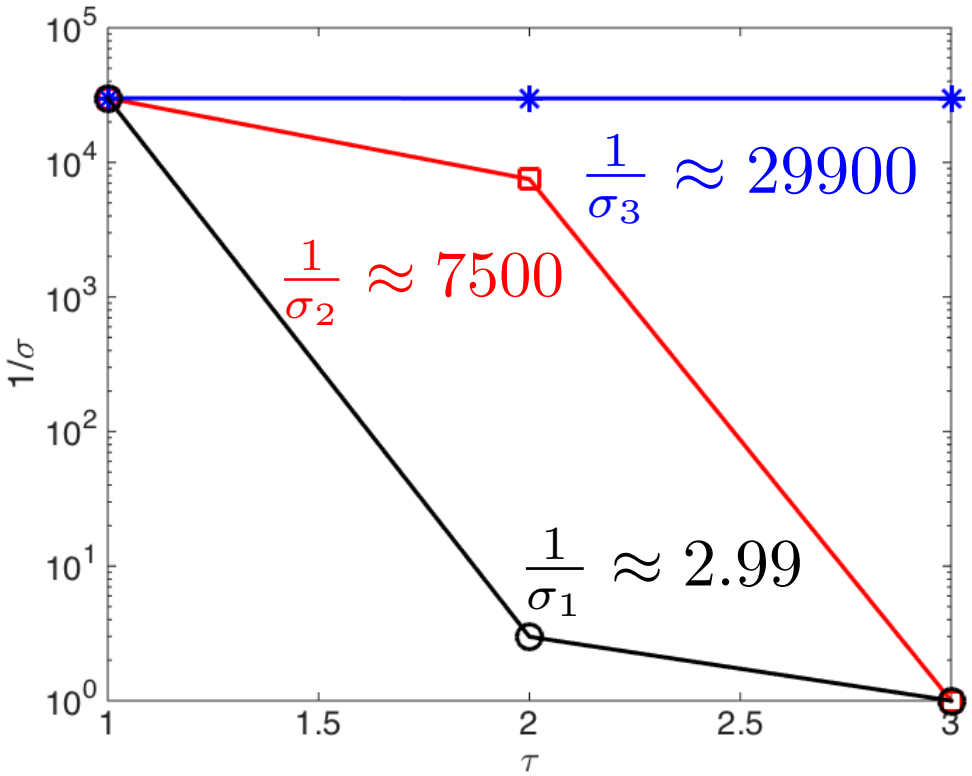
9.1
Motivation

The Power of Curvature

$$\min_{x \in \mathbb{R}^3} \left[f(x) = \frac{1}{2} x^T \mathbf{M} x + b^\top x + c \right]$$

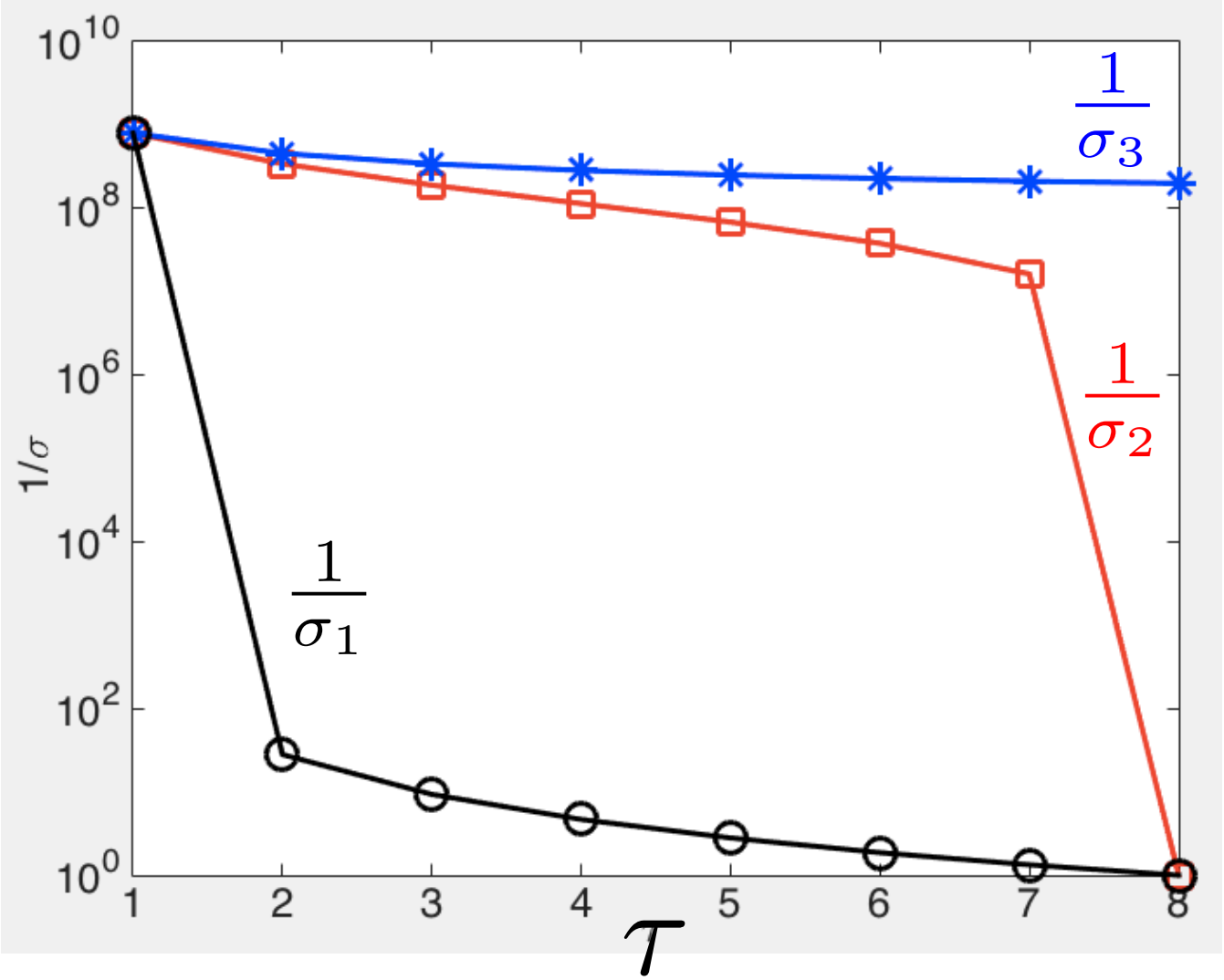
$$\mathbf{M} = \begin{pmatrix} 1.0000 & 0.9900 & 0.9999 \\ 0.9900 & 1.0000 & 0.9900 \\ 0.9999 & 0.9900 & 1.0000 \end{pmatrix}$$

condition number $\approx 3 \times 10^4$



- Phenom. described in [Qu et al 15]
- Two points of view: "Exact line search in higher dimensional subspaces" or "inversion of random submatrices of the Hessian"
- Applied to ERM dual: **SDNA** (Stochastic Dual Newton Ascent)

8D Quadratic



9.2
Three Methods

The Problem & Assumptions

$$\min_{x \in \mathbb{R}^n} f(x)$$

Large dimension

Strong convexity

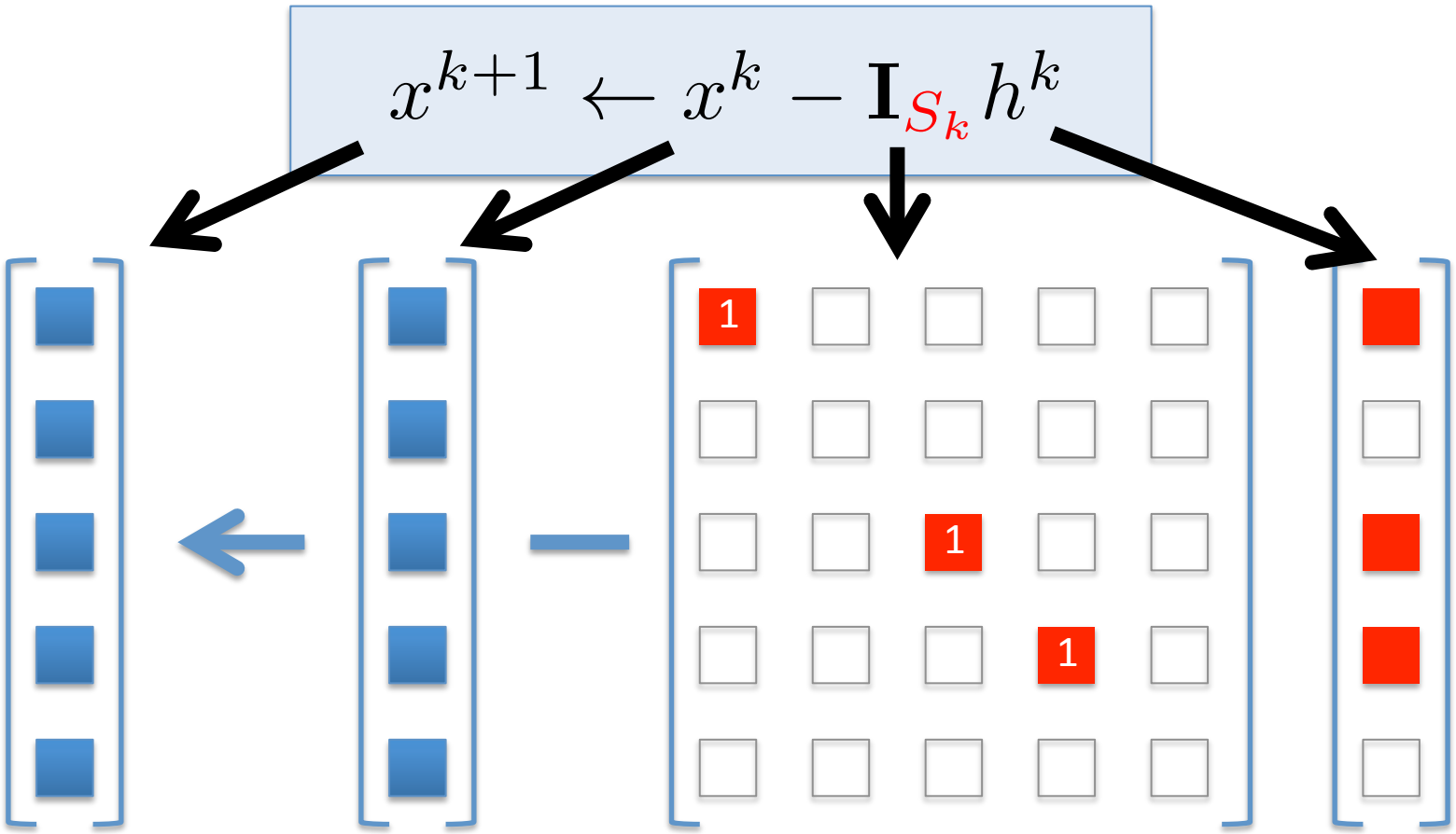
$$f(x) + (\nabla f(x))^\top h + \frac{1}{2} h^\top \mathbf{G} h \leq f(x+h)$$

Positive definite matrices

Smoothness

$$f(x+h) \leq f(x) + (\nabla f(x))^\top h + \frac{1}{2} h^\top \mathbf{M} h$$

Randomized Update



Method 3

P.R. and Martin Takáč

On optimal probabilities in stochastic coordinate descent methods

In NIPS Workshop on Optimization for Machine Learning, 2013

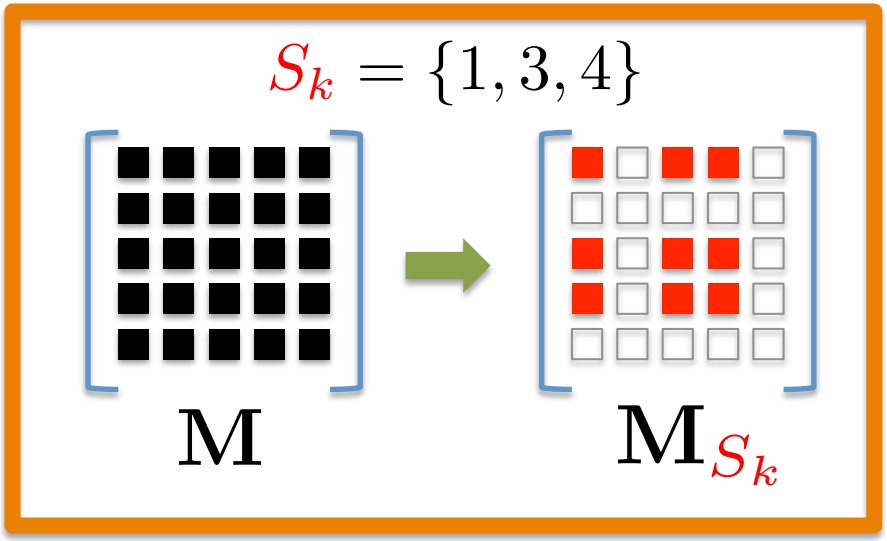
Optimization Letters 2015 (arXiv:1310.3438)

Key Inequality

$$f(x+h) \leq f(x) + (\nabla f(x))^\top h + \frac{1}{2} h^\top \mathbf{M} h$$

$$x \leftarrow x^k$$

$$h \leftarrow \mathbf{I}_{S_k} h = \sum_{i \in S_k} h_i e_i$$



$$f(x^k + \mathbf{I}_{S_k} h) \leq f(x^k) + (\nabla f(x^k))^\top (\mathbf{I}_{S_k} h) + \frac{1}{2} \underbrace{(\mathbf{I}_{S_k} h)^\top \mathbf{M} (\mathbf{I}_{S_k} h)}_{h^\top \mathbf{M}_{S_k} h}$$

Method 3

$$f(x^k + \mathbf{I}_{S_k} h) \leq f(x^k) + \left(\mathbf{I}_{S_k} \nabla f(x^k)\right)^\top h + \frac{1}{2} h^\top \mathbf{M}_{S_k} h$$

1. take expectations on both sides

$p_i = \mathbb{P}(i \in S_k)$

$$\mathbb{E}[f(x^k + \mathbf{I}_{S_k} h)] \leq f(x^k) + (\mathbf{Diag}(p)\nabla f(x^k))^\top h + \frac{1}{2} h^\top \mathbb{E}[\mathbf{M}_{S_k}] h$$

2. diagonalize

$\mathbb{E}[\mathbf{M}_{S_k}] \preceq \mathbf{Diag}(p \circ v)$

$$\mathbb{E}[f(x^k + \mathbf{I}_{S_k} h)] \leq f(x^k) + (\mathbf{Diag}(p)\nabla f(x^k))^\top h + \frac{1}{2} h^\top \mathbf{Diag}(p \circ v) h$$

3. minimize the RHS in *h*

$$x^{k+1} \leftarrow x^k - \mathbf{I}_{S_k} (\mathbf{Diag}(v))^{-1} \nabla f(x^k)$$

Method 3

i.i.d. with arbitrary distribution

Choose a random set S_k of coordinates

For $i \in S_k$ do

$$x_i{}^{k+1} \leftarrow x_i{}^{k} - \frac{1}{v_i}(\nabla f(x^k))^\top e_i$$

For $i \notin S_k$ do

$$x_i{}^{k+1} \leftarrow x_i{}^{k}$$

Convergence

Theorem (RT'13)

$$\mathbb{E}[f(x^k) - f(x^*)] \leq (1 - \sigma_3)^k (f(x^0) - f(x^*))$$

$$\sigma_3 = \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{Diag}(p \circ v^{-1})\mathbf{G}^{1/2}\right)$$

Alternative formulation:

$$k \geq \frac{1}{\sigma_3} \log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right) \quad \Rightarrow \quad \mathbb{E}[f(x^k) - f(x^*)] \leq \epsilon$$

Uniform vs Optimal Sampling

Special case:

$$\mathbf{G} = \lambda \mathbf{I} \quad \Rightarrow \quad \frac{1}{\sigma_3} = \max_i \frac{v_i}{\lambda p_i}$$

$$\mathbb{P}\left(|S_k| = 1\right) = 1 \quad \Rightarrow \quad v_i = \mathbf{M}_{ii}$$

$$p_i = \frac{1}{n} \quad \Rightarrow \quad \frac{1}{\sigma_3} = \frac{n \max_i \mathbf{M}_{ii}}{\lambda}$$

$$p_i = \frac{\mathbf{M}_{ii}}{\sum_i \mathbf{M}_{ii}} \quad \Rightarrow \quad \frac{1}{\sigma_3} = \frac{\sum_{i=1}^n \mathbf{M}_{ii}}{\lambda}$$

Method 2

Method 2

$$f(x^k + \mathbf{I}_{S_k} h) \leq f(x^k) + \left(\mathbf{I}_{S_k} \nabla f(x^k)\right)^\top h + \frac{1}{2} h^\top \mathbf{M}_{S_k} h$$

1. take expectations on both sides

$$\mathbb{E}[f(x^k + \mathbf{I}_{S_k} h)] \leq f(x^k) + (\mathbf{Diag}(p)\nabla f(x^k))^\top h + \frac{1}{2} h^\top \mathbb{E}[\mathbf{M}_{S_k}] h$$

$$p_i = \mathbb{P}(i \in S_k)$$

2. minimize the RHS in *h*

$$x^{k+1} \leftarrow x^k - \mathbf{I}_{S_k} \left(\mathbb{E}[\mathbf{M}_{S_k}]\right)^{-1} \mathbf{Diag}(p) \nabla f(x^k)$$

Convergence of Method 2

Theorem (QRTF'15)

$$\mathbb{E}[f(x^k) - f(x^*)] \leq (1 - \sigma_2)^k (f(x^0) - f(x^*))$$

$$\sigma_2 = \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{Diag}(p)\left(\mathbb{E}\left[\mathbf{M}_{S_k}\right]\right)^{-1}\mathbf{Diag}(p)\mathbf{G}^{1/2}\right)$$

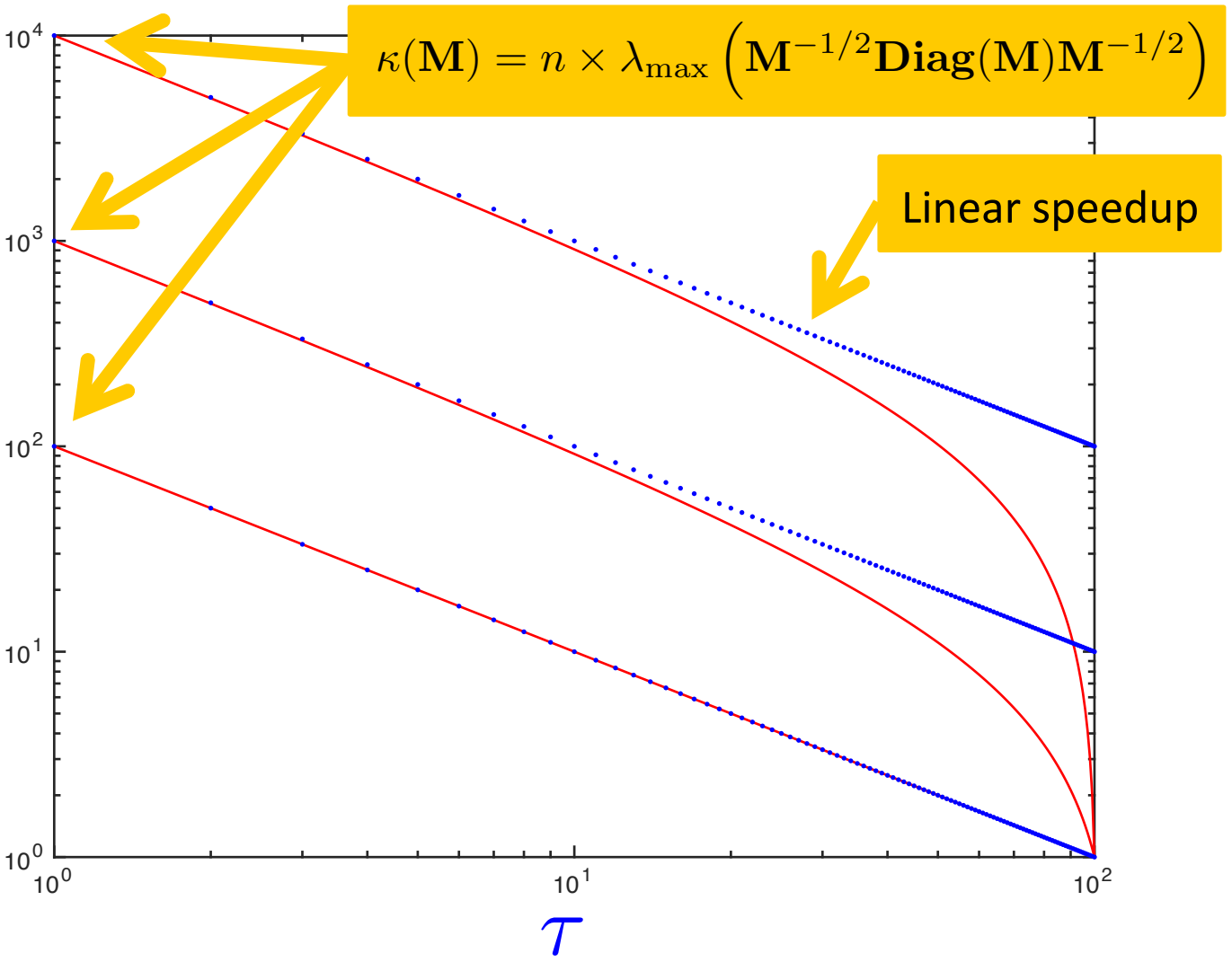
Alternative formulation:

$$k \geq \frac{1}{\sigma_2}\log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right) \quad \Rightarrow \quad \mathbb{E}[f(x^k) - f(x^*)] \leq \epsilon$$

Leading term in the complexity of Method 2 as a function of $\tau = \mathbb{E}[|S_k|]$

$$\frac{1}{\sigma_2(\tau)} = \frac{n}{n-1}\lambda_{\max}\left(\mathbf{G}^{-1/2}\left[\left(\frac{n}{\tau}-1\right)\mathbf{Diag}(\mathbf{M}) + \left(1-\frac{1}{\tau}\right)\mathbf{M}\right]\mathbf{G}^{-1/2}\right)$$

$$\frac{1}{\sigma_2(\tau)}$$



Method 1
Randomized Newton Method

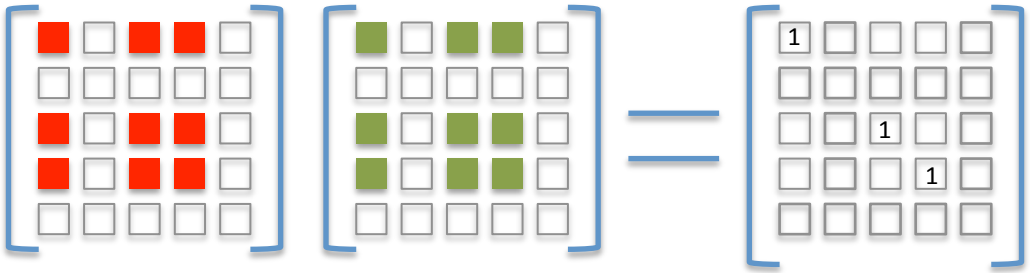
Method 1: Randomized Newton

$$f(x^k + \mathbf{I}_{S_k} h) \leq f(x^k) + \left(\mathbf{I}_{S_k} \nabla f(x^k)\right)^\top h + \frac{1}{2} h^\top \mathbf{M}_{S_k} h$$

minimize the RHS in *h*

$$x^{k+1} \leftarrow x^k - (\mathbf{M}_{S_k})^{-1} \nabla f(x^k)$$

$$S_k = \{1, 3, 4\}$$



$\mathbf{M}_{S_k}$ $\quad (\mathbf{M}_{S_k})^{-1}$ $\quad \mathbf{I}_{S_k}$

Convergence of Method 1 (Randomized Newton Method)

Theorem (QRTF'15)

$$\mathbb{E}[f(x^k) - f(x^*)] \leq (1 - \sigma_1)^k (f(x^0) - f(x^*))$$

$$\sigma_1 = \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbb{E}\left[(\mathbf{M}_{S_k})^{-1}\right]\mathbf{G}^{1/2}\right)$$

Alternative formulation:

$$k \geq \frac{1}{\sigma_1}\log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right) \quad \Rightarrow \quad \mathbb{E}[f(x^k) - f(x^*)] \leq \epsilon$$

9.3
Three Convergence Rates

3 Convergence Rates

Theorem [QRTF'15]

$$0 < \sigma_3 \leq \sigma_2 \leq \sigma_1 \leq 1$$

$$\sigma_1(1) = \sigma_2(1) = \sigma_3(1)$$

$$\sigma_1(n) = \sigma_2(n) = \frac{1}{\kappa_f}$$

$$\sigma_2(\tau) \geq \tau\sigma_2(1)$$

$$\sigma_3(\tau) \leq \tau\sigma_3(1)$$

$$\kappa_f = \lambda_{\max}\left(\mathbf{G}^{-1/2}\mathbf{M}\mathbf{G}^{-1/2}\right)$$

The 3 methods coincide if we update 1 coordinate at a time

Methods 1 and 2 coincide if we update all coordinates

Randomized Newton: **superlinear speedup**

Randomized Coordinate Descent: **sublinear speedup**

9.4
Application to ERM

Primal Problem

$$|\phi_i'(a) - \phi_i'(b)| \leq \tfrac{1}{\gamma}|a-b| \quad \forall a, b \in \mathbb{R}$$

P = Regularized Empirical Risk

$1/\gamma$ - smooth & convex functions ("risk")

positive regularization parameter

$$\min_{w \in \mathbb{R}^d} \left[P(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \lambda g(w) \right]$$

d = # features (parameters)

w = linear predictor

n data vectors ("examples")

1 - strongly convex function ("regularizer")

$$g(w) \geq g(w') + \langle \nabla g(w'), w - w' \rangle + \frac{1}{2}\|w - w'\|^2, \quad w, w' \in \mathbb{R}^d$$

Dual Problem

n dual variables: as many as # examples in the primal

$$\in \mathbb{R}^d$$

$$\max_{\alpha \in \mathbb{R}^n} \left[D(\alpha) \equiv -\lambda g^* \left(\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i \right) - \frac{1}{n} \sum_{i=1}^{n} \phi_i^*(-\alpha_i) \right]$$

1 – smooth & convex

γ - strongly convex

$$g^*(w') = \max_{w \in \mathbb{R}^d} \left\{ (w')^\top w - g(w) \right\}$$

$$\phi_i^*(a') = \max_{a \in \mathbb{R}^m} \left\{ (a')^\top a - \phi_i(a) \right\}$$

SDNA

Initialization: $\alpha^0 \in \mathbb{R}^n \quad \bar{\alpha}^0 = \frac{1}{\lambda n}\mathbf{A}\alpha^0$

$$\mathbf{A} = [A_1, A_2, \ldots, A_n] \in \mathbb{R}^{d\times n}$$

$$\mathbf{X} = \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A}$$

Iterate:

Primal update: $w^k = \nabla g^*(\bar{\alpha}^k)$

Generate a random set S_k

Compute:

$$h^k = \arg\min_{h\in\mathbf{R}^n} \left((\mathbf{A}^\top w^k)_{S_k}\right)^\top h + \tfrac{1}{2}h^\top \mathbf{X}_{S_k} h + \sum_{i\in S_k} \phi_i^*(-\alpha_i^k - h_i)$$

Dual update: $\alpha^{k+1} \leftarrow \alpha^k + \sum_{i\in S_k} h_i^k e_i$

Maintain average: $\bar{\alpha}^{k+1} = \bar{\alpha}^k + \frac{1}{\lambda n}\sum_{i\in S_k} h_i^k A_i$

Convergence of SDNA

Better rate than SDCA

Theorem (QRTF'15)

Assume that S_k is uniform

$$\mathbb{E}[P(w^k) - D(\alpha^k)] \leq (1 - \sigma_1^{prox})^k \frac{D(\alpha^*) - D(\alpha^0)}{\theta(S_k)}$$

Expected duality gap after *k* iterations

$$\sigma_1^{prox} = \frac{\tau}{n} \min\{1, s_1\}$$

$$\tau = \mathbb{E}[|S_k|] \qquad s_1 = \lambda_{\min}\left[\left(\frac{1}{\tau\gamma\lambda}\mathbb{E}[(\mathbf{A}^\top\mathbf{A})_{S_k}] + \mathbf{I}\right)^{-1}\right]$$

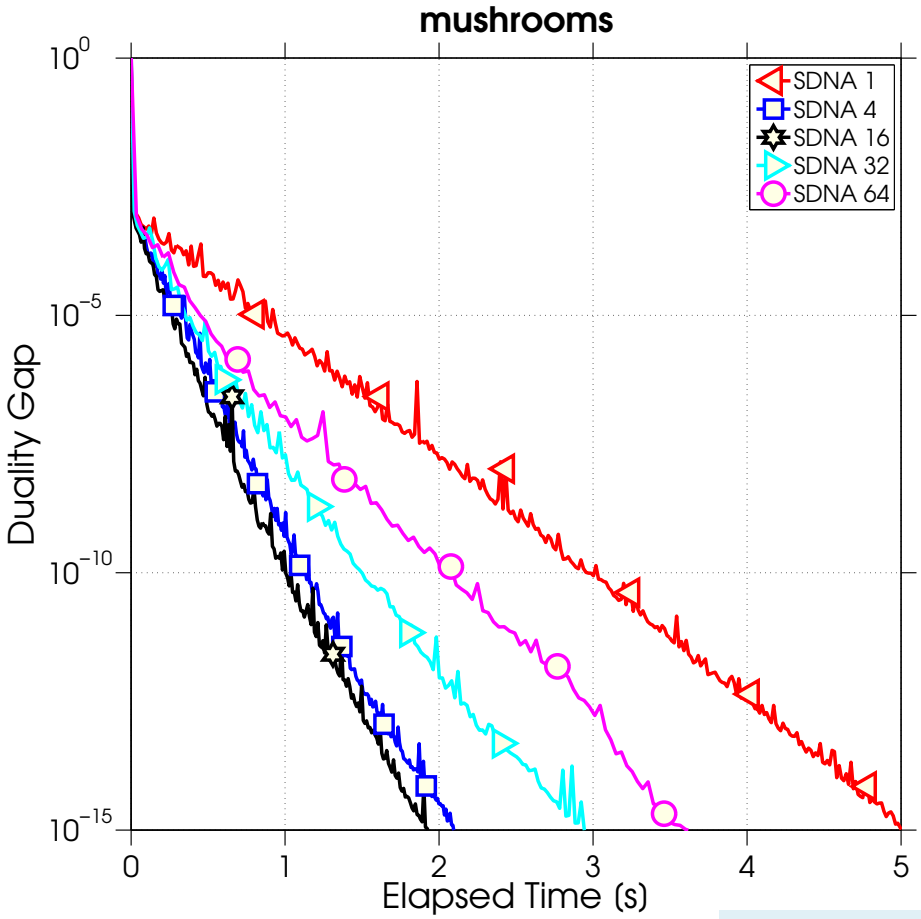
9.5
Experiments

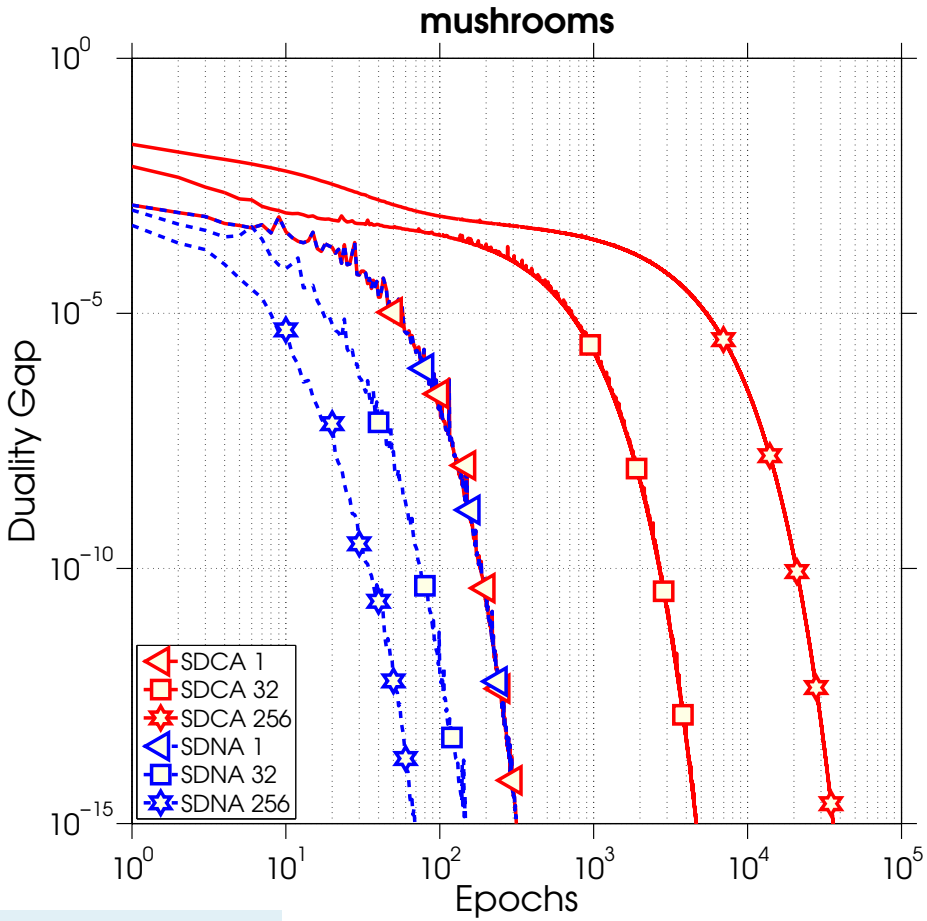
Real Dataset:

`mushrooms`

$d = 112$ $n = 8{,}124$

Sampling "Smallish" Submatrices of the Hessian Helps





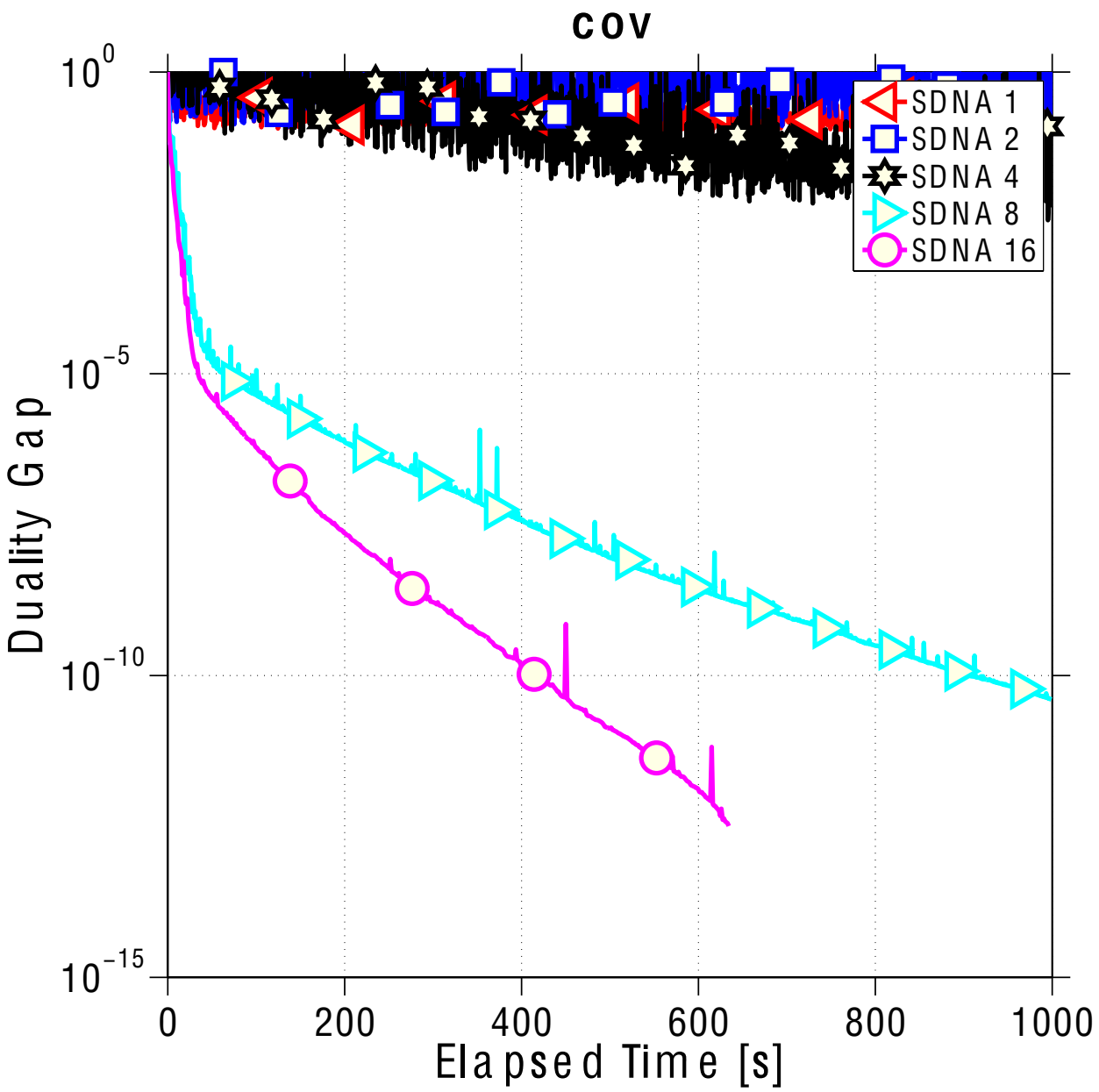
features: $d = 112$

examples: $n = 8124$

Real Dataset:

cov

$d = 54$ $n = 581{,}012$


cov
SDNA 1
SDNA 2
SDNA 4
SDNA 8
SDNA 16
Duality Gap
Elapsed Time [s]
10^{0}
10^{-5}
10^{-10}
10^{-15}
0
200
400
600
800
1000

9.6
Summary

Summary

- Can combine curvature & randomization and get complexity rates
- Curvature is utilized by doing exact computations in small but multidimensional subspaces
- Randomized "Newton" (Method 1):
  - Superlinear speedup (always)
  - Expensive iterations: Needs to solve a "small" but potentially dense linear system in each step
- Randomized Coordinate Descent (Method 3):
  - Sublinear speedup (gets better with sparsity or good spectral properties)
  - Cheap iterations: Needs to solve a small diagonal linear system in each step
- Can apply to the dual of ERM: SDNA
  - Coincides with SDCA if minibatch size = 1
  - Improves on SDCA when minibatch size is "small enough"
  - New effect: # passes over data decreases as minibatch size increases
- Previous work: Stochastic quasi-Newton [Schraudolph, Yu, Gunter '07] [Bordes, Bottou, Gallinari '09] [Byrd, Hansen, Nocedal, Singer '14] Newton sketch [Pilanci & Wainwright '15]

Randomized Methods with Arbitrary Sampling

Method 3

P.R. and Martin Takáč. **On optimal probabilities in stochastic coordinate descent methods.** *Optimization Letters, 2015 (arXiv:1310.3438)*

Zheng Qu, P.R. and Tong Zhang. **Randomized dual coordinate ascent with arbitrary sampling.** *arXiv:1411.5873*

Zheng Qu and P.R. **Coordinate descent with arbitrary sampling I: algorithms and complexity.** *arXiv:1412.8060*

Zheng Qu and P.R. **Coordinate descent with arbitrary sampling II: ESO.** *arXiv:1412.8063*

Zheng Qu, P.R., Martin Takáč and Olivier Fercoq. **SDNA: Stochastic Dual Newton Ascent for empirical risk minimization.** *arXiv:1502.02268*

Dominik Csiba and P.R. **Primal method for ERM with flexible mini-batching schemes and non-convex losses.** ICML 2015 *(arXiv:1502.02268)*

Robert M. Gower and P.R. **Randomized iterative methods for linear systems.** *arXiv: 1502.02268*

BIBLIOGRAPHY
(Coordinate Descent)

Citation	Algorithm	Paper
[Leventhal & Lewis 08]	RCD	**Randomized methods for linear constraints: convergence rates and conditioning.** *Mathematics of OR* 35(3), 641-654, 2010 (arXiv:0806.3015)
[S-Shwartz & Tewari 09]	SCD	**Stochastic methods for L1-regularized loss minimization.** *ICML* 2009
[Nesterov 10]	UCDM, RCDM, ACDM	**Efficiency of coordinate descent methods on huge-scale optimization problems.** *SIAM J. on Optimization*, 22(2):341–362, 2012 (CORE Discussion Paper 2010/2)
[Bradley et al 11]	Shotgun 	**Parallel coordinate descent for L1-regularized loss minimization**. *ICML*, 2011 (arXiv: 1105.5379)
[R & Takáč 11a]	SCD	**Efficient serial and parallel coordinate descent methods for huge-scale truss topology design.** *Operations Research Proceedings*, 27-32, 2012 (Opt Online 08/2011)
[R & Takáč 11b]	UCDC, RCDC	**Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function.** *Mathematical Programming* 144(2), 1-38, 2014 (arXiv:1107.2848)
[R & Takáč 12]	PCDM	**Parallel coordinate descent methods for big data optimization.** *Mathematical Programming*, 2015 (arXiv:1212.0873)
[S-Shwartz & Zhang 12]	SDCA	**Stochastic dual coordinate ascent methods for regularized loss minimization.** *JMLR* 14, 567-599, 2013 (arXiv:1209.1873)
[Necoara & Clipici 13]	RCD	**A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints.** COAP 57(2), 303-337, 2014 (arXiv: 1302.3074)
[Takáč et al 13]	mSDCA	**Mini-batch primal and dual methods for SVMs.** *ICML* 2013 (arXiv:1303.2314)
[Tappenden, R, & Gondzio 13]	ICD	**Inexact coordinate descent.** arXiv:1304.5530, 2013
[S-Shwartz & Zhang 13a]	ASDCA	**Accelerated mini-batch stochastic dual coordinate ascent.** *NIPS* 2013 (arXiv: 1305.2581)

Citation	Algorithm	Paper
[Lu & Xiao 13]	RBCD	**On the complexity analysis of randomized block-coordinate descent methods**. *Mathematical Programming,* 2014 (arXiv:1305.4723)
[Patrascu & Necoara 13]		**Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization**. *J of Global Optimization* 61(1), 19-46 (arXiv:1305.4027)
[Lee & Sidford 13]	ACDM	**Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems.** *FOCS* 2013 (arXiv:1305.1922)
[Tappenden, R & Buke 13]	DQA vs PCDM	**Separable approximations and decomposition methods for the augmented Lagrangian.** *Optimization Methods and Software* 30(3), 643-668, 2015 (arXiv: 1308.6774)
[S-Shwartz & Zhang 13b]	Acc Prox-SDCA	**Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization.** *Mathematical Programming* 2014 (arXiv:1309.2375)
[Fercoq & R 13a]	SPCDM	**Smooth minimization of nonsmooth functions with parallel coordinate descent methods.** arXiv:1309.5885, 2013
[R & Takáč 13a]	HYDRA	**Distributed coordinate descent method for learning with big data.** arXiv:1310.2059, 2013
[R & Takáč 13b]	SYNC	**On optimal probabilities in stochastic coordinate descent methods.** *Opt. Letters*, 2015 (arXiv:1310.3438)
[Liu et al 13]	AsySCD	**An asynchronous parallel stochastic coordinate descent algorithm.** *ICML* 2014 (arXiv: 1311.1873)
[Shalit & Chechik 13]	RCM	**Efficient coordinate-descent for orthogonal matrices through Givens rotations.** *ICML* 2014 (arXiv:1312.0624)
[Fercoq & R 13b]	APPROX	**Accelerated, parallel and proximal coordinate descent.** arXiv:1312.5799, 2013
[Yang 13]	DisDCA	**Trading computation for communication: distributed stochastic dual coordinate ascent.** *NIPS* 2013
[Zhao & Zhang 14]	I-Prox SDCA, I-Prox SGD	**Stochastic optimization with importance sampling.** ICML 2015, arXiv:1401.2753, 2014

Citation	Algorithm	Paper
[Liu & Wright 14]	AsySPCD	**Asynchronous stochastic coordinate descent: parallelism and convergence properties.** *SIAM J. on Optimization*, 25(1), 351–376, 2015 (arXiv:1403.3862)
[Mahajan, Keerthi & Sundararajan 14]	DBCD	**A distributed block coordinate descent method for training l1 regularized linear classifiers.** arXiv:1405.4544, 2014
[Fercoq et al 14]	Hydra2	**Fast distributed coordinate descent for non-strongly convex losses.** *MLSP* 2014 (arXiv:1405.5300)
[Mareček, R and Takáč 14]	DBCD	**Distributed block coordinate descent for minimizing partially separable functions.** Numerical Analysis and Opt., Springer Proc. in Math. and Stat. (arXiv:1406.0238)
[Lin, Lu & Xiao 14]	APCG	**An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization.** *NIPS* 2014 (arXiv:1407.1296)
[Zhang & Xiao 14]	SPDC	**Stochastic primal-dual coordinate method for regularized empirical risk minimization.** *ICML* 2015 (arXiv:1409.3257)
[Jaggi, Smith, Takáč et al 14]	CoCoA	**Communication-efficient distributed dual coordinate ascent.** *NIPS* 2014 (arXiv: 1409.1458)
[Qu, R & Zhang 14]	 QUARTZ	**Randomized dual coordinate ascent with arbitrary sampling.** arXiv:1411.5873, 2014
[Konečný, Q & R 14]	S2CD	**Semi-stochastic coordinate descent**. *NIPS* Optimization Workshop, 2014 (arXiv: 1412.6293)
[Qu and R 14a]	ALPHA α	**Coordinate descent with arbitrary sampling I: algorithms and complexity.** arXiv: 1412.8060, 2014
[Qu and R 14b]		**Coordinate descent with arbitrary sampling II: expected separable overapproximation.** arXiv:1412.8063, 2014
[Qu et al 15]	SDNA	**SDNA: Stochastic dual newton ascent for empirical risk minimization.** arXiv: 1502.02268, 2015
[Ma, Smith, Jaggi et al 15]	CoCoA+	**Adding vs. averaging in distributed primal-dual optimization.** *ICML* 2015

Citation	Algorithm	Paper
[Tappenden, Takáč & R 15]	PCDM	**On the complexity of parallel coordinate descent.** arXiv:1503.03033, 2015
[Csiba, Qu & R 15]	AdaSDCA	**Stochastic dual coordinate ascent with adaptive probabilities.** *ICML* 2015
[Ene & Nguyen 15]	RCDM, APPROX	**Random coordinate descent methods for minimizing decomposable submodular functions.** *ICML* 2015 (arXiv:1502.02643)
[S-Shwartz 15]	SDCA	**SDCA without duality.** arXiv:1502.06177, 2015
[Csiba & R 15]	dfSDCA	**Primal method for ERM with flexible mini-batching schemes and non-convex losses**. arXiv:1506.02227, 2015
[Wright 15]		**Coordinate descent algorithms.** *Mathematical Programming* 151(1), 3-34, 2015 (arXiv:1502.04759)
[Gower & R 15]		**Randomized iterative methods for linear systems.** arXiv:1506.03296, 2015
[Nutini et al 15]		**Coordinate descent converges faster with the Gauss-Southwell rule than random selection.** *ICML* 2015

THE END