Randomized Optimization Methods

Peter Richtárik

King Abdullah University of Science and Technology

Paris, Aug 28-Sept 1, 2017

Outline

1. Supervised Learning
   - Prediction, loss functions, regularizers, ERM
   - Convexity, strong convexity and smoothness
   - ERM duality, convex conjugation
   - 4 + 4 problem classes
   - Linear systems as ERM
2. Standard Algorithmic Toolbox in Optimization
   - 8 tools: GD, Acceleration, Proximal Trick, Randomized Decomposition (SGD/RCD), Minibatching, Variance Reduction, Importance Sampling, Duality
   - Summary
3. Stochastic Methods for Linear Systems
   - Stochastic reformulations
   - Basic, parallel and accelerated methods
   - Dual method
   - Extra topics: special cases, stochastic preconditioning, stochastic matrix inversion

Part 1
Supervised Learning

The Idea

Prediction of Object Labels

Set of "natural" objects $\mathcal{A}$	Set of labels	$\mathcal{B}$	Prediction task
NYT articles 	**Article category**	**(finite set)**	**Multi-class classification**
E-mails	Spam / not-spam	$\{-1, 1\}$	Binary classification
Images	Image category	(finite set)	Multi-class classification
Surveillance videos	Probability of a threat	$[0, 1]$	Regression
User clicks	Age	$(0, 150]$	Regression

Statistical Model of Objects & Labels

We assume that object-label pairs occur in nature according to some (unknown) distribution:

$$(a_i, b_i) \sim \mathcal{D}$$

GOAL: Given a sampled object a_i predict the unknown label b_i

Feature Map: Vector Representation of Natural Objects

features

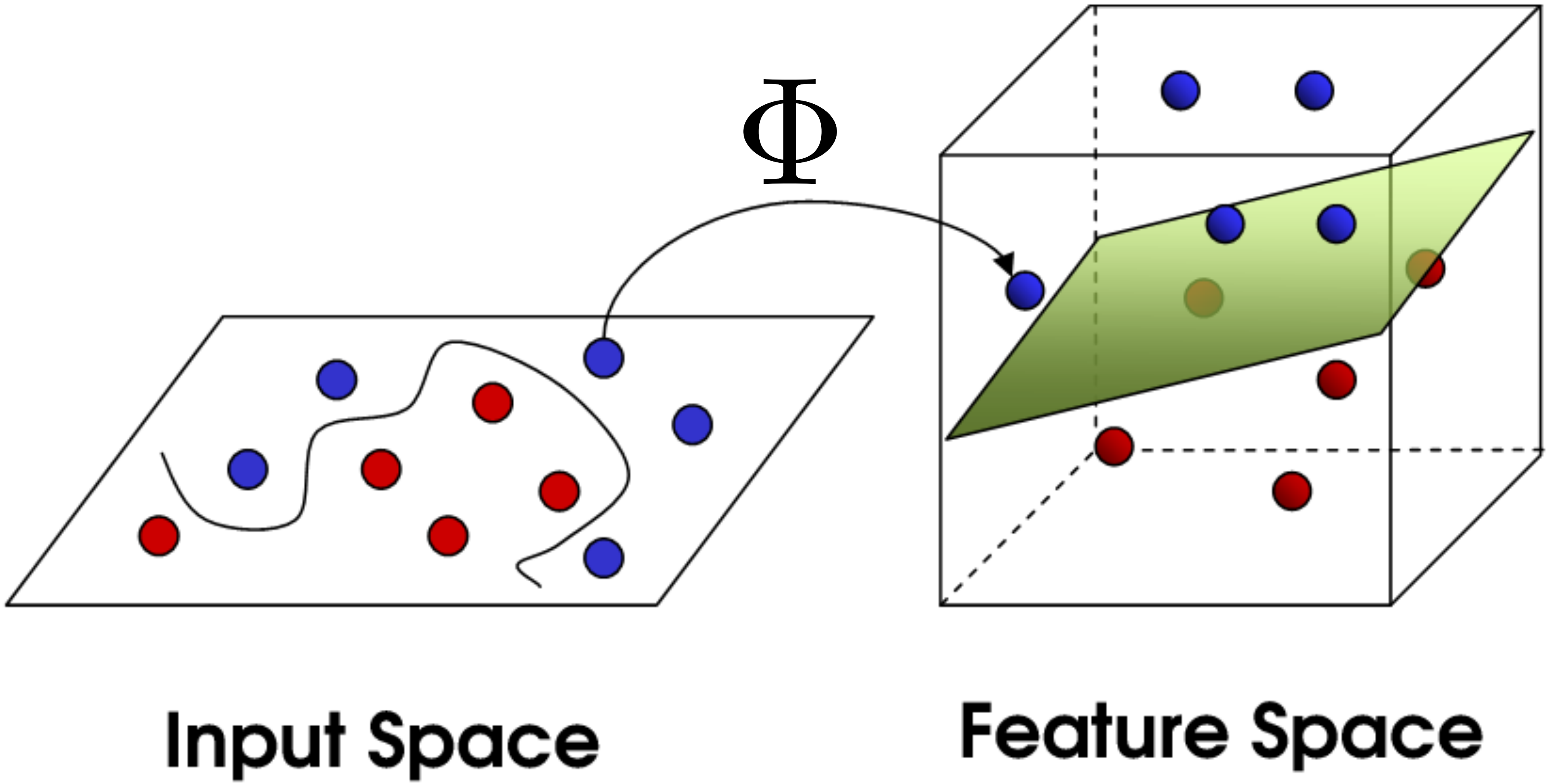
$$\mathcal{A} \xmapsto{\Phi} \mathbb{R}^d$$

Vector representation

Feature engineering (manual design)

Representation learning (automatic design)

Kernel Trick

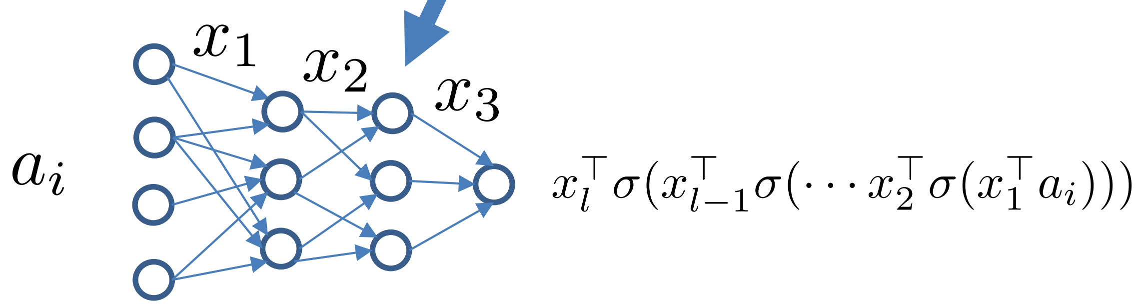


Predictor

Parameter defining the predictor

$$h_x : \mathcal{A} \mapsto \mathbb{R}, \qquad x \in \mathbb{R}^d$$

	$h_x(a_i)$	Feature map
Linear Predictor	$x^\top \Phi(a_i)$	$\Phi(a_i)$ explicit
Neural Network	$x_l^\top \sigma(x_{l-1}^\top \sigma(\cdots x_2^\top \sigma(x_1^\top a_i)))$	learned $\sigma(x_{l-1}^\top \sigma(\cdots x_2^\top \sigma(x_1^\top a_i)))$



Loss and Expected Loss

$$loss(h_x(a_i), b_i)$$

Predicted label

True label

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

$$\min_{x \in \mathbb{R}^d} \mathbf{E}_{(a_i, b_i) \sim \mathcal{D}} \left[loss(h_x(a_i), b_i) \right]$$

Empirical Risk Minimization

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

$$(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n) \sim \mathcal{D}$$

Monte-Carlo integration (sample average approximation)

Output predictor which minimizes the Empirical Risk:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} loss(h_x(a_i), b_i) + g(x)$$

From now on, let: $h_x(a_i) = \Phi(a_i)^\top x$ (linear predictor)

$\Phi(a_i) = a_i$ (objects are already represented as vectors)

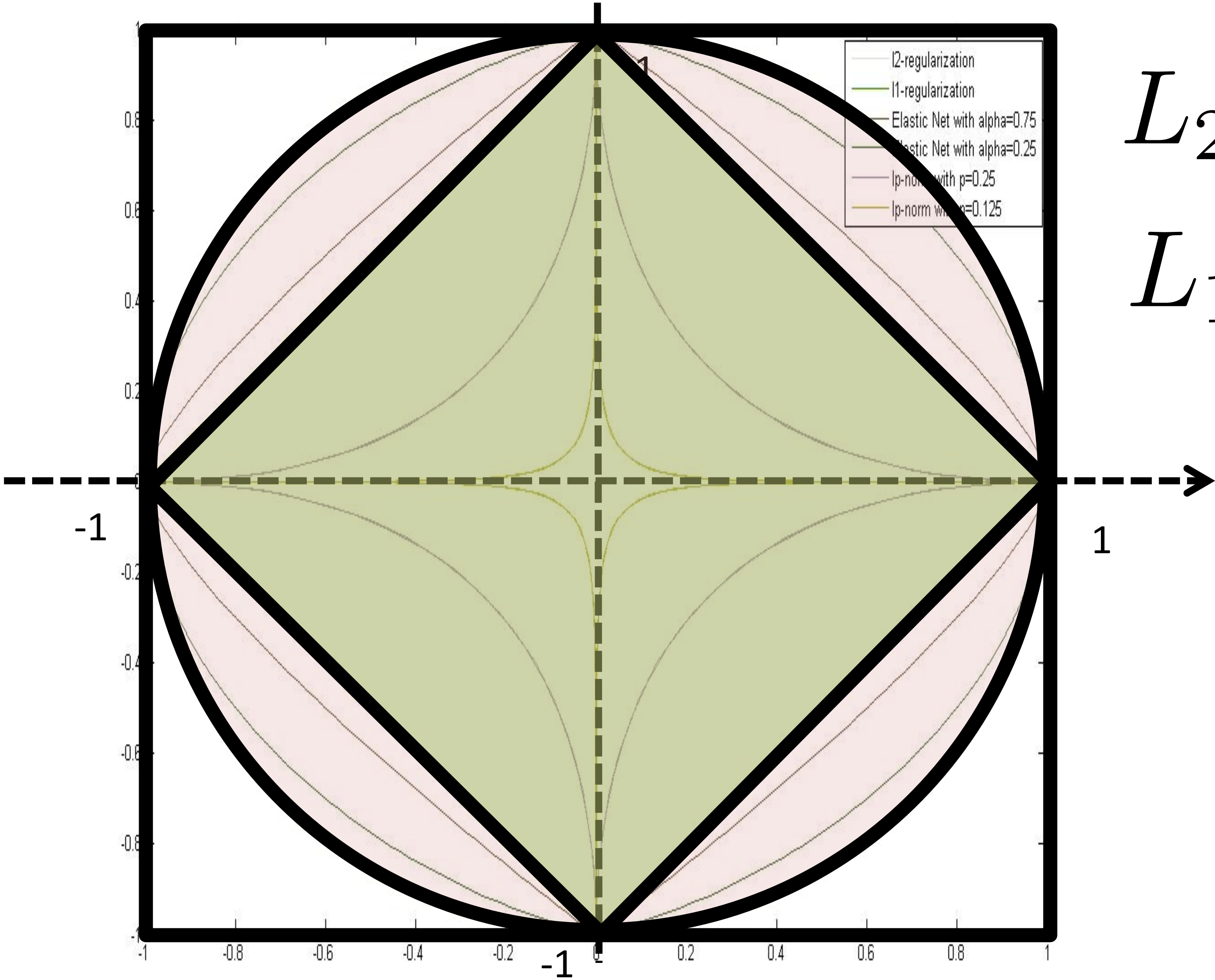
$f_i(a_i^\top x) \stackrel{\text{def}}{=} loss(a_i^\top x, b_i)$ (hiding the label)

Loss Functions

&

Regularizers

Regularizers



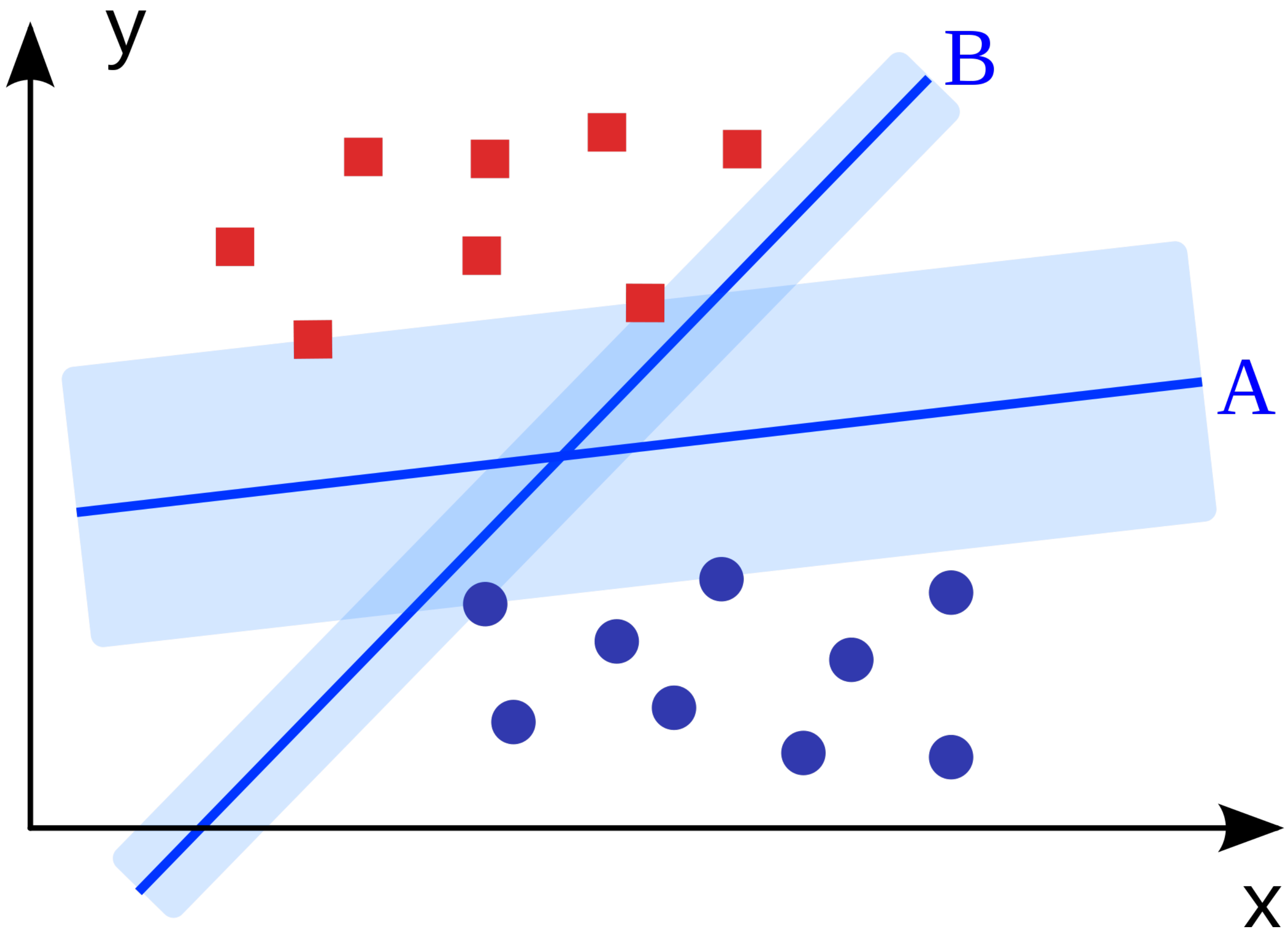
L_2 $\frac{\mu}{2}\|x\|_2^2$

L_1 $\mu\|x\|_1$

Examples of ERM Problems

	$f_i(t)$	$g(x)$
Least Squares	$\frac{1}{2}(t-b_i)^2$	0
Ridge Regression	$\frac{1}{2}(t-b_i)^2$	$\frac{\mu}{2}\|\|x\|\|_2^2$ $\|x\|_2 = \sqrt{x^\top x}$
LASSO	$\frac{1}{2}(t-b_i)^2$	$\mu\|\|x\|\|_1$ $\|x\|_1 = \sum_i \|x_i\|$
Non-negative Least Squares Regression	$\frac{1}{2}(t-b_i)^2$	$1_{x\geq 0}(x) = \begin{cases} 0 & x \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$
SVM	$\max\{0, 1-b_i \cdot t\}$	$\frac{\mu}{2}\|\|x\|\|_2^2$
Logistic Regression	$\log(1+e^{-b_i t})$	$\frac{\mu}{2}\|\|x\|\|_2^2$
Linear System (Best Approximation)	$1_{\{b_i\}}(t) = \begin{cases} 0 & t = b_i, \\ +\infty & \text{otherwise.} \end{cases}$	$\frac{1}{2}\|\|x - x^0\|\|_B^2$ $\|x\|_B = \sqrt{x^\top B x}$
L1 Regression	$\|t-b_i\|$	0

SVM: Support Vector Machine



Source: wikipedia

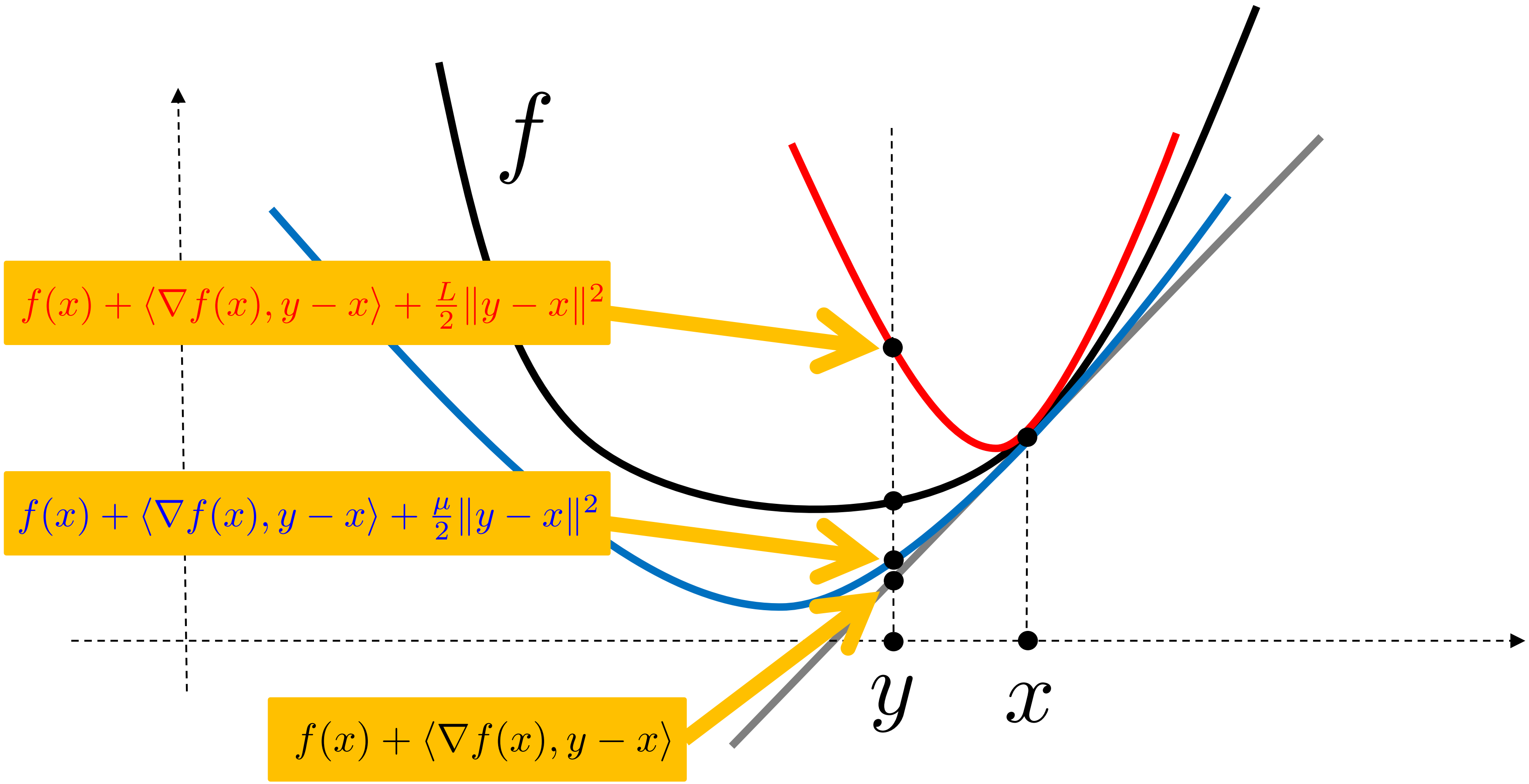
Typical Function Classes

$f : \mathbb{R}^d \to \mathbb{R}$	Defining property	If twice differentiable
convex	$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ **If continuously differentiable:** $f(x) + \langle \nabla f(x), y - x \rangle \leq f(y)$ $0 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle$	$0 \preceq \nabla^2 f(x)$
μ-strongly convex	$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) - \frac{\mu}{2}\alpha(1-\alpha)\|x-y\|^2$ **If continuously differentiable:** $f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y-x\|^2 \leq f(y)$ $\mu\|x-y\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle$	$\mu \cdot I \preceq \nabla^2 f(x)$
L-smooth	$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$ $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y-x\|^2$	$\nabla^2 f(x) \leq L \cdot I$

Visualizing Smoothness and Strong Convexity

$$\mu \cdot I \preceq \nabla^2 f(x) \preceq L \cdot I$$

$$f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$$



Empirical Risk Minimization

Primal Problem

loss function

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x) + g(x) \right]$$

d = # features (parameters)

n = # samples

regularizer

Convex Conjugate (Legendre-Fenchel Transform)

- Convex conjugate of a function is the generalization of the Legendre transform
- Convex conjugation was 200 years later studied by Werner Fenchel
- It is a key tool in optimization duality

Adrien-Marie Legendre

1820 watercolor caricature of Adrien-Marie Legendre by French artist Julien-Leopold Boilly (see portrait debacle), the only existing portrait known[1]

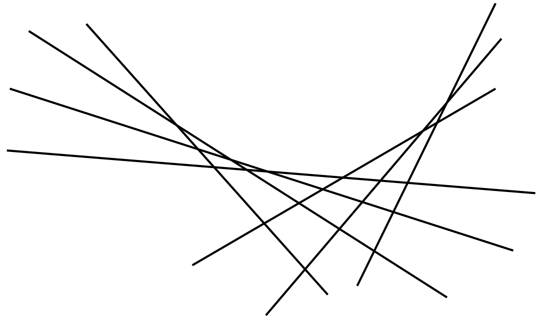
Born	18 September 1752 Paris, France
Died	10 January 1833 (aged 80) Paris, France
Residence	France
Nationality	French
Fields	Mathematician
Institutions	École Militaire École Normale École Polytechnique
Alma mater	Collège Mazarin
Known for	Legendre transformation Legendre polynomials Legendre transform Elliptic functions Introducing the character ∂[2]

Moritz Werner Fenchel

Werner Fenchel, 1972

Born	3 May 1905 Berlin, Germany
Died	24 January 1988 (aged 82) Copenhagen, Denmark
Residence	Germany, Denmark, USA
Citizenship	German
Fields	Mathematics: Geometry Optimization
Institutions	University of Copenhagen University of Göttingen
Alma mater	University of Berlin
Doctoral advisor	Ludwig Bieberbach
Doctoral students	Birgit Grodal Peter Scherk Troels Jørgensen
Known for	Alexandrov–Fenchel inequality Legendre–Fenchel transformation Fenchel's duality theorem

Convex Conjugate



$$f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$$

$$f^*(z) \stackrel{\text{def}}{=} \sup_{x \in \mathbb{R}^d} \{\langle z, x \rangle - f(x)\}$$

Theorem

$$f \text{ is } L\text{-smooth} \quad \Leftrightarrow \quad f^* \text{ is } \frac{1}{L}\text{-strongly convex}$$

$$f \text{ is } \mu\text{-strongly convex} \quad \Leftrightarrow \quad f^* \text{ is } \frac{1}{\mu}\text{-smooth}$$

Examples: $f(x) = \frac{1}{2}\|x\|_B^2 \quad \Rightarrow \quad f^*(x) = \frac{1}{2}\|x\|_{B^{-1}}^2$

$f(x) = 1_C(x) \quad \Rightarrow \quad f^*(z) = \sup_{x \in C} \langle z, x \rangle$

$$f^*(z) \stackrel{\text{def}}{=} \sup_{x\in\mathbb{R}^d} \{\langle z,x\rangle - f(x)\}$$

Primal and Dual Problems

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x) + g(x)\right]$$

$$\max_{y\in\mathbb{R}^n}\left[D(y) \stackrel{\text{def}}{=} -\frac{1}{n}\sum_{i=1}^{n} f_i^*(-y_i) - g^*\left(\frac{1}{n} A^\top y\right)\right]$$

concave

$$A^\top = (a_1 \quad a_2 \quad \cdots \quad a_n)$$

$$A = \begin{pmatrix} a_1^\top \\ a_2^\top \\ \vdots \\ a_n^\top \end{pmatrix}$$

Duality

Weak Duality: $P(x) \geq D(y)$ (Always)

Strong Duality: $P(x^*) = D(y^*)$ (Under suitable assumptions)

Optimal solutions

If *g* is strongly convex, we can recover primal optimal solution from dual optimal solution:

$$x^* = \nabla g^* \left(\tfrac{1}{n} A^\top y^* \right)$$

Weak Duality & Optimality Conditions

$$P(x) - D(y) \quad = \quad g(x) + g^*\left(\tfrac{1}{n}A^\top y\right) + \frac{1}{n}\sum_{i=1}^{n}\{f_i(a_i^\top x) + f_i^*(-y_i)\} =$$

$$\underbrace{g(x) + g^*\left(\tfrac{1}{n}A^\top y\right) - \langle x, \tfrac{1}{n}A^\top y\rangle}_{\geq 0} + \frac{1}{n}\sum_{i=1}^{n}\underbrace{\{f_i(a_i^\top x) + f_i^*(-y_i) + \langle a_i^\top x, y_i\rangle\}}_{\geq 0}$$

Weak duality

Optimality conditions

$$x = \nabla g^*\left(\frac{1}{n}A^\top y\right)$$

$$y_i = -\nabla f_i(a_i^\top x) \qquad \forall i$$

4 Interesting Classes of Convex ERM Problems

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x) + g(x) \right]$$

$$\max_{y\in\mathbb{R}^n} \left[D(y) \stackrel{\text{def}}{=} -\frac{1}{n}\sum_{i=1}^{n} f_i^*(-y_i) - g^*\left(\frac{1}{n}A^\top y\right) \right]$$

f_i, g convex

f_i \ g	$\mu > 0$	$\mu = 0$
L-smooth	**Ridge regression** $\frac{1}{2}(t-b_i)^2$ $\frac{\mu}{2}\|x\|_2^2$ **Logistic regression** $\log(1+e^{-b_i t})$ $\frac{\mu}{2}\|x\|_2^2$	**LASSO** $\frac{1}{2}(t-b_i)^2$ $\mu\|x\|_1$ **Least Squares Regression** $\frac{1}{2}(t-b_i)^2$ 0
not L-smooth	**Linear systems** $1_{\{b_i\}}(t)$ $\frac{1}{2}\|x-x^0\|_B^2$ **SVM** $\max\{0, 1-b_i\cdot t\}$ $\frac{\mu}{2}\|x\|_2^2$	**L1-SVM** $\max\{0, 1-b_i\cdot t\}$ $\mu\|x\|_1$ **L1 regression** $\|t-b_i\|$ 0

4 Interesting Classes of ERM Problems Based on Dimensions

n \ d	SMALL	BIG
SMALL	**Deterministic methods will do fine:** *GD, AGD, Newton, quasi-Newton, …*	“Big Model” Setting **Decompose *d*** *Primal: RCD-type*
BIG	“Big Data” Setting **Decompose *n*** *Primal: SGD-type* *Dual: RCD-type*	?

Example:
Solving Linear Systems

Solving Linear Systems

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

$$\text{Solve } Ax = b$$

$A \in \mathbb{R}^{n \times d}$

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

$$A = \begin{pmatrix} a_1^\top \\ a_2^\top \\ \vdots \\ a_n^\top \end{pmatrix}$$

$$\text{Think: } n \gg d$$

Interesting Cases

$$\min_{x\in\mathbb{R}^d}\left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x) + g(x)\right]$$

f_i, g convex

$$\max_{y\in\mathbb{R}^n}\left[D(y) \stackrel{\text{def}}{=} -\frac{1}{n}\sum_{i=1}^{n} f_i^*(-y_i) - g^*\left(\frac{1}{n}A^\top y\right)\right]$$

f_i \ g	$\mu > 0$	$\mu = 0$
L-smooth	**Ridge regression** $\frac{1}{2}(t-b_i)^2$ $\frac{\mu}{2}\\|x\\|_2^2$ **Logistic regression** $\log(1+e^{-b_i t})$ $\frac{\mu}{2}\\|x\\|_2^2$	**LASSO**
not L-smooth	**Linear systems** $1_{\{b_i\}}(t)$ $\frac{1}{2}\\|x-x^0\\|_B^2$ **SVM** $\max\{0,1-b_i\cdot t\}$ $\frac{\mu}{2}\\|x\\|_2^2$	

Loss Functions and Regularizers

	$f_i(t)$	$g(x)$	
es	$\frac{1}{2}(t-b_i)^2$	0	
idge Regression	$\frac{1}{2}(t-b_i)^2$	$\frac{\mu}{2}\\|x\\|_2^2$	$\\|x\\|_2 = \sqrt{x^\top x}$
LASSO	$\frac{1}{2}(t-b_i)^2$	$\mu\\|x\\|_1$	$\\|x\\|_1 = \sum_i \|x_i\|$
Non-negative Least Squares Regression	$\frac{1}{2}(t-b_i)^2$	$1_{x\geq 0}(x) = \begin{cases} 0 & x \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$	
SVM	$\max\{0, 1-b_i\cdot t\}$	$\frac{\mu}{2}\\|x\\|_2^2$	
Logistic Regression	$\log(1+e^{-b_i t})$	$\frac{\mu}{2}\\|x\\|_2^2$	
Linear System (Best Approximation)	$1_{\{b_i\}}(t) = \begin{cases} 0 & t = b_i, \\ +\infty & \text{otherwise.} \end{cases}$	$\frac{1}{2}\\|x-x^0\\|_B^2$	$\\|x\\|_B = \sqrt{x^\top B x}$
L1 Regression	$\|t-b_i\|$	0	

Linear Systems (Best Approximation Version) as a Primal ERM Problem

$$g(x) = \frac{1}{2}\|x - x^0\|_B^2$$

$$\min_{x \in \mathbb{R}^d} \left[P(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^\top x) + g(x) \right]$$

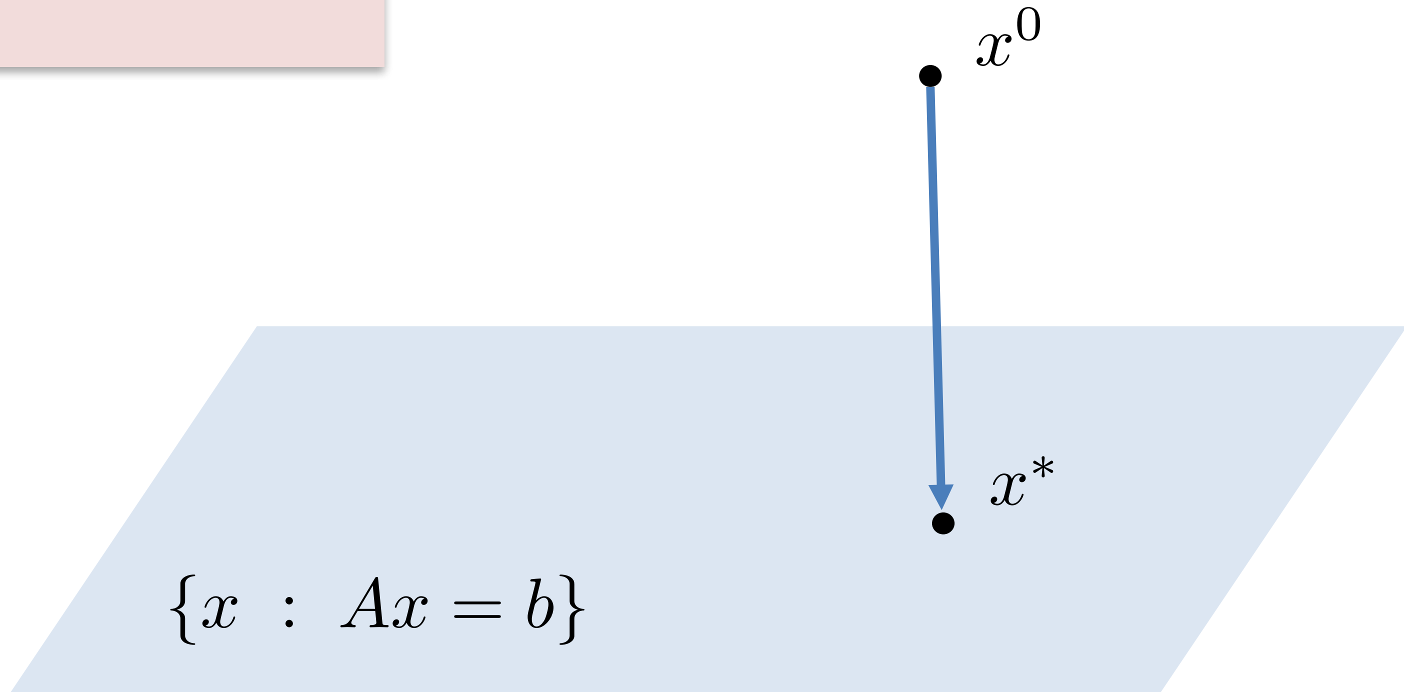
$$f_i(t) = 1_{\{b_i\}}(t) \overset{\text{def}}{=} \begin{cases} 0 & \text{for } t = b_i, \\ +\infty & \text{otherwise.} \end{cases}$$

Primal Problem: Best Approximation

$$\min_{x \in \mathbb{R}^d} \frac{1}{2}\|x - x^0\|_B^2$$

$$\text{Subject to } Ax = b$$

$$\|x\|_B = \sqrt{x^\top B x}$$



Dual Problem

Recall convex conjugate:

$$f^*(z) \overset{\text{def}}{=} \sup_{x \in \mathbb{R}^d} \{\langle z, x\rangle - f(x)\}$$

$$f_i(t) = 1_{\{b_i\}}(t) \qquad f_i^*(t) = b_i t$$

$$g(x) = \frac{1}{2}\|x - x^0\|_B^2 \qquad g^*(x) = \langle x^0, x\rangle + \frac{1}{2}\|x\|_{B^{-1}}^2$$

$$\max_{y \in \mathbb{R}^n} \left[D(y) \overset{\text{def}}{=} \left\langle b - Ax^0, \frac{y}{n} \right\rangle - \frac{1}{2}\left\| A^\top \frac{y}{n} \right\|_{B^{-1}}^2 \right]$$

Unconstrained (non-strongly) concave quadratic maximization

Recovering Primal Solution from Dual Solution

Recall:

$$x^* = \nabla g^* \left(\tfrac{1}{n} A^\top y^*\right)$$

$$g^*(x) = \langle x^0, x\rangle + \tfrac{1}{2}\|x\|^2_{B^{-1}}$$

$$\Downarrow$$

$$\nabla g^*(x) = x^0 + B^{-1}x$$

$$\Downarrow$$

$$x^* = x^0 + \tfrac{1}{n} B^{-1} A^\top y^*$$

Further Reading on Randomized Methods for Linear Systems

Primal View:

Robert M. Gower and P.R.
Randomized Iterative Methods for Linear Systems
SIAM J. on Matrix Analysis and Applications 36(4), 1660-1690, 2015

Most Downloaded SIMAX Paper

Dual View:

Robert M. Gower and P.R.
Stochastic Dual Ascent for Solving Linear Systems
arXiv:1512.06890, 2015

Inverting Matrices & Connection to Quasi-Newton Methods:

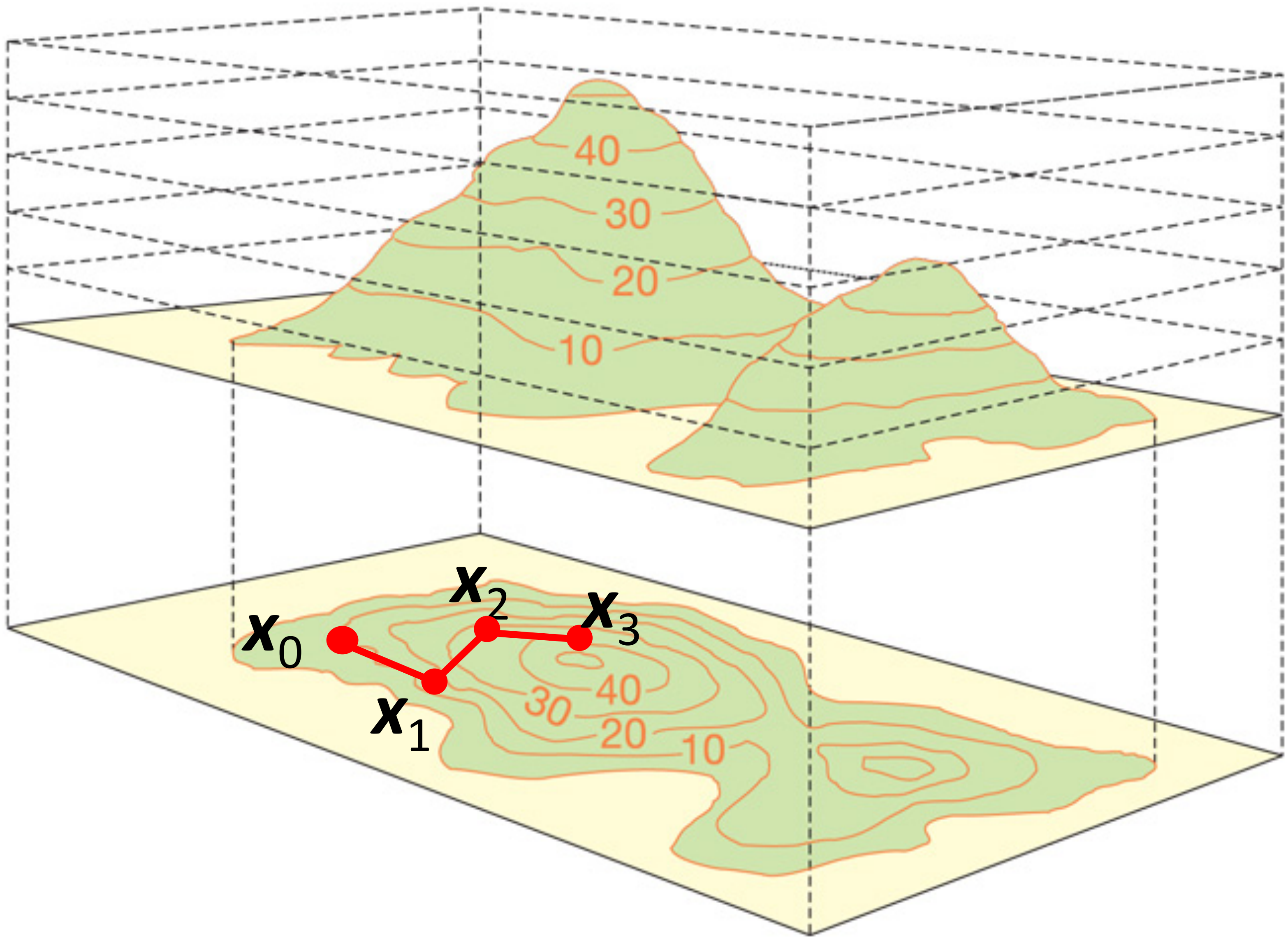
Robert M. Gower and P.R.
Randomized Quasi-Newton Updates are Linearly Convergent Matrix Inversion Algorithms
arXiv:1602.01768, 2016

Part 2
Standard
Algorithmic Toolbox


Optimization with Big Data
= Extreme* Mountain Climbing
* in a billion dimensional space on a foggy day

God’s Algorithm = Teleportation

Mortals Have to Walk...



Algorithmic Tools

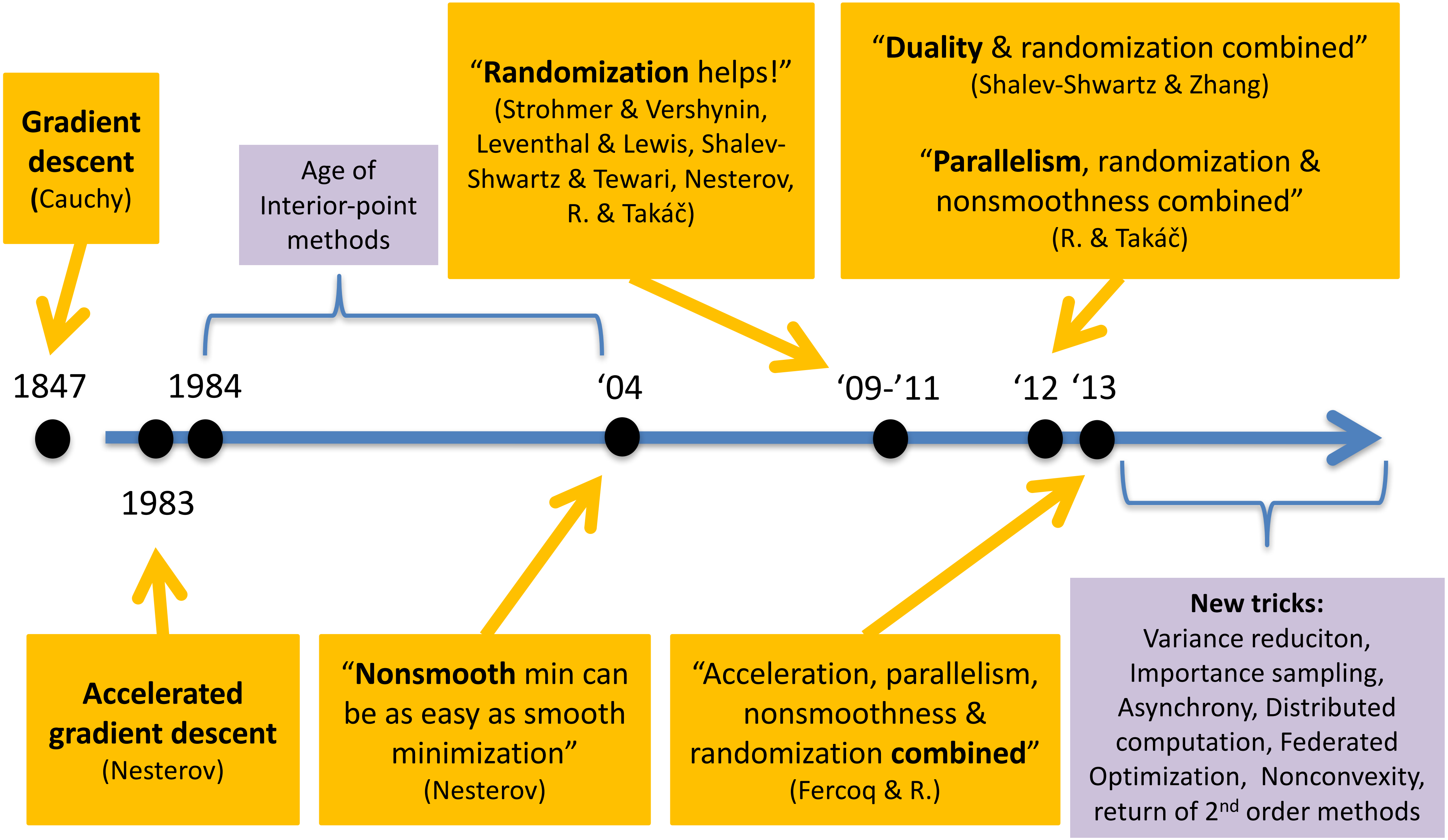
1. Gradient descent
2. Handling non-smoothness via the proximal trick
3. Acceleration
4. Randomized decomposition
5. Parallelism / mini-batching

More tools:

- Variance reduction
- Importance sampling
- Asynchrony
- Curvature
- Line search

All these tools can be combined!

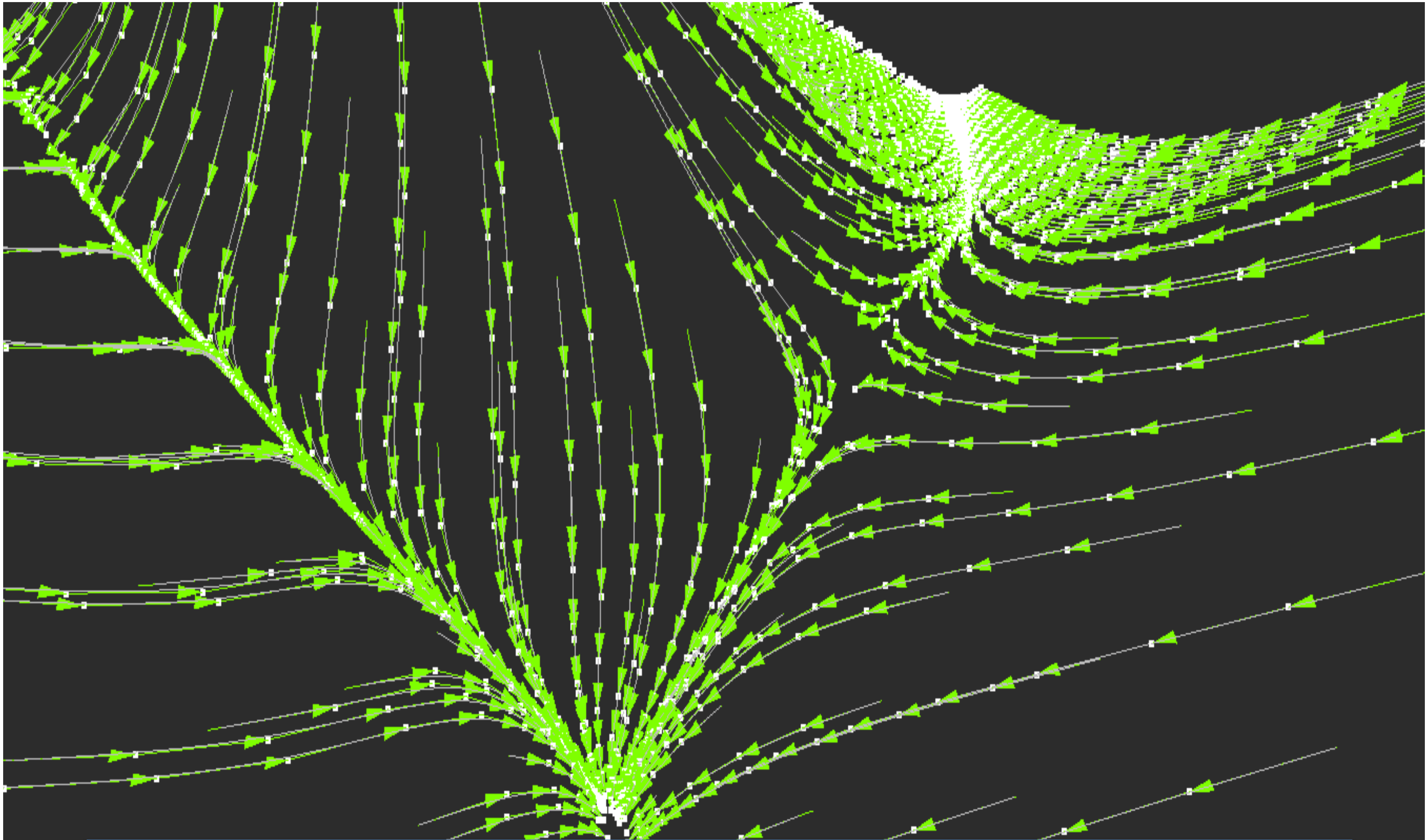
Brief, Biased and Severely Incomplete History of Big Data Optimization



Tool 1

Gradient Descent (1847)

“Just follow a ball rolling down the hill”



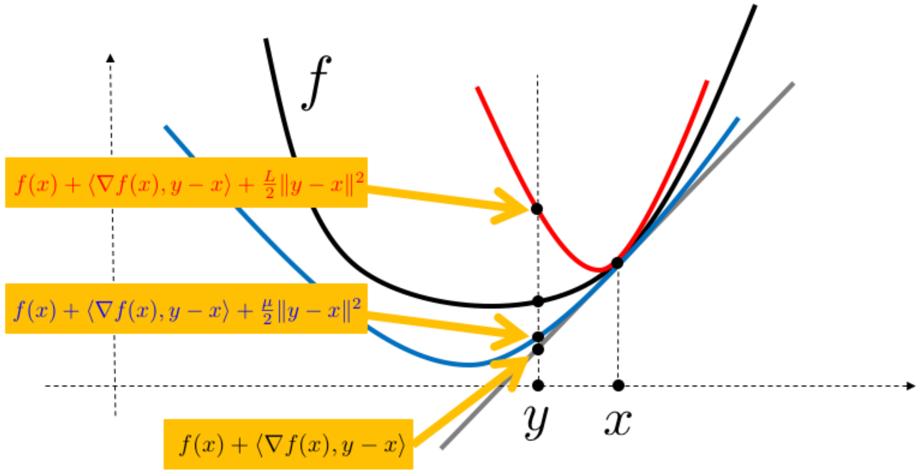
Augustin Cauchy

Méthode générale pour la résolution des systèmes d'équations simultanées, *pp. 536–538,* 1847

The Problem

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

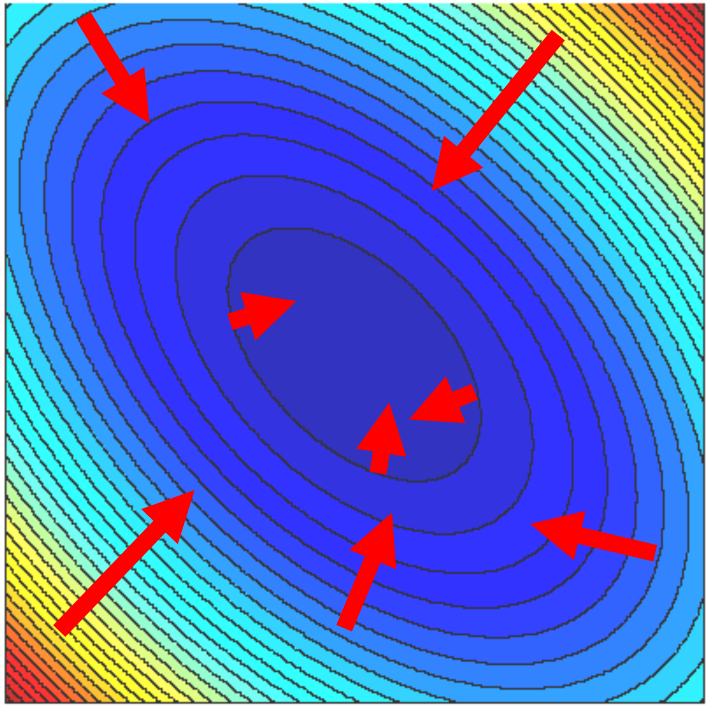
L-smooth, μ-strongly convex



$$f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$$

Gradient Descent (GD)

$$x^{t+1} = x^t - \frac{1}{L}\nabla f(x^t)$$



iterations

condition number of f

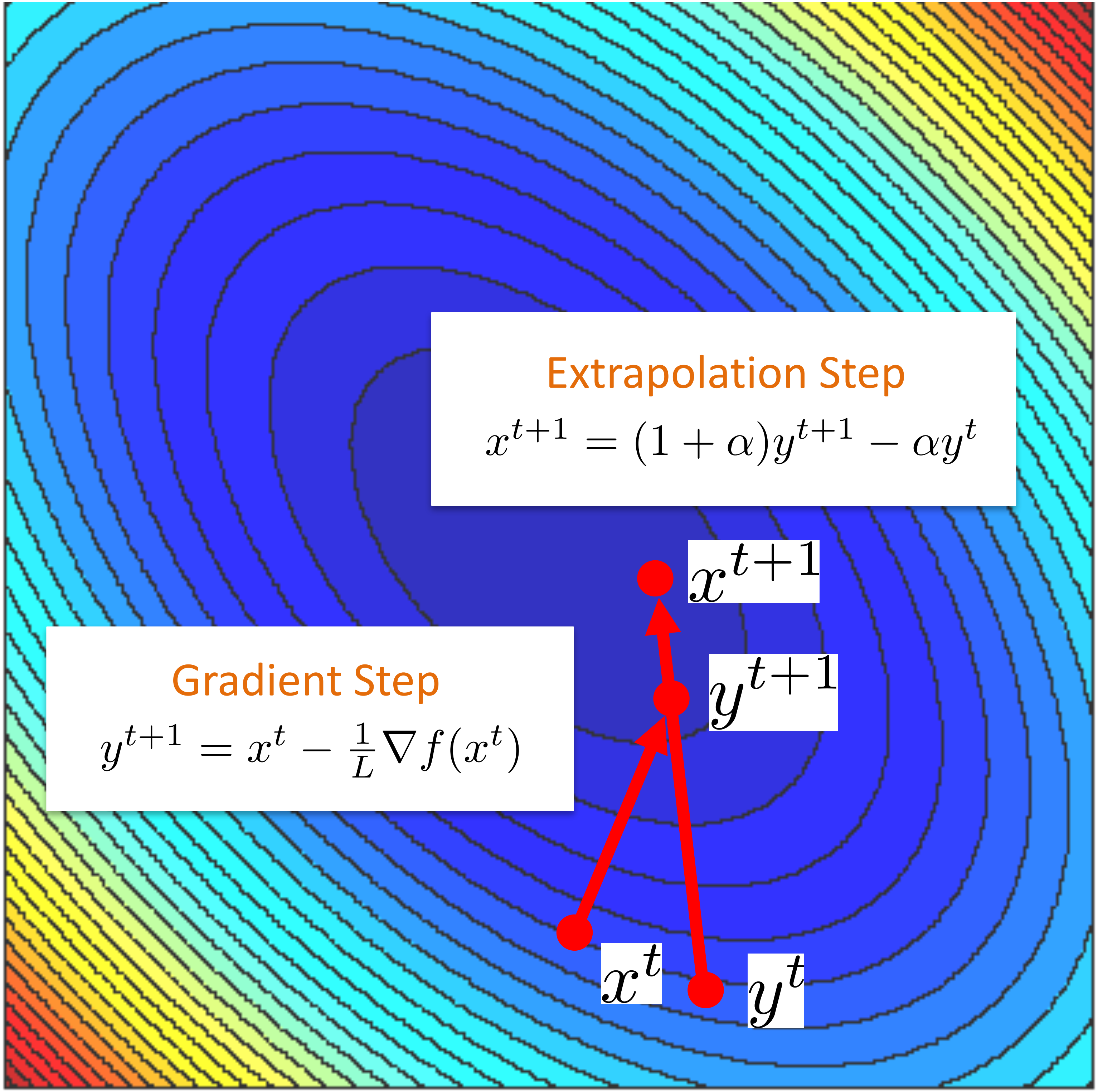
$$t \geq \frac{L}{\mu} \log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right)$$

$$f(x^t) - f(x^*) \leq \epsilon$$

Tool 2

Acceleration (1983/2003)

"Gradient descent can be made much faster!"



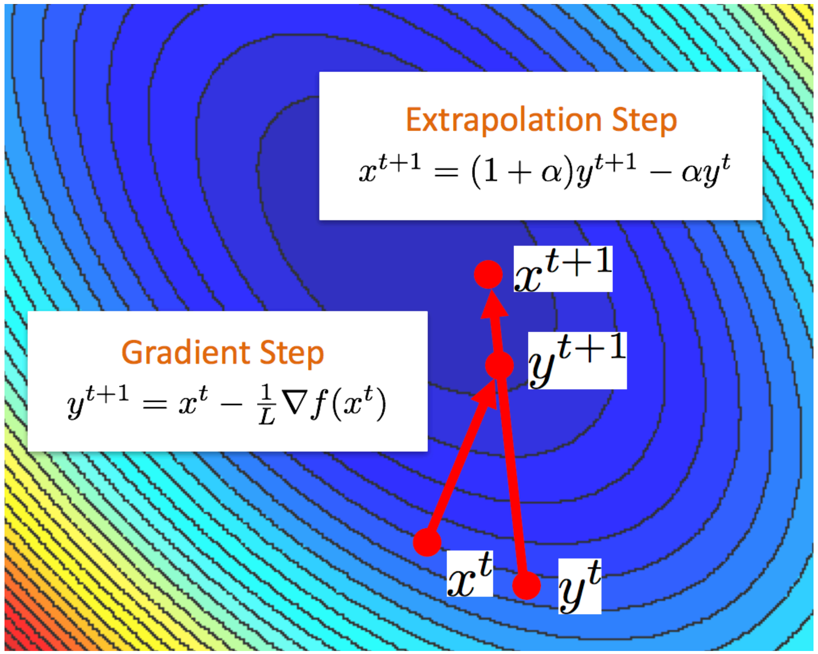
Extrapolation Step
$x^{t+1} = (1+\alpha)y^{t+1} - \alpha y^t$
Gradient Step
$y^{t+1} = x^t - \frac{1}{L}\nabla f(x^t)$
x^{t+1}
y^{t+1}
x^t
y^t

Accelerated Gradient Descent (AGD)

Gradient step: $$y^{t+1} = x^t - \frac{1}{L}\nabla f(x^t)$$

Extrapolation: $$x^{t+1} = (1+\alpha)y^{t+1} - \alpha y^t$$

$$\alpha = \frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}$$



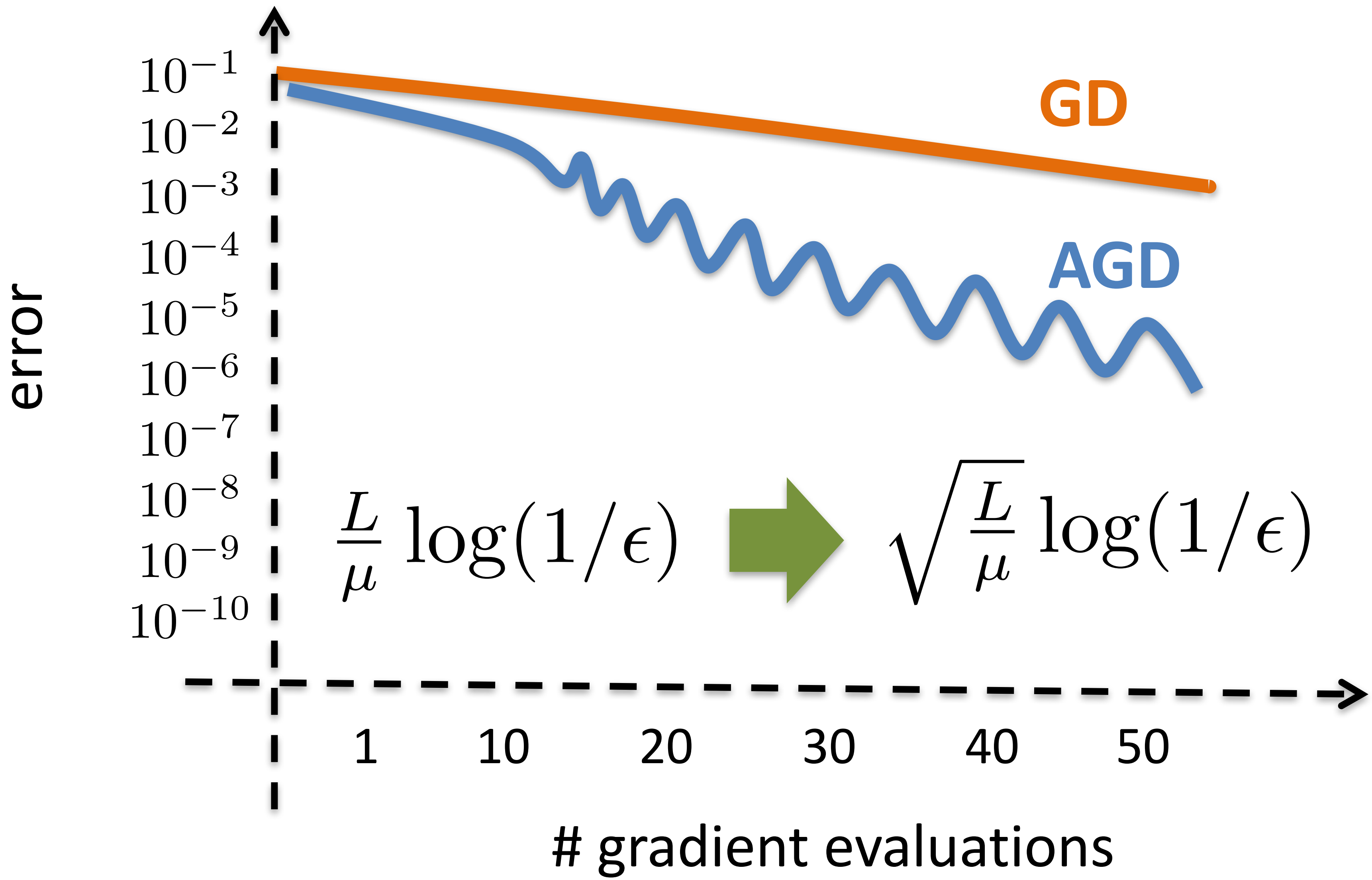
iterations

Square root of the condition number of *f*

$$t \geq \sqrt{\frac{L}{\mu}} \log\left(\frac{C}{\epsilon}\right)$$

$$f(x^t) - f(x^*) \leq \epsilon$$

Acceleration Works (Somewhat Mysteriously)



Acceleration and ODEs

ODE for Gradient Descent

$$\dot{X}(t) + \nabla f(X(t)) = 0$$

ODE for Accelerated Gradient Descent

$$\ddot{X}(t) + \frac{3}{t}\dot{X}(t) + \nabla f(X(t)) = 0$$

Weijie Su, Stephen Boyd and Emmanuel J. Candes
A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights
NIPS, 2014

Acceleration

- Reignited interest in gradient methods
- Called momentum in deep neural networks literature
- Oscillation can be tamed (e.g., by restarting)
- Approaches:
  - Early work [Nesterov, 1983, 2003, 2005]
  - ODEs [Su-Boyd-Candes, 2014]
  - Geometry/ellipsoid method [Bubeck-Lee-Singh, 2014]
  - Linear coupling [AllenZhu-Orecchia, 2014]
  - Katalyst [Mairal-Zarchaoui, 2015]
  - Optimal averaging [Scieur-D'Aspremont-Bach, 2016]

PDF Adobe

Yurii Nesterov
Introductory Lectures on Convex Optimization: a Basic Course
Kluwer, Boston, 2003

Strongly convex case

PDF Adobe

Yurii Nesterov
A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence O(1 / k^2)
Soviet Math. Doklady 269, 543-547, 1983

Weakly convex case

Tool 3

Proximal Trick (2004)

“Some nonsmooth problems are as easy as smooth problems”

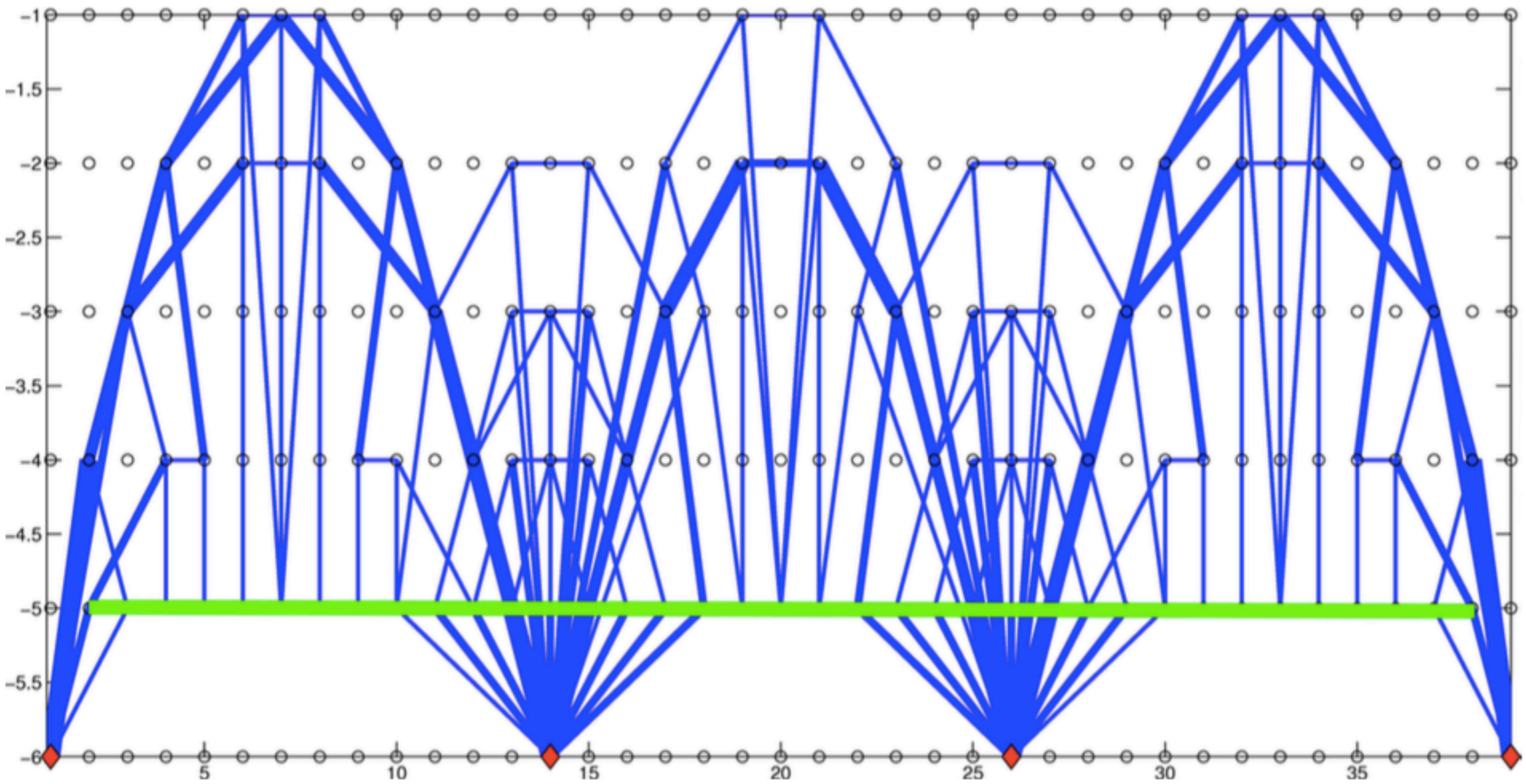
The Problem

$$\min_{x \in \mathbb{R}^d} f(x) + g(x)$$

L-smooth, convex

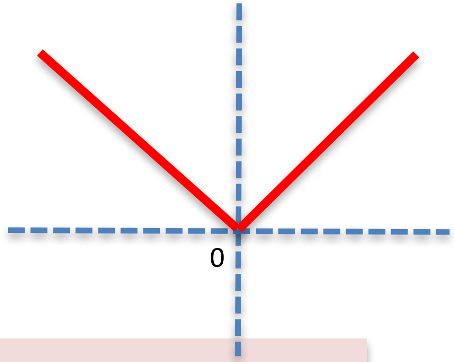
Convex,
but can be
nonsmooth

Truss Topology Design



P.R. and Martin Takáč. **Efficient Serial and Parallel Coordinate Descent Methods for Huge-Scale Truss Topology Design.** *Operations Research Proceedings*, pp 27-32, 2012

Truss Topology Design: “LASSO” Problem



Encodes all potential bars

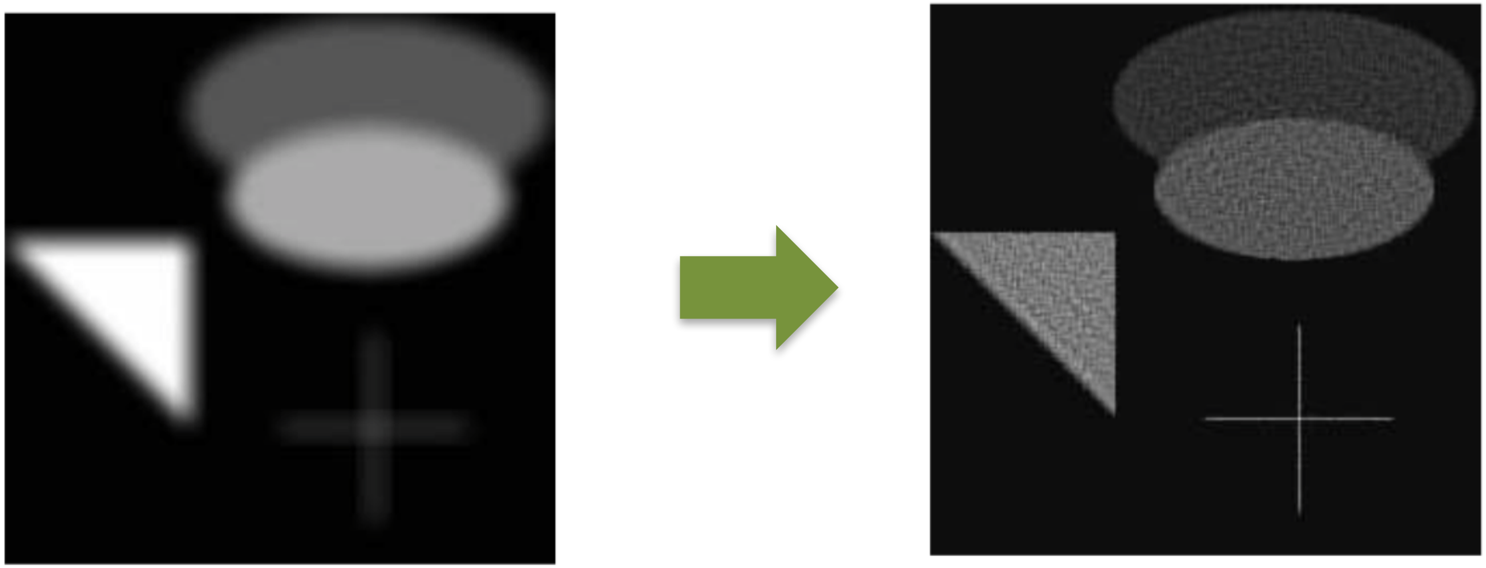
$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{2}\|Ax-b\|_2^2}_{\text{Least-squares (convex, smooth, quadratic)}} + \underbrace{\lambda\|x\|_1}_{\text{L1 norm (convex, nonsmooth, but “simple”)}}$$

potential bars (quadratic in mesh size)

Least-squares (convex, smooth, quadratic)

L1 norm (convex, nonsmooth, but “simple”)

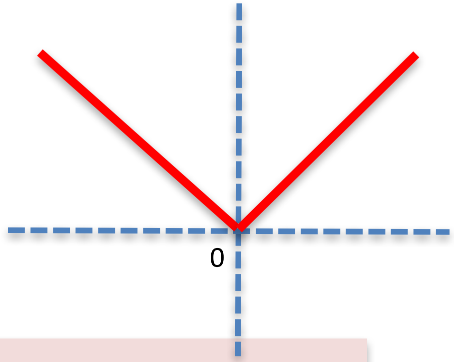
Image Deblurring



Amir Beck and Marc Teboulle. **A Fast Iterative Shrinking-Thresholding Algorithm for Linear Inverse Problems.** *SIAM J. Imaging Sciences* 2(1), 183-202, 2009

Jakub Konečný, Jie Liu, P.R., Martin Takáč. **Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting.** *IEEE Journal of Selected Topics in Signal Processing* 10(2), 242-255, 2016

Image Deblurring: “LASSO” Problem



blurred image

$$\min_{x \in \mathbb{R}^d} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1$$

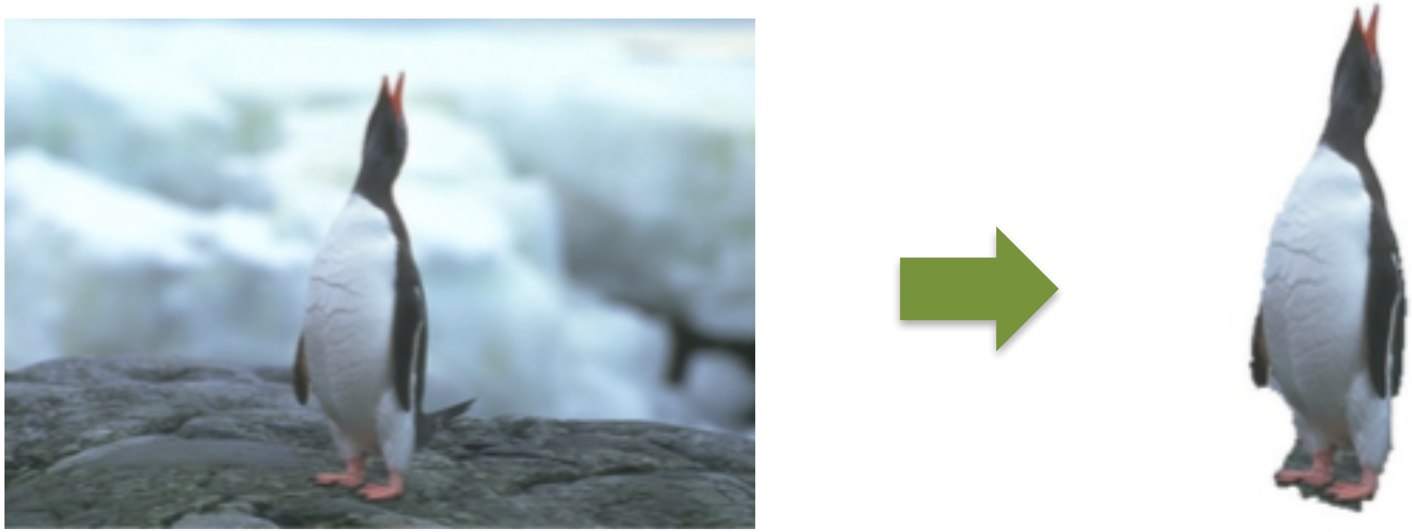
image

pixels in the image

Blurring matrix multiplied by a wavelet basis matrix

Encourages sparsity in the wavelet basis

Image Segmentation



Alina Ene and Huy L. Nguyen. **Random Coordinate Descent Methods for Minimizing Decomposable Submodular Functions.** *ICML* 2015

Olivier Fercoq and P.R. **Accelerated, Parallel and Proximal Coordinate Descent.** *SIAM Journal on Optimization* 25(4), 1997-2023, 2015

Image Segmentation: (Reformulated) Submodular Optimization

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\left\|\sum_{i=1}^{d} x_i\right\|^2 \\ \text{subject to} \quad & x_i \in P_i,\ i = 1, 2, \ldots, d \end{aligned}$$

Smooth, convex, quadratic

polytope

grows with the image size

Image Segmentation: (Reformulated) Submodular Optimization

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|\sum_{i=1}^{d} x_i\right\|^2 \\ \text{subject to} & x_i \in P_i,\ i = 1, 2, \ldots, d \end{array}$$

?

$$\min_{x \in \mathbb{R}^d} f(x) + g(x)$$

$$f(x) = \frac{1}{2}\left\|\sum_{i=1}^{d} x_i\right\|^2$$

$$g(x) = 1_{P_1 \cap P_2 \cap \cdots \cap P_d}(x) = \sum_{i=1}^{d} 1_{P_i}(x) = \begin{cases} 0 & x \in P_1 \cap P_2 \cap \cdots \cap P_d, \\ +\infty & \text{otherwise.} \end{cases}$$

Proximal Gradient Descent (PGD)

STEP 1: Pretend there is no regularizer

$$z^{t+1} = x^t - \tfrac{1}{L}\nabla f(x^t)$$

STEP 2: Take a “proximal” step with respect to *g*

$$x^{t+1} = \arg\min_{x\in\mathbb{R}^d} \frac{1}{2}\|x - z^{t+1}\|_2^2 + \frac{1}{L}g(x)$$

- Gradient Descent is a special case for *g* = 0
- Even though this is a nonsmooth problem, # steps is the same as for Gradient Descent!
- Efficient if Step 2 is easy to do

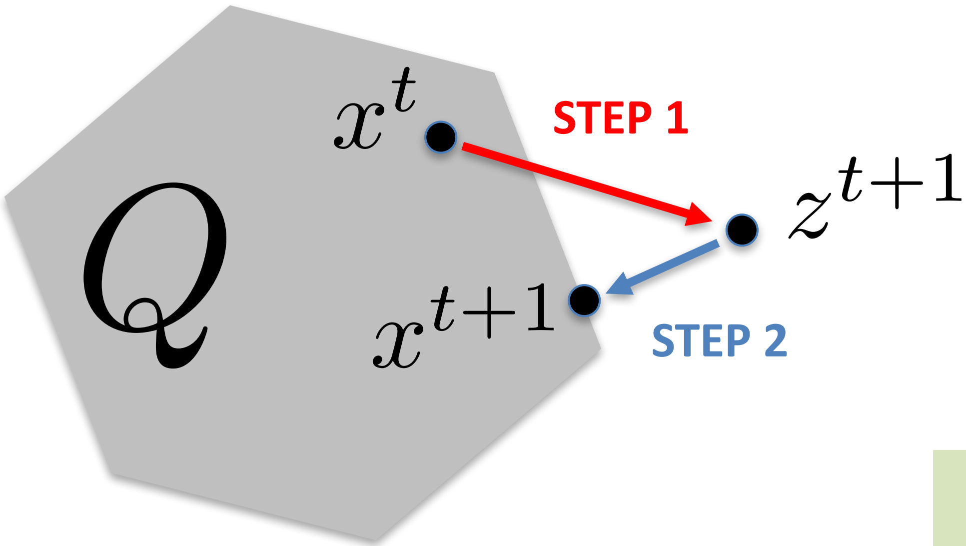
$$\frac{L}{\mu}\log(1/\epsilon)$$

Example: Projected Gradient Descent

$$\min_{x \in Q} f(x) \quad \Leftrightarrow \quad \min_x f(x) + g(x)$$

Convex set

$$g(x) = 1_Q(x) \stackrel{\text{def}}{=} \begin{cases} 0 & x \in Q \\ +\infty & x \notin Q \end{cases}$$



$$z^{t+1} = x^t - \frac{1}{L}\nabla f(x^t)$$

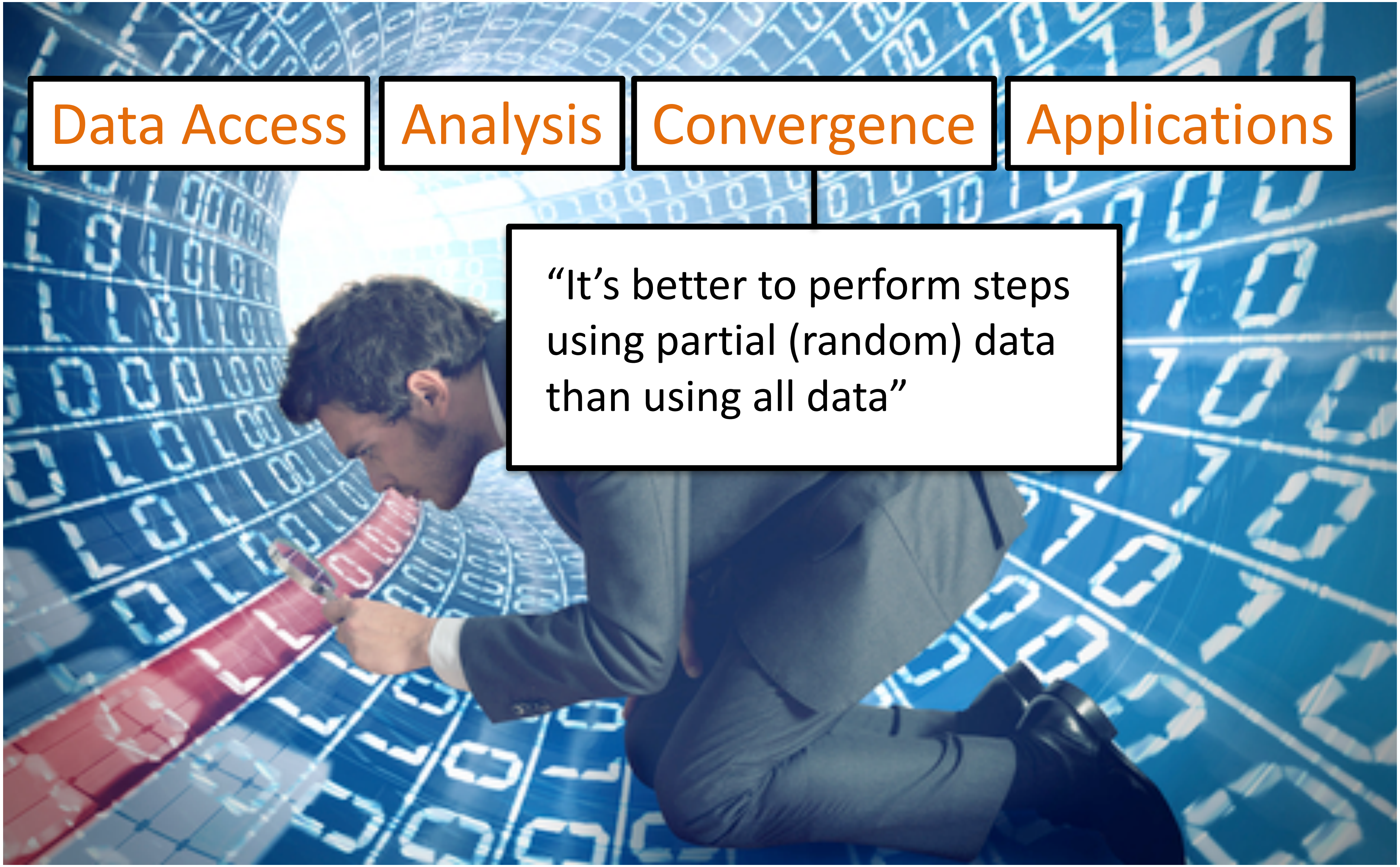
$$x^{t+1} = \arg\min_{x \in \mathbb{R}^d} \frac{1}{2}\|x - z^{t+1}\|_2^2 + \frac{1}{L} g(x)$$

Tool 4

Randomized Decomposition

“Doing many simple decisions is better than doing a few smart ones”

Why Randomize?



Decomposition Principles

$$\min_{x \in Q} f(x)$$

Decompose f

additive: $f = \sum_i f_i$

Example:
Stochastic Gradient Descent

Decompose Q

additive: $Q = \mathbb{R}^d = \bigoplus_{i=1}^s Q_i$

Example:
Randomized Coordinate Descent

multiplicative: $Q = \bigcap_{i=1}^s Q_i$

Example:
Stochastic Projection Method

Primal ERM Problem: Stochastic Gradient Descent

H. Robbins and S. Monro
A Stochastic Approximation Method
Annals of Mathematical Statistics 22, pp. 400–407, 1951

The Problem

n is big

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

$$\min_{x \in \mathbb{R}^d} \quad \left[P(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^\top x) + \cancel{g(x)} \right]$$

Stochastic Gradient Descent (SGD)

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right\}$$

stepsize

$$x^{t+1}=x^t-h^t\nabla f_i(x^t)$$

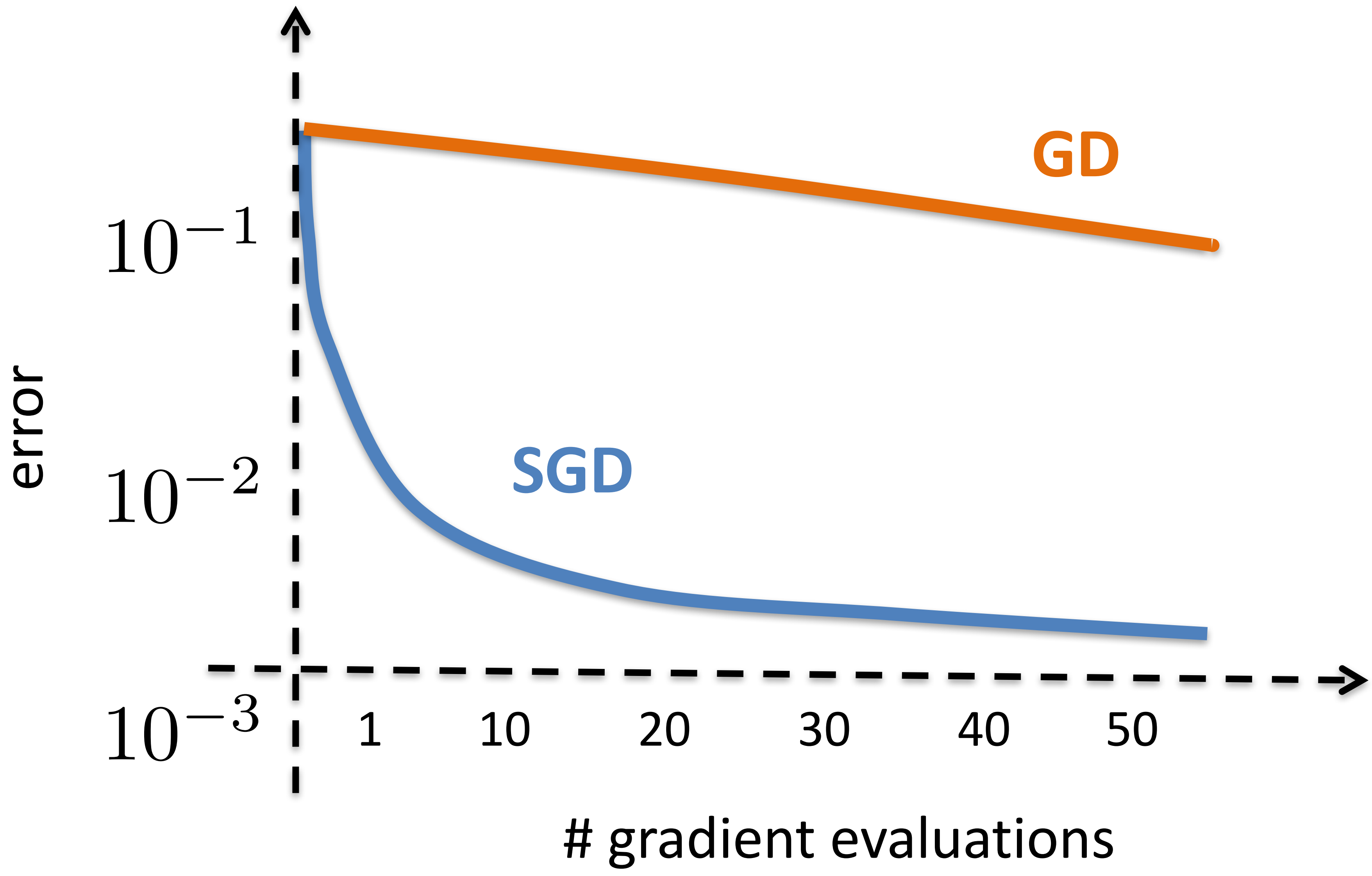
$$\mathbf{E}[\nabla f_i(x)]=\nabla f(x)$$

Unbiased estimate of the gradient

i = chosen uniformly at random

1 iteration of SGD is *n* times cheaper than 1 iteration of GD !

Stochastic Gradient Descent vs Gradient Descent



Dual ERM Problem: Randomized Coordinate Descent

Yurii Nesterov
Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems
SIAM Journal on Optimization, 22(2), 341–362, 2012

P.R. and Martin Takáč
Iteration Complexity of Randomized Block Coordinate Descent Methods for Minimizing a Composite Function
Mathematical Programming 144(2), 1-38, 2014 *(arXiv:1107.2848)*

INFORMS Computing Society Best Student Paper Prize (runner up), 2012

How to Handle Big Dimensions?

Primal ERM:

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x) + g(x) \right]$$

What if *d* is big?

Dual ERM:

$$\max_{y\in\mathbb{R}^n} \left[D(y) \stackrel{\text{def}}{=} -\frac{1}{n}\sum_{i=1}^{n} f_i^*(-y_i) - g^*\left(\frac{1}{n}A^\top y\right) \right]$$

What if *n* is big?

Solution:

Decompose the dimension!

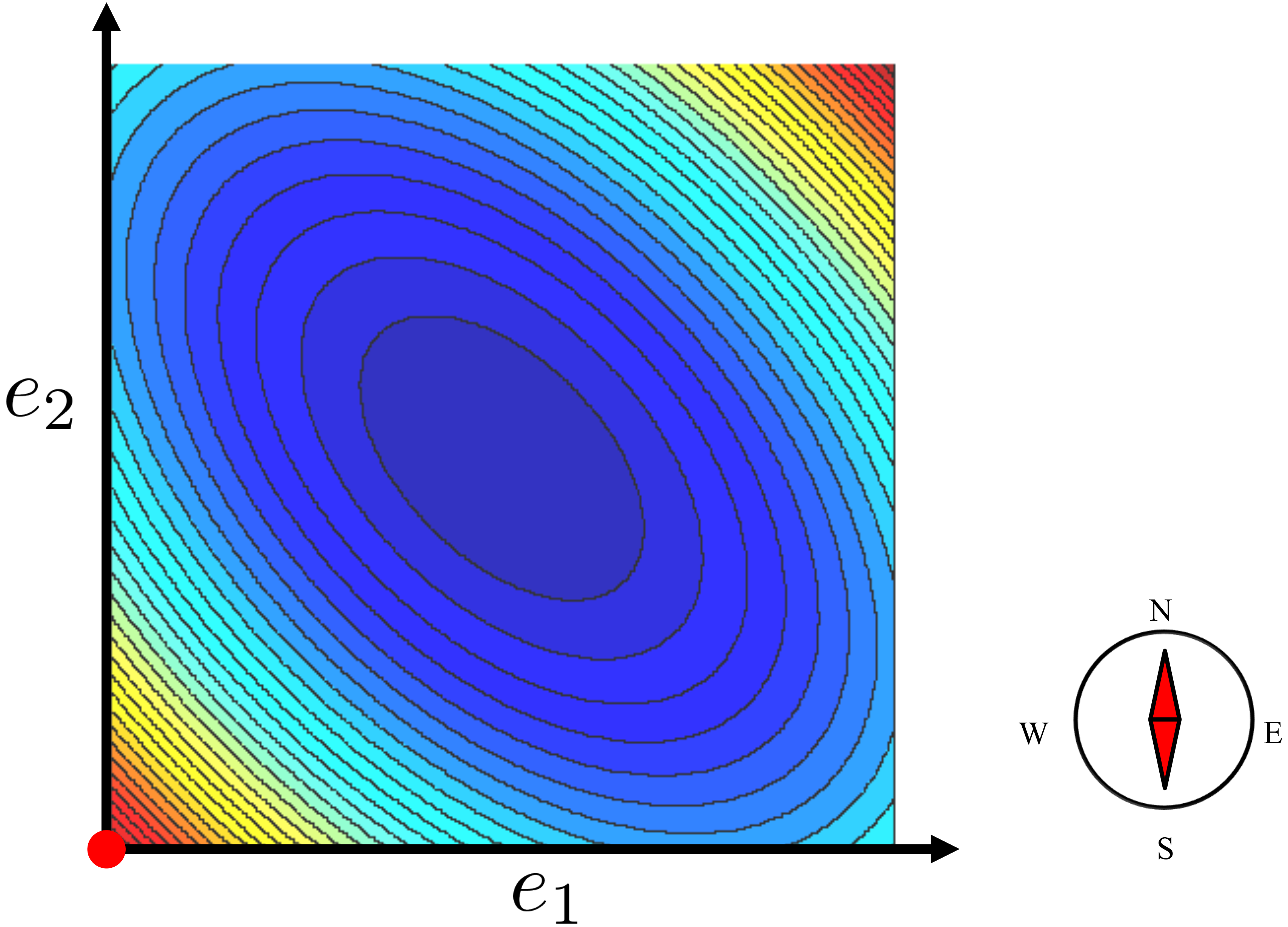
The Problem

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

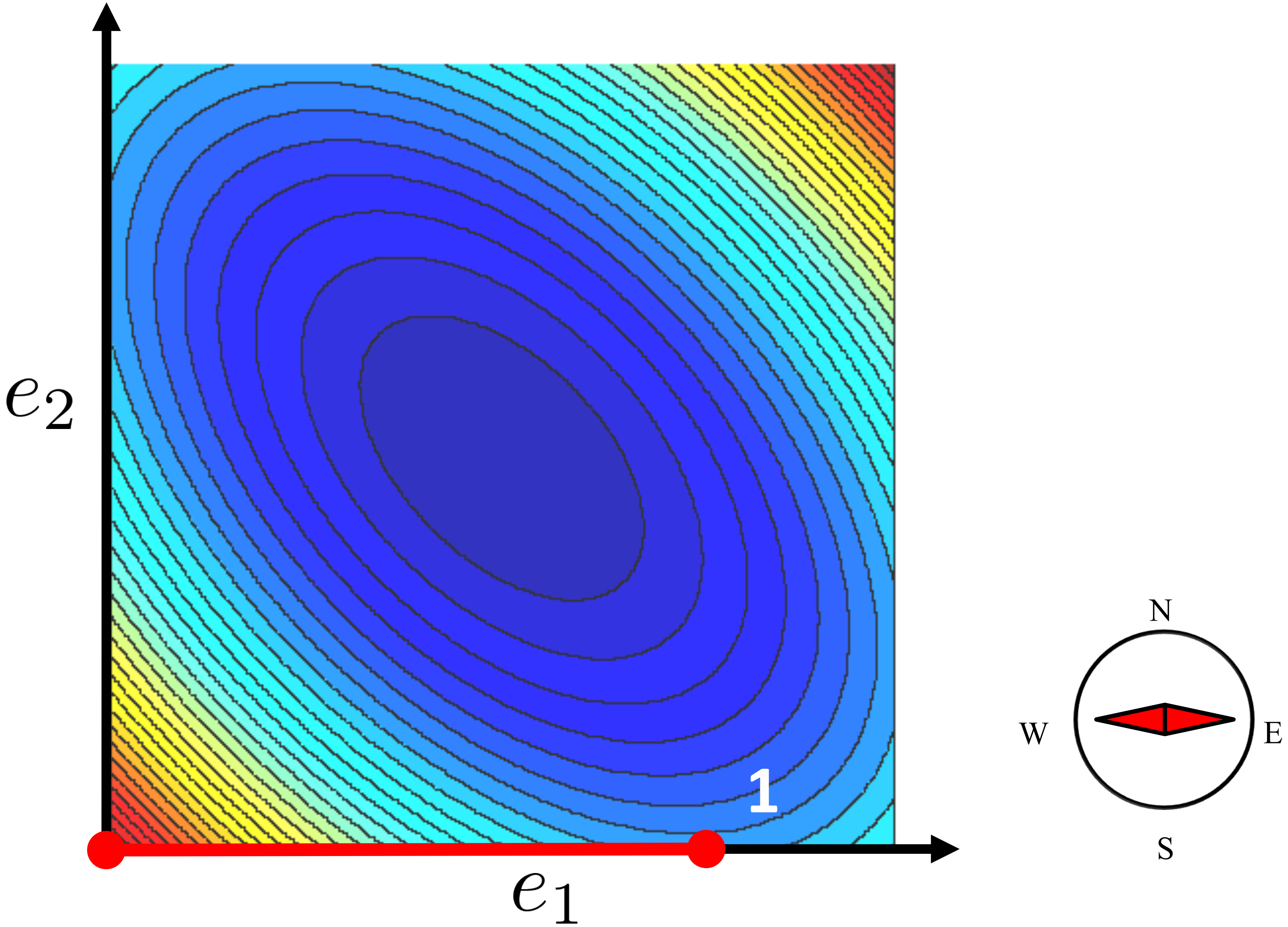
n is BIG

L-smooth, μ-strongly convex

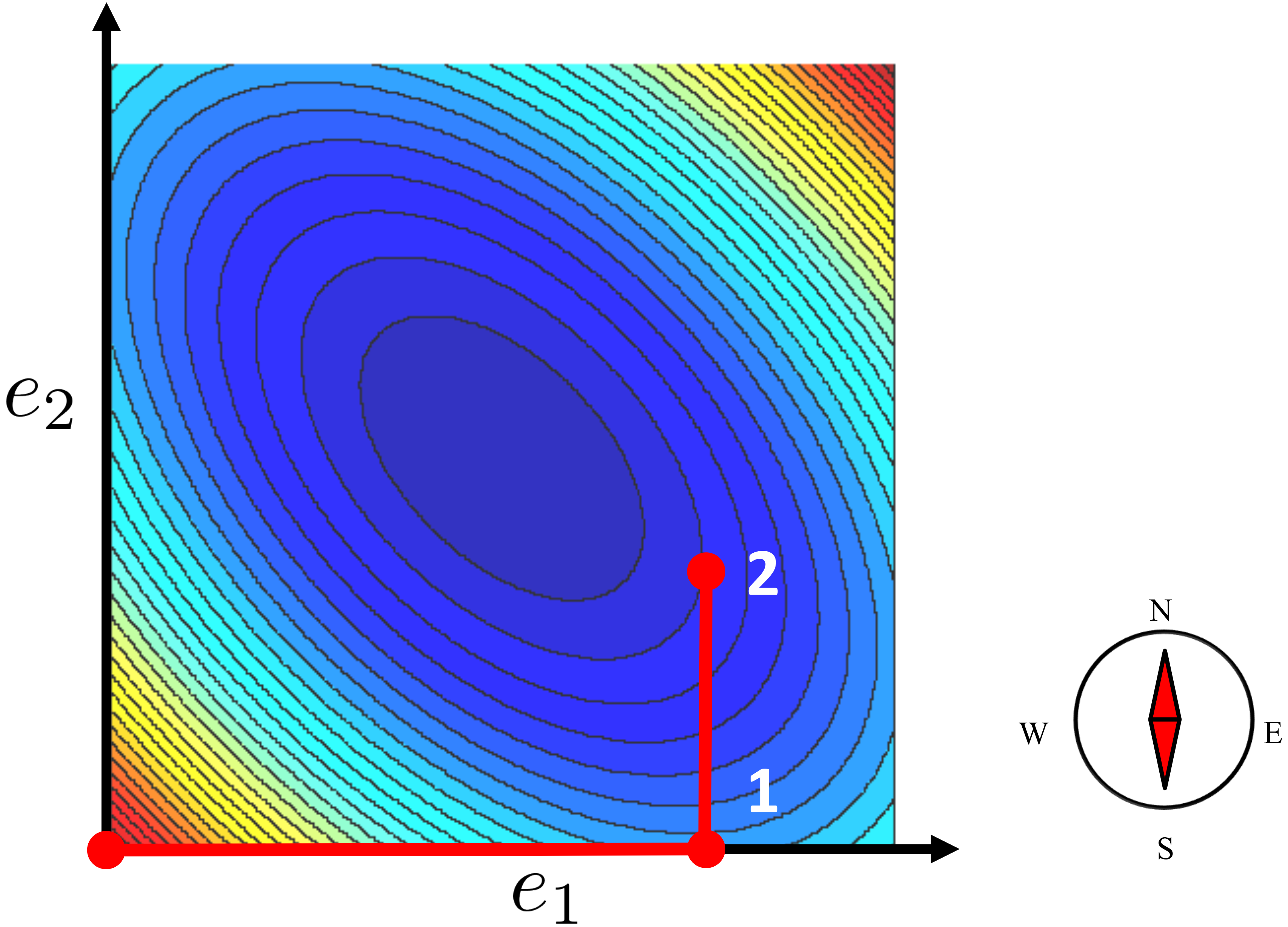
Randomized Coordinate Descent in 2D



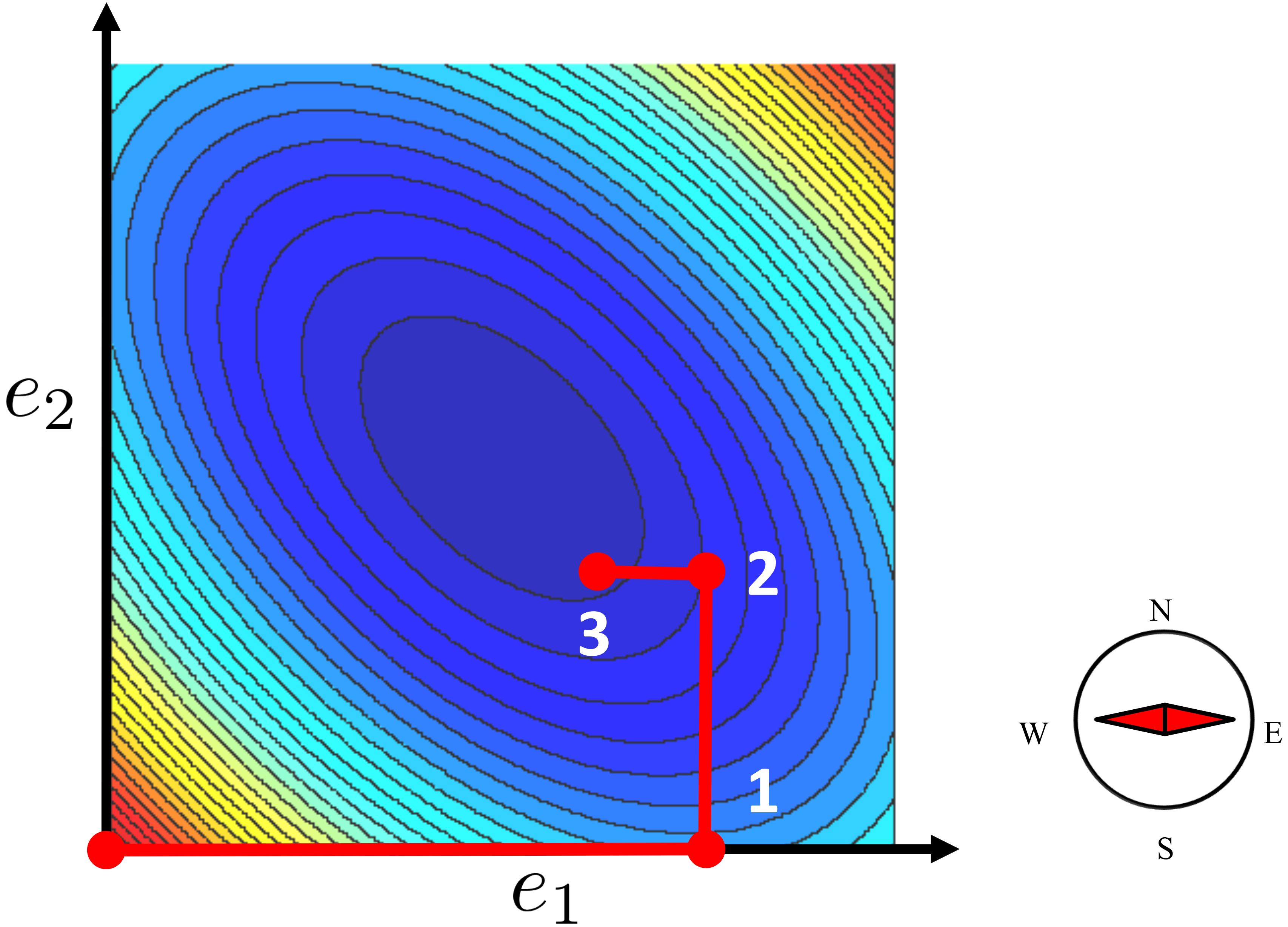
Randomized Coordinate Descent in 2D



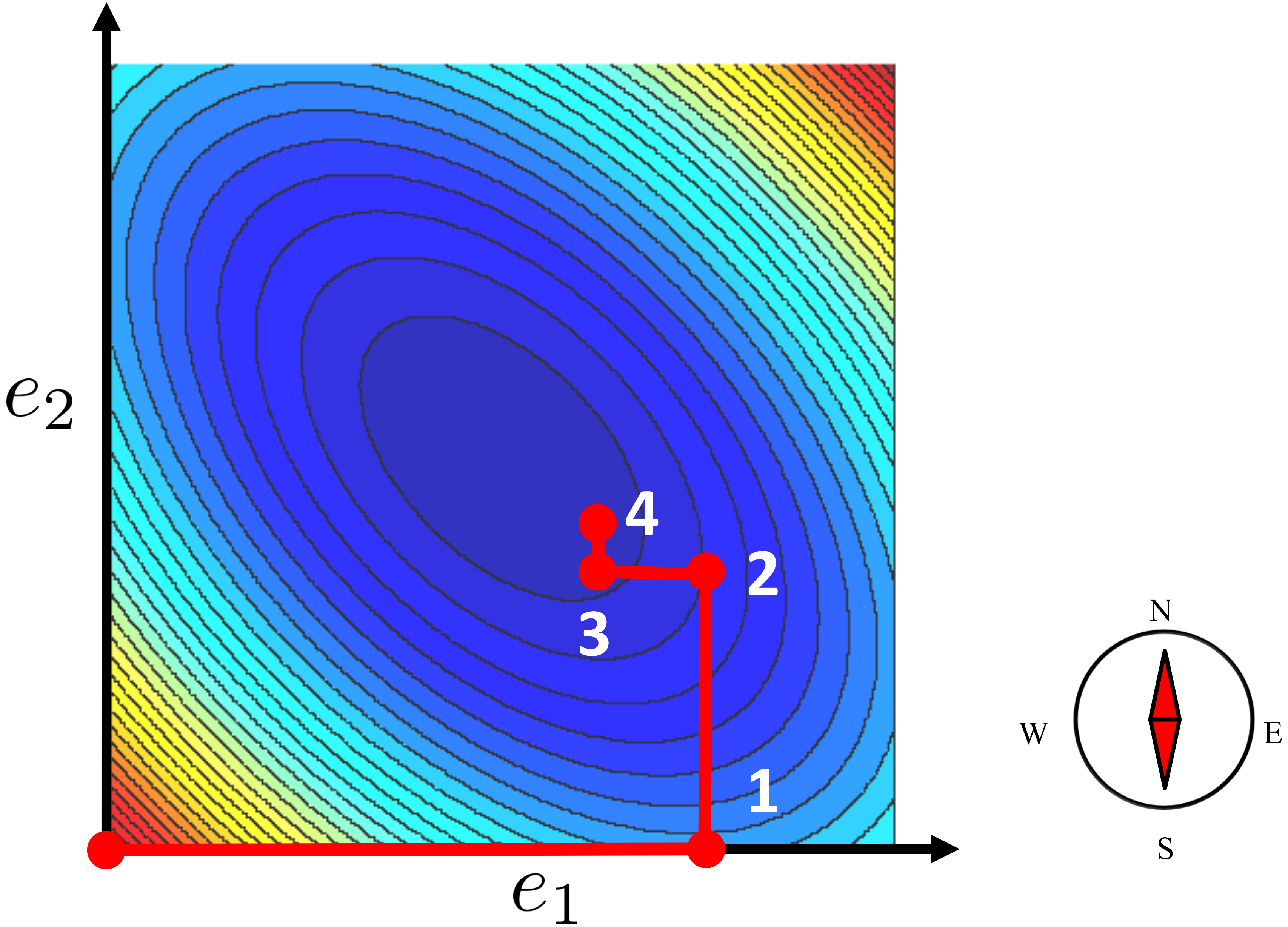
Randomized Coordinate Descent in 2D



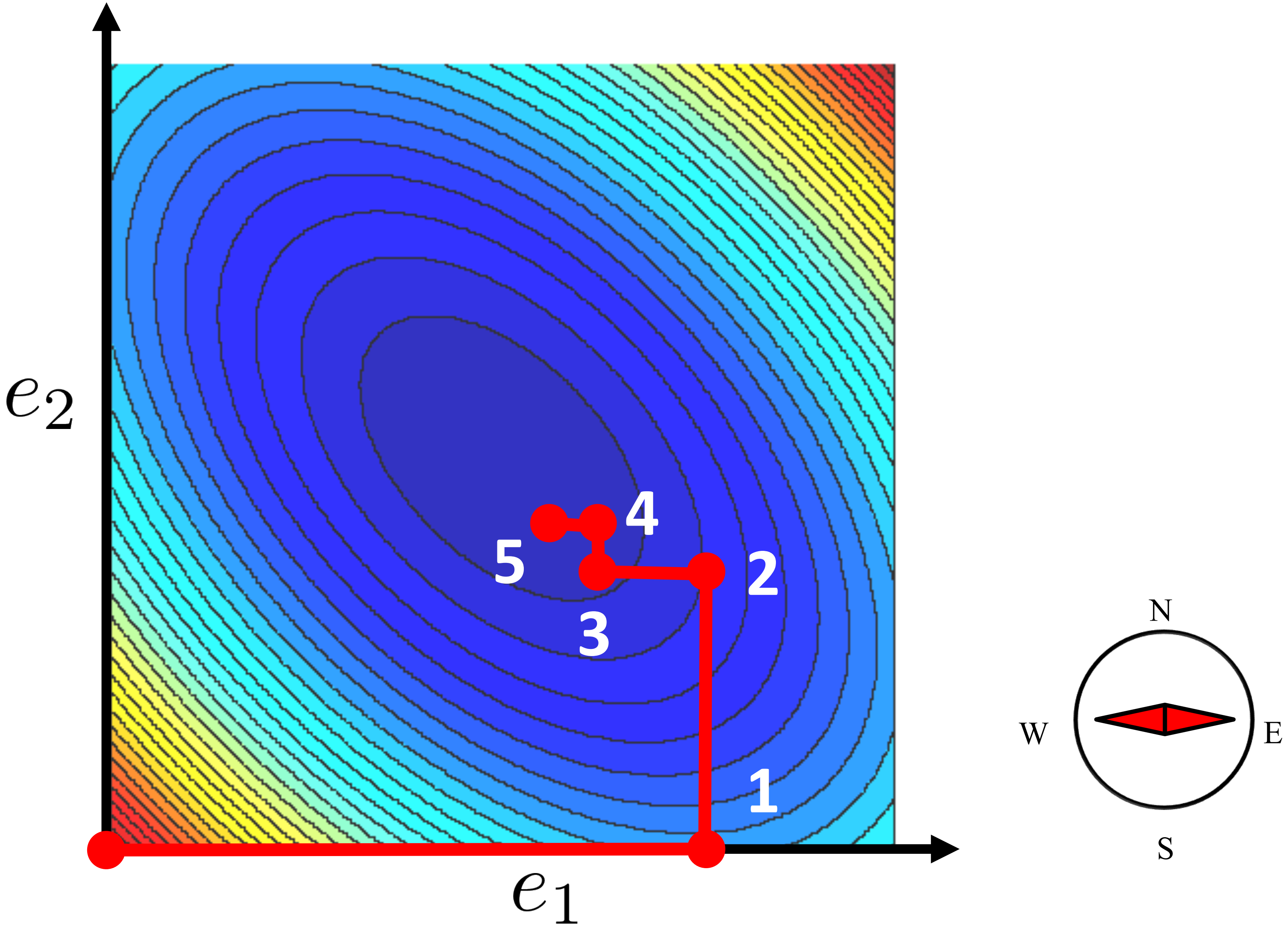
Randomized Coordinate Descent in 2D



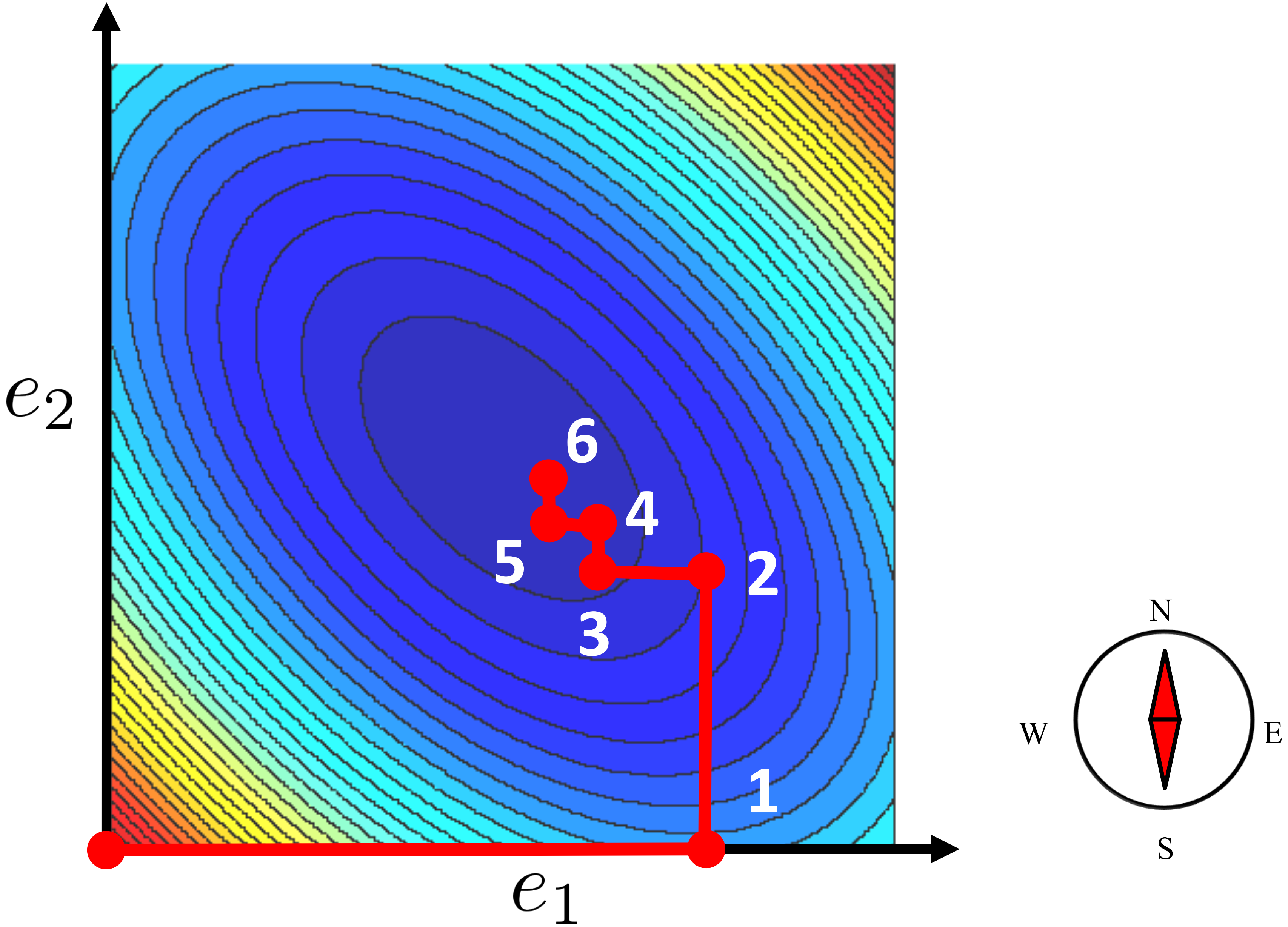
Randomized Coordinate Descent in 2D



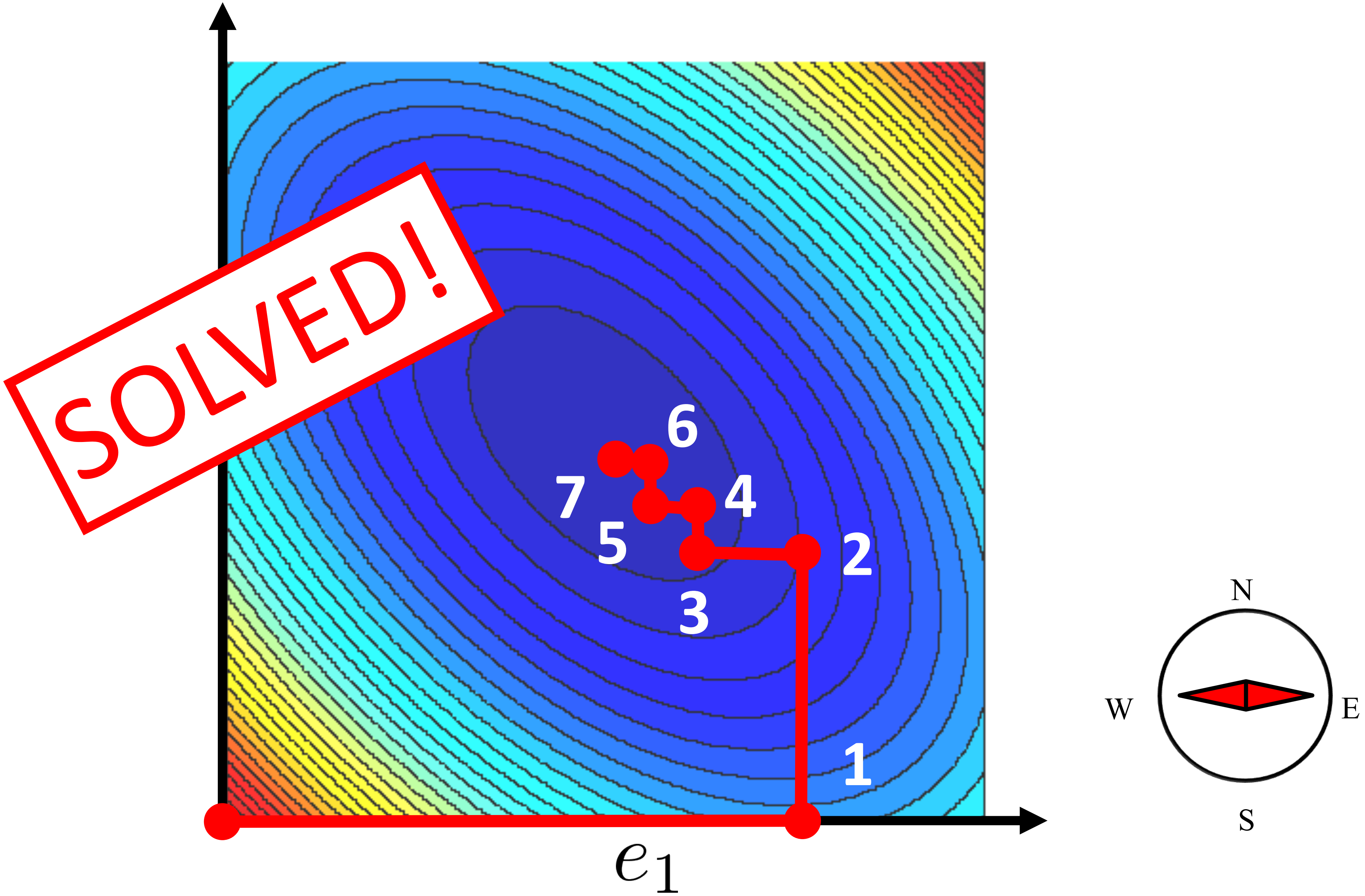
Randomized Coordinate Descent in 2D



Randomized Coordinate Descent in 2D



Randomized Coordinate Descent in 2D



Randomized Coordinate Descent

Partial derivative of f

i^{th} standard unit basis vector in $\mathbb{R}^n$

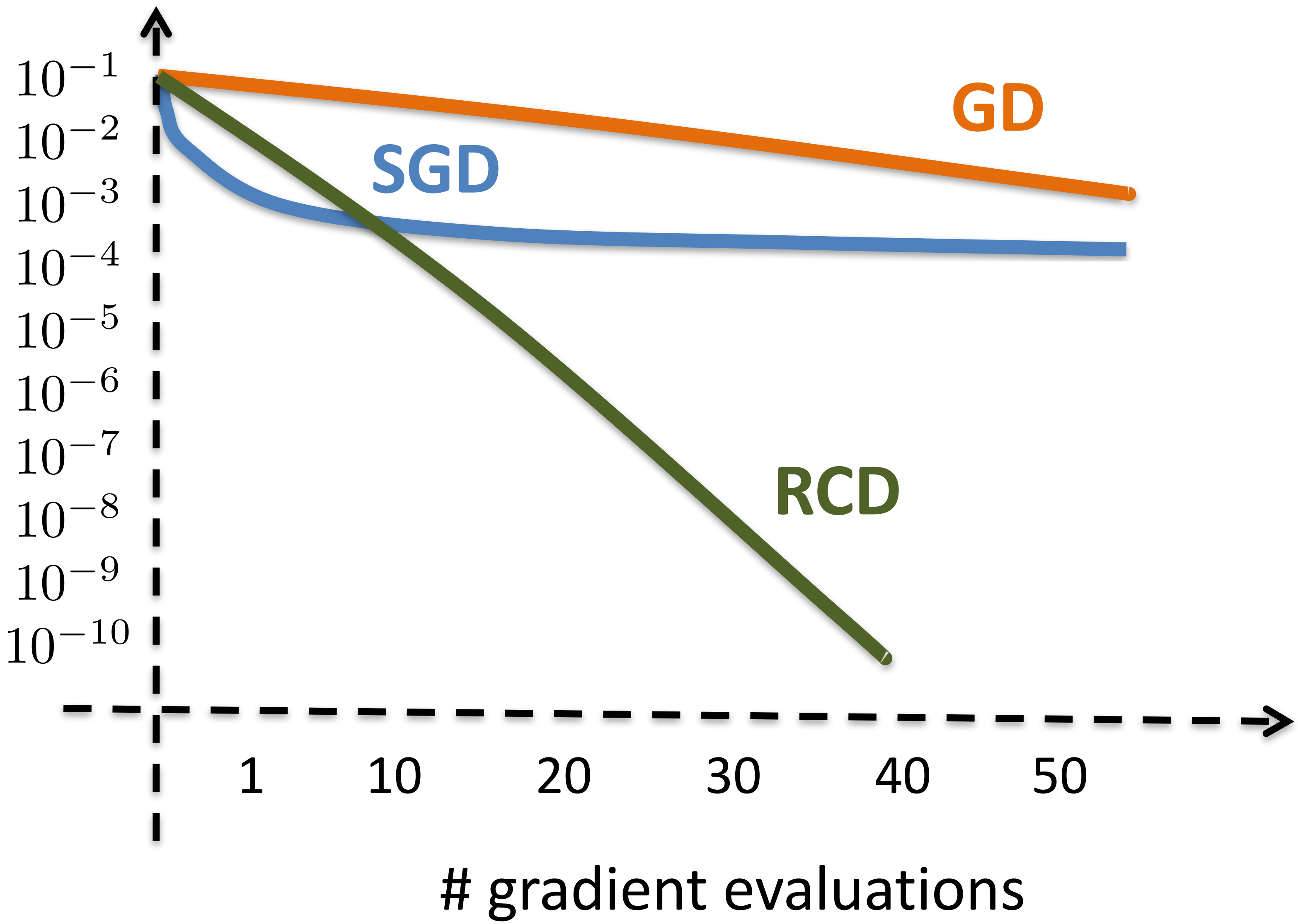
$$x^{t+1} = x^t - \frac{1}{L_i} \nabla_i f(x^t) e_i$$

f is L_i-smooth along e_i:
$|\nabla_i f(x + te_i) - \nabla_i f(x)| \leq L_i |t|$

Often, each iteration is n times cheaper.
However, complexity is not n times worse!
So, RCD is better than GD!

$$t \geq \left(\frac{\max_i L_i}{\mu} \right) \log \left(\frac{C}{\epsilon} \right) \Rightarrow \mathbf{E}[f(x^t) - f(x^*)] \leq \epsilon$$

SGD vs GD vs RCD



LASSO: 1 Billion Rows & 100 Million Variables

source: [R. & Takáč, arXiv 2011, MAPR 2014]

$$A \in \mathbf{R}^{10^9 \times 10^8}$$

t/n	error	# nonzeros in x_k	time [s]
0.01	$< 10^{18}$	18,486	1.32
9.35	$< 10^{14}$	99,837,255	1294.72
11.97	$< 10^{13}$	99,567,891	1657.32
14.78	$< 10^{12}$	98,630,735	2045.53
17.12	$< 10^{11}$	96,305,090	2370.07
20.09	$< 10^{10}$	86,242,708	2781.11
22.60	$< 10^{9}$	58,157,883	3128.49
24.97	$< 10^{8}$	19,926,459	3455.80
28.62	$< 10^{7}$	747,104	3960.96
31.47	$< 10^{6}$	266,180	4325.60
34.47	$< 10^{5}$	175,981	4693.44
36.84	$< 10^{4}$	163,297	5004.24
39.39	$< 10^{3}$	160,516	5347.71
41.08	$< 10^{2}$	160,138	5577.22
43.88	$< 10^{1}$	160,011	5941.72
45.94	$< 10^{0}$	160,002	6218.82
46.19	$< 10^{-1}$	160,001	6252.20
46.25	$< 10^{-2}$	160,000	6260.20
46.89	$< 10^{-3}$	160,000	6344.31
46.91	$< 10^{-4}$	160,000	6346.99
46.93	$< 10^{-5}$	160,000	6349.69

Tool 5

Parallelism / Minibatching

“Work on random subsets”

The Problem

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

n is BIG

L-smooth, μ-strongly convex

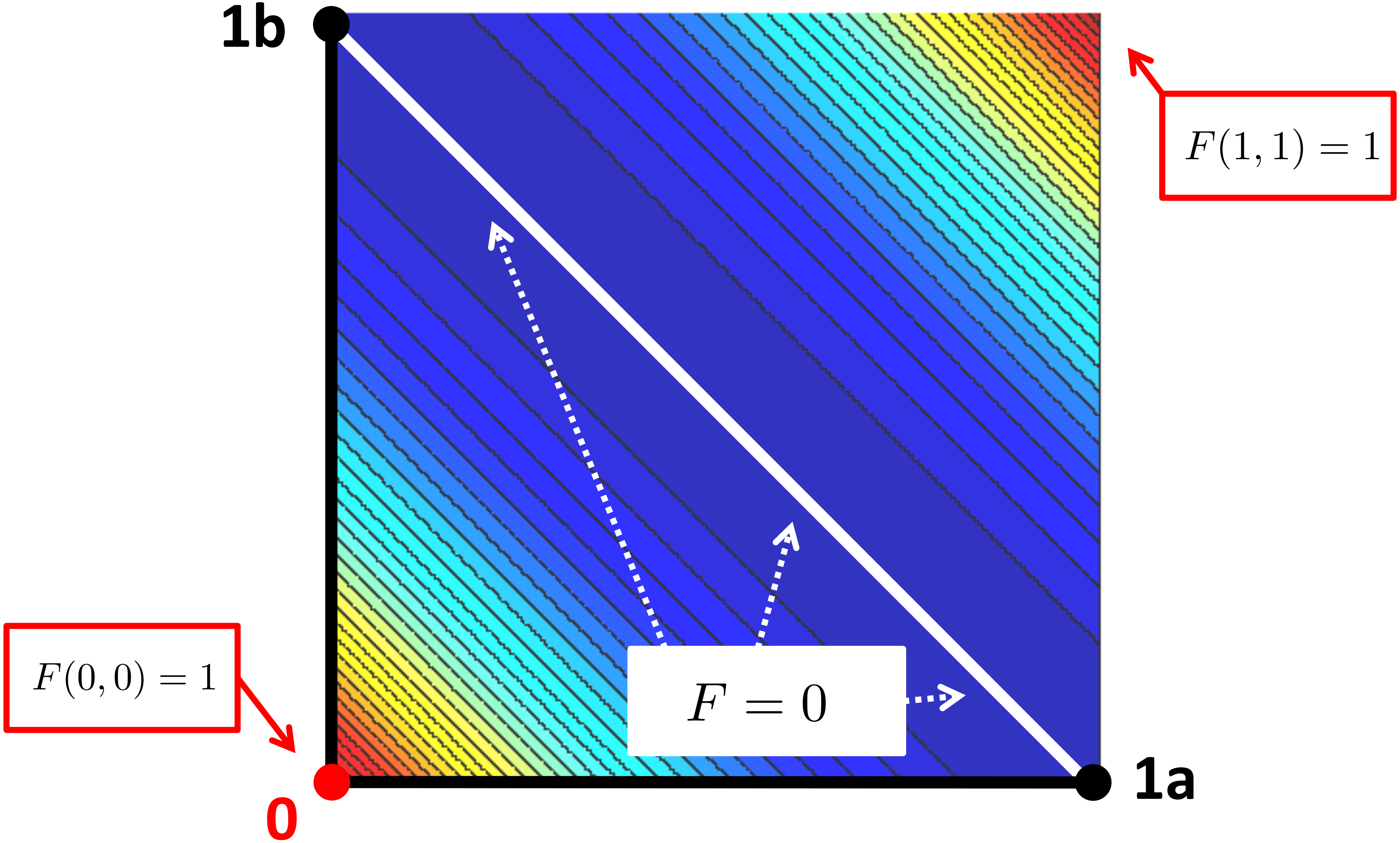
Parallel Randomized Coordinate Descent

P.R. and Martin Takáč
Parallel Coordinate Descent Methods for Big Data Optimization
Mathematical Programming 156(1), 433-484, 2016

16th IMA Leslie Fox Prize (2nd), 2013
Most downloaded MAPR paper

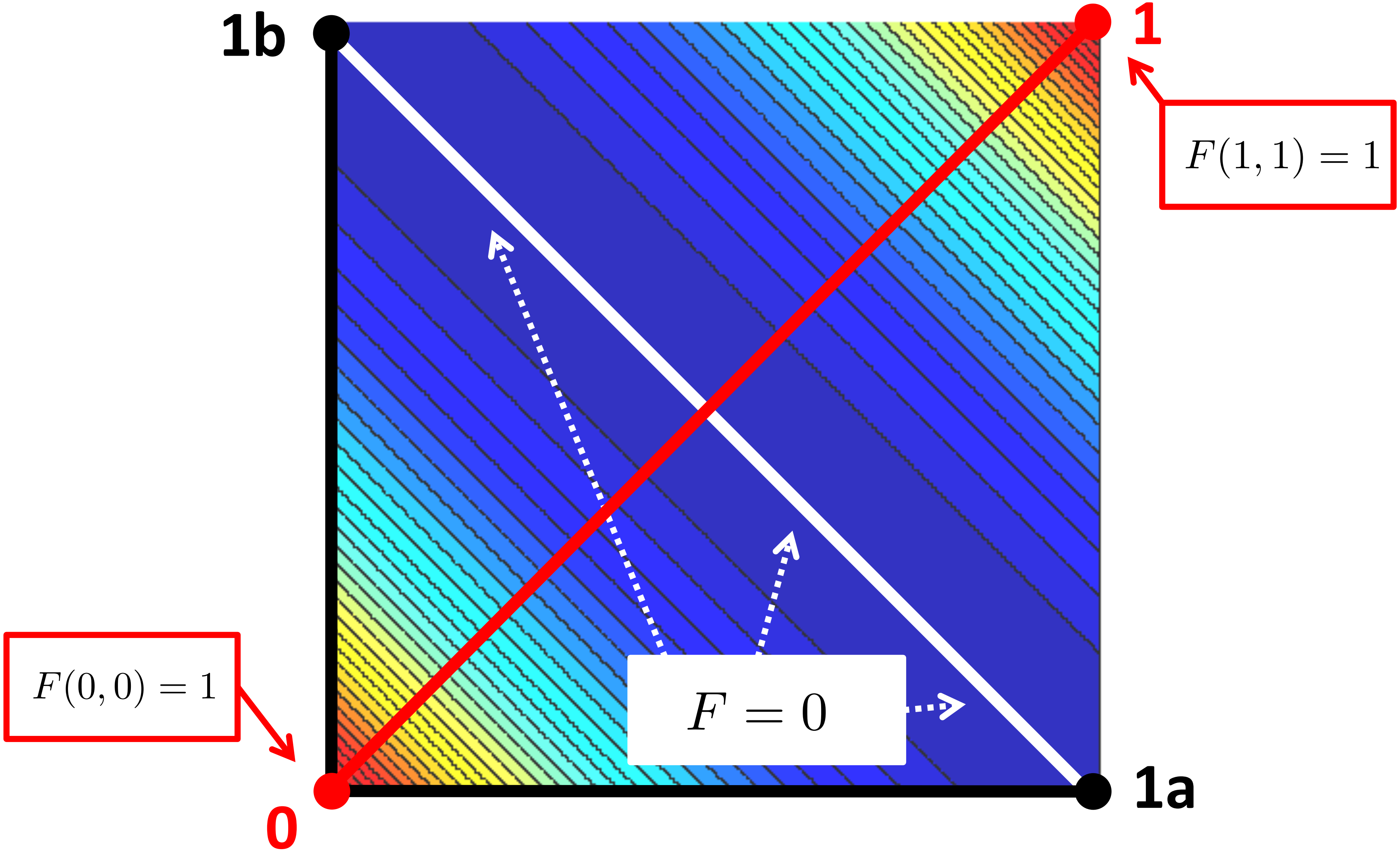
Additive Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$



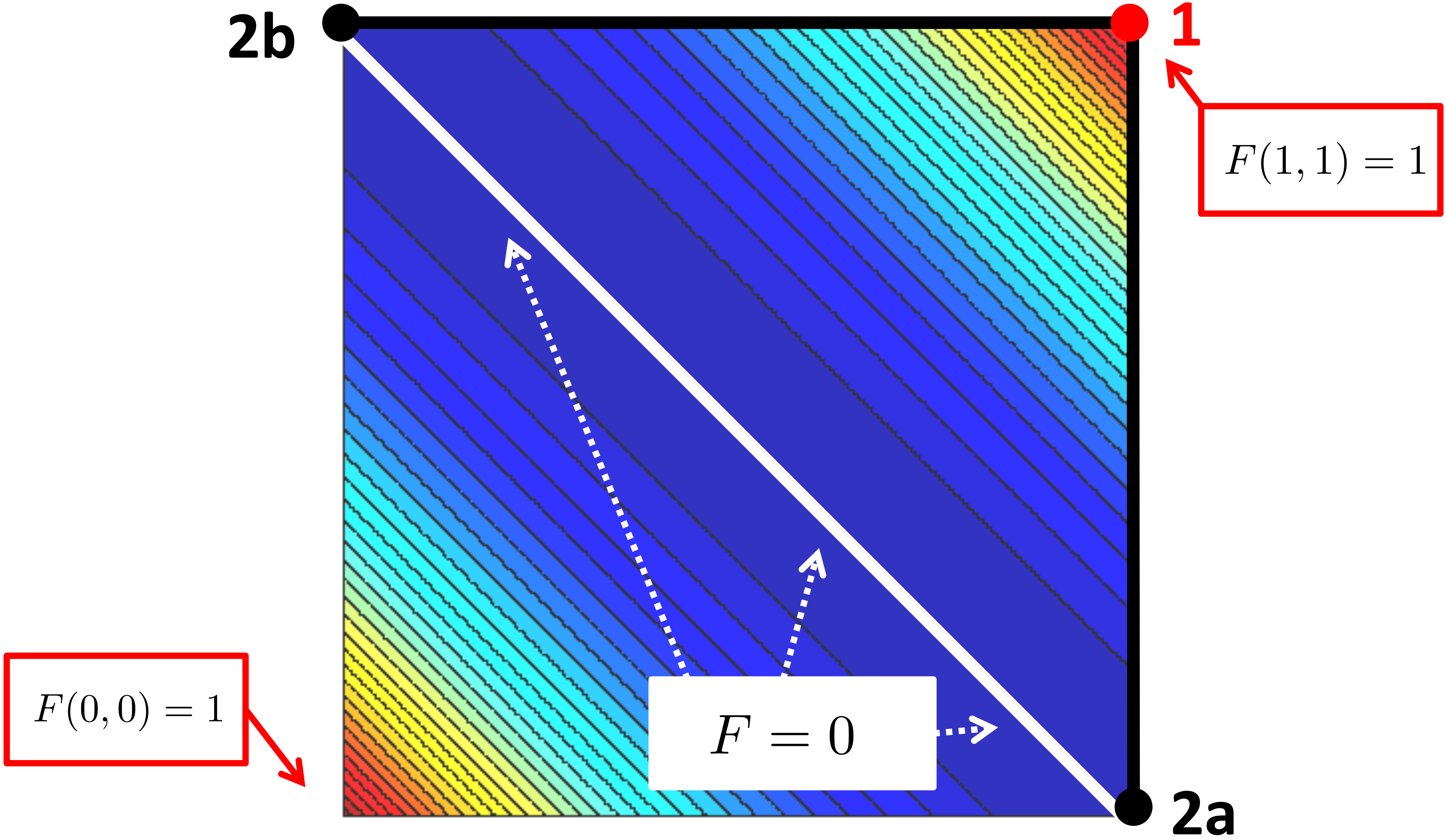
Additive Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$



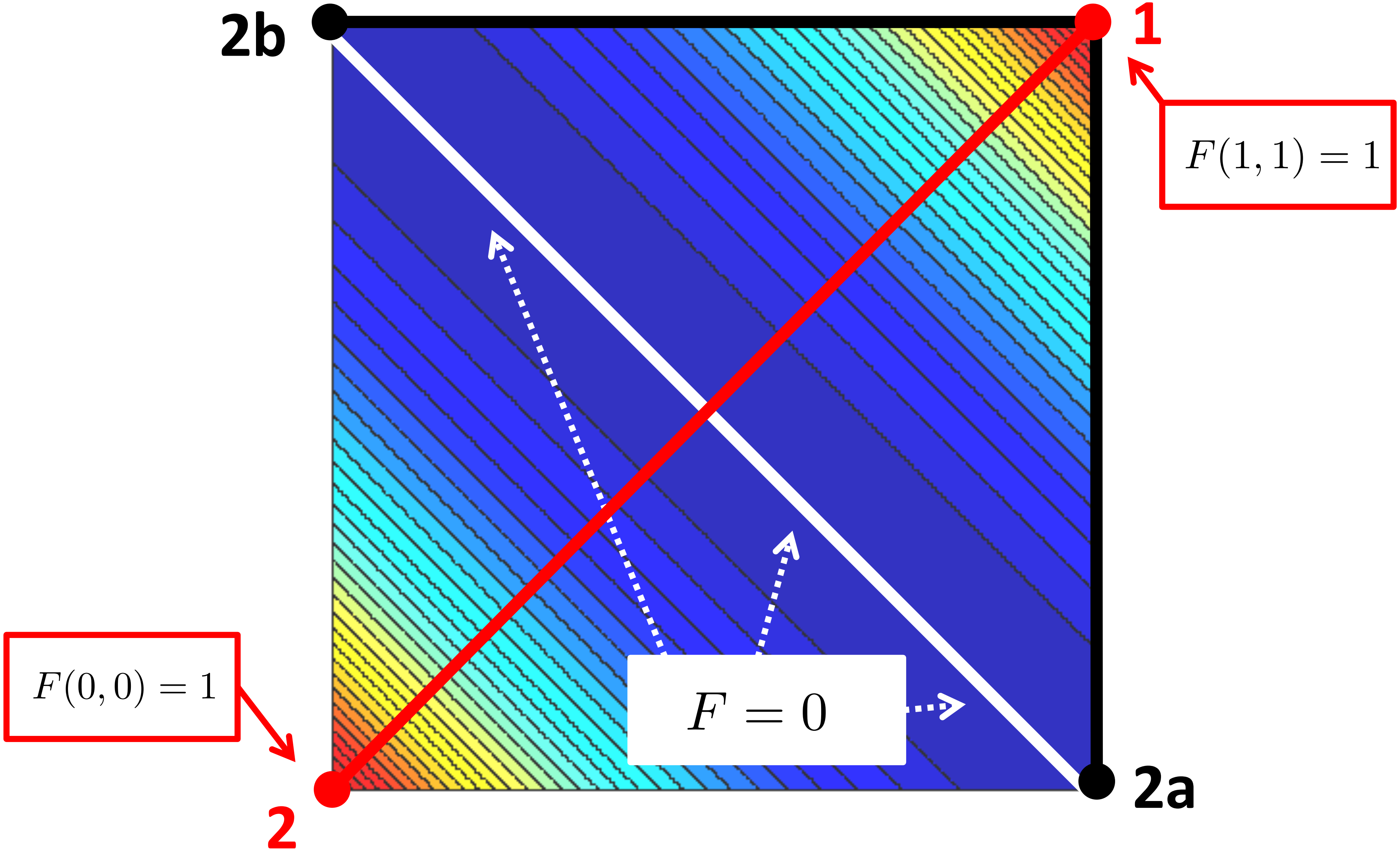
Additive Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$



Additive Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$



Additive Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$

OOPS!

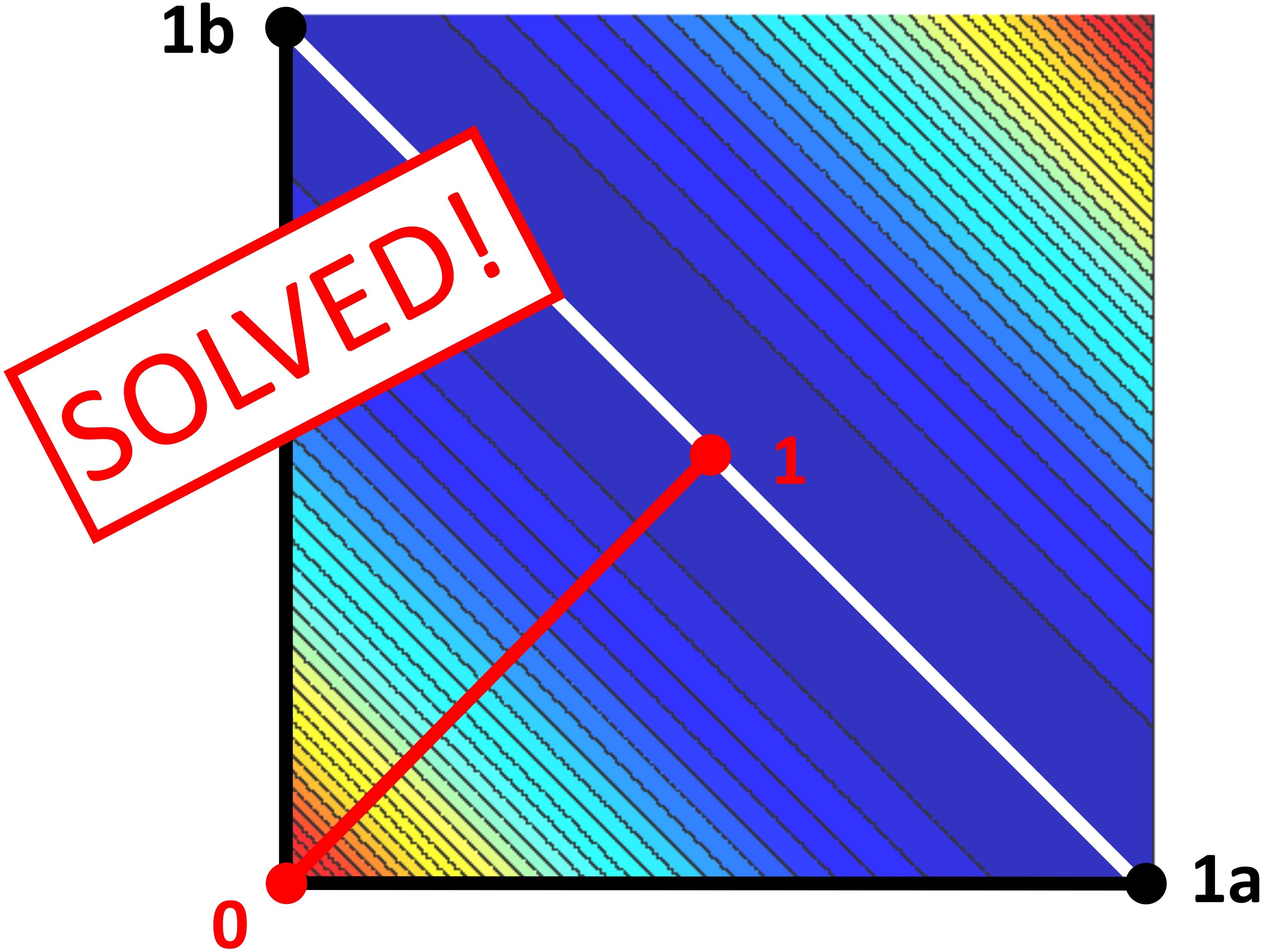
$\boxed{F(1,1) = 1}$

$\boxed{F(0,0) = 1}$

2

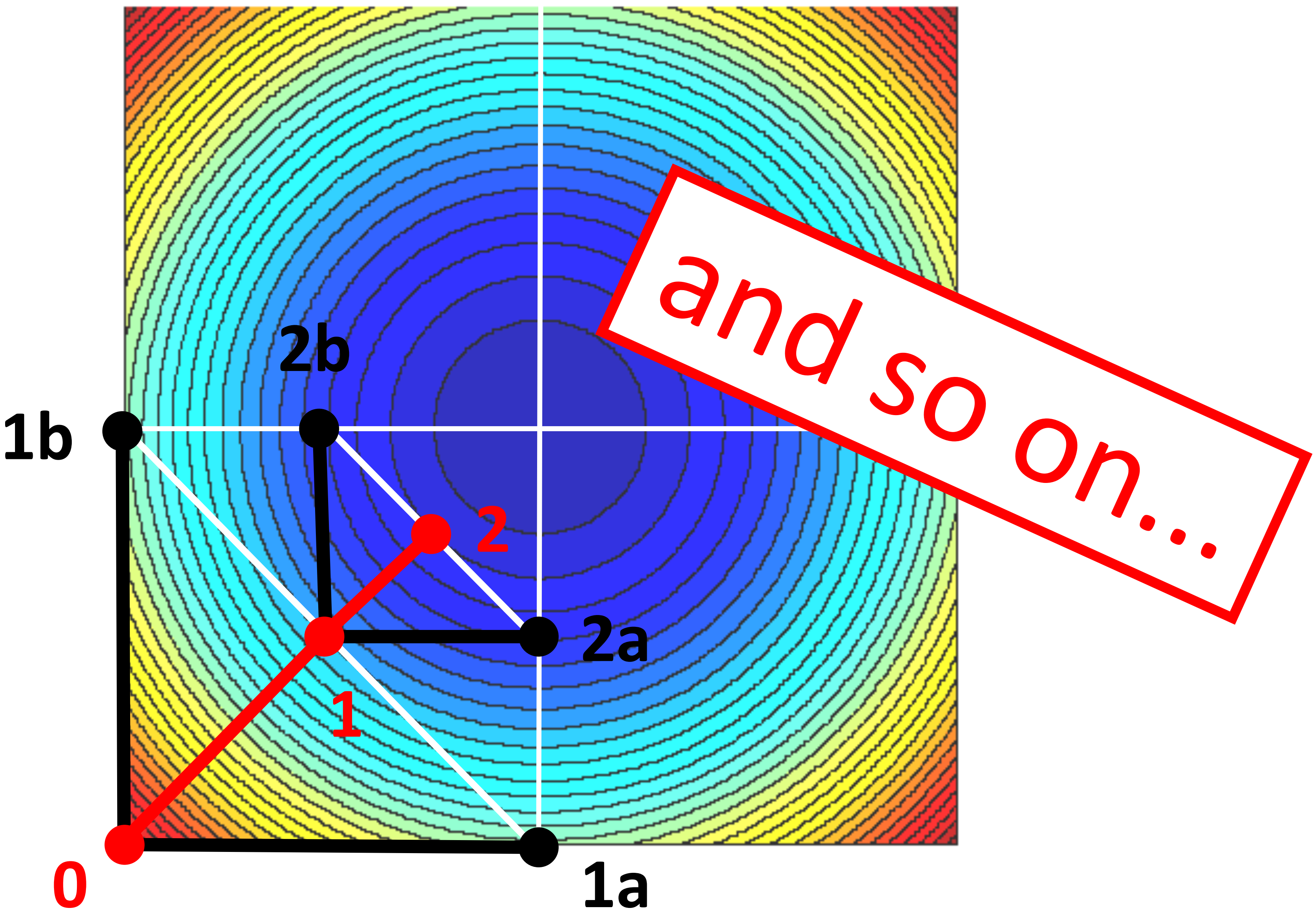
Averaging Strategy

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 + x_2 - 1)^2$$



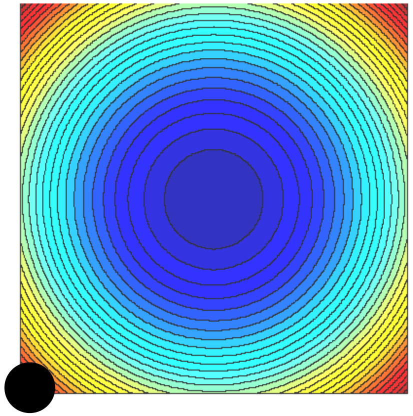
Averaging Can Be Bad, Too!

$$x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



Actually, Averaging Can Be Very Bad!

$$f(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + \cdots + (x_n - 1)^2$$



$$x^0 = 0 \in \mathbb{R}^n \;\Rightarrow\; f(x^0) = n$$

BAD!!!

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

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

WANT

How to Combine the Updates?

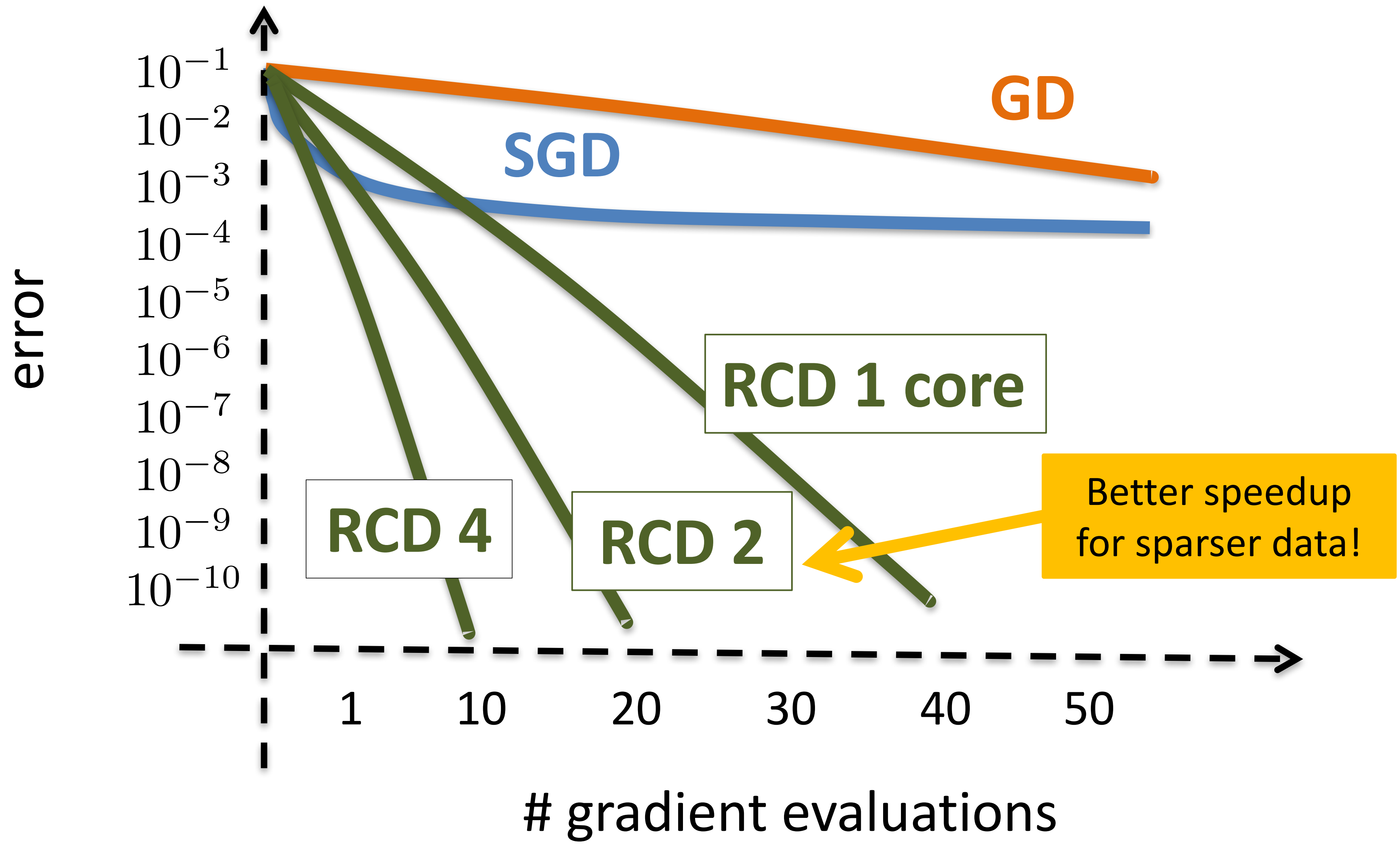
- We should do data-dependent combination of the results obtained in parallel
- There is rich theory for this now

Averaging (no speedup) | Adding (perfect speedup)

Dense data | Sparse data

Zheng Qu and P.R.
Coordinate Descent with Arbitrary Sampling II: Expected Separable Overapproximation
Optimization Methods and Software 31(5), 858-884, 2016

Performance



Problem with 1 Billion Variables

source: [R. & Takáč, arXiv 2011, MAPR 2014]

	Error $f(x^t) - f(x^*)$			Elapsed Time		
$(t \cdot \tau)/n$	1 core	8 cores	16 cores	1 core	8 cores	16 cores
0	6.27e+22	6.27e+22	6.27e+22	0.00	0.00	0.00
1	2.24e+22	2.24e+22	2.24e+22	0.89	0.11	0.06
2	2.25e+22	3.64e+19	2.24e+22	1.97	0.27	0.14
3	1.15e+20	1.94e+19	1.37e+20	3.20	0.43	0.21
4	5.25e+19	1.42e+18	8.19e+19	4.28	0.58	0.29
5	1.59e+19	1.05e+17	3.37e+19	5.37	0.73	0.37
6	1.97e+18	1.17e+16	1.33e+19	6.64	0.89	0.45
7	2.40e+16	3.18e+15	8.39e+17	7.87	1.04	0.53
⋮	⋮	⋮	⋮	⋮	⋮	⋮
26	3.49e+02	4.11e+01	3.68e+03	31.71	3.99	2.02
27	1.92e+02	5.70e+00	7.77e+02	33.00	4.14	2.10
28	1.07e+02	2.14e+00	6.69e+02	34.23	4.30	2.17
29	6.18e+00	2.35e-01	3.64e+01	35.31	4.45	2.25
30	4.31e+00	4.03e-02	2.74e+00	36.60	4.60	2.33
31	6.17e-01	3.50e-02	6.20e-01	37.90	4.75	2.41
32	1.83e-02	2.41e-03	2.34e-01	39.17	4.91	2.48
33	3.80e-03	1.63e-03	1.57e-02	40.39	5.06	2.56
34	7.28e-14	7.46e-14	1.20e-02	41.47	5.21	2.64
35	-	-	1.23e-03	-	-	2.72
36	-	-	3.99e-04	-	-	2.80
37	-	-	7.46e-14	-	-	2.87

Tools 1-5

Summary

Tools 1-5 Summary

Method	**# iterations**	**Cost of 1 iter.**
Gradient Descent (GD)	$\frac{L}{\mu}\log(1/\epsilon)$	n
Accelerated Gradient Descent (AGD)	$\sqrt{\frac{L}{\mu}}\log(1/\epsilon)$	n
Proximal Gradient Descent (PGD)	$\frac{L}{\mu}\log(1/\epsilon)$	n + Prox Step
Stochastic Gradient Descent (SGD)	$\left(\frac{\max_i L_i}{\mu} + \frac{\sigma^2}{\mu^2 \epsilon}\right)\log(1/\epsilon)$	1
Randomized Coordinate Descent (RCD)	$\frac{\max_i L_i}{\mu}\log(1/\epsilon)$	1

Suffers from high variance of stochastic gradient

Tool 6

Variance Reduction

"SGD is too noisy, fix it!"

Variance Reduction

	Decreasing stepsizes	Mini-batching	Adjusting the direction	Importance sampling
How does it work?	Scaling down the noise	More samples, less variance	Duality (SDCA) or Control Variate (SVRG)	Sample more important data (or parameters) more often
CONS:	Slow down; Hard to tune the stepsize	More work per iteration	A bit (SVRG) or a lot (SDCA) more memory needed	Might overfit probabilities to outliers
PROS:	Still converges Widely known	Parallelizable	Improved dependence on epsilon	Improved condition number for "variable" data

Good news: All tricks can be combined!

Tool 7

Importance Sampling

"Sample important data more often"

The Problem

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

Smooth and μ-strongly convex

SYNC

P.R. and Martin Takáč
On optimal probabilities in stochastic coordinate descent methods
Optimization Letters 10(6), 1233-1243, 2016 *(arXiv:1310.3438)*

ARBITRARY SAMPLING:
i.i.d. subset of {1, 2,..., *n*} with arbitrary distribution

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

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

Example $n = 3$

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Key Assumption

Parameters $v_1, \ldots, v_n$ satisfy:

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

Inequality must hold for all

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

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

Complexity Theorem

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

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

strong convexity constant

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

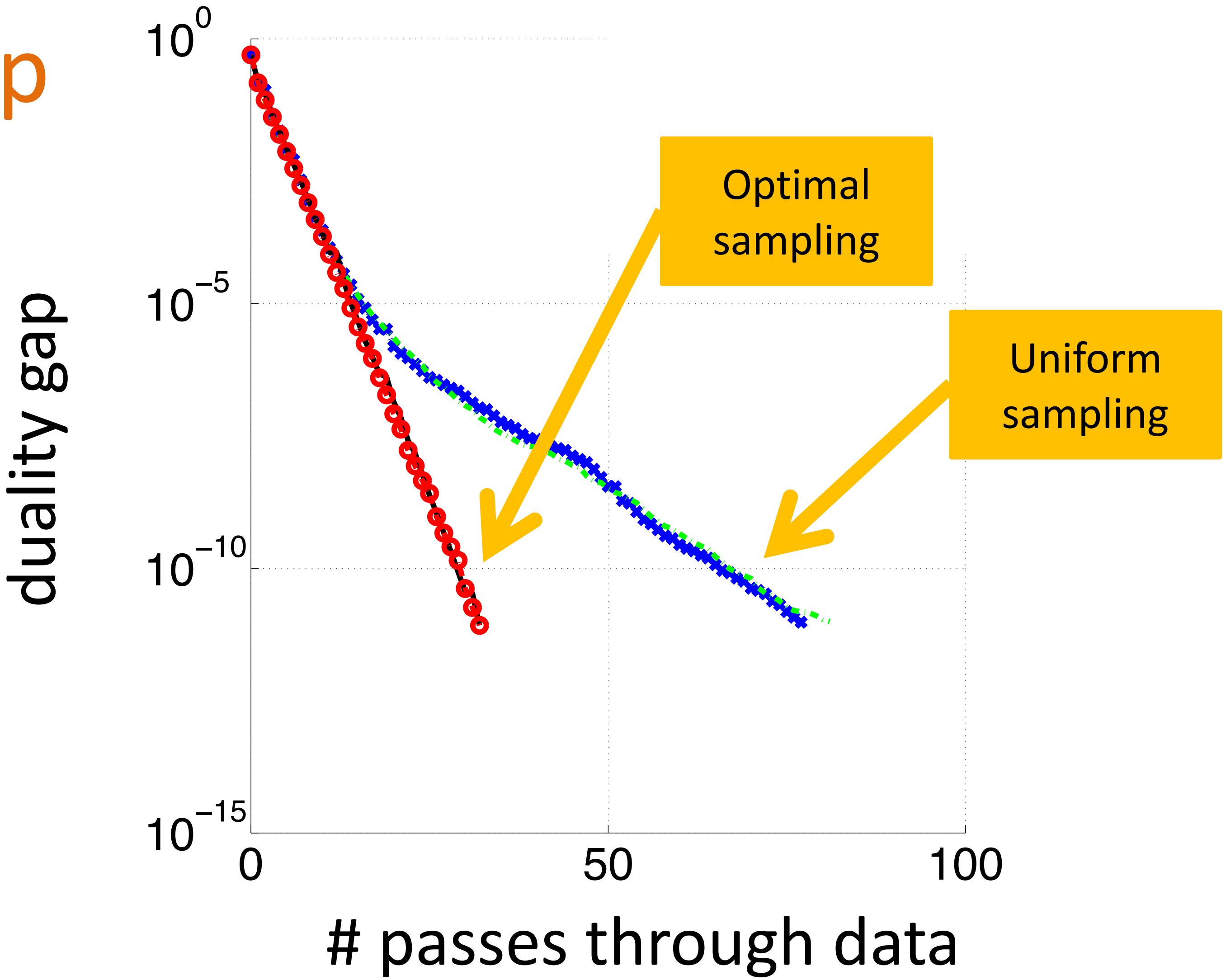
Uniform vs Optimal Sampling

$$p_i = \frac{1}{n} \quad \Rightarrow \quad \max_i \frac{v_i}{p_i \mu} = \frac{n \max_i v_i}{\mu}$$

$$p_i = \frac{v_i}{\sum_i v_i} \quad \Rightarrow \quad \max_i \frac{v_i}{p_i \mu} = \frac{\sum_i v_i}{\mu}$$

Logistic Regression: Laptop

Zheng Qu, P.R. and Tong Zhang. **Quartz: Randomized Dual Coordinate Ascent with Arbitrary Sampling.** In *Advances in Neural Information Processing Systems* 28, 2015



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

More Work on Arbitrary Sampling

Zheng Qu, P.R. and Tong Zhang
Quartz: Randomized dual coordinate ascent with arbitrary sampling
In *Advances in Neural Information Processing Systems* 28, 2015

Zheng Qu and P.R.
Coordinate descent with arbitrary sampling I: algorithms and complexity
Optimization Methods and Software 31(5), 829-857, 2016

Zheng Qu and P.R.
Coordinate descent with arbitrary sampling II: expected separable overapproximation
Optimization Methods and Software 31(5), 858-884, 2016

Tool 8

Duality

“Solve the dual instead”

3-in1: Three Variance Reduction Strategies in 1 Method

Variance Reduction

	Decreasing stepsizes	Mini-batching	Adjusting the direction	Importance sampling
How does it work?	Scaling down the noise	More samples, less variance	Duality (SDCA) or Control Variate (SVRG)	Sample more important data (or parameters) more often
CONS:	Slow down; Hard to tune the stepsize	More work per iteration	A bit (SVRG) or a lot (SDCA) more memory needed	Might overfit probabilities to outliers
PROS:	Still converges Widely known	Parallelizable	Improved dependence on epsilon	Improved condition number for “variable” data

Good news: All tricks can be combined!

The Problem

$$\min_{x \in \mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^\top x) + g(x) \right]$$

Convex and L-smooth

$\frac{\mu}{2} \|x\|_2^2$

We will discuss duality without actually considering the dual problem. The basic proof technique (due to Shai Shalev-Shwartz, 2015) is dual-free.

Motivation I

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n f_i(a_i^\top x) + g(x) \right]$$

x^* is optimal

$$0 = \nabla P(x^*) = \left(\frac{1}{n}\sum_{i=1}^n a_i \nabla f_i(a_i^\top x^*) \right) + \mu x^*$$

$$x^* = \frac{1}{\mu n}\sum_{i=1}^n a_i y_i^*$$

$$y_i^* := -\nabla f_i(a_i^\top x^*)$$

Motivation II

Algorithmic Ideas:

1. Simultaneously search for both x^* and $y_1^*, \ldots, y_n^*$

2. Try to do "something like"

$$y_i^{t+1} \leftarrow -\nabla f_i(a_i^\top x^t)$$

Does not quite work: too "greedy"

3. Maintain the relationship

$$x^t = \frac{1}{\mu n} \sum_{i=1}^{n} a_i y_i^t$$

The Algorithm: dfSDCA

Initialize the relationship

STEP 0: INITIALIZE

Choose $y_1^0, \ldots, y_n^0 \in \mathbb{R}$

$$x^0 = \frac{1}{\mu n} \sum_{i=1}^{n} a_i y_i^0$$

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

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

$$y_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right) y_i^t + \frac{\theta}{p_i} \left(-\nabla f_i(a_i^\top x^t)\right)$$

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

STEP 2: PRIMAL UPDATE

$$x^{t+1} \leftarrow x^t + \sum_{i \in S_t} \frac{\theta}{n \mu p_i} a_i \left(-\nabla f_i(a_i^\top x^t) + y_i^t\right)$$

This is just maintaining the relationship

Complexity

“ESO constants”: similar definition as for NSync

Theorem [Csiba & R ‘15]

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

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

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

Relevant Papers

Shai Shalev-Shwartz
SDCA without duality
arXiv:1502.06177, 2015

Dual–free SDCA idea

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

dfSDCA

Same theoretical result, but for general g and using duality

Zheng Qu and P.R.
Coordinate descent with arbitrary sampling II: expected separable overapproximation
Optimization Methods and Software 31(5), 858-884, 2016

Standard Tools: Final Remarks

Methods / Tools	GD 1847	AGD ‘83 ‘03	PGD ‘05	SGD ‘51	RCD ‘10	PCDM ‘12	SDCA ‘12	SVRG ‘14
1.Gradient Descent	YES	YES	YES	YES	YES	YES	YES	YES
2. Acceleration	NO	YES	NO	NO Katyusha ‘17	NO APPROX ‘13 ALPHA ‘14	NO	NO AccProx-SDCA ‘13 APCG ‘14	NO
3. Proximal Trick	NO PGM ‘05	NO	YES	NO	NO RCDC ‘11 APPROX ‘13	NO* PCDM ‘12	YES	NO ProxSVRG ‘14
4. Randomized Decomposition	NO	NO	NO	YES	YES	YES	YES	YES
5. Parallelism (Minibatching)	YES	YES	YES*	NO mSGD ‘13	NO PCDM ‘12 APPROX ’13 ALPHA ‘14	YES	NO QUARTZ ‘15	NO mS2GD ‘14
6. Variance Reduction				NO SAG ’11 SVRG ’13 S2GD ’13 SDCA ’12	YES	YES	YES	YES
7. Duality	NO	NO	YES	YES	NO RCDC ‘11	NO PCDM ‘12	YES	NO
8. Importance Sampling				NO Iprox-SMD ‘13	YES NSync ’13 RCDC ‘11 ALPHA ’14	NO ALPHA ‘14	NO QUARTZ ‘15	NO
9. Curvature	NO	NO	NO	NO	NO SDNA ‘15	NO SDNA ‘15	NO SDNA ‘15	NO SBFGS ‘15

Tools \ Methods	NSync '13	dfSDCA '15
1.Gradient Descent	YES	YES
2. Acceleration	NO	NO
3. Proximal Trick	NO	NO QUARTZ '15
4. Randomized Decomposition	YES	YES
5. Parallelism (Minibatching)	YES	YES
6. Variance Reduction	YES	YES
7. Duality	NO	NO* QUARTZ '15
8. Importance Sampling	YES	YES
9. Curvature	NO	NO

SVRG	Accelerating stochastic gradient descent using predictive variance reduction R Johnson, T Zhang Advances in neural information processing systems, 315-323	480	2013
S2GD	Semi-stochastic gradient descent methods J Konečný, P Richtárik Frontiers in Applied Mathematics and Statistics	107 *	2017
ProxSVRG	A proximal stochastic gradient method with progressive variance reduction L Xiao, T Zhang SIAM Journal on Optimization 24 (4), 2057-2075	213	2014
mSGD	Mini-batch primal and dual methods for SVMs M Takáč, A Bijral, P Richtárik, N Srebro 30th International Conference on Machine Learning (ICML)	102 *	2013
QUARTZ	Quartz: Randomized dual coordinate ascent with arbitrary sampling Z Qu, P Richtárik, T Zhang Advances in Neural Information Processing Systems 28, 865--873	67	2015
SAG	Minimizing finite sums with the stochastic average gradient M Schmidt, N Le Roux, F Bach Mathematical Programming (MAPR), 2017.	293 *	2013
ALPHA	Coordinate descent with arbitrary sampling I: algorithms and complexity Z Qu, P Richtárik Optimization Methods and Software 31 (5), 829-857	56	2016
NSync	On optimal probabilities in stochastic coordinate descent methods P Richtárik, M Takáč Optimization Letters 10 (6), 1233-1243	46	2016
SPDC	Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization. Y Zhang, L Xiao ICML, 353-361	78	2015

RCDC	Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function P Richtarik, M Takáč Mathematical Programming 144 (2), 1-38	355	2014
PCDM	Parallel coordinate descent methods for big data optimization P Richtárik, M Takáč Mathematical Programming 156 (1), 433-484	228	2016
APPROX	Accelerated, parallel and proximal coordinate descent O Fercoq, P Richtárik SIAM Journal on Optimization 25 (4), 1997-2023	143	2015
ProxSVRG	A proximal stochastic gradient method with progressive variance reduction L Xiao, T Zhang SIAM Journal on Optimization 24 (4), 2057-2075	213	2014
CoCoA+	Adding vs. averaging in distributed primal-dual optimization C Ma, V Smith, M Jaggi, MI Jordan, P Richtárik, M Takáč 32nd International Conference on Machine Learning (ICML)	45	2015
SDCA	Stochastic dual coordinate ascent methods for regularized loss minimization S Shalev-Shwartz, T Zhang Journal of Machine Learning Research 14 (Feb), 567-599	428	2013
Katyusha	Katyusha: The first direct acceleration of stochastic gradient methods Z Allen-Zhu arXiv preprint arXiv:1603.05953	51 *	2016
Iprox-SMD	Stochastic optimization with importance sampling for regularized loss minimization P Zhao, T Zhang Proceedings of the 32nd International Conference on Machine Learning (ICML ...	89	2015

GD, AGD	Introductory lectures on convex optimization: A basic course Y Nesterov Springer Science & Business Media	2564	2013
AGD	Smooth minimization of non-smooth functions Y Nesterov Mathematical programming 103 (1), 127-152	1686	2005
PGD	Gradient methods for minimizing composite objective function Y Nesterov Core	1288 *	2007
RCD	Efficiency of coordinate descent methods on huge-scale optimization problems Y Nesterov SIAM Journal on Optimization 22 (2), 341-362	581	2012
SBFGS	Stochastic block BFGS: squeezing more curvature out of data RM Gower, D Goldfarb, P Richtárik 33rd International Conference on Machine Learning (ICML)	25	2016
APCG	An accelerated proximal coordinate gradient method Q Lin, Z Lu, L Xiao Advances in Neural Information Processing Systems, 3059-3067	74	2014
Acc Prox-SDCA	Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization S Shalev-Shwartz, T Zhang International Conference on Machine Learning, 64-72	135	2014
mS2GD	Mini-batch semi-stochastic gradient descent in the proximal setting J Konečný, J Liu, P Richtárik, M Takáč IEEE Journal of Selected Topics in Signal Processing 10 (2), 242-255	68	2015

Part 3
Stochastic Methods for Linear Systems

The Plan

Plan

- Quick recall of ERM formulation of linear systems
- **Four stochastic reformulations** (not related to ERM)
- **Basic method** (solves primal ERM)
- **Parallel** and **accelerated methods** (solve primal ERM)
- **Duality** (method for solving dual ERM)
- **EXTRA TOPIC: Special cases** (specializing some parameters of the method)
- **EXTRA TOPIC: Stochastic preconditioning** (vast generalization of importance sampling)
- **EXTRA TOPIC: Stochastic matrix inversion**

We will (mostly) follow this paper

P.R. and Martin Takáč
Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory
arXiv:1706.01108, 2017

Algorithms

Basic Method
- Stochastic gradient descent
- Stochastic Newton method
- Stochastic proximal point method
- Stochastic preconditioning method
- Stochastic fixed point method
- Stochastic projection method

Dual of the Basic Method
- Stochastic dual subspace ascent

Parallel Methods

Accelerated Methods

Selected Special Cases (Basic Method)
- Randomized Kaczmarz Method
- Stochastic coordinate descent
- Randomized Newton method
- Stochastic Gaussian descent
- Stochastic spectral descent

Quick Recall:
Linear Systems as ERM

Solving Linear Systems

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

$$\text{Solve } Ax = b$$

$A \in \mathbb{R}^{n \times d}$

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

$$A = \begin{pmatrix} a_1^\top \\ a_2^\top \\ \vdots \\ a_n^\top \end{pmatrix}$$

$$\text{Think: } n \gg d$$

Linear Systems (Best Approximation Version) as a Primal ERM Problem

$$g(x) = \frac{1}{2}\|x - x^0\|_B^2$$

$$\min_{x\in\mathbb{R}^d} \left[P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n f_i(a_i^\top x) + g(x) \right]$$

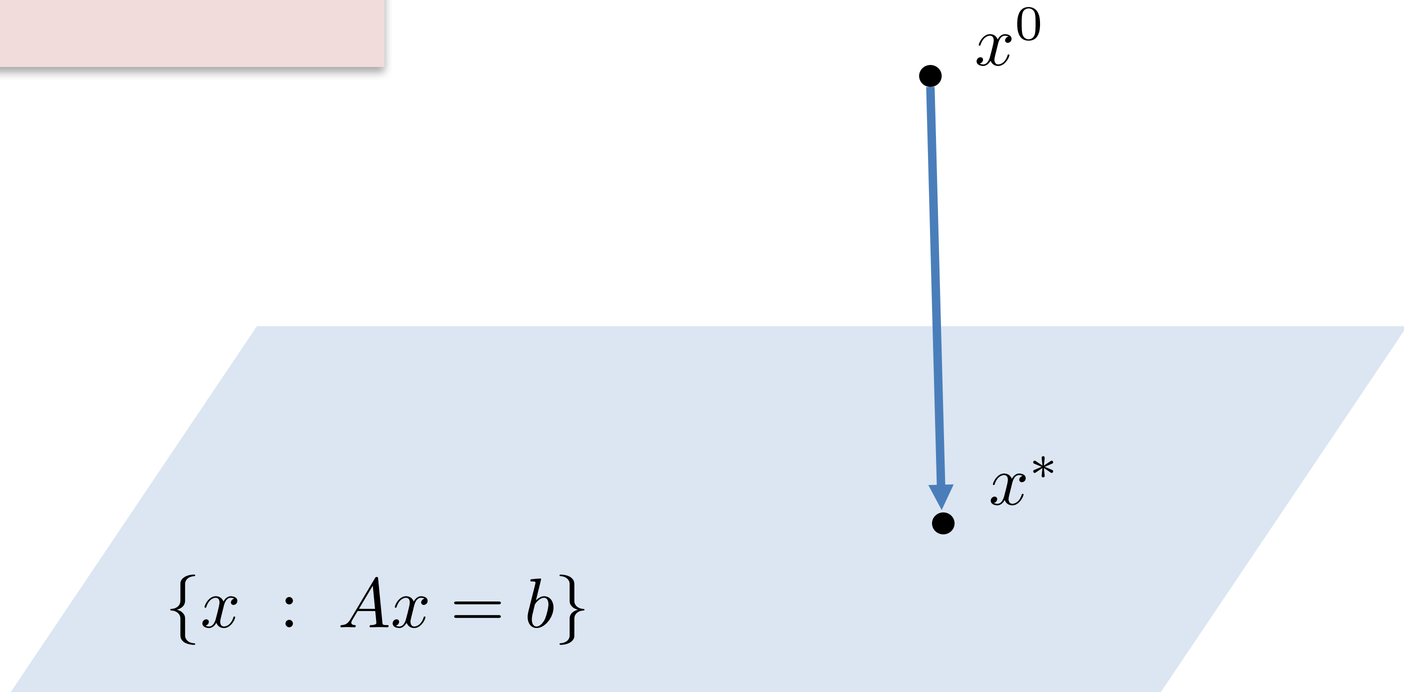
$$f_i(t) = 1_{\{b_i\}}(t) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{for } t = b_i, \\ +\infty & \text{otherwise.} \end{cases}$$

Primal Problem: Best Approximation

$$\min_{x \in \mathbb{R}^d} \frac{1}{2}\|x - x^0\|_B^2$$

$$\text{Subject to } Ax = b$$

$$\|x\|_B = \sqrt{x^\top B x}$$



Dual Problem

Recall convex conjugate:

$$f^*(z) \stackrel{\text{def}}{=} \sup_{x \in \mathbb{R}^d} \{\langle z, x\rangle - f(x)\}$$

$$f_i(t) = 1_{\{b_i\}}(t) \qquad f_i^*(t) = b_i t$$

$$g(x) = \frac{1}{2}\|x - x^0\|_B^2 \qquad g^*(x) = \langle x^0, x\rangle + \frac{1}{2}\|x\|_{B^{-1}}^2$$

$$\max_{y \in \mathbb{R}^n} \left[D(y) \stackrel{\text{def}}{=} \left\langle b - Ax^0, \frac{y}{n} \right\rangle - \frac{1}{2}\left\| A^\top \frac{y}{n} \right\|_{B^{-1}}^2 \right]$$

Unconstrained (non-strongly) concave quadratic maximization

Recovering Primal Solution from Dual Solution

Recall:

$$x^* = \nabla g^* \left(\tfrac{1}{n} A^\top y^*\right)$$

$$g^*(x) = \langle x^0, x \rangle + \tfrac{1}{2}\|x\|^2_{B^{-1}}$$

$$\Downarrow$$

$$\nabla g^*(x) = x^0 + B^{-1}x$$

$$\Downarrow$$

$$x^* = x^0 + \tfrac{1}{n} B^{-1} A^\top y^*$$

Reformulation 1: Stochastic Optimization

Change of Notation

$$\cancel{n}\left\{\overbrace{A}^{\cancel{d}}x = b\right.$$

A System of Linear Equations

m equations with n unknowns

$$A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m$$

$$m\left\{ \overbrace{A}^{n} x = b \right.$$

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

Stochastic Reformulations of Linear Systems

$n \times n$ pos def

distribution over $m \times q$ matrices

$$Ax = b \xrightarrow{B, \mathcal{D}}$$

1. Stochastic Optimization
2. Stochastic Linear System
3. Stochastic Fixed Point
4. Probabilistic Intersection

Example: $B = \text{identity}$
$\mathcal{D} = \text{uniform over } e_1, \ldots, e_m$ (unit basis vectors in $\mathbb{R}^m$)

Theorem

a) These 4 problems have the same solution sets
b) Weak necessary & sufficient conditions for the solution set to be equal to $\{x \;:\; Ax = b\}$

Reformulation 1: Stochastic Optimization

Stochastic Optimization

Stochastic function
(unbiased estimator of function *f*)

$$\text{Minimize } f(x) \stackrel{\text{def}}{=} \mathbf{E}_{S\sim\mathcal{D}}[f_S(x)]$$

$$f_S(x) = \frac{1}{2}\|x - \Pi^B_{\mathcal{L}_S}(x)\|^2_B = \frac{1}{2}(Ax-b)^\top H_S(Ax-b)$$

$$\mathcal{L}_S = \{x \;:\; S^\top Ax = S^\top b\}$$

Sketched system

$$H_S \stackrel{\text{def}}{=} S(S^\top AB^{-1}A^\top S)^\dagger S^\top$$

Special Case

$\mathcal{D}$ is defined by: $S = e_i$ with probability $1/m$

$B = I$ (identity matrix)

$$m = 3 \quad \Rightarrow \quad e_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad e_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}, \quad e_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

Expectation becomes average over *m* functions:

$$\text{Minimize} \quad f(x) := \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{\|A_{:i}\|^2} (A_{:i} x - b_i)^2}_{f_i(x)}$$

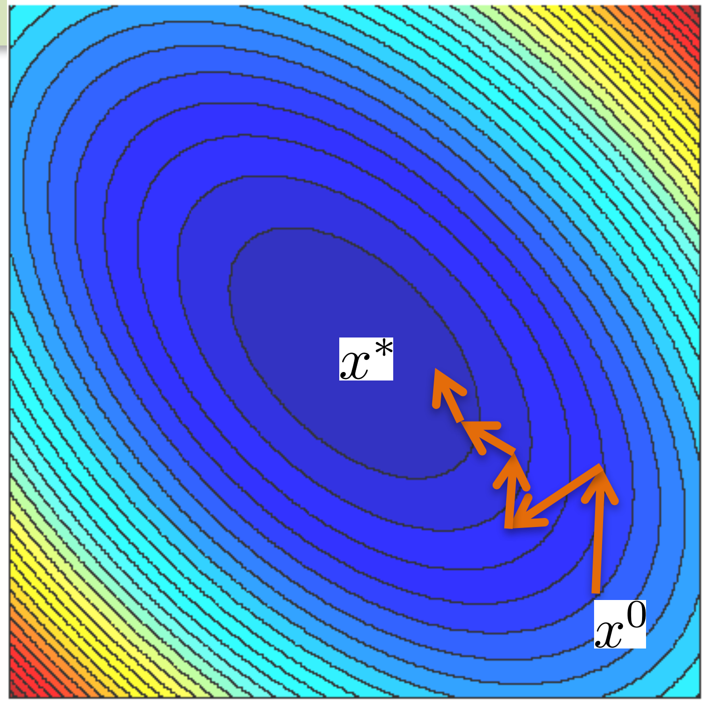
Special Case: Randomized Algorithm

Algorithm (Stochastic Gradient Descent)

1. Choose random $i \in \{1, 2, \ldots, m\}$
2. $x^{t+1} = x^t - \nabla f_i(x^t)$

Stochastic gradient (unbiased estimator of the gradient):

$$\mathbf{E}[\nabla f_i(x)] = \nabla f(x)$$



Reformulation 2:
Stochastic Linear System

Stochastic Linear System

Instead of $Ax = b$ we solve the preconditioned system:

$$H_S \stackrel{\text{def}}{=} S(S^\top A B^{-1} A^\top S)^\dagger S^\top$$

$$\text{Solve } \underbrace{B^{-1}A^\top \mathbf{E}_{S\sim\mathcal{D}}[H_S]}_{\text{Preconditioner } P} Ax = \underbrace{B^{-1}A^\top \mathbf{E}_{S\sim\mathcal{D}}[H_S]}_{\text{Preconditioner } P} b$$

Special Case

$\mathcal{D}$ is defined by: $S = e_i$ with probability $1/m$
$B = I$ (identity matrix)

$$\text{Solve } PAx = Pb$$

$$P := \frac{1}{m} \sum_{i=1}^{m} \underbrace{A^\top \frac{e_i e_i^\top}{\|A_{i:}\|^2}}_{P_i}$$

Special Case: Algorithm

Algorithm (Stochastic Preconditioning Method)

1. Choose random $i \in \{1, 2, \ldots, m\}$
2. $x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \{\|x - x^t\| \ : \ P_i A x = P_i b\}$

See also: Sketch & Project Method [Gower & Richtarik, 2015]

Stochastic preconditioner (unbiased estimator of the preconditioner *P*)

$$\mathbf{E}[P_i] = P$$

Reformulation 3: Stochastic Fixed Point Problem

Stochastic Fixed Point Problem

$$\text{Solve } x = \underbrace{\mathbf{E}_{S\sim\mathcal{D}}\left[\Pi^B_{\mathcal{L}_S}(x)\right]}_{\phi(x)}$$

Projection in B-norm onto $\mathcal{L}_S = \{x \;:\; S^\top Ax = S^\top b\}$

Special Case

$\mathcal{D}$ is defined by: $S = e_i$ with probability $1/m$
$B = I$ (identity matrix)

$$\text{Solve } x = \phi(x)$$

$$\phi(x) := x - P(Ax - b) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{x - P_i(Ax - b)}_{\phi_i(x)}$$

Special Case: Algorithm

Algorithm (Stochastic Fixed Point Method)

1. Choose random $i \in \{1, 2, \ldots, m\}$
2. $x^{t+1} = \phi_i(x^t)$

Stochastic operator (unbiased estimator of the fixed point operator)

$$\mathbf{E}[\phi_i(x)] = \phi(x)$$

Reformulation 4: Stochastic Intersection Problem

Stochastic Intersection of Sets

$S \sim \mathcal{D}$

"Sketched" system: $$S^\top A x = S^\top b$$

Stochastic set: $$\mathcal{L}_S = \{x \;:\; S^\top A x = S^\top b\}$$

Definition

Stochastic intersection of the sets $\{\mathcal{L}_S\}_{S\sim\mathcal{D}}$ is the set

$$\bigcap_{S\sim\mathcal{D}} \mathcal{L}_S \overset{\text{def}}{=} \{x \;:\; \mathbf{P}(x \in \mathcal{L}_S) = 1\}$$

Discrete Case: Stochastic Intersection = Classical Intersection

$\mathcal{D}$ is discrete:

$S = S_i$ with probability $p_i > 0$

$$\{x \;:\; \mathbf{P}(x \in \mathcal{L}_S) = 1\} = \bigcap_i \mathcal{L}_{S_i}$$

Stochastic intersection of sets

"Classical" intersection of sets

Indicator Function of a Set

$$1_{\mathcal{M}}(x) = \begin{cases} 0 & x \in \mathcal{M} \\ +\infty & \text{otherwise.} \end{cases}$$

Indicator function of the stochastic set:

$$1_{\mathcal{L}_S}(x) = \begin{cases} 0 & x \in \mathcal{L}_S \\ +\infty & \text{otherwise.} \end{cases}$$

Stochastic Intersection

$$1_{\mathcal{L}_S}(x) = \begin{cases} 0 & x \in \mathcal{L}_S \\ +\infty & \text{otherwise.} \end{cases}$$

Lemma

$$\mathbf{E}_{S\sim\mathcal{D}}\left[1_{\mathcal{L}_S}(x)\right] = \begin{cases} 0 & \mathbf{P}(x \in \mathcal{L}_S) = 1 \\ +\infty & \text{otherwise.} \end{cases}$$

That is, the expectation of the indicator functions of the stochastic sets is an indicator function of the stochastic intersection those sets:

$$\mathbf{E}_{S\sim\mathcal{D}}\left[1_{\mathcal{L}_S}(x)\right] = 1_{\bigcap_{S\sim\mathcal{D}}\mathcal{L}_S}(x)$$

Stochastic Intersection Problem

Stochastic set:

$$\mathcal{L}_S = \{x \ : \ S^\top A x = S^\top b\}$$

$$\text{Find } x \in \bigcap_{S\sim\mathcal{D}} \mathcal{L}_S$$

Lemma Under some weak assumptions (e.g., $\mathbf{E}[H_S] \succ 0$ is sufficient)

$$\mathcal{L} = \bigcap_{S\sim\mathcal{D}} \mathcal{L}_S$$

Solution set of the linear system:

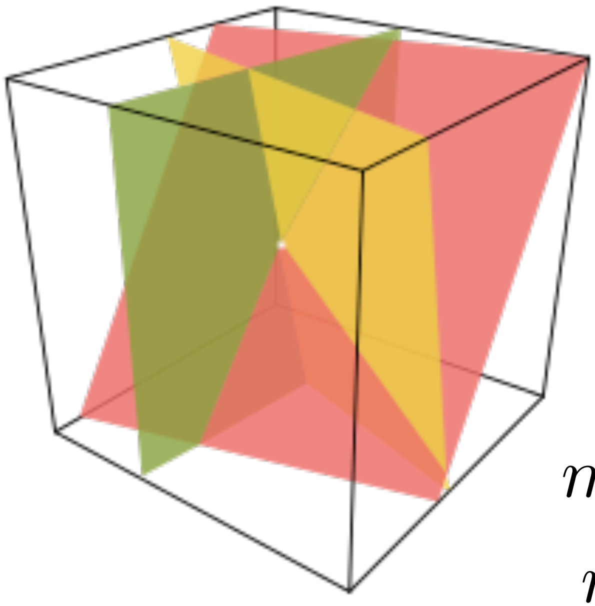
$$\mathcal{L} \stackrel{\text{def}}{=} \{x \ : \ Ax = b\}$$

Special Case

$\mathcal{D}$ is defined by: $S = e_i$ with probability $1/m$

$B = I$ (identity matrix)

$$\text{Find } x \in \bigcap_{i=1}^{m} \mathcal{L}_i$$



$m = 3$

$n = 3$

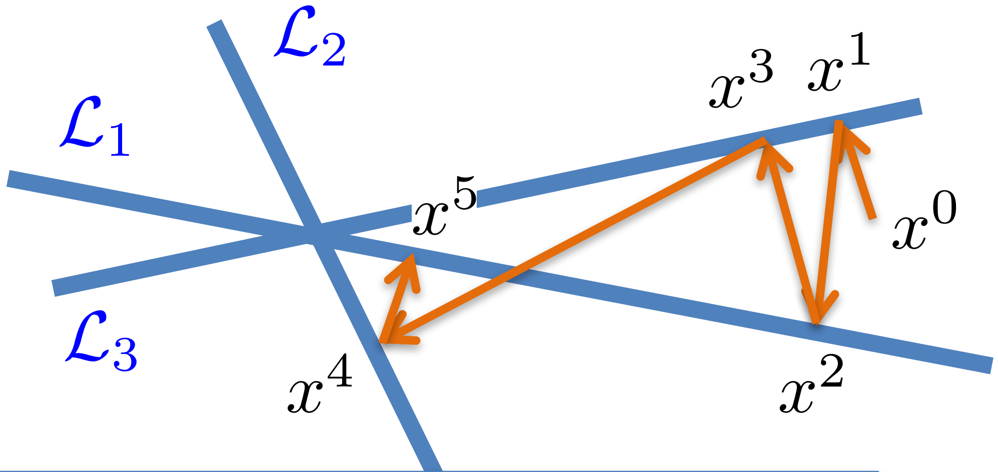
$$\mathcal{L}_i \stackrel{\text{def}}{=} \{x \; : \; a_i^\top x = b_i\}$$

Special Case: Algorithm

Algorithm (Stochastic Projection Method)

1. Choose random $i \in \{1, 2, \ldots, m\}$
2. $x^{t+1} = \Pi_{\mathcal{L}_i}(x^t)$

Projection onto $\mathcal{L}_i$
(Stochastic set)



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)

Summary

Deterministic concept	Decomposition	Stochastic estimate
Function f	$f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$	Stochastic function $f_i(x)$
Gradient $\nabla f(x)$	$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x)$	Stochastic gradient $\nabla f_i(x)$
Hessian $\nabla^2 f(x)$	$\nabla^2 f(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla^2 f_i(x)$	Stochastic Hessian $\nabla^2 f_i(x)$
Preconditioned system $PAx = Pb$	$P = \frac{1}{m}\sum_{i=1}^{m} P_i$	Stochastic system $P_i Ax = P_i b$
Preconditioner P	$P = \frac{1}{m}\sum_{i=1}^{m} P_i$	Stochastic preconditioner P_i
Operator $\phi(x)$	$\phi(x) = \frac{1}{m}\sum_{i=1}^{m} \phi_i(x)$	Stochastic operator $\phi_i(x)$
Set $\mathcal{L}$	$\mathcal{L} = \bigcap_{i=1}^{m} \mathcal{L}_i$	Stochastic set $\mathcal{L}_i$

Stochastic Reformulations

Reformulation	Key concepts	Algorithm (special case)
Stochastic optimization problem Minimize $\frac{1}{m}\sum_{i=1}^m f_i(x)$	stochastic function stochastic gradient stochastic Hessian	**Stochastic gradient descent** $x^{t+1} = x^t - \nabla f_i(x^t)$
Stochastic linear system Solve $\left(\frac{1}{m}\sum_{i=1}^m P_i\right) Ax = \left(\frac{1}{m}\sum_{i=1}^m P_i\right) b$	stochastic system stochastic precondition.	**Stochastic precond. method** $x^{t+1} = \arg\min_{x\,:\,P_iAx=P_ib} \\|x - x^t\\|$
Stochastic fixed point problem Solve $x = \frac{1}{m}\sum_{i=1}^m \phi_i(x)$	stochastic operator	**Stochastic fixed point method** $x^{t+1} = \phi_i(x^t)$
Stochastic intersection problem Find $x \in \bigcap_{i=1}^m \mathcal{L}_i$	stochastic set	**Stochastic projection method** $x^{t+1} = \Pi_{\mathcal{L}_i}(x^t)$

Basic Method

Methods Beyond the Special Case

We proposed some "natural" methods in the special case:

$\mathcal{D}$ is defined by: $S = e_i$ with probability $1/m$

$B = I$ (identity matrix)

We now proceed to the general case:

- General $\mathcal{D}$
- General B
- Introduction of a stepise $\omega > 0$
- more methods: stochastic Newton, stochastic proximal point method

Basic Method

Stochastic Gradient Descent

a key method in stochastic optimization
& machine learning

Stochastic Optimization Problem

$$\text{Minimize } f(x) \overset{\text{def}}{=} \mathbf{E}_{S\sim\mathcal{D}}[f_S(x)]$$

constant stepsize

$S^t \sim \mathcal{D}$

$$x^{t+1} = x^t - \omega \nabla f_{S^t}(x^t)$$

stochastic gradient

Stochastic Newton Method

Stochastic Optimization Problem

$$\text{Minimize } f(x) \overset{\text{def}}{=} \mathbf{E}_{S\sim\mathcal{D}}[f_S(x)]$$

$$S^t \sim \mathcal{D}$$

Constant stepsize

stochastic gradient

$$x^{t+1} = x^t - \omega(\nabla^2 f_{S^t})^{\dagger_B} \nabla f_{S^t}(x^t)$$

B- pseudoinverse of the **stochastic Hessian**

Stochastic Proximal Point Method

Stochastic Optimization Problem

$$\text{Minimize } f(x) \overset{\text{def}}{=} \mathbf{E}_{S\sim\mathcal{D}}[f_S(x)]$$

$$S^t \sim \mathcal{D}$$

$$x^{t+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ f_{S^t}(x) + \frac{1-\omega}{2\omega}\|x - x^t\|_B^2 \right\}$$

Stochastic function
(unbiased estimate of *f*)

Term encouraging proximity
to the last iterate

Stochastic Preconditioning Method

Stochastic Linear System

$$\text{Solve } PAx = Pb$$

$$P = \mathbf{E}_{S\sim\mathcal{D}}[B^{-1}A^\top H_S]$$

$$S^t \sim \mathcal{D}$$

$$x^{t+1} = \arg\min_{x \,:\, P_{S^t}Ax = P_{S^t}b} \|x - x^t\|_B$$

Stochastic preconditioner
(unbiased estimator of *P*)

Stochastic Fixed Point Method

Stochastic Fixed Point Problem

$$\text{Solve } x = \phi(x)$$

$$\phi(x) = \mathbf{E}_{S\sim\mathcal{D}}\left[\phi_S(x)\right]$$

$$\phi_S(x) = \Pi^B_{\mathcal{L}_S}(x)$$

Stochastic operator
(unbiased estimator of the fixed point operator $\phi(x)$)

$$S^t \sim \mathcal{D}$$

$$x^{t+1} = \omega\phi_{S^t}(x^t) + (1-\omega)x^t$$

Relaxation parameter

Stochastic Projection Method

Stochastic Intersection Problem

Find $x \in \bigcap_{S \sim \mathcal{D}} \mathcal{L}_S$

Stochastic projection map

$$x^{t+1} = \omega \Pi^B_{\mathcal{L}_{S^t}}(x^t) + (1-\omega)x^t$$

Stochastic set
"unbiased" estimator of the set
$\bigcap_{S \sim \mathcal{D}} \mathcal{L}_S$

Relaxation parameter

Equivalence & Exactness

Equivalence of Reformulations

Theorem

The 4 stochastic reformulations are equivalent

set of minimizers of the stochastic optimization problem
=
set of solutions of the stochastic linear system
=
set of fixed points of the stochastic fixed point problem
=
set of solutions of the stochastic intersection problem

Equivalence of Algorithms

Theorem

All algorithms we described are equivalent

1. Stochastic Gradient Descent
2. Stochastic Newton Method
3. Stochastic Proximal Point Method
4. Stochastic Preconditioning Method
5. Stochastic Fixed Point Method
6. Stochastic Projection Method

Exactness of Reformulations

Theorem

$$\mathbf{E}[H_S] \succ 0$$

The set of solutions of all 4 stochastic problems is

$$\mathcal{L} \overset{\text{def}}{=} \{x \; : \; Ax = b\}$$

set of minimizers of the stochastic optimization problem
=
set of solutions of the stochastic linear system
=
set of fixed points of the stochastic fixed point problem
=
set of solutions of the stochastic intersection problem

Summary

Deterministic concept	**Decomposition**	**Stochastic estimate**
Function f	$f(x) = \mathbf{E}[f_S(x)]$	Stochastic function $f_S(x) = \frac{1}{2}\\|Ax-b\\|_{H_S}^2$
Gradient $\nabla f(x)$	$\nabla f(x) = \mathbf{E}[\nabla f_S(x)]$	Stochastic gradient $\nabla f_S(x) = A^\top H_S(Ax-b)$
Hessian $\nabla^2 f(x)$	$\nabla^2 f(x) = \mathbf{E}\left[\nabla^2 f_S(x)\right]$	Stochastic Hessian $\nabla^2 f_S(x) = A^\top H_S A$
Preconditioner P	$P = \mathbf{E}[P_S]$	Stochastic preconditioner $P_S = B^{-1}A^\top H_S$
Preconditioned system $PAx = Pb$	$PA = \mathbf{E}[P_S A]$ $Pb = \mathbf{E}[P_S b]$	Stochastic system $P_S A x = P_S b$
Operator $\phi(x)$	$\phi(x) = \mathbf{E}\left[\Pi^B_{\mathcal{L}_S}(x)\right]$	Stochastic operator $\phi_S(x) = \Pi^B_{\mathcal{L}_S}(x)$
Set $\mathcal{L}$	$\mathcal{L} = \bigcap_{S\sim\mathcal{D}} \mathcal{L}_S$ $\mathbf{E}_{S\sim\mathcal{D}}\left[1_{\mathcal{L}_S}(x)\right] = 1_{\bigcap_{S\sim\mathcal{D}}\mathcal{L}_S}(x)$	Stochastic set $\mathcal{L}_S = \{x \ : \ S^\top A x = S^\top b\}$

REFORMULATION	BASIC METHOD
Stochastic optimization problem Minimize $\quad f(x)$ $f(x) = \mathbf{E}[f_S(x)]$	**SGD** $\quad x^{t+1} = x^t - \omega \nabla f_{S^t}(x^t)$ **SNM** $\quad x^{t+1} = x^t - \omega (\nabla^2 f_{S^t})^{\dagger_B} \nabla f_{S^t}(x^t)$ **SPPM** $\quad x^{t+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ f_{S^t}(x) + \frac{1-\omega}{2\omega}\|x - x^t\|_B^2 \right\}$
Stochastic linear system Solve $\quad PAx = Pb$ $P = \mathbf{E}[P_S]$	**Stochastic Preconditioning Method (SPM)** $x^{t+1} = \arg\min_{x \,:\, P_{S^t}Ax = P_{S^t}b} \|x - x^t\|_B$
Stochastic fixed point problem Solve $\quad x = \phi(x)$ $\phi(x) = \mathbf{E}[\phi_S(x)]$	**Stochastic Fixed Point Method (SFPM)** $x^{t+1} = \omega \phi_{S^t}(x^t) + (1-\omega)x^t$
Stochastic intersection problem Find $\quad x \in \mathcal{L}$ $\mathcal{L} = \bigcap_{S\sim\mathcal{D}} \mathcal{L}_S$	**Stochastic Projection Method (SPM)** $x^{t+1} = \omega \Pi^B_{\mathcal{L}_{S^t}}(x^t) + (1-\omega)x^t$

Convergence

Key Matrix

(captures the convergence of the basic method)

$$W \overset{\text{def}}{=} B^{-1/2} A^\top \mathbf{E}_{S\sim\mathcal{D}}[H_S] A B^{-1/2}$$

$$H_S = S(S^\top A B^{-1} A^\top S)^\dagger S^\top$$

$$W = U \Lambda U^\top = \sum_{i=1}^{n} \lambda_i u_i u_i^\top$$

Eigenvalue decomposition

Smallest nonzero eigenvalue: $\lambda_{\min}^+$

Largest eigenvalue: $\lambda_{\max}$

Basic Method: Complexity

Theorem [R & Takáč, 2017]

$$\mathbf{E}[U^\top B^{1/2}(x^t - x^*)] = (I - \omega\Lambda)^t U^\top B^{1/2}(x^0 - x^*)$$

stepsize / relaxation parameter

$$W \stackrel{\text{def}}{=} B^{-1/2}A^\top \mathbf{E}_{S\sim\mathcal{D}}[H_S]AB^{-1/2} = U\Lambda U^\top$$

Basic Method: Complexity

Convergence of Expected Iterates

$$t \geq \frac{1}{\lambda_{\min}^{+}} \log\left(\frac{1}{\epsilon}\right) \xrightarrow{\omega = 1} \|\mathbf{E}[x^t - x^*]\|_B^2 \leq \epsilon$$

$$t \geq \frac{\lambda_{\max}}{\lambda_{\min}^{+}} \log\left(\frac{1}{\epsilon}\right) \xrightarrow{\omega = 1/\lambda_{\max}} \|\mathbf{E}[x^t - x^*]\|_B^2 \leq \epsilon$$

L2 Convergence

$$t \geq \frac{1}{\lambda_{\min}^{+}} \log\left(\frac{1}{\epsilon}\right) \xrightarrow{\omega = 1} \mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \epsilon$$

Parallel &
Accelerated Methods

Parallel Method

Parallel Method

"Run 1 step of the basic method from x^t
several times independently,
and average the results."

i.i.d.

$$x^{t+1} = \frac{1}{\tau}\sum_{i=1}^{\tau} \underbrace{\phi_\omega(x^t, S_i^t)}_{\text{One step of the basic method from } x^t}$$

Parallel Method: Complexity

L2 Convergence

$$\tau = 1 \qquad\qquad \tau = +\infty$$

$$t \geq \frac{1}{\lambda_{\min}^{+}} \log\left(\frac{1}{\epsilon}\right)$$

or

$$t \geq \frac{\lambda_{\max}}{\lambda_{\min}^{+}} \log\left(\frac{1}{\epsilon}\right)$$

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

Accelerated Method

Accelerated Method

Acceleration parameter (between 1 and 2)

$S^t, S^{t-1} \sim \mathcal{D}$ (independent)

$$x^{t+1} = \gamma \underbrace{\phi_\omega(x^t, S^t)}_{\text{One step of the basic method from } x^t} + (1-\gamma)\underbrace{\phi_\omega(x^{t-1}, S^{t-1})}_{\text{One step of the basic method from } x^{t-1}}$$

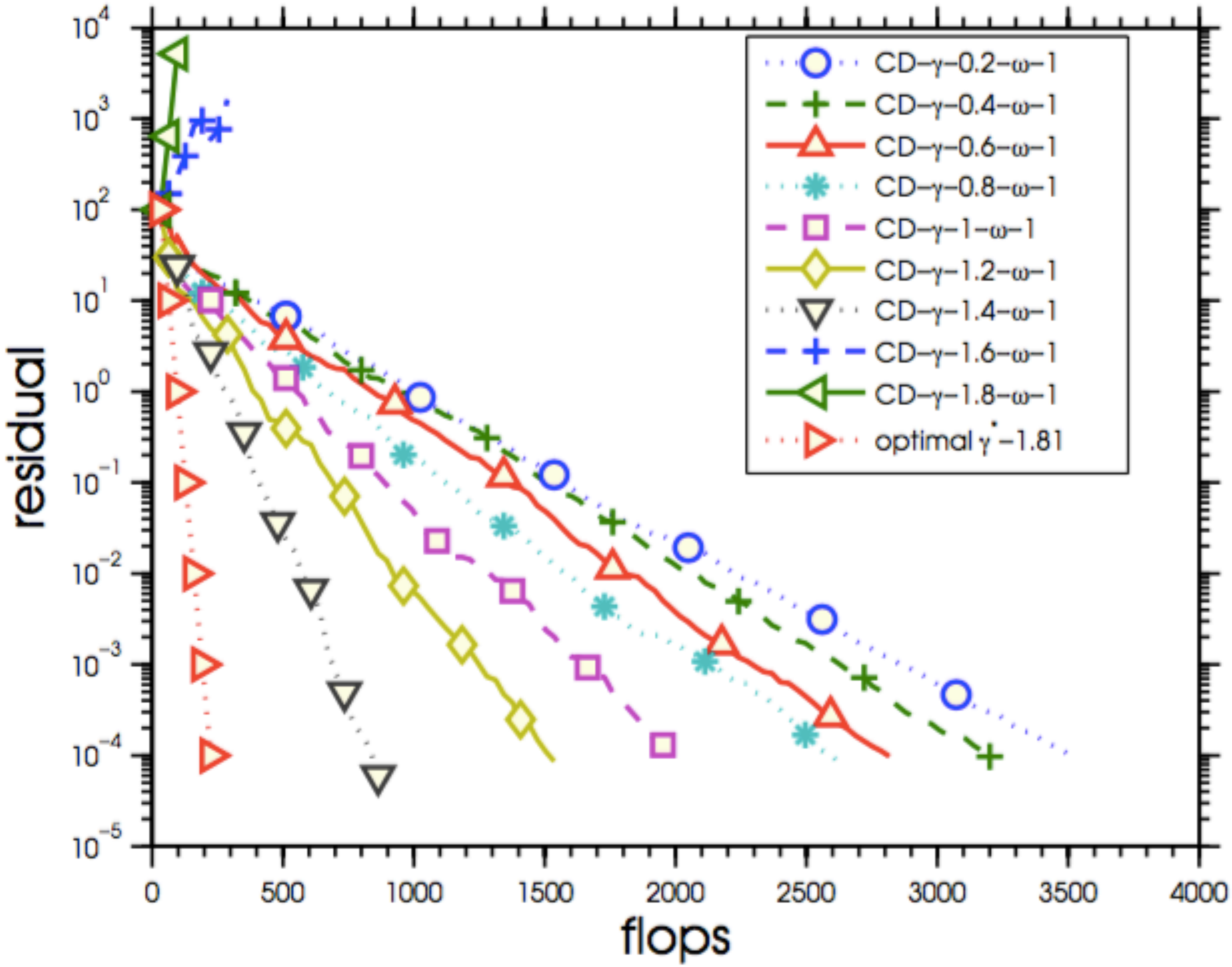
Accelerated Method: Complexity

Convergence of Iterates

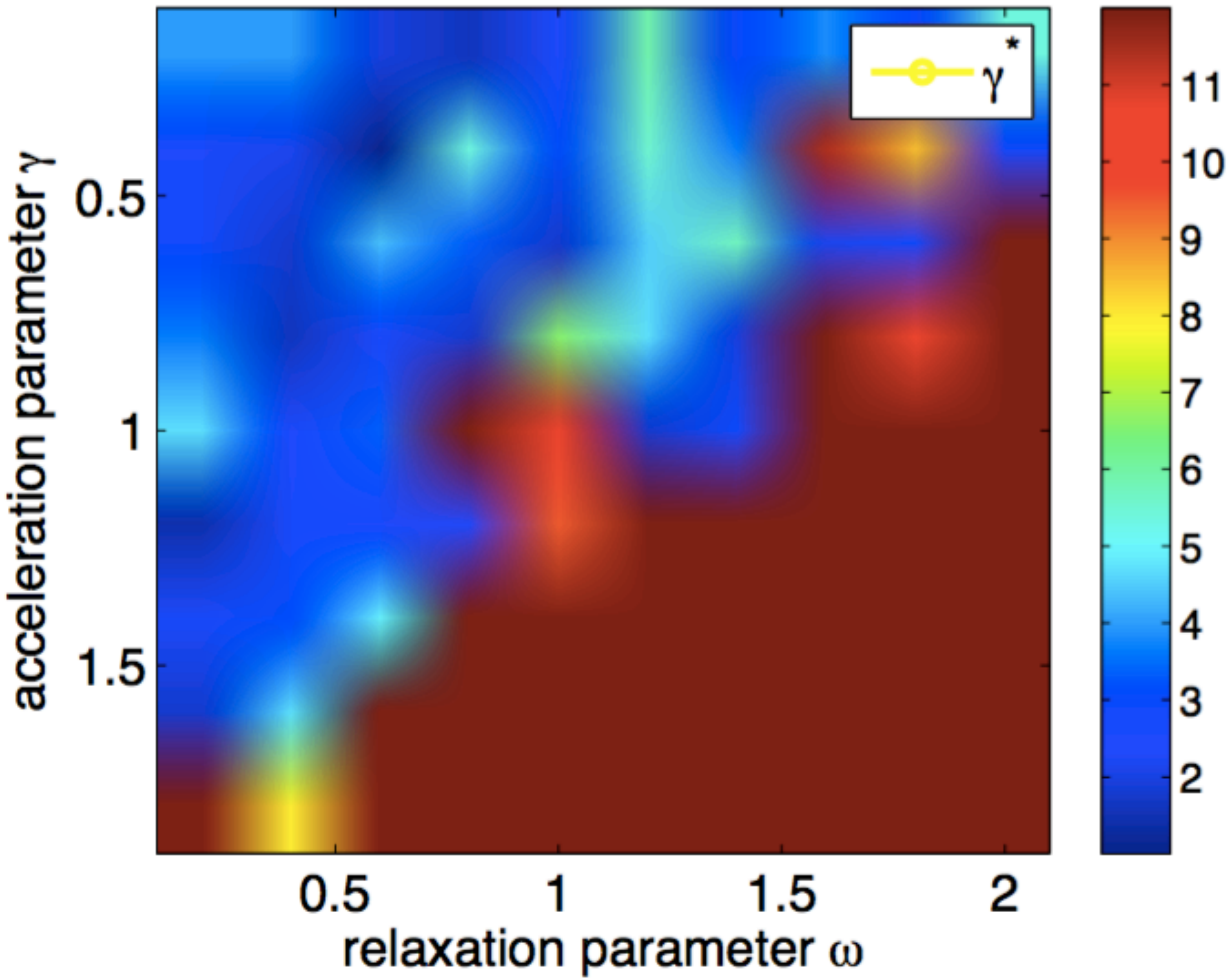
$$t \geq \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}^{+}}} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \|\mathbf{E}[x^t - x^*]\|_B^2 \leq \epsilon$$

Basic Method depends on $\frac{\lambda_{\max}}{\lambda_{\min}^{+}}$!

Acceleration Accelerates



More Relaxation Requires More Acceleration



Detailed Complexity Results

Alg.	ω	τ	γ	Quantity	Rate	Complexity	Theorem
1	1	-	-	$\lVert \mathrm{E}[x_k - x_*] \rVert_{\mathbf{B}}^2$	$(1-\lambda_{\min}^+)^{2k}$	$1/\lambda_{\min}^+$	4.3, 4.4, 4.6
1	$1/\lambda_{\max}$	-	-	$\lVert \mathrm{E}[x_k - x_*] \rVert_{\mathbf{B}}^2$	$(1-1/\zeta)^{2k}$	ζ	4.3, 4.4, 4.6
1	$\frac{2}{\lambda_{\min}^+ + \lambda_{\max}}$	-	-	$\lVert \mathrm{E}[x_k - x_*] \rVert_{\mathbf{B}}^2$	$(1-2/(\zeta+1))^{2k}$	ζ	4.3, 4.4, 4.6
1	1	-	-	$\mathrm{E}\left[\lVert x_k - x_* \rVert_{\mathbf{B}}^2\right]$	$(1-\lambda_{\min}^+)^{k}$	$1/\lambda_{\min}^+$	4.8
1	1	-	-	$\mathrm{E}[f(x_k)]$	$(1-\lambda_{\min}^+)^{k}$	$1/\lambda_{\min}^+$	4.10
2	1	τ	-	$\mathrm{E}\left[\lVert x_k - x_* \rVert_{\mathbf{B}}^2\right]$	$\left(1-\lambda_{\min}^+(2-\xi(\tau))\right)^{k}$		5.1
2	$1/\xi(\tau)$	τ	-	$\mathrm{E}\left[\lVert x_k - x_* \rVert_{\mathbf{B}}^2\right]$	$\left(1-\frac{\lambda_{\min}^+}{\xi(\tau)}\right)^{k}$	$\xi(\tau)/\lambda_{\min}^+$	5.1
2	$1/\lambda_{\max}$	∞	-	$\mathrm{E}\left[\lVert x_k - x_* \rVert_{\mathbf{B}}^2\right]$	$(1-1/\zeta)^{k}$	ζ	5.1
3	1	-	$\frac{2}{1+\sqrt{0.99\lambda_{\min}^+}}$	$\lVert \mathrm{E}[x_k - x_*] \rVert_{\mathbf{B}}^2$	$\left(1-\sqrt{0.99\lambda_{\min}^+}\right)^{2k}$	$\sqrt{1/\lambda_{\min}^+}$	5.3
3	$1/\lambda_{\max}$	-	$\frac{2}{1+\sqrt{0.99/\zeta}}$	$\lVert \mathrm{E}[x_k - x_*] \rVert_{\mathbf{B}}^2$	$\left(1-\sqrt{0.99/\zeta}\right)^{2k}$	$\sqrt{\zeta}$	5.3

Table 1: Summary of the main complexity results. In all cases, $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$ (the projection of the starting point onto the solution space of the linear system). "Complexity" refers to the number of iterations needed to drive "Quantity" below some error tolerance $\epsilon > 0$ (we suppress a $\log(1/\epsilon)$ factor in all expressions in the "Complexity" column). In the table we use the following expressions: $\xi(\tau) = \frac{1}{\tau} + (1-\frac{1}{\tau})\lambda_{\max}$ and $\zeta = \lambda_{\max}/\lambda_{\min}^+$.

Summary

Summary

- 4 Equivalent stochastic reformulations of a linear system
  - Stochastic optimization
  - Stochastic fixed point problem
  - Stochastic linear system
  - Probabilistic intersection
- 3 Algorithms
  - Basic (SGD, stochastic Newton method, stochastic fixed point method, stochastic proximal point method, stochastic projection method, …)
  - Parallel
  - Accelerated
- Iteration complexity guarantees for various measures of success
  - Expected iterates (closed form)
  - L1 / L2 convergence
  - Convergence of *f;* ergodic …

Related Work

Basic method with unit stepsize and full rank A:

Robert Mansel Gower and P.R.
Randomized Iterative Methods for Linear Systems
SIAM J. Matrix Analysis & Applications 36(4):1660-1690, 2015

- 2017 IMA Fox Prize (2^{nd} Prize) in Numerical Analysis
- Most downloaded SIMAX paper

Removal of full rank assumption + duality:

Robert Mansel Gower and P.R.
Stochastic Dual Ascent for Solving Linear Systems
arXiv:1512.06890, 2015

We now move here

Inverting matrices & connection to Quasi-Newton updates:

Robert Mansel Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
arXiv:1602.01768, 2016

Computing the pseudoinverse:

Robert Mansel Gower and P.R.
Linearly Convergent Randomized Iterative Methods for Computing the Pseudoinverse
arXiv:1612.06255, 2016

Application in machine learning:

Robert Mansel Gower, Donald Goldfarb and P.R.
Stochastic Block BFGS: Squeezing More Curvature out of Data
ICML 2016

Duality: Basic Method

Robert Mansel Gower (Edinburgh -> INRIA)

[GR'15a]

Robert Mansel Gower and P.R.
Randomized Iterative Methods for Linear Systems
SIAM Journal on Matrix Analysis and Applications 36(4):1660-1690, 2015

[GR'15b]

Robert Mansel Gower and P.R.
Stochastic Dual Ascent for Solving Linear Systems
arXiv:1512.06890, 2015

Recall the Initial Problem:
Solve a Linear System

$$m\left\{\overbrace{A}^{n}\,\overset{\in \mathbb{R}^n}{x}\right. = \left. b\right\}m$$

Assumption 1

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

Optimization Formulation

Primal Problem

$B \succ 0$

$$\begin{array}{ll} \text{minimize} & P(x) := \frac{1}{2}\|x - c\|_B^2 \\ \text{subject to} & Ax = b \\ & x \in \mathbb{R}^n \end{array}$$

$A \in \mathbb{R}^{m \times n}$

$\frac{1}{2}(x - c)^\top B(x - c)$

Dual Problem

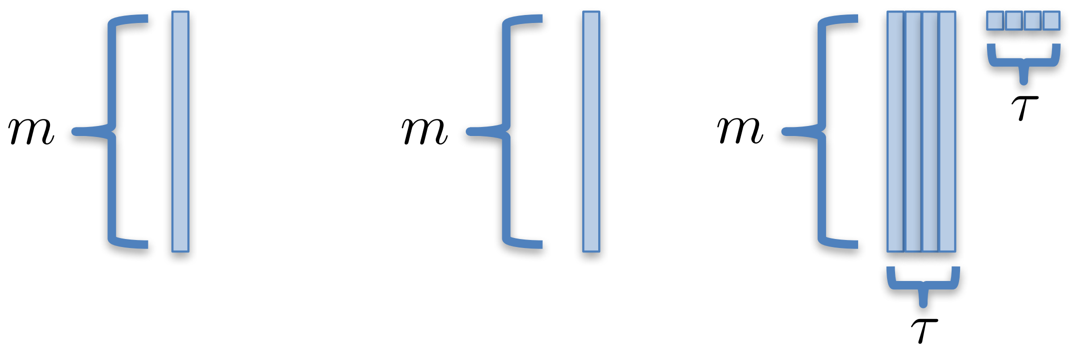
Unconstrained non-strongly concave quadratic maximization problem

$$\begin{array}{ll} \text{maximize} & D(y) := (b - Ac)^\top y - \frac{1}{2}\|A^\top y\|_{B^{-1}}^2 \\ \text{subject to} & y \in \mathbb{R}^m \end{array}$$

Stochastic Dual Subspace Ascent

A random $m \times \tau$ matrix drawn i.i.d. in each iteration $S \sim \mathcal{D}$

$$y^{t+1} = y^t + S\lambda^t$$



Moore-Penrose pseudo-inverse of a small $\tau \times \tau$ matrix

$$\lambda^t := \arg\min_{\lambda \in Q^t} \|\lambda\|_2$$

$$Q^t := \arg\max_{\lambda} D(y^t + S\lambda)$$

$$\lambda^t = \left(S^\top A B^{-1} A^\top S\right)^\dagger S^\top \left(b - A\left(c + B^{-1} A^\top y^t\right)\right)$$

$$x^* = \nabla g^*(A^\top y^*)$$

Dual Correspondence Lemma

Lemma

Affine mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$

$$x(y) := c + B^{-1}A^\top y$$

(Any) dual optimal point

Primal optimal point

$$\underbrace{D(y^*) - D(y)}_{\text{Dual error (in function values)}} = \underbrace{\tfrac{1}{2}\|x(y) - x^*\|_B^2}_{\text{Primal error (in distance)}}$$

Primal Method = Linear Image of the Dual Method

$$x^t := x(y^t) = c + B^{-1}A^\top y^t$$

Corresponding primal iterates

Dual iterates produced by SDA

Convergence

Main Assumption

Assumption 2

The matrix

$$\mathbf{E}_{S\sim\mathcal{D}}\left[\underbrace{S\left(S^\top A B^{-1} A^\top S\right)^\dagger S^\top}_{H_S}\right]$$

is nonsingular

Complexity of SDSA

$$\rho := 1 - \lambda_{\min}^{+}\left(B^{-1/2}A^{\top}\mathbf{E}[H]AB^{-1/2}\right)$$

$$U_0 = \frac{1}{2}\|x^0 - x^*\|_B^2$$

Theorem [Gower & R., 2015]

Primal iterates: $\mathbf{E}\left[\frac{1}{2}\|x^t - x^*\|_B^2\right] \leq \rho^t U_0$ GR'15a

Residual: $\mathbf{E}[\|Ax^t - b\|_B] \leq \rho^{t/2}\|A\|_B\sqrt{2 \times U_0}$

Dual error: $\mathbf{E}[OPT - D(y^t)] \leq \rho^t U_0$

Primal error: $\mathbf{E}[P(x^t) - OPT] \leq \rho^t U_0 + 2\rho^{t/2}\sqrt{OPT \times U_0}$

Duality gap: $\mathbf{E}[P(x^t) - D(y^t)] \leq 2\rho^t U_0 + 2\rho^{t/2}\sqrt{OPT \times U_0}$

The Rate: Lower and Upper Bounds

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

Theorem

$$0 \leq 1 - \frac{\mathbf{Rank}(S^\top A)}{\mathbf{Rank}(A)} \leq \rho < 1$$

Insight:

$\rho \leq 1$ always
$\rho < 1$ if Assumption 2 holds

Insight: The lower bound is good when:
i) the dimension of the search space in the "constrain and approximate" viewpoint is large,
ii) the rank of *A* is small

Extensions

Extensions 1

Robert Mansel Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
arXiv:1602.01768, 2016

Matrix Inversion
& Quasi-Newton Updates

Nicolas Loizou and P.R.
A New Perspective on Randomized Gossip Algorithms
In *Proceedings of The 4th IEEE Global Conference on Signal Processing*, 2016

Randomized Gossip
Algorithms

Extensions 2

Robert Mansel Gower, Donald Goldfarb and P.R.
Stochastic Block BFGS: Squeezing More Curvature Out of Data
In: *Proceedings of the 33th International Conference on Machine Learning, pp 1869-1878,* 2016

ERM

P.R. and Martin Takáč
Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory
arXiv:1706.01108, 2017

Stuff I talked about earlier…

Duality:
More Insights

1. Relaxation Viewpoint
"Sketch and Project"

$$\|x\|_B^2 = x^\top B x$$

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

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

S = identity matrix ➡ convergence in 1 step

$$\min_x\{\|x - x^0\| \ : \ Ax = 0\}$$

E.S. Coakley, V. Rokhlin and M. Tygert. **A Fast Randomized Algorithm for Orthogonal Projection**. *SIAM Journal on Scientific Computing* 33(2), pp. 849–868, 2011

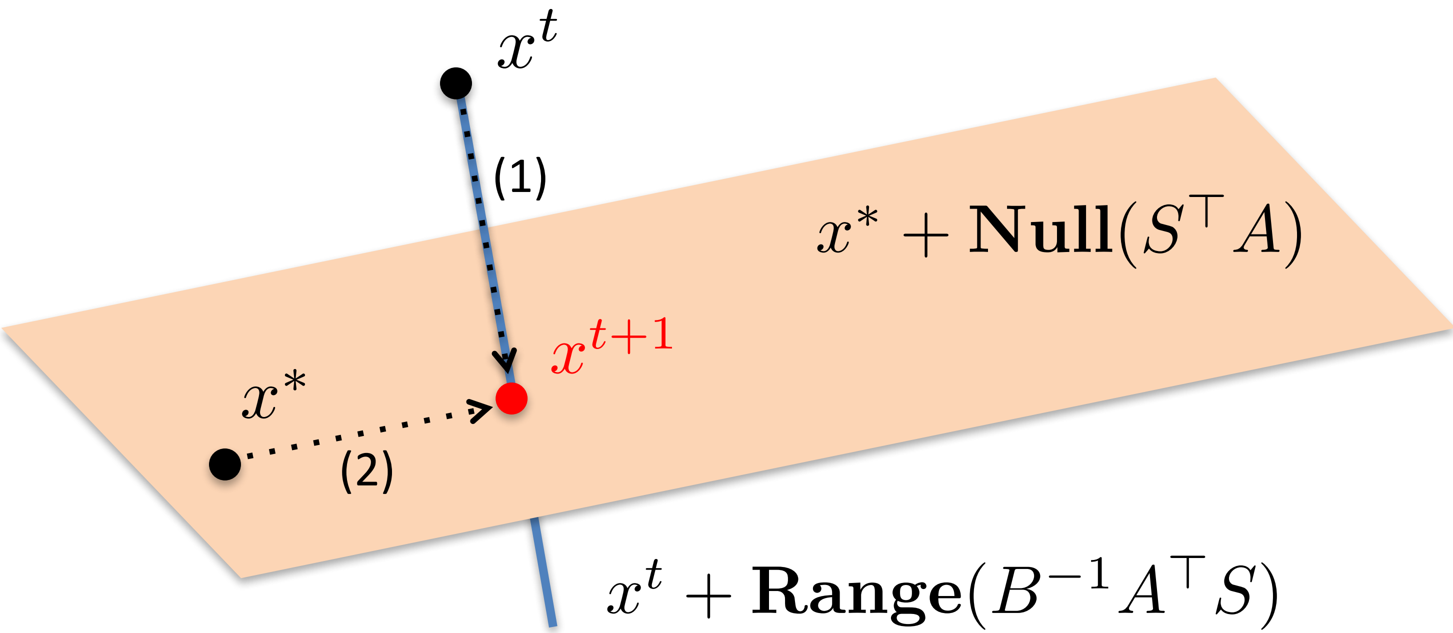
2. Approximation Viewpoint
"Constrain and Approximate"

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

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

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

3. Geometric Viewpoint "Random Intersect"



$$(1) \quad x^{t+1} = \arg\min_x \|x - x^t\|_B \quad \text{subject to} \quad S^\top A x = S^\top b$$

$$(2) \quad x^{t+1} = \arg\min_x \|x - x^*\|_B \quad \text{subject to} \quad x = x^t + B^{-1}A^\top S\lambda$$

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

4. Algebraic Viewpoint
"Random Linear Solve"

$$x^{t+1} \quad = \quad \text{solution in } x \text{ of the linear system}$$

$$S^\top A x = S^\top b$$

$$x = x^t + B^{-1} A^\top S \lambda$$

Unknown

Unknown

5. Algebraic Viewpoint
“Random Update”

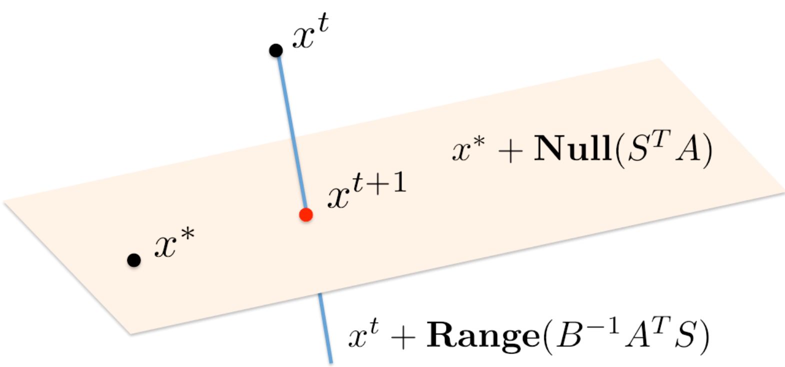
$$x^{t+1} = x^t - \overbrace{B^{-1}A^\top S(S^\top A B^{-1} A^\top S)^\dagger S^\top (Ax^t - b)}^{\text{Random Update Vector}}$$

Moore-Penrose pseudo-inverse

6. Analytic Viewpoint
"Random Fixed Point"

$$Z := A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top A$$

$$x^{t+1} - x^* = \underbrace{(I - B^{-1}Z)}_{\text{Random Iteration Matrix}}(x^t - x^*)$$



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

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

EXTRA TOPIC:
Special Cases

Special Case 1: 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:

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

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

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

RK: Derivation and Rate

General Method

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

Special Choice of Parameters

$$B = I$$

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

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

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

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

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

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

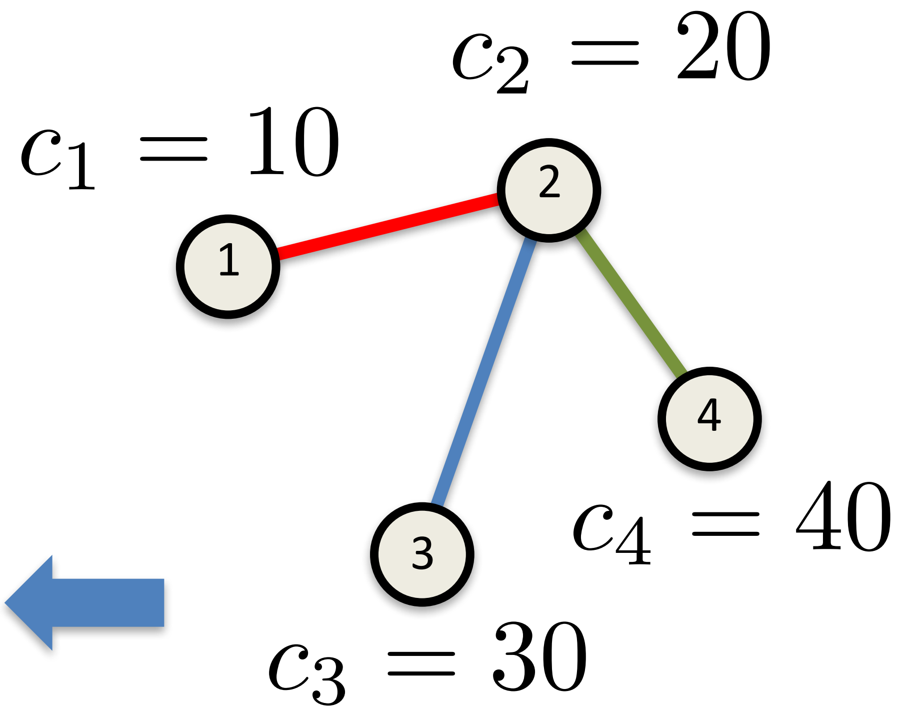
$$x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|_2^2 \quad \text{s.t.} \quad x = x^t + \boxed{y}(A_{i:})^T, \quad y \in \mathbb{R}$$

Application: Average Consensus

$$\min_{x \in \mathbb{R}^4} \frac{1}{2}\|x - c\|_2^2$$

$$\text{subject to } Ax = 0$$

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$



Insight: Randomized Kaczmarz = Randomized Gossip

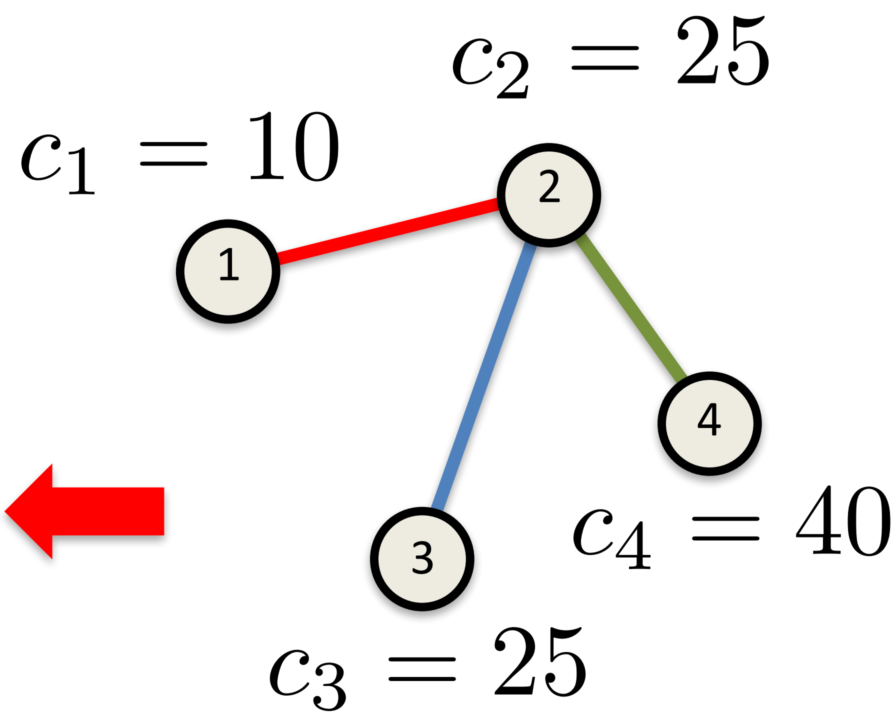
Now also have: dual interpretation, block variants, …

Application: Average Consensus

$$\min_{x\in\mathbb{R}^4} \frac{1}{2}\|x-c\|_2^2$$

$$\text{subject to } Ax = 0$$

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$



Insight: Randomized Kaczmarz = Randomized Gossip

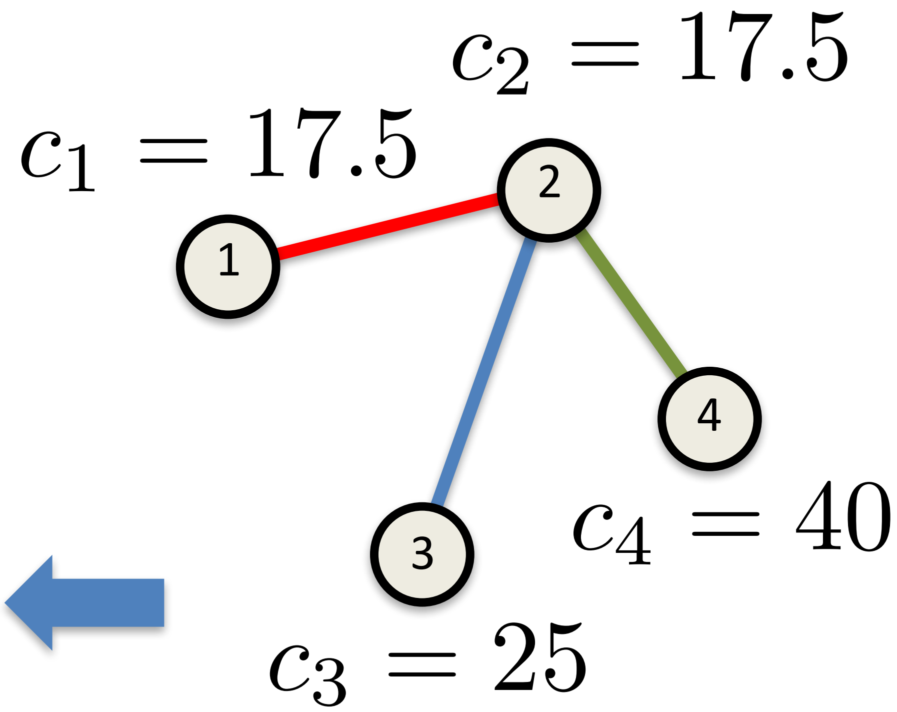
Now also have: dual interpretation, block variants, …

Application: Average Consensus

$$\min_{x\in\mathbb{R}^4} \frac{1}{2}\|x - c\|_2^2$$

$$\text{subject to } Ax = 0$$

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$



Insight: Randomized Kaczmarz = Randomized Gossip

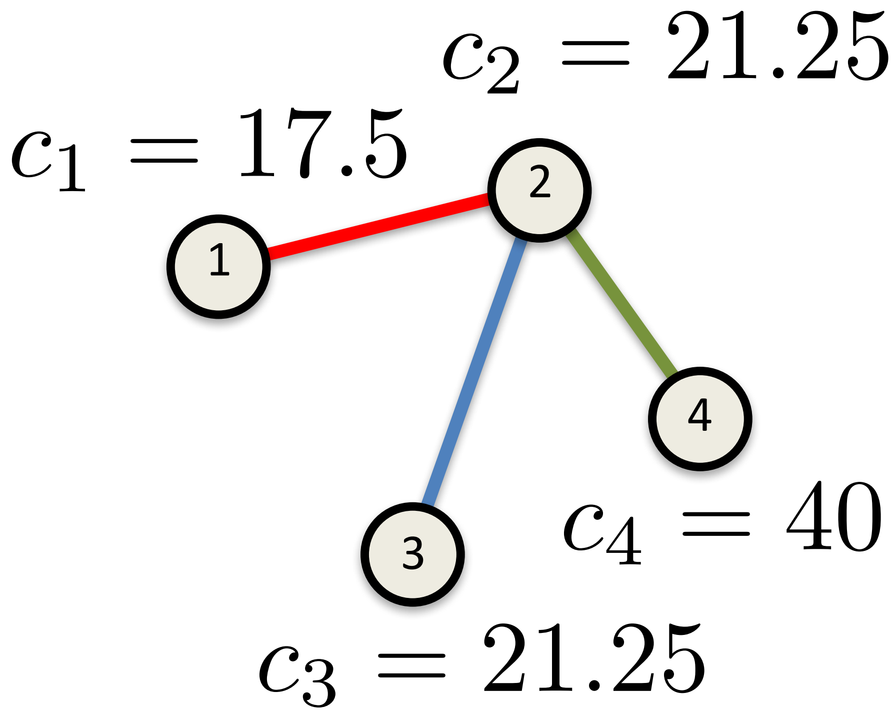
Now also have: dual interpretation, block variants, …

Application: Average Consensus

$$\min_{x\in\mathbb{R}^4} \frac{1}{2}\|x-c\|_2^2$$

$$\text{subject to } Ax = 0$$

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$



Insight: Randomized Kaczmarz = Randomized Gossip

Now also have: dual interpretation, block variants, …

RK: Further Reading

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

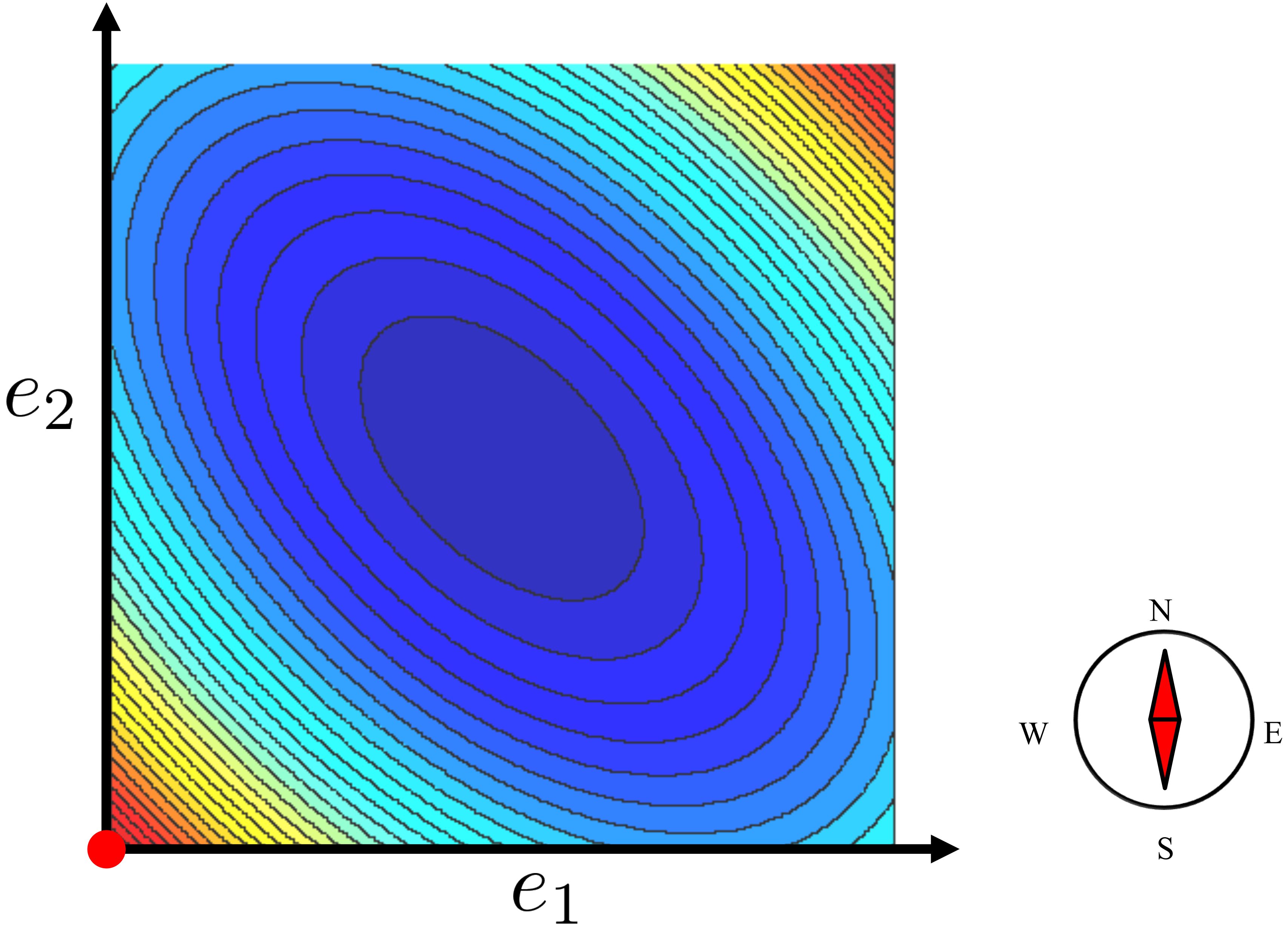
D. Needell and J. Tropp. **Paved with good intentions: analysis of a randomized block Kaczmarz method.** *Linear Algebra and its Applications* 441:199-221, 2012

D. Needell, N. Srebro and R. Ward. **Stochastic gradient descent, weighted sampling and the randomized Kaczmarz algorithm.** *Mathematical Programming* 155(1-2):549-573, 2016

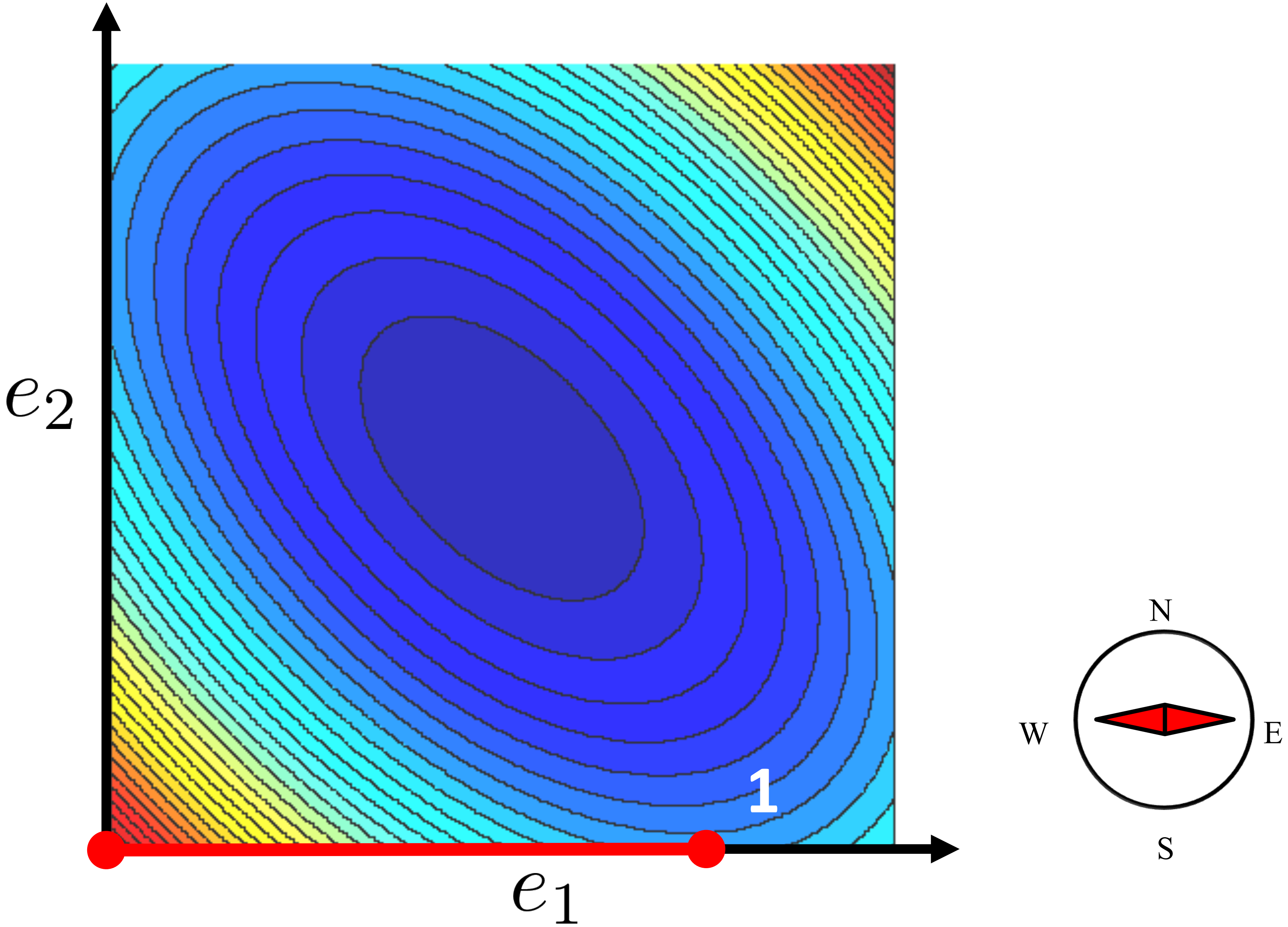
A. Ramdas. **Rows vs Columns for Linear Systems of Equations – Randomized Kaczmarz or Coordinate Descent?** *arXiv:1406.5295*, 2014

Special Case 2: Randomized Coordinate Descent

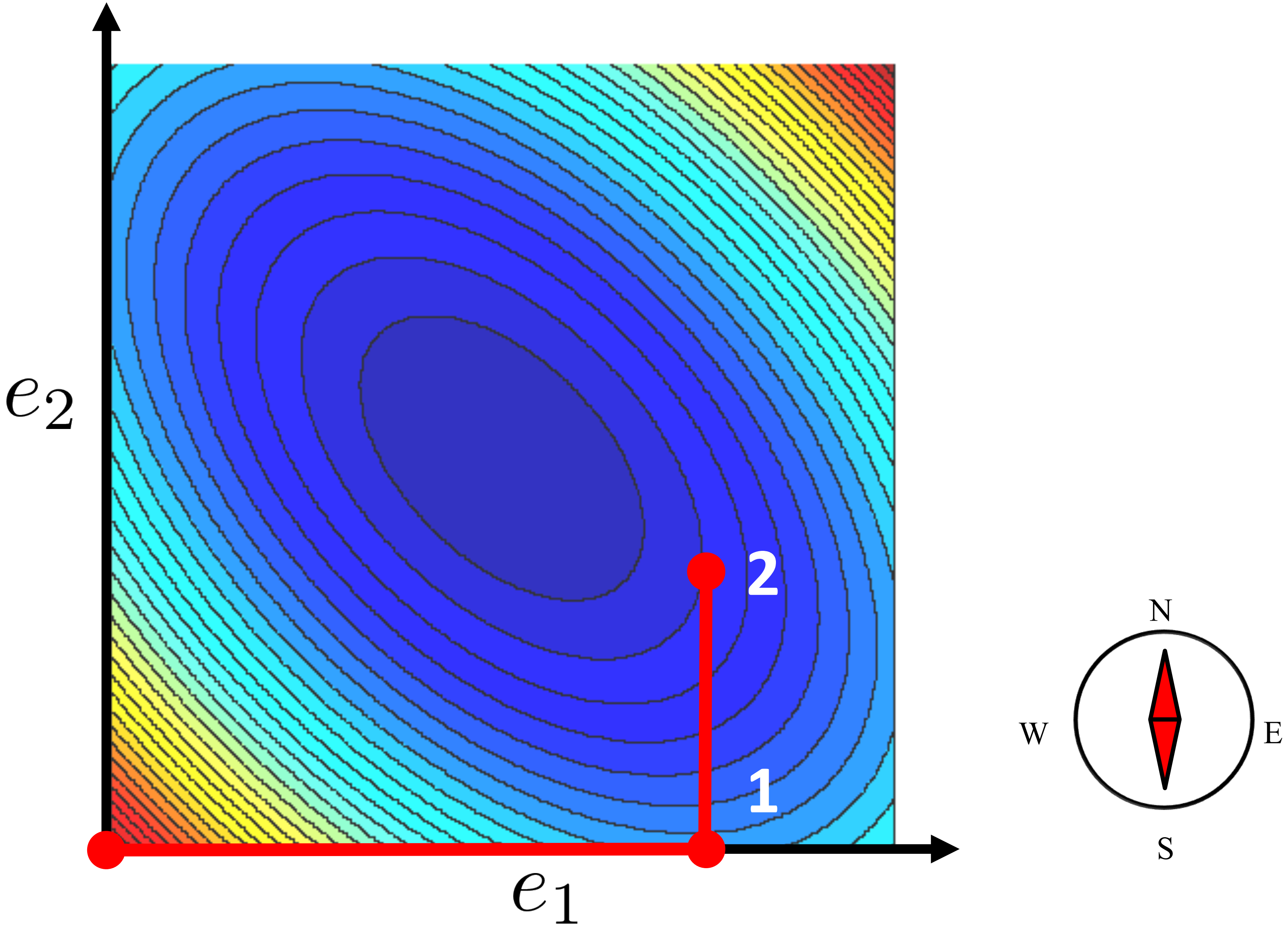
Randomized Coordinate Descent in 2D



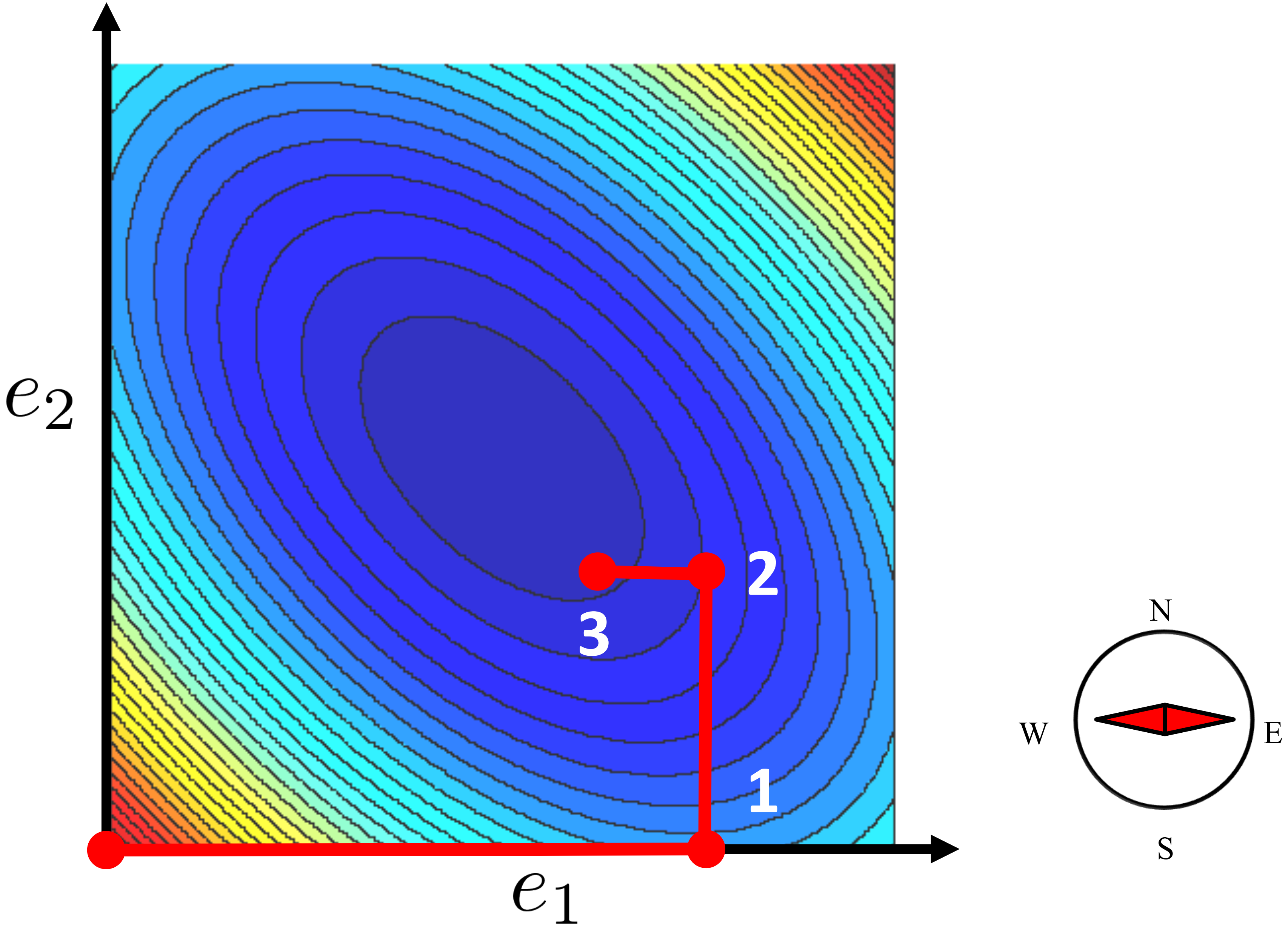
Randomized Coordinate Descent in 2D



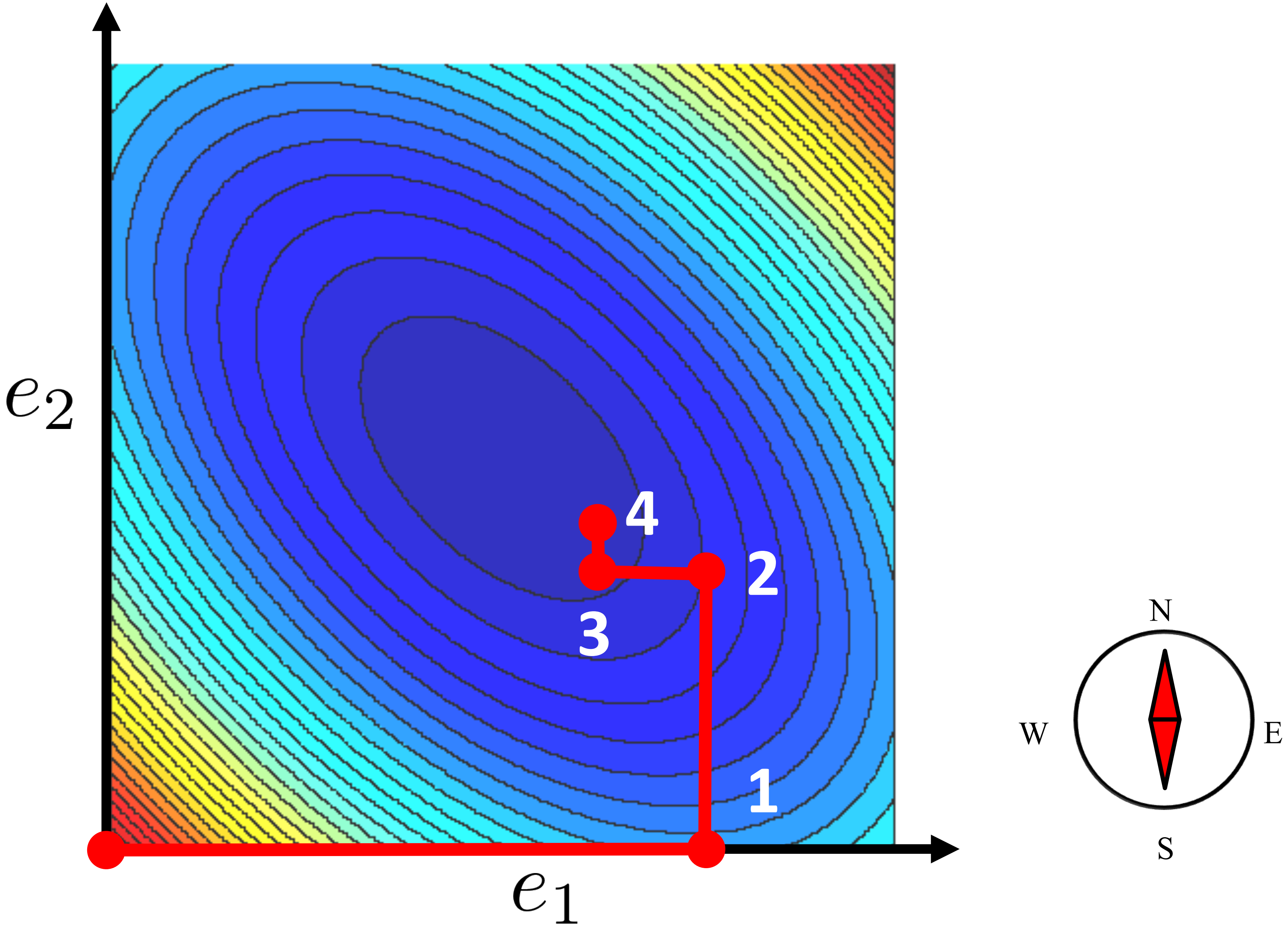
Randomized Coordinate Descent in 2D



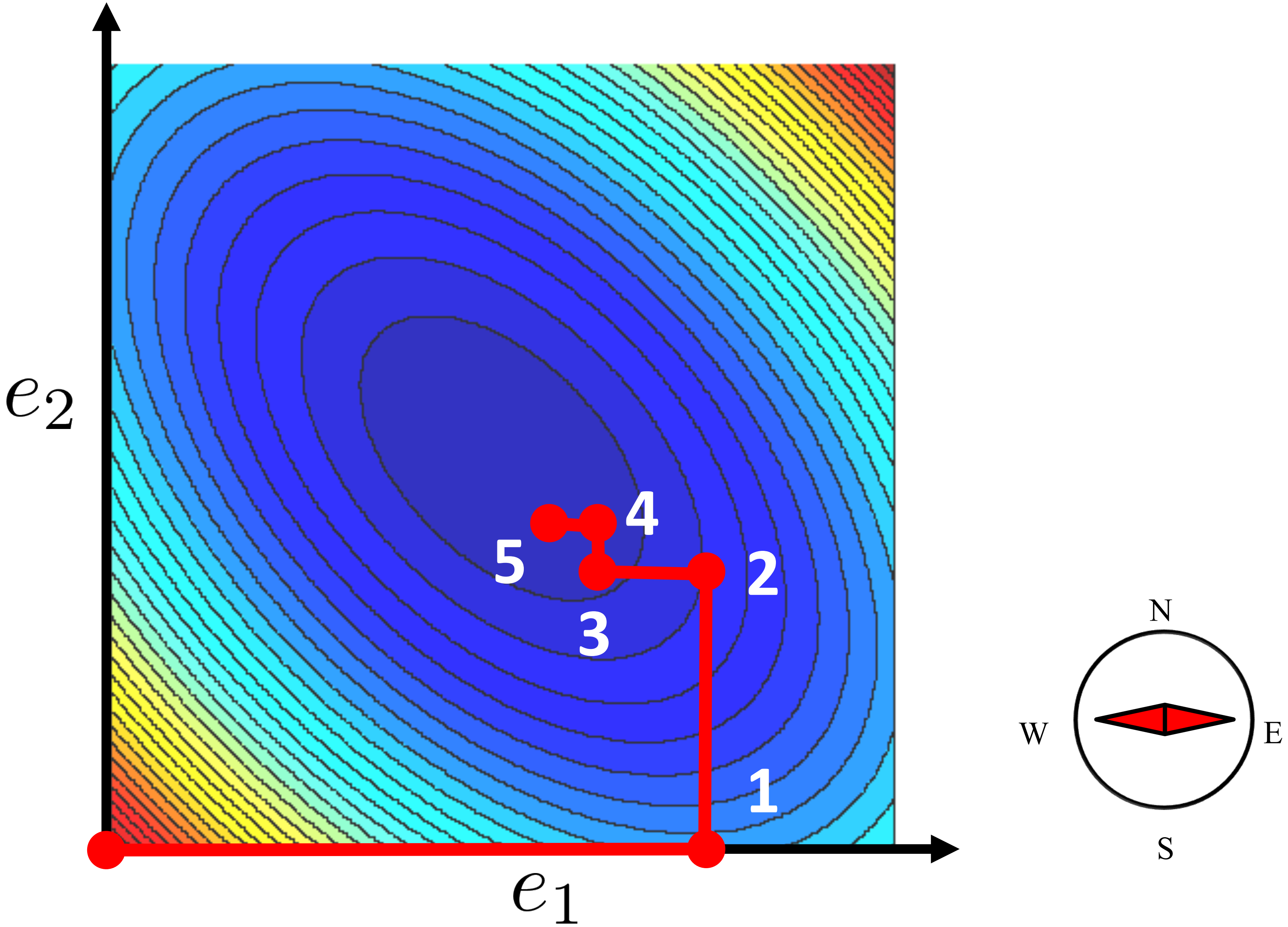
Randomized Coordinate Descent in 2D



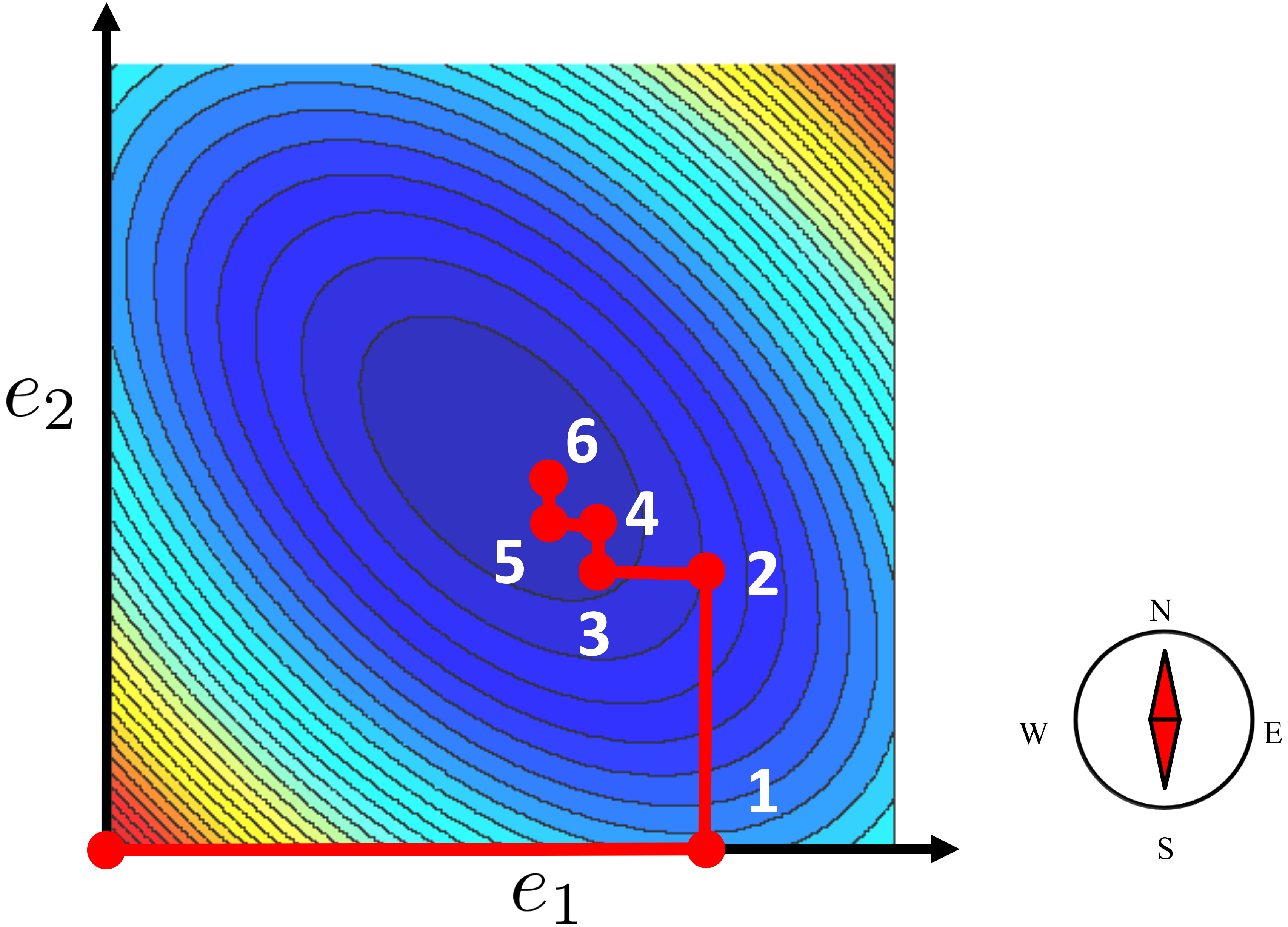
Randomized Coordinate Descent in 2D



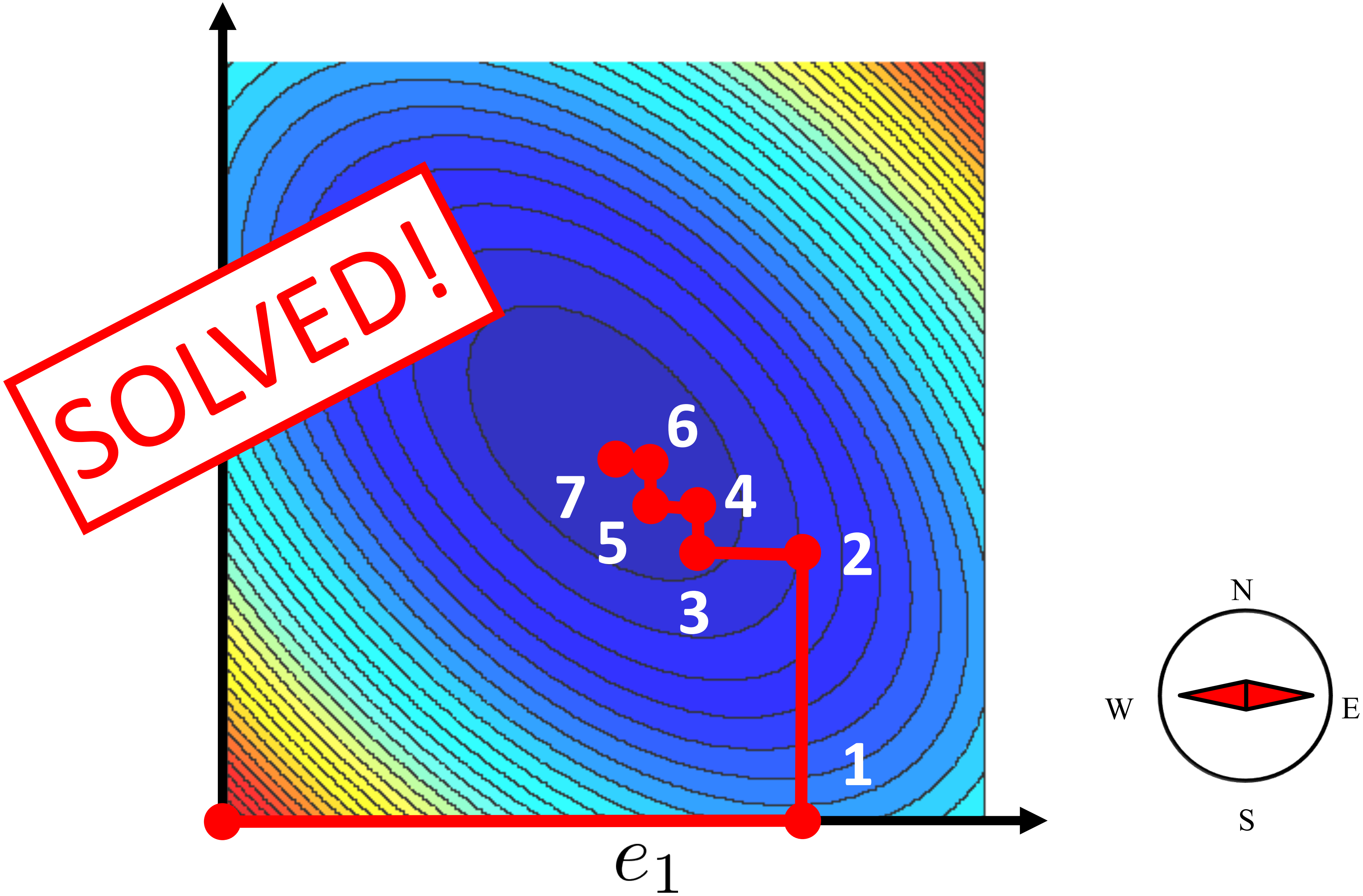
Randomized Coordinate Descent in 2D



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)

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

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

Assume: Positive definite

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

$B = A$ $\qquad$ $S = e^i = (0,\dots,0,1,0,\dots,0)$ 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

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

Special Choice of Parameters

$$B = A$$

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

$$S = e^i$$

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

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: "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*:

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

$$f(x) = \frac{1}{2} x^T A x - b^T x \quad \Rightarrow$$

$$L_i = A_{ii} \qquad \nabla_i f(x) = (A_{i:})^T x - b_i$$

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:

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

Experiment

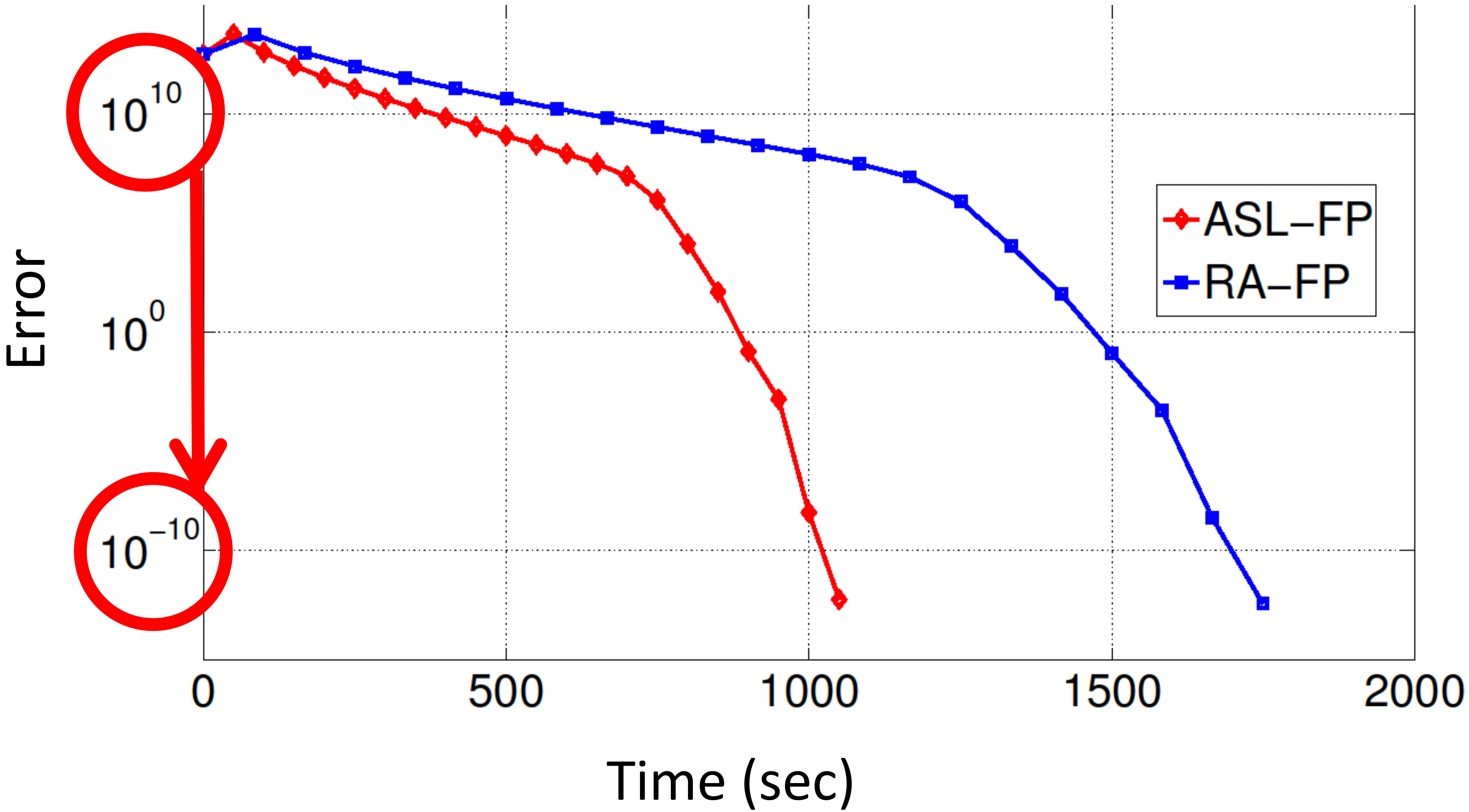
Machine: 128 nodes of Hector Supercomputer (4096 cores)

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

HYDRA

P.R. and Martin Takáč. **Distributed coordinate descent for learning with big data.** *Journal of Machine Learning Research* 17(75):1-25, 2016 (*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)

HYDRA²

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

Special Case 3: Randomized Newton Method

Randomized Newton (RN)

Z. Qu, PR, M. Takáč and O. Fercoq. **Stochastic Dual Newton Ascent for Empirical Risk Minimization.** *ICML* 2016

SDNA

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

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

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

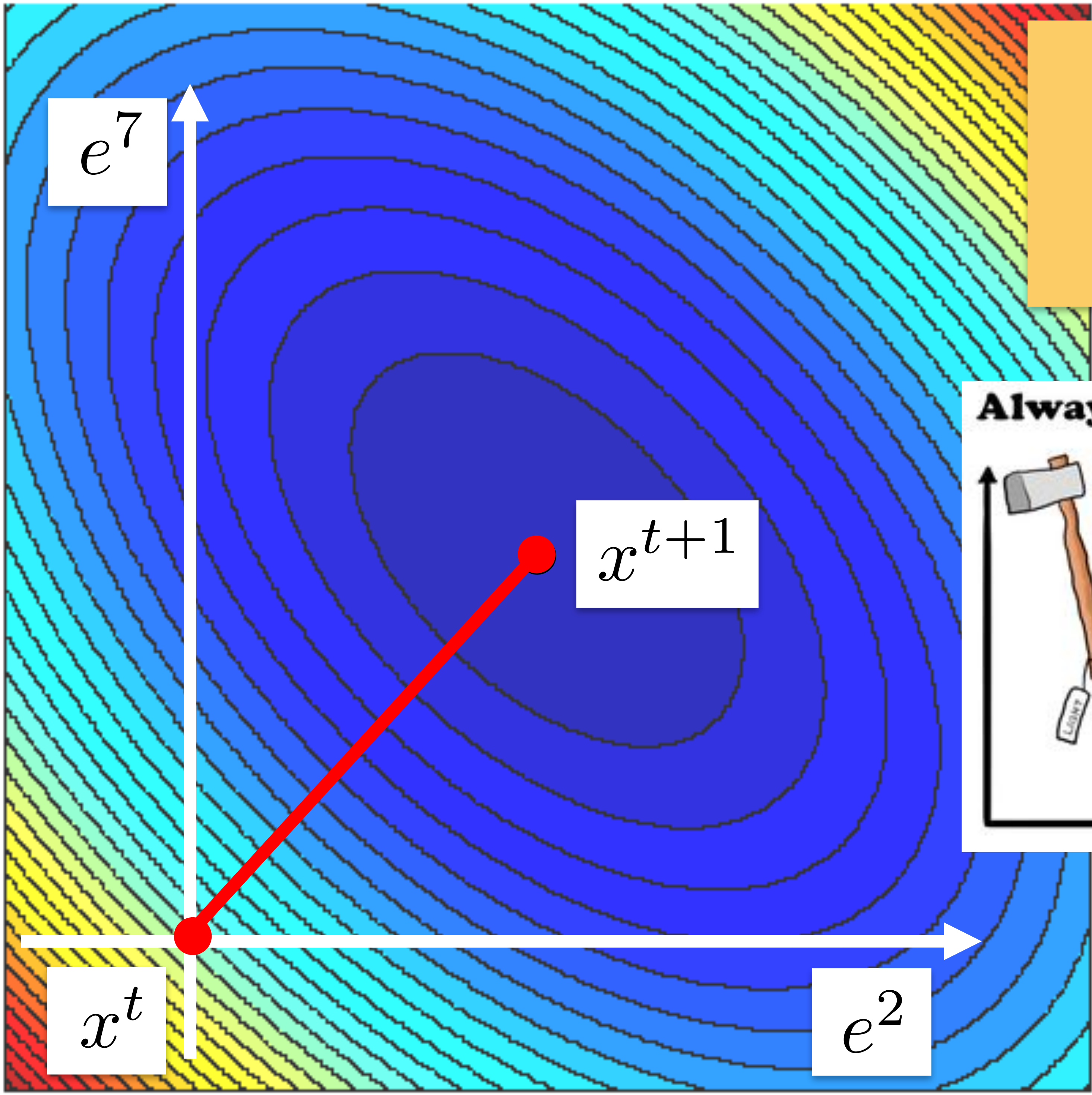
$$x^{t+1} = x^t - B^{-1}A^T S (S^T A B^{-1} A^T S)^{\dagger} S^T (Ax^t - b)$$

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*



$$C = \{2, 7\}$$

$$|C| = 2$$

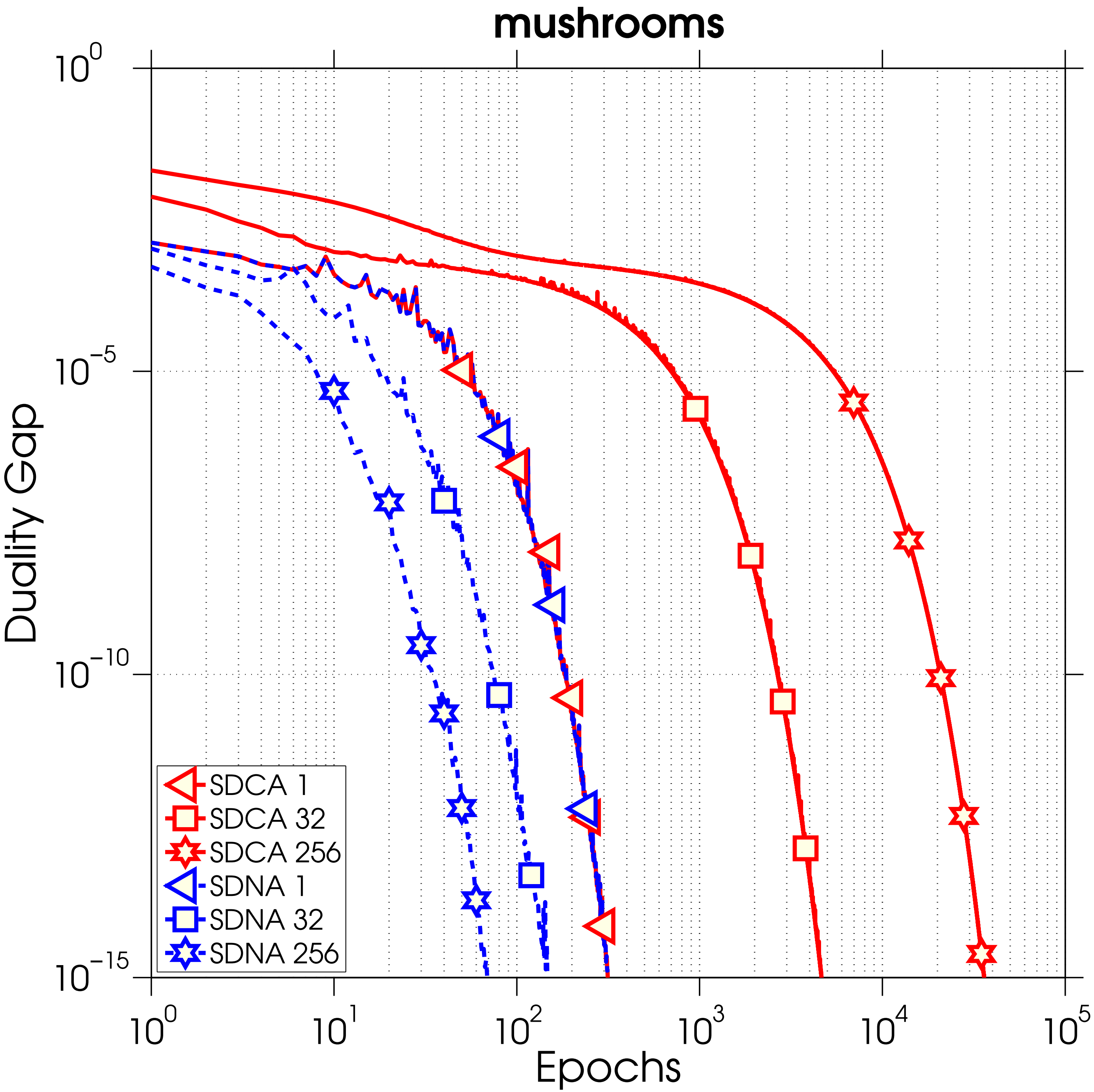
Experiment 4

Machine: laptop

Problem: Ridge Regression, n = 8124, d = 112

SDNA

Zheng Qu, P.R., Martin Takáč and Olivier Fercoq, **SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization.** *ICML*, 2016


mushrooms
Duality Gap
Epochs
10^0
10^-5
10^-10
10^-15
10^0
10^1
10^2
10^3
10^4
10^5
SDCA 1
SDCA 32
SDCA 256
SDNA 1
SDNA 32
SDNA 256

Special Case 4: Gaussian Descent

Gaussian Descent

General Method

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

Special Choice of Parameters

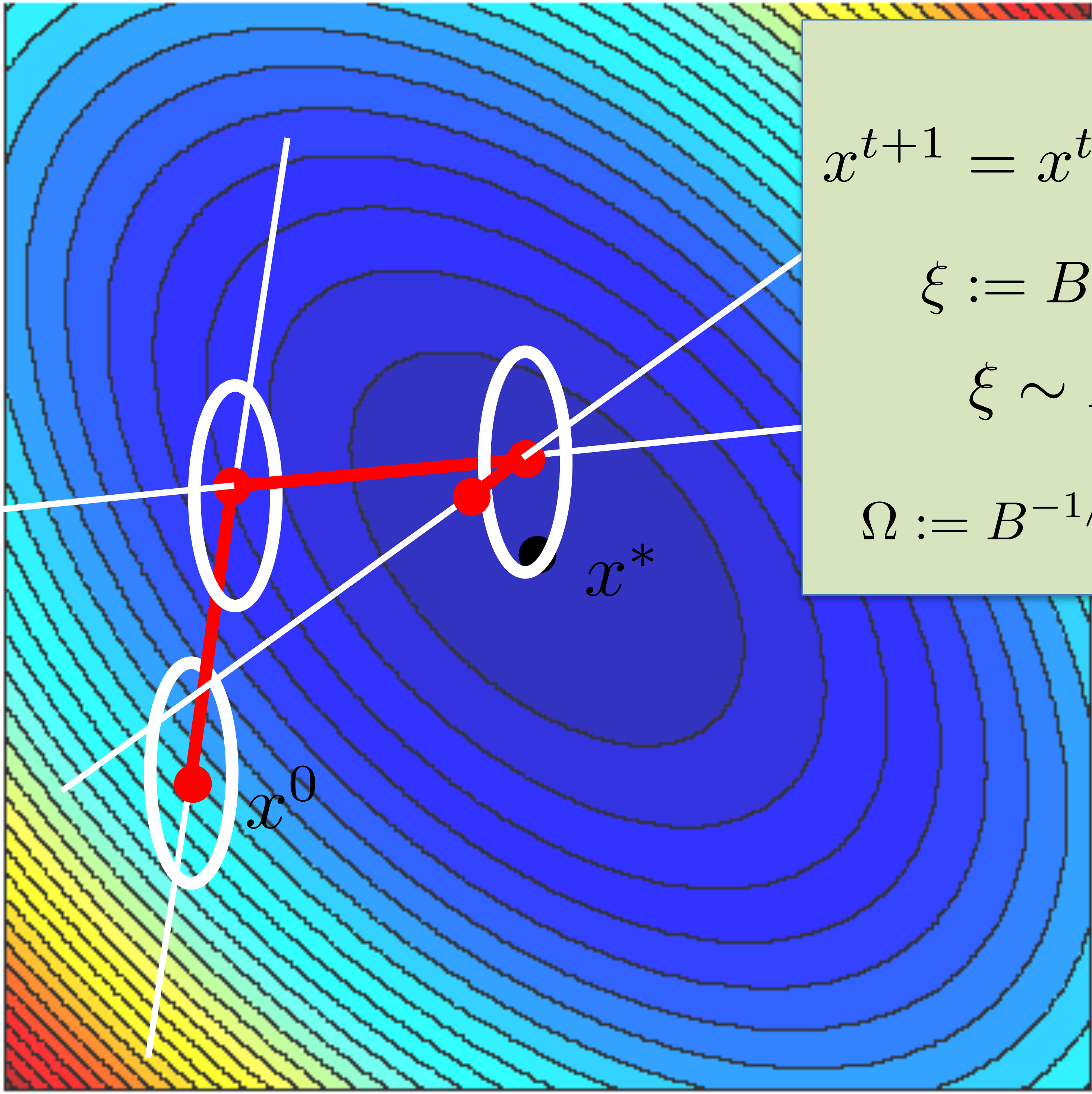
$$S \sim N(0, \Sigma)$$

Positive definite covariance matrix

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

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

Lemma [Gower & R, 2015]

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

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

This follows from the general lower bound

Gaussian Descent: Further Reading

Yurii Nesterov and Vladimir Spokoiny. **Random gradient-free minimization of convex functions.** *Foundations of Computational Mathematics* 17(2):527-566, 2017

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

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

EXTRA TOPIC: Stochastic Preconditioning

Stochastic Preconditioning

Definition [R & Takáč, 2017]

Given a family of randomized algorithms for solving some problem, indexed by a set of randomization strategies defining the family, how to choose the best method in the family?

Our context:

$$\text{How to choose } \mathcal{D} \text{ and } B?$$

Fixing Probabilities, Choosing Matrices

Formalizing the Problem

Consider family of distributions $\mathcal{D}$ parameterised as follows:

$$S = S_i \in \mathbb{R}^m \text{ (for } i = 1, 2, \ldots, m) \text{ with probability } 1/m$$

These vectors can be chosen !

Probabilities are fixed !

For simplicity, assume A is $n \times n$ and positive definite
Choose $B = A$

Recall:

Theorem [Gower & R, 2015] For the basic method we have

$$t \geq \frac{1}{\lambda_{\min}^+} \log\left(\frac{1}{\epsilon}\right) \quad \overset{\omega = 1}{\Longrightarrow} \quad \mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \epsilon$$

We will focus on maximizing this

Problem and Solution

$$W \stackrel{\text{def}}{=} B^{-1/2} A^\top \mathbf{E}_{S\sim\mathcal{D}}[H_S] A B^{-1/2}$$

$$\max_{S_1,\dots,S_m \in \mathbb{R}^m} \lambda_{\min}^{+}(W)$$

Theorem [Gower & R, 2015]

The optimal vectors $S_1, \dots, S_m$ are the eigenvectors of A.

Moreover, $W = \frac{1}{m} I$, and hence $\lambda_i = \frac{1}{m}$ for all i

Corollary

$$t \geq m \log\left(\frac{1}{\epsilon}\right) \quad \xrightarrow{\omega = 1} \quad \mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \epsilon$$

"Spectral" basic method (complexity independent of condition number)

Comments

- The spectral basic method is impractical in its pure form
  - Need to compute eigenvectors of *A*!
  - We ignore the fact that choice of *D* influences the cost of 1 iteration
- However, it highlights the potential power of stochastic preconditioning
- In generalizations (to convex/nonconvex opt), it only makes sense to consider a small family of distributions

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

It is natural to randomize over *i*.
This corresponds to the family:

$$S = e_i \text{ with probability } p_i > 0$$

$$x^{t+1} = x^t - \omega \nabla f_i(x^t)$$

Importance Sampling: Fixing Matrices, Choosing Probabilities

Formalizing the Problem

Consider family of distributions $\mathcal{D}$ parameterised as follows:

$$S = S_i \in \mathbb{R}^m \text{ (for } i = 1, 2, \ldots, r) \text{ with probability } p_i \geq 0$$

These vectors are fixed !

Probabilities can be chosen !

Theorem [Gower & R, 2015] For the basic method we have

$$t \geq \frac{1}{\lambda_{\min}^+} \log\left(\frac{1}{\epsilon}\right) \xRightarrow{\omega = 1} \mathbf{E}\left[\|x^t - x^*\|_B^2\right] \leq \epsilon$$

Again, we will focus on maximizing this

Problem and Solution

$$W \overset{\text{def}}{=} B^{-1/2} A^\top \mathbf{E}_{S \sim \mathcal{D}}[H_S] A B^{-1/2}$$

$$\max_{p_1,\ldots,p_r \geq 0,\ \sum_i p_i = 1} \lambda_{\min}^+(W)$$

Sometimes we know that $\lambda_{\min} > 0$

Then we can reformulate the above as a **semidefinite program:**

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

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

Leads to different (better) probabilities than ”Lipschitz” or “uniform” probabilities known in convex optimization. This is because we have more structure to exploit.

RCD: Optimal Probabilities can Lead to a Remarkable Improvement

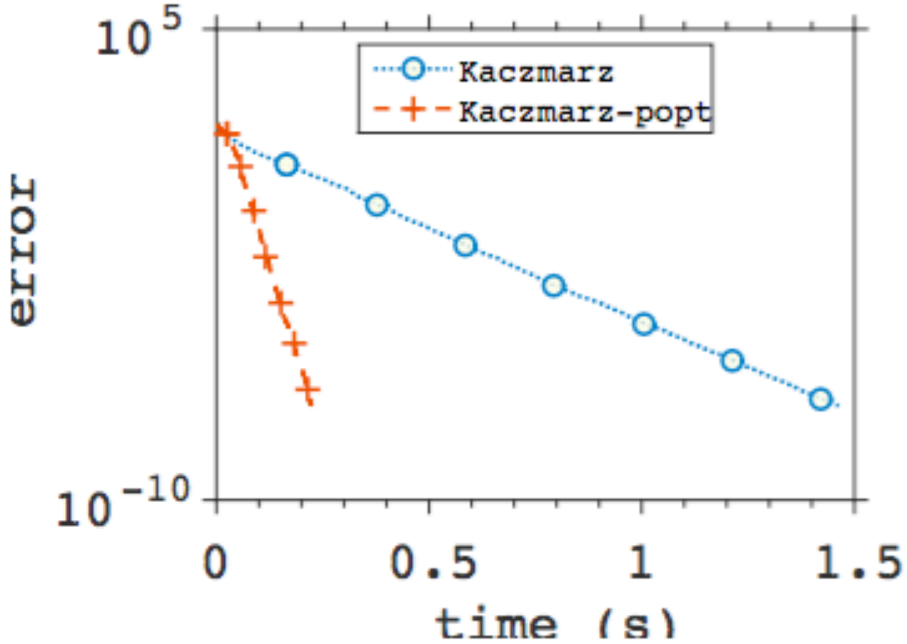
Rate for **convenient** (standard) **probabilities**

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

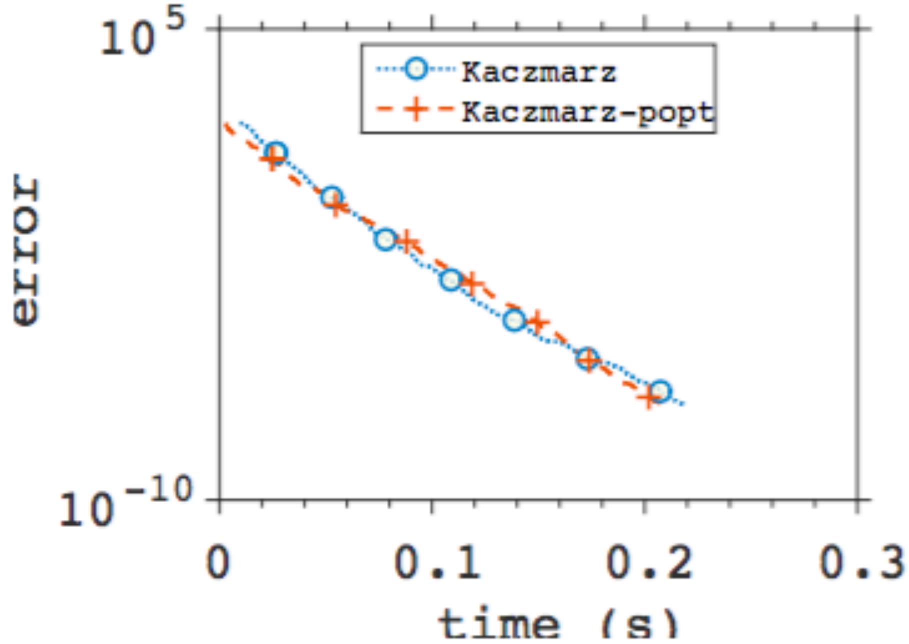
Lower bound on the rate

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	$1 - 2.17 \cdot \boxed{10^{-7}}$	$1 - 1.26 \cdot \boxed{10^{-4}}$	$1 - 7.81 \cdot 10^{-3}$
liver-disorders-ridge	$1 - 5.16 \cdot 10^{-4}$	$1 - 8.25 \cdot 10^{-3}$	$1 - 1.67 \cdot 10^{-1}$
covtype.binary-ridge	$1 - 7.57 \cdot \boxed{10^{-14}}$	$1 - 1.48 \cdot \boxed{10^{-6}}$	$1 - 1.85 \cdot 10^{-2}$

RK: Convenient vs Optimal

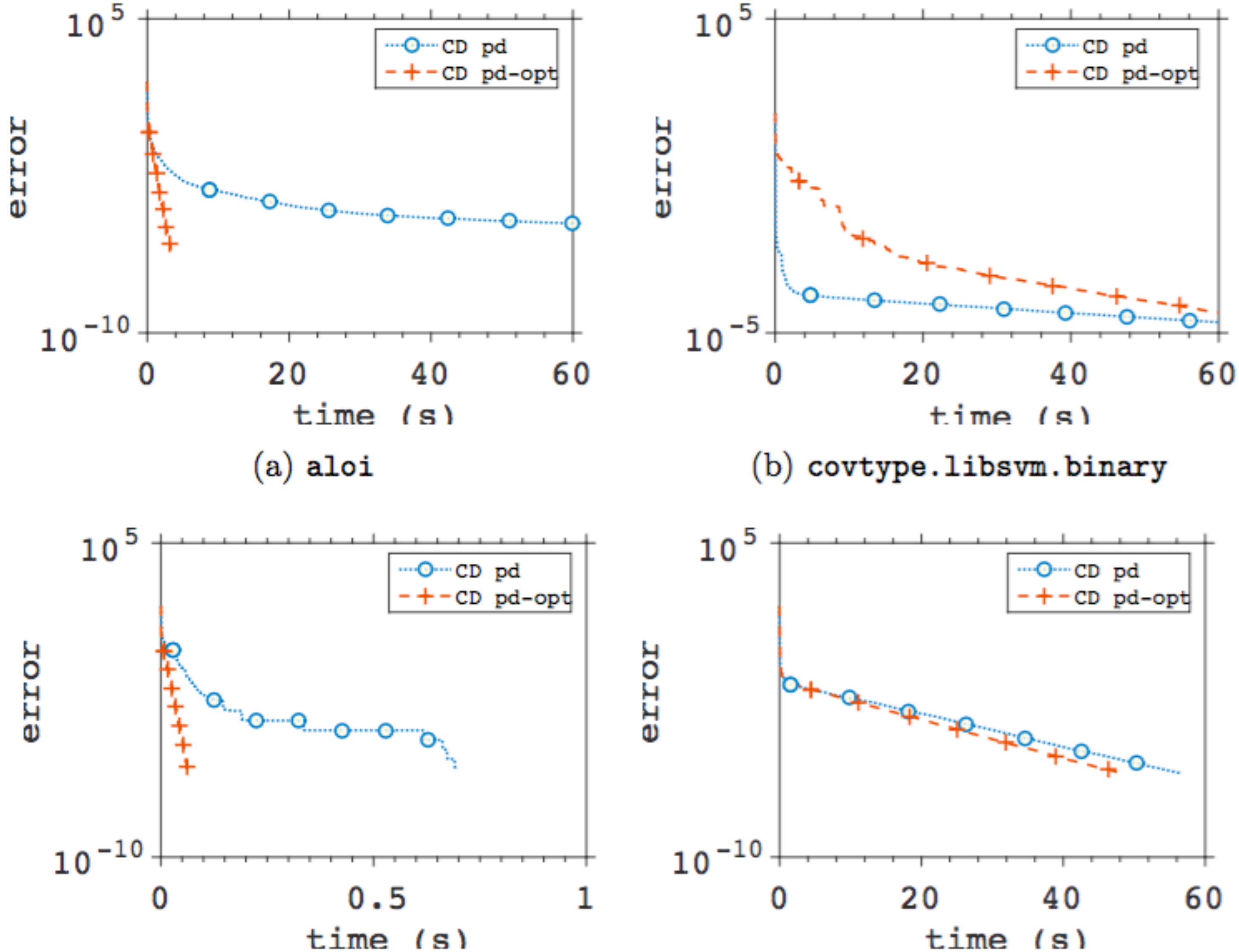


(a) liver-disorders-popt-k



(b) rand(500,100)

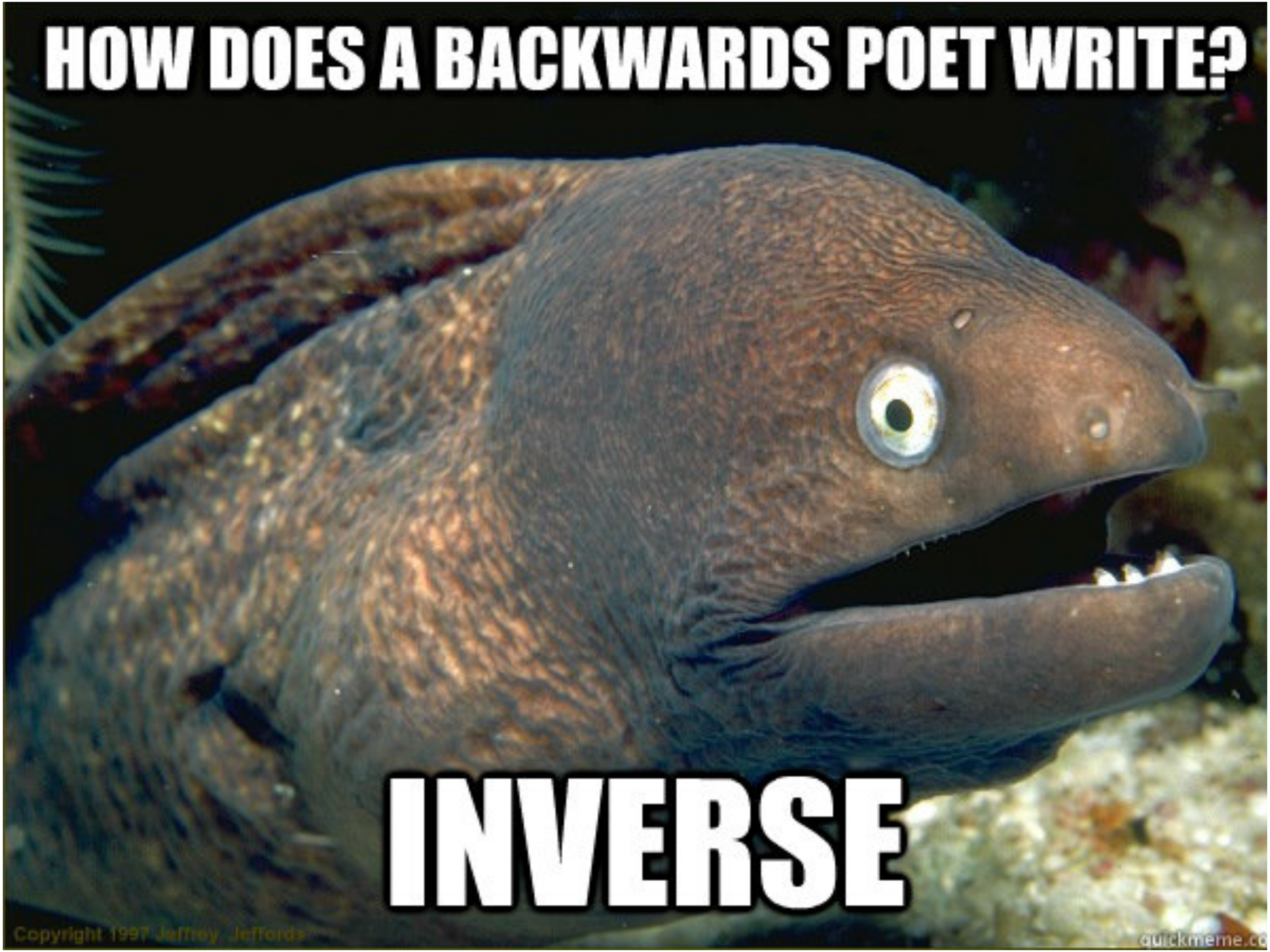
RCD: Convenient vs Optimal



(a) **aloi** (b) **covtype.libsvm.binary**

(c) **liver-disorders-ridge** (d) **mushrooms-ridge-opt**

EXTRA TOPIC: Randomized Matrix Inversion


HOW DOES A BACKWARDS POET WRITE?
INVERSE
Copyright 1997 Jeffrey Jeffords
quickmeme.co

Robert Mansel Gower (Edinburgh -> Paris)

Robert Mansel Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
arXiv:1602.01768, 2016

The Problem: Invert a Matrix

$$\overbrace{A}^{n}X = I, \quad A, X \in \mathbb{R}^{n \times n}$$

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

Identity matrix

Assumption 1 Matrix A is invertible

Inverting Symmetric Matrices

1. Sketch and Project

$$\|X\|_{F(B)} := \sqrt{\mathbf{Tr}(X^\top B X B)}$$

$$X^{t+1} = \arg \min_{X \in \mathbb{R}^{n \times n}} \|X - X^t\|^2_{F(B)}$$

$$\text{subject to} \quad S^\top A X = S^\top, \quad X = X^\top$$

- Quasi-Newton updates are of this form: *S* = deterministic column vector
- We get **randomized block** version of quasi-Newton updates!
- **Randomized quasi-Newton updates are linearly convergent matrix inversion methods**
- Interpretation**: Gaussian Inference** (Henning, 2015)

Donald Goldfarb. **A Family of Variable-Metric Methods Derived by Variational Means.** *Mathematics of Computation* 24(109), 1970

Gaussian Inference

Philipp Henning
Probabilistic Interpretation of Linear Solvers
SIAM Journal on Optimization 25(1):234-260, 2015

The new iterate X_{k+1} can be interpreted as

- the mean of a posterior distribution
- under a Gaussian prior with mean X_k and
- noiseless (and random) linear observation of A^{-1}

Randomized QN Updates

B	**Equation**	**Method**
I	$AX = I$	Powel-Symmetric-Broyden (PSB)
A^{-1}	$XA^{-1} = I$	Davidon-Fletcher-Powell (DFP)
A	$AX = I$	Broyden-Fletcher-Goldfarb-Shanno (BFGS)

- All these QN methods arise as **special cases of the framework**
- All are **linearly convergent,** with explicit convergence rates
- We also recover **non-symmetric updates** such as **Bad Broyden** and **Good Broyden**
- We get **block versions**
- We get randomized versions of **new QN updates**

2. Constrain and Approximate

$$X^{t+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \|X - A^{-1}\|^2_{F(B)}$$

$$\text{s.t.} \quad X = X^t + \Lambda S^\top A B^{-1} + B^{-1} A^\top S \Lambda^\top$$

$$\Lambda \in \mathbb{R}^{n\times\tau} \text{ is free}$$

New formulation even for standard QN methods

Randomized BFGS: $B = A,\ \tau = 1$

$$X^{t+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \|X - A^{-1}\|^2_{F(A)} = \boxed{\|AX - I\|^2_F}$$

$$\text{s.t.} \quad X = X^t + \boxed{\lambda S^\top + S\lambda^\top}$$

$$\lambda \in \mathbb{R}^n \text{ is free}$$

RBFGS performs "best" symmetric rank-2 update

4. Random Update

$$H = S(S^\top AB^{-1}A^\top S)^\dagger S^\top$$

$$X^{t+1} = X^t - (X^t A - I)HAB^{-1} + B^{-1}AH(AX^t - I)(AHAB^{-1} - I)$$

6. Random Fixed Point

$$X^{t+1} - A^{-1} = (I - B^{-1}A^\top HA)(X^t - A^{-1})(I - AHA^\top B^{-1})$$

Complexity / Convergence

Theorem [GR'16]

$$\|M\|_B := \|B^{1/2} M B^{1/2}\|_2$$

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

2. $$\mathbf{E}[H] \succ 0 \Rightarrow \rho < 1$$

$$\mathbf{E}\left[\|X^t - A^{-1}\|_{F(B)}^2\right] \leq \rho^t \|X^0 - A^{-1}\|_{F(B)}^2$$

Summary: Matrix Inversion

- Block version of QN updates
- New points of view (constrain and approximate, ...)
- New link between QN and approx. inverse preconditioning
- First time randomized QN updates are proposed
- First stochastic method for matrix inversion (with complexity bounds)?
- Linear convergence under weak assumptions
- Did not talk about:
  - Nonsymmetric variants
  - Theoretical bounds for discretely distributed S
  - Adaptive randomized BFGS
  - Limited memory and factored implementations
  - Experiments (Newton-Schultz; MinRes)
  - Use in empirical risk minimization [Gower, Goldfarb & R. 2016]
  - Extension: computation of the pseudoinverse [Gower & R. 2016]

Extensions

Ongoing work:
- Distributed, accelerated and adaptive variants
- Optimization with linear constraints, …

Matrix Inversion

Robert M. Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
arXiv:1602.01768, 2016

$$\text{Solve } AX = I$$

Machine Learning

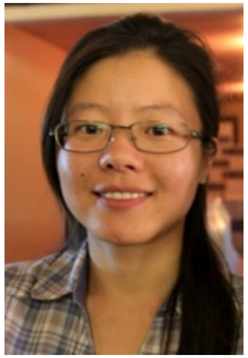
Robert M. Gower, Donald Goldfarb and P.R.
Stochastic Block BFGS: Squeezing More Curvature out of Data
ICML, 2016

Zheng Qu, P.R., Martin Takáč and Olivier Fercoq
Stochastic Dual Newton Ascent for Empirical Risk Minimization
ICML, 2016

The End

Martin Takáč
(Lehigh)

Jakub Mareček
(IBM)



Zheng Qu
(Hong Kong)

Olivier Fercoq
(Telecom ParisTech)

Rachael Tappenden
(Johns Hopkins)

Robert M Gower
(Edinburgh)

Virginia Smith
(Berkeley)

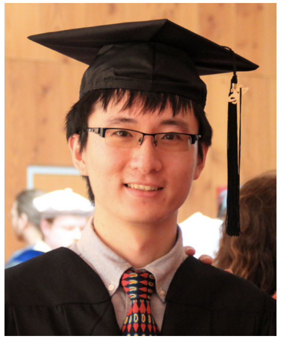
Jakub Konečný
(Edinburgh)

Jie Liu
(Lehigh)

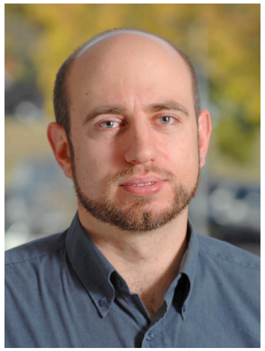
Michael Jordan
(Berkeley)

Dominik Csba
(Edinburgh)

Tong Zhang
(Rutgers & Baidu)



Zeyuan Allen-Zhu
(Princeton)



Nati Srebro
(TTI Chicago)

Donald Goldfarb
(Columbia)

Chenxin Ma
(Lehigh)

Martin Jaggi
(ETH Zurich)