SEGA: Variance Reduction via Gradient Sketching

Peter Richtárik

Joint work with Filip Hanzely (KAUST) and Konstantin Mishchenko (KAUST)

Numerical Algorithms in Nonsmooth Optimization
Erwin Schrödinger International Institute for Mathematics and Physics
Vienna, February 26, 2019

NIPS Proceedings$^{\beta}$ | Books | 2018

SEGA: Variance Reduction via Gradient Sketching

Part of: Advances in Neural Information Processing Systems 31 (NIPS 2018)

[PDF] [BibTeX] [Supplemental] [Reviews]

Authors

- Filip Hanzely
- Konstantin Mishchenko
- Peter Richtarik

Filip Hanzely

Konstantin
Mishchenko

Optimization & Machine Learning Group @ KAUST

SEGA: Variance Reduction via Gradient Sketching

Filip Hanzely[1] Konstantin Mishchenko [1] Peter Richtárik[1, 2, 3]

[1]KAUST [2]University of Edinburgh [3]Moscow Institute of Physics and Technology

Problem and Assumptions

Regularized Optimization

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + R(x) \tag{1}$$

- f: $\mathbf{M}$–smooth & μ–strongly convex convex:
$$f(x+h) \le f(x) + \langle \nabla f(x), h\rangle + \tfrac{1}{2}\langle \mathbf{M}h, h\rangle$$
$$f(x) + \langle \nabla f(x), h\rangle + \tfrac{\mu}{2}\|h\|^2 \le f(x+h)$$
(natural assumptions for ERM with linear predictors)
- R non-smooth, convex & proximable

New Oracle: Gradient Sketch

We do not have direct access to $\nabla f(x)$. Instead, we have access to a random linear transformation of the gradient:

$$\mathbf{S}^\top \nabla f(x) \in \mathbb{R}^b, \qquad \mathbf{S} \sim \mathcal{D} \tag{2}$$

- $\mathbf{S}$: random $n \times b$ matrix (b small)
- $\mathcal{D}$: distribution from which $\mathbf{S}$ is drawn

Goal

Design a proximal stochastic gradient-type method for solving (1) using the gradient sketch oracle (2).

Simple Algorithmic Idea

$$x^{k+1} = \mathrm{prox}_{\alpha R}(x^k - \alpha g^k), \tag{3}$$

α = stepsize; g^k = a "nice" estimator of $\nabla f(x^k)$.

How to design a good gradient estimator g^k?

Key Challenges:

- In the case when $\mathcal{D}$ is a distribution over standard basis vectors $e_1, \dots, e_n$ in $\mathbb{R}^n$, i.e., if we have access to random partial derivatives of f, then we can use
$$g^k = e_i^\top \nabla f(x^k) e_i,$$
and (3) reduces to proximal randomized coordinate descent (CD). However, CD does not work with non-separable regularizers R. So, we have an issue even in this simple case! How to resolve it?
- How to deal with gradient sketches coming from any distribution $\mathcal{D}$?

Resolution: The SEGA estimator. We will iteratively learn an unbiased variance-reduced estimator g^k of the gradient $\nabla f(x^k)$ by incorporating the latest information provided by the gradient sketch.

Constructing the SEGA Estimator

SEGA Estimator

1. Ask oracle for a gradient sketch at x^k: $\mathbf{S}_k^\top \nabla f(x^k)$
2. Define h^{k+1} as the closest (in some energy norm $\|h\|^2_{\mathbf{B}} \overset{\text{def}}{=} h^\top \mathbf{B} h$, where $\mathbf{B} \succ 0$) vector to h^k consistent with the gradient sketch:
$$h^{k+1} = \arg\min_{h\in\mathbb{R}^n} \|h - h^k\|^2_{\mathbf{B}} \quad \text{subject to} \quad \mathbf{S}_k^\top h = \mathbf{S}_k^\top \nabla f(x^k) \tag{4}$$
Closed-form solution of (4):
$$h^{k+1} = h^k + \mathbf{B}^{-1}\mathbf{Z}_k(\nabla f(x^k) - h^k); \qquad \mathbf{Z}_k \overset{\text{def}}{=} \mathbf{S}_k\left(\mathbf{S}_k^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top$$
3. Define the SEGA estimator:
$$g^k = h^k + \theta_k \mathbf{B}^{-1}\mathbf{Z}_k(\nabla f(x^k) - h^k) \tag{5}$$
(θ_k is a random variable ensuring that g^k is unbiased)

Key property: As $x_k \to x^*$, we get $g^k \to 0$, and hence SEGA estimator is variance-reduced.

Variants:

- `biasSEGA` estimator: use h^{k+1} instead of g^k
- `subspaceSEGA` estimator: If $f(x) = \phi(\mathbf{A}x)$ for some matrix $\mathbf{A} \in \mathbb{R}^{d\times n}$, we can improve the SEGA estimator by exploiting the fact that ∇f lies in Range($\mathbf{A}^\top$). We do this by adding the constraint $h \in$ Range($\mathbf{A}^\top$) to (4).

SEGA (SkEtched GrAdient descent)

`SEGA` = Method (3) + `SEGA` estimator (5)
`biasSEGA` = Method (3) + `biasSEGA` estimator (4)
`subspaceSEGA` = Method (3) + `subspaceSEGA` estimator

Convergence of SEGA

Let $\mathcal{D}$ be the uniform distribution over standard basis vectors $e_1, \dots, e_n \in \mathbb{R}^n$, and choose $\mathbf{B} = \mathbf{I}$. Then with stepsize $\alpha = \Omega(\frac{1}{n\lambda_{\max}(\mathbf{M})})$ and some constant $\sigma > 0$, we have

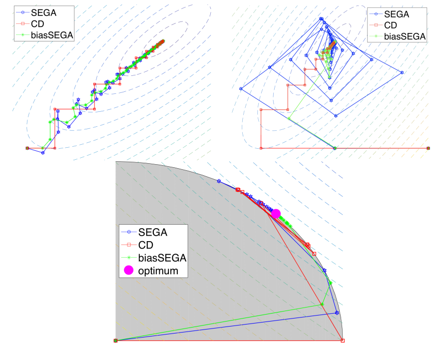
$$\mathbb{E}\left[\Phi^k\right] \le (1-\alpha\mu)^k \mathbb{E}\left[\Phi^0\right],$$

where $\Phi^k \overset{\text{def}}{=} \|x^k - x^*\|^2 + \sigma\alpha\|h^k - \nabla f(x^*)\|^2$, $x^* = \arg\min_x F(x)$.

- Note that $x^k \to x^*$ and $h^k \to \nabla f(x^*)$
- General convergence result for any $\mathbf{B} \succ \mathbf{0}$ and any $\mathcal{D}$ can be found in the paper [1].
- `subspaceSEGA`: If $\mathcal{D}$ samples from the columns of $\mathbf{A}^\top$, the rate can be $\Omega(\frac{n}{d})$ faster than standard `SEGA`.
- For coordinate sketches, we designed an accelerated `SEGA`, and established accelerated rate (read next).

Iterates of SEGA (in 2D)

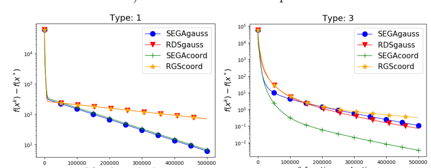
Iterates evolution of `SEGA`, `CD` and `biasSEGA` (updates made using h^{k+1} instead of g^k).



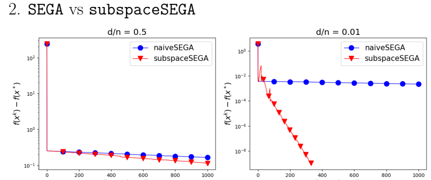
Bottom plot: R is the indicator function of the unit ball. While `CD` does not converge, `SEGA` does!

Experiments

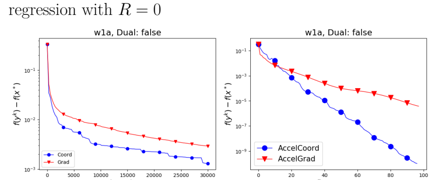
1. `SEGA` vs Random Direct Search (`RDS`) [2] (coordinate and Gaussian sketches) for derivative-free optimization



2. `SEGA` vs `subspaceSEGA`



3. `SEGA` vs Coordinate Descent (`CD`) [3] (left) and `ASEGA` vs Accelerated Coordinate Descent (`ACD`) [4, 5] (right) on ridge regression with $R = 0$



SEGA with Coordinate Sketches

Setup:

- $\mathbf{S}$ are column submatrices of the identity matrix
- Probability vector $p \in \mathbb{R}^n$: $p_i \overset{\text{def}}{=} \mathrm{Prob}(e_i \in \mathbf{S})$
- Probability matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$: $\mathbf{P}_{ij} \overset{\text{def}}{=} \mathrm{Prob}(e_i \in \mathbf{S}, e_j \in \mathbf{S})$
- ESO vector $v \in \mathbb{R}^n$ (for mini-batching) defined by:
$$\mathbf{P} \bullet \mathbf{M} \preceq \mathrm{Diag}(p \bullet v)$$

Acceleration: For coordinate sketches we also designed an accelerated variant of SEGA:

Algorithm Accelerated SEGA (ASEGA)

$x^0 = y^0 = z^0 \in \mathbb{R}^n$; $h^0 \in \mathbb{R}^n$; params $\alpha, \beta, \tau, \mu > 0$
for $k = 1, 2, \dots$ **do**
  $x^k = (1-\tau)y^{k-1} + \tau z^{k-1}$
  Sample $\mathbf{S}_k \sim \mathcal{D}$, and compute g^k and h^{k+1}
  $y^k = x^k - \alpha p^{-1} \bullet g^k$
  $z^k = \frac{1}{1+\beta\mu}(z^k + \beta\mu x^k - \beta g^k)$
end for

Rates: We prove the following iteration complexity bounds of `SEGA` and `ASEGA` with coordinate sketches:

Method	Complexity
`SEGA` importance sampling	$8.55 \cdot \frac{\mathrm{Tr}(\mathbf{M})}{\mu} \log\frac{1}{\epsilon}$
`SEGA` arbitrary sampling	$8.55 \cdot \left(\max_i \frac{v_i}{p_i\mu}\right) \log\frac{1}{\epsilon}$
`ASEGA` importance sampling	$9.8 \cdot \frac{\sum_i \sqrt{\mathbf{M}_{ii}}}{\sqrt{\mu}} \log\frac{1}{\epsilon}$
`ASEGA` arbitrary sampling	$9.8 \cdot \sqrt{\max_i \frac{v_i}{p_i^2\mu}} \log\frac{1}{\epsilon}$

Up to the constant factors 8.55 and 9.5, these rates are exactly the same as the rates of coordinate descent [3] and accelerated coordinate descent [4, 5]. So, we extend the reach of coordinate descent methods to problem (1) with a non-separable regularizer (e.g., arbitrary convex constraint)

References

[1] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: Variance reduction via gradient sketching. In *NeurIPS*, 2018.

[2] El Houcine Bergou, Peter Richtárik, and Eduard Gorbunov. Stochastic three point method for minimizing nonconvex, convex and strongly convex functions. *Manuscript*, 2018.

[3] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016.

[4] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *ICML*, 2016.

[5] Filip Hanzely and Peter Richtárik. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. *arXiv:1809.09354*, 2018.

Outline

1. Introduction
    a) The problem
    b) SEGA Oracle
    c) A “Gutless” Method
2. SEGA Estimator
    a) Sketch & Project
    b) Correcting for Bias
    c) Examples
3. SEGA Algorithm
    a) Variants
    b) Complexity
4. Experiments

1. Introduction

The Problem

Composite Minimization

Smoothness: $f(x+h) \le f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2}\langle \mathbf{L} h, h \rangle$

Strong convexity: $f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2}\langle \mu \mathbf{I} h, h \rangle \le f(x+h)$

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + R(x)$$

Dimension *n:*
very large

convex & closed
(and not necessarily separable)

New Stochastic First-Order Oracle

New Stochastic First Order Oracle

SkEtched GrAdient (SEGA) Oracle

Access to a random linear transformation (i.e., "sketch") of the gradient:

$$\mathbf{S}^\top \nabla f(x)$$

$$\mathbf{S} = [s_1, s_2, \dots, s_b] \in \mathbb{R}^{n \times b}$$

$$\mathbf{S} \sim \mathcal{D}$$

$$\mathbf{S}^\top \nabla f(x) = \begin{pmatrix} \langle \nabla f(x), s_1 \rangle \\ \langle \nabla f(x), s_2 \rangle \\ \vdots \\ \langle \nabla f(x), s_b \rangle \end{pmatrix} \in \mathbb{R}^b$$

Examples

1 Gaussian sketch $\quad \mathbf{S} = s \sim \mathcal{N}(0, \mathbf{\Omega})$

$$\mathbf{S}^\top \nabla f(x) = \langle \nabla f(x), s \rangle = \lim_{t \to 0} \frac{f(x + ts) - f(x)}{t}$$

2 Coordinate sketch $\quad \mathbf{S} = e_i \text{ with probability } p_i > 0$

$$\mathbf{S}^\top \nabla f(x) = \langle \nabla f(x), e_i \rangle = (\nabla f(x))_i$$

A “Gutless” Method

Proximal Stochastic Gradient Descent

$$\mathrm{prox}_{\alpha R}(z) \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^n} \left(\alpha R(x) + \frac{1}{2}\|x - z\|^2 \right)$$

stepsize

$$x^{k+1} = \mathrm{prox}_{\alpha R}(x^k - \alpha g^k)$$

"Good" estimator of the gradient

Key question:

How to construct a "good" estimator using the SkEtched GrAdient (SEGA) oracle?

2. SEGA: The Estimator

What Do We Want?

What is a “Good” Estimator?

1. Implementable given the information provided by the gradient sketch oracle

2. Unbiased

$$\mathbb{E}_{\mathbf{S}_k \sim \mathcal{D}}\left[g^k \mid x^k\right] = \nabla f(x^k)$$

3. Diminishing variance

$$\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \to 0$$

Sketch & Project

Sketch & Project

New estimator of the gradient

Previous estimator of the gradient

$$h^{k+1} = \arg\min_{h\in\mathbb{R}^n} \|h - h^k\|^2$$

$$\text{subject to } \mathbf{S}_k^\top h = \underbrace{\mathbf{S}_k^\top \nabla f(x^k)}_{}$$

Sketched Gradient

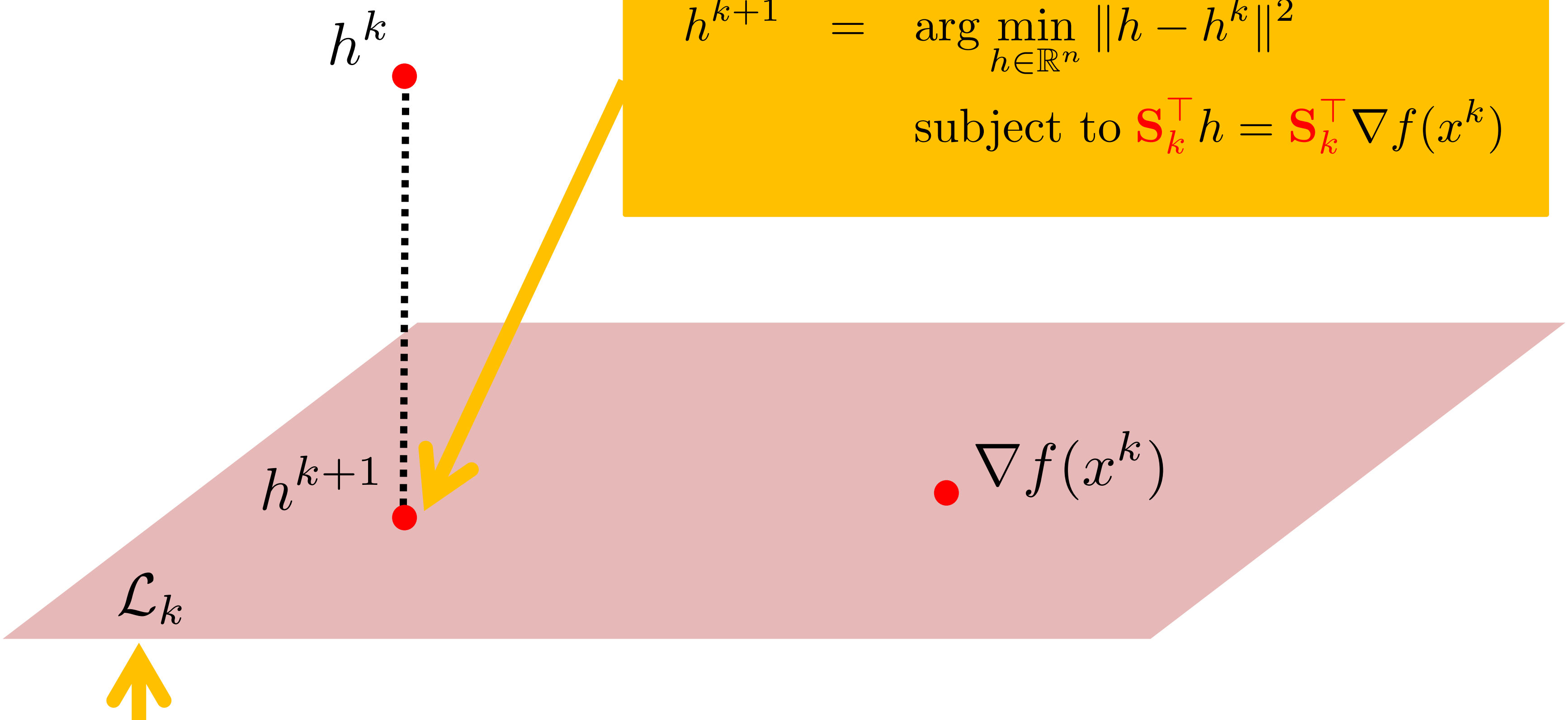
Closed-form solution:

$$h^{k+1} = h^k + \mathbf{Z}_k(\nabla f(x^k) - h^k)$$

$$\mathbf{Z}_k \overset{\text{def}}{=} \mathbf{S}_k \left(\mathbf{S}_k^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top$$

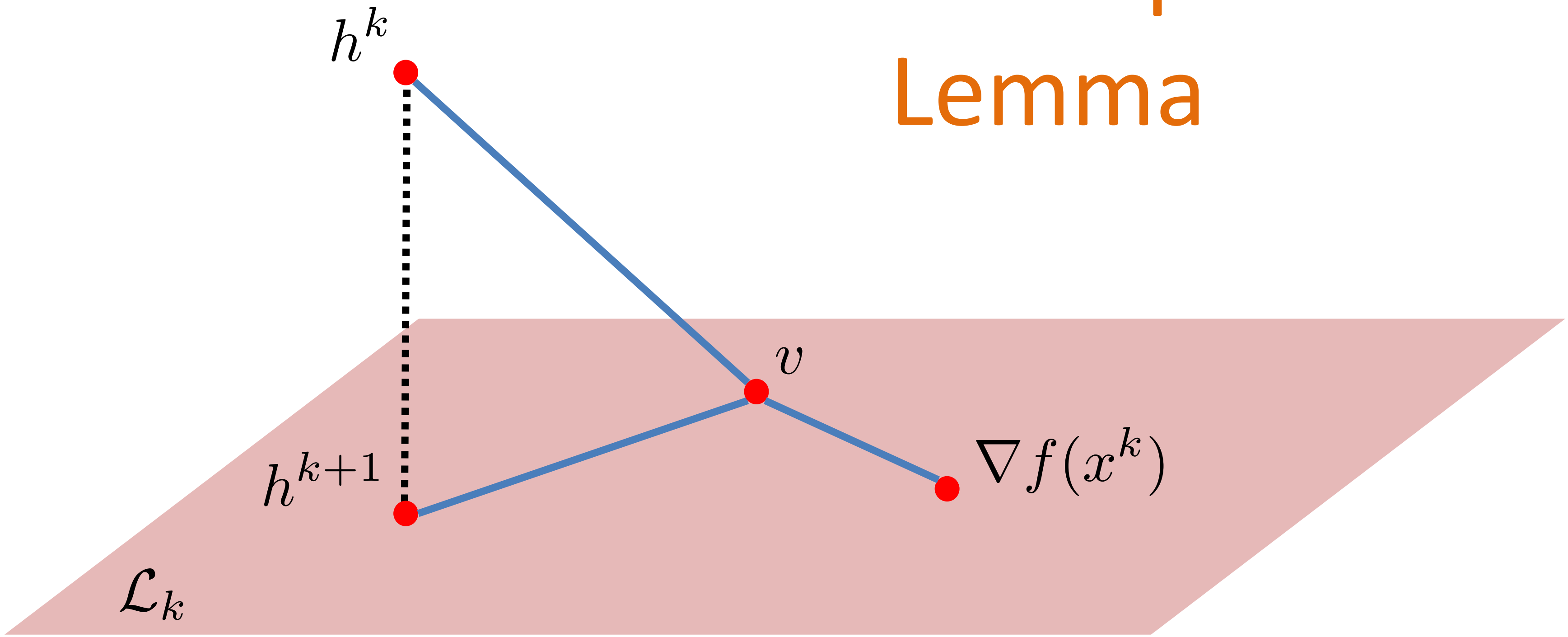
Sketch & Project: Visualization

$$h^{k+1} = \arg\min_{h\in\mathbb{R}^n} \|h - h^k\|^2 \quad \text{subject to } \mathbf{S}_k^\top h = \mathbf{S}_k^\top \nabla f(x^k)$$



$$\mathcal{L}_k \overset{\text{def}}{=} \left\{ h \in \mathbb{R}^n \mid \mathbf{S}_k^\top h = \mathbf{S}_k^\top \nabla f(x^k) \right\} = \nabla f(x^k) + \mathrm{Null}(\mathbf{S}_k^\top)$$

Error Decomposition Lemma



Lemma For any $v \in \mathbb{R}^n$

$$\mathbb{E}_{\mathcal{D}}\left[\|h^{k+1} - v\|^2_{\mathbf{I}}\right] = \|h^k - v\|^2_{\mathbf{I}-\mathbb{E}_{\mathcal{D}}[\mathbf{Z}]} + \|\nabla f(x^k) - v\|^2_{\mathbb{E}_{\mathcal{D}}[\mathbf{Z}]}$$

Sketch and Project I

Original sketch and project

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

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

Removal of full rank assumption + duality

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

Inverting matrices & connection to quasi-Newton updates

New understanding of Quasi-Newton Rules

Robert Mansel Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
SIAM J. on Matrix Analysis and Applications 38(4), 1380-1409, 2017

Computing the pseudoinverse

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

Application to machine learning

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

My course from last week

Sketch and project revisited: stochastic reformulations of linear systems

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

Sketch and Project II

Linear convergence of the stochastic heavy ball method

Nicolas Loizou and P.R.
Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods
arXiv:1712.09677, 2017

Stochastic projection methods for convex feasibility

Extension to Convex Feasibility

Ion Necoara, Andrei Patrascu and P.R.
Randomized Projection Methods for Convex Feasibility Problems: Conditioning and Convergence Rates
arXiv:1801.04873, 2018

Stochastic spectral & conjugate descent

Dmitry Kovalev, Eduard Gorbunov, Elnur Gasanov and P.R.
Stochastic Spectral and Conjugate Descent Methods
NeurIPS 2018

Accelerated stochastic matrix inversion

Robert Mansel Gower, Filip Hanzely, P.R. and Sebastian Stich
Accelerated Stochastic Matrix Inversion: General Theory and Speeding up BFGS Rules for Faster Second-Order Optimization
NeurIPS 2018

Acceleration

SAGD: a "strange" special case of JacSketch

Adel Bibi, Alibek Sailanbayev, Bernard Ghanem, Robert Mansel Gower and P.R.
Improving SAGA via a Probabilistic Interpolation with Gradient Descent
arXiv:1806.05633, 2018

Unbiasedness: SEGA for Coordinate Sketches

$n = 2$

2D Example

$$\mathbf{S} = \begin{cases} e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{with probability} \quad p_1 \in (0,1) \\ e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{with probability} \quad p_2 = 1 - p_1 \end{cases}$$

$$\mathbf{S}_k = e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \Longrightarrow \quad \mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_1 = (\nabla f(x^k))_1\}$$

$$\mathbf{S}_k = e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_2 = (\nabla f(x^k))_2\}$$

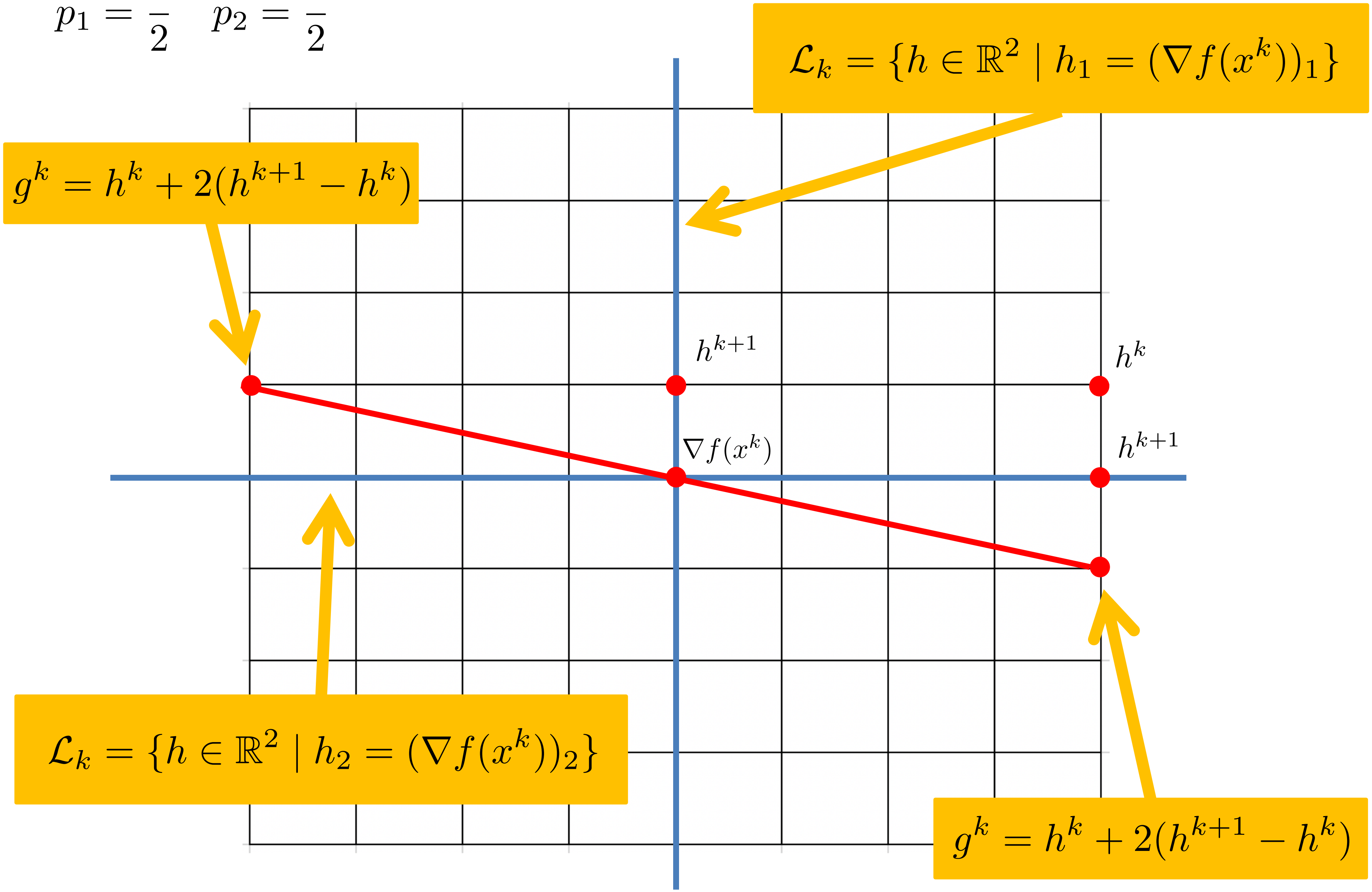
Case 1

$$p_1 = \frac{1}{2} \qquad p_2 = \frac{1}{2}$$

SEGA Estimator

$n = 2$

$p_1 = \frac{1}{2} \quad p_2 = \frac{1}{2}$



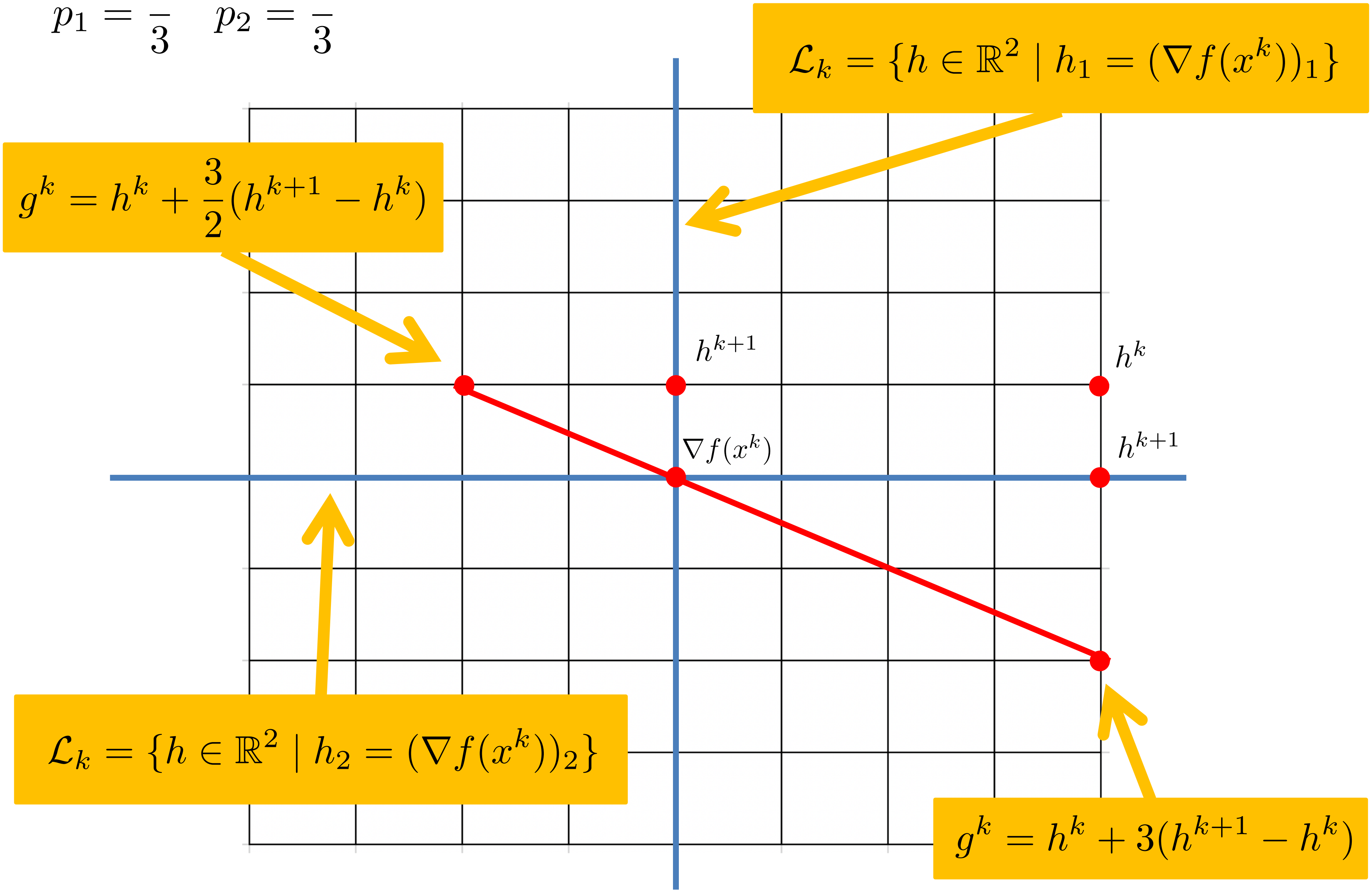
Case 2

$$p_1 = \frac{2}{3} \qquad p_2 = \frac{1}{3}$$

SEGA Estimator

$n = 2$

$p_1 = \frac{2}{3} \quad p_2 = \frac{1}{3}$



SEGA for General Sketches

SEGA Estimator

SEGA estimator

Bias correcting random variable

$$\mathbb{E}\left[\theta_k \mathbf{Z}_k\right] = \mathbf{I}$$

$$\begin{aligned} g^k &\stackrel{\text{def}}{=} h^k + \theta_k (h^{k+1} - h^k) \\ &= h^k + \theta_k \mathbf{Z}_k (\nabla f(x^k) - h^k) \end{aligned}$$

$$\mathbb{E}_{\mathcal{D}}\left[g^k\right] = \nabla f(x^k)$$

3. SEGA: The Algorithm

The Algorithm

The SEGA Algorithm

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + R(x)$$

0 Choose $x^0, h^0 \in \mathrm{dom} F$

For $k \geq 0$ **REPEAT**

1 Ask **SEGA Oracle** for $\mathbf{S}_k^\top \nabla f(x^k)$

Sketched Gradient

Perform **Sketch & Project**

$$h^{k+1} = \arg\min_{h \in \mathbb{R}^n} \|h - h^k\|^2$$

$$\text{subject to } \mathbf{S}_k^\top h = \mathbf{S}_k^\top \nabla f(x^k)$$

2 Compute the **SEGA Estimator**

$$g^k = h^k + \theta_k (h^{k+1} - h^k)$$

3 Perform **Proximal SGD** step

$$x^{k+1} = \mathrm{prox}_{\alpha R}(x^k - \alpha g^k)$$

Variants of SEGA

$$x^{k+1} = \mathrm{prox}_{\alpha R}(x^k - \alpha g^k)$$

$$\mathbb{E}_{\mathcal{D}}\left[\theta_k \mathbf{Z}_k\right] = \mathbf{I}$$

1. SEGA $\quad g^k = h^k + \theta_k (h^{k+1} - h^k)$

2. Biased SEGA $\quad$ Use $\theta_k \equiv 1 \Rightarrow g^k = h^{k+1}$

3. Subspace SEGA

$$f(x) = \phi(Ax) \Rightarrow \nabla f(x) \in \mathrm{Range}(A^\top)$$

$$\begin{aligned} h^{k+1} \quad = \quad & \arg\min_{h \in \mathbb{R}^n} \|h - h^k\|^2 \\ & \text{subject to } \mathbf{S}_k^\top h = \mathbf{S}_k^\top \nabla f(x^k) \\ & \qquad h \in \mathrm{Range}(A^\top) \end{aligned}$$

4. Accelerated SEGA

Complexity: General Sketch

Complexity for General Sketches

Strong convexity:

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

Theorem

$$\mathbb{E}\left[\Phi^k\right] \leq (1 - \alpha\mu)^k \Phi^0$$

Lyapunov function: $x^0, h^0 \in \mathrm{dom} F$

$$\Phi^k \overset{\text{def}}{=} \|x^k - x^*\|^2 + \sigma\alpha\|h^k - \nabla f(x^*)\|^2$$

Stepsize can't be too large:

$$\alpha(2(\mathbf{C} - \mathbf{I}) + \sigma\mu\mathbf{I}) \leq \sigma\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{Z}\right]$$

$$2\alpha\mathbf{C} + \sigma\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{Z}\right] \leq \mathbf{L}^{-1}$$

$$\mathbf{C} \overset{\text{def}}{=} \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\theta^2\mathbf{Z}\right]$$

Complexity: Coordinate Sketch

Coordinate Sketch: Arbitrary Sampling Setup

Random subset of {1, … , *n*}

- $\mathbf{S} = \mathbf{I}_{:C}$ (random column submatrix of the identity matrix)
- Probability vector $p \in \mathbb{R}^n$: $p_i \stackrel{\text{def}}{=} \text{Prob}(i \in C)$
- Probability matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$: $\mathbf{P}_{ij} \stackrel{\text{def}}{=} \text{Prob}(i \in C \;\&\; j \in C)$
- ESO vector $v \in \mathbb{R}^n$ (for mini-batching) defined by:

$$\mathbf{P} \bullet \mathbf{M} \preceq \text{Diag}(p \bullet v)$$

Hadamard product

Complexity Results

$$f(x+h) \leq f(x) + \langle \nabla f(x), h\rangle + \tfrac{1}{2}\langle \mathbf{L}h, h\rangle$$

$$f(x) + \langle \nabla f(x), h\rangle + \tfrac{1}{2}\langle \mu \mathbf{I}h, h\rangle \leq f(x+h)$$

$$R \equiv 0$$

Method	Complexity
SEGA importance sampling	$8.55 \cdot \frac{\mathrm{Tr}(\mathbf{L})}{\mu} \log \frac{1}{\epsilon}$
SEGA arbitrary sampling	$8.55 \cdot \left(\max_i \frac{v_i}{p_i \mu}\right) \log \frac{1}{\epsilon}$
ASEGA importance sampling	$9.8 \cdot \frac{\sum_i \sqrt{\mathbf{L}_{ii}}}{\sqrt{\mu}} \log \frac{1}{\epsilon}$
ASEGA arbitrary sampling	$9.8 \cdot \sqrt{\max_i \frac{v_i}{p_i^2 \mu}} \log \frac{1}{\epsilon}$

Up to the constant factors 8.55 and 9.5, these rates are exactly the same as the rates of CD [R. & Takáč '16] and accelerated CD [Allen-Zhu et al '16, Hanzely & R. '19].

Coordinate Descent

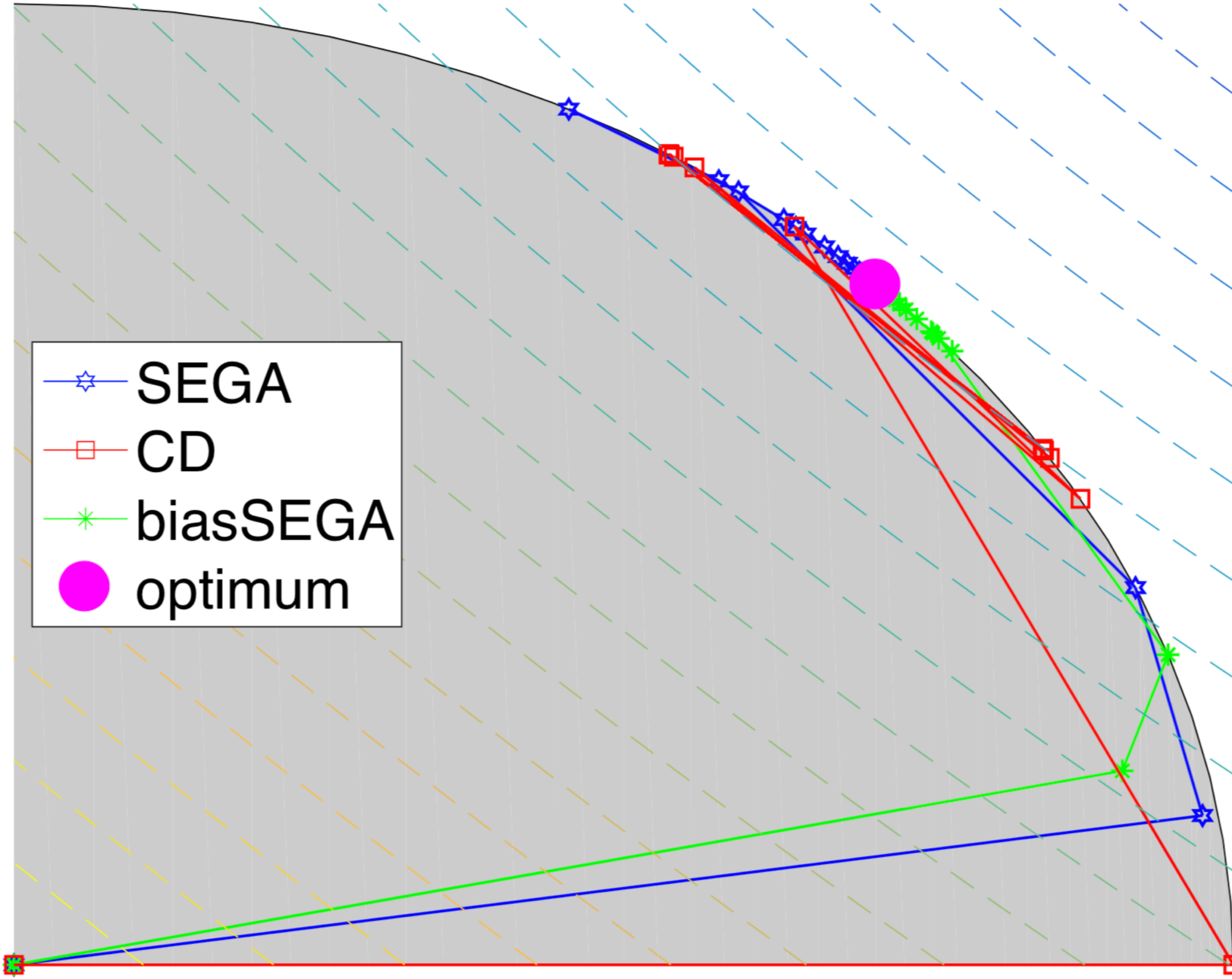
P.R. and Martin Takáč
On optimal probabilities in stochastic coordinate descent methods
Optimization Letters 10(6), 1223-1243, 2016

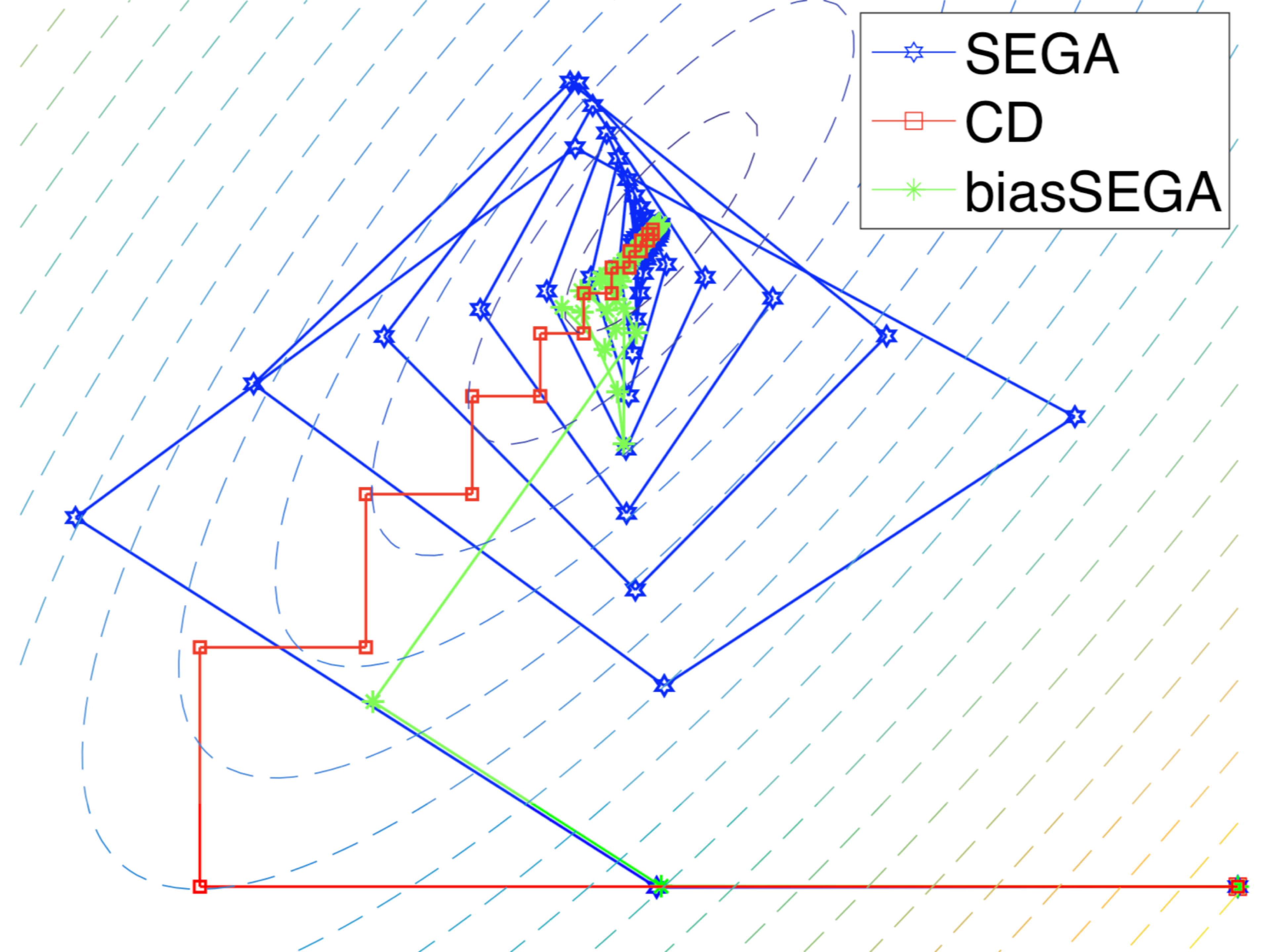
Zeyuan Allen-Zhu, Zheng Qu, P.R. and Yang Yuan
Even faster accelerated coordinate descent using non-uniform sampling
ICML 2016

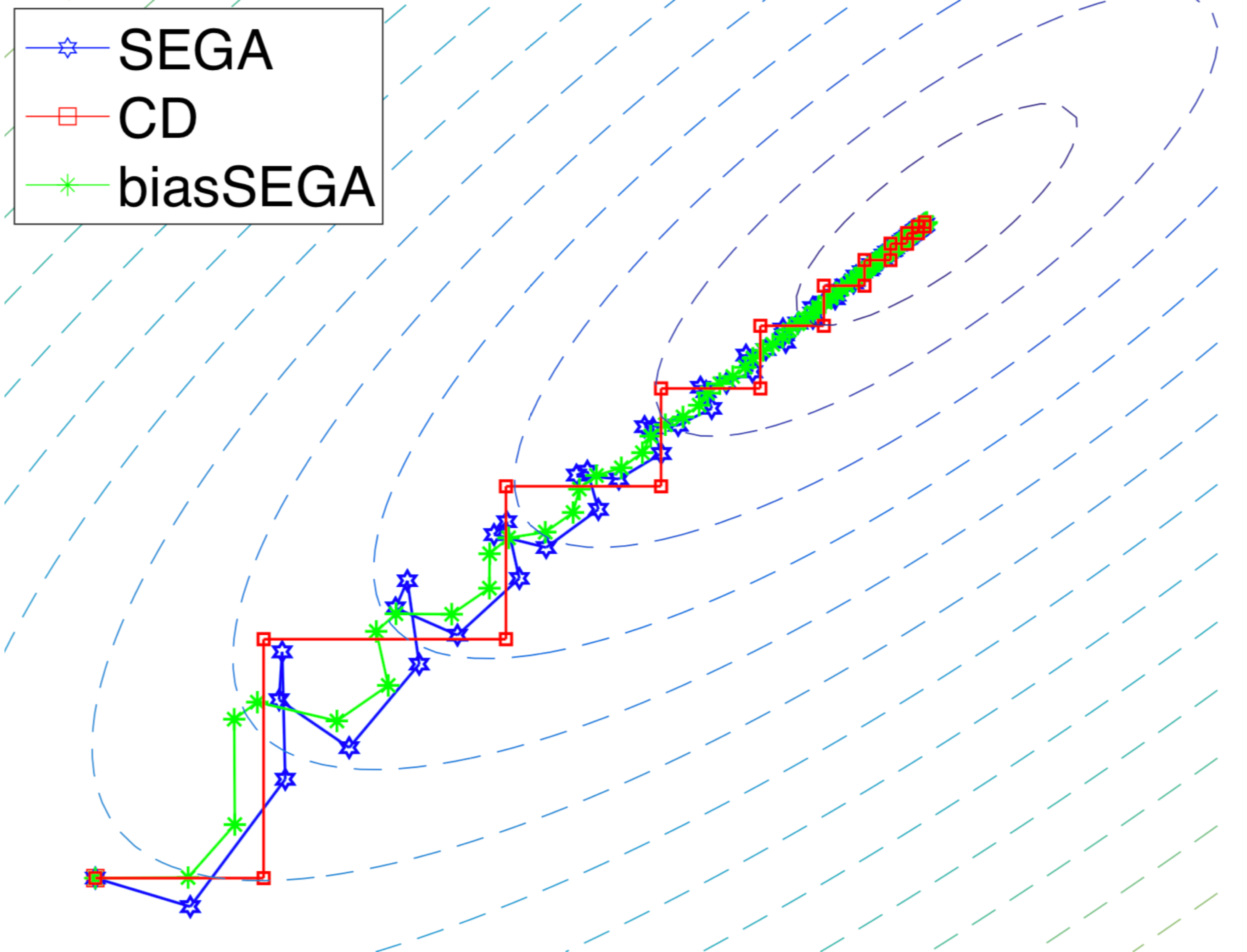
Filip Hanzely and P.R.
Accelerated coordinate descent with arbitrary sampling and best rates for minibatches
AISTATS 2019

4. Experiments

Illustration in 2D


SEGA
CD
biasSEGA
optimum

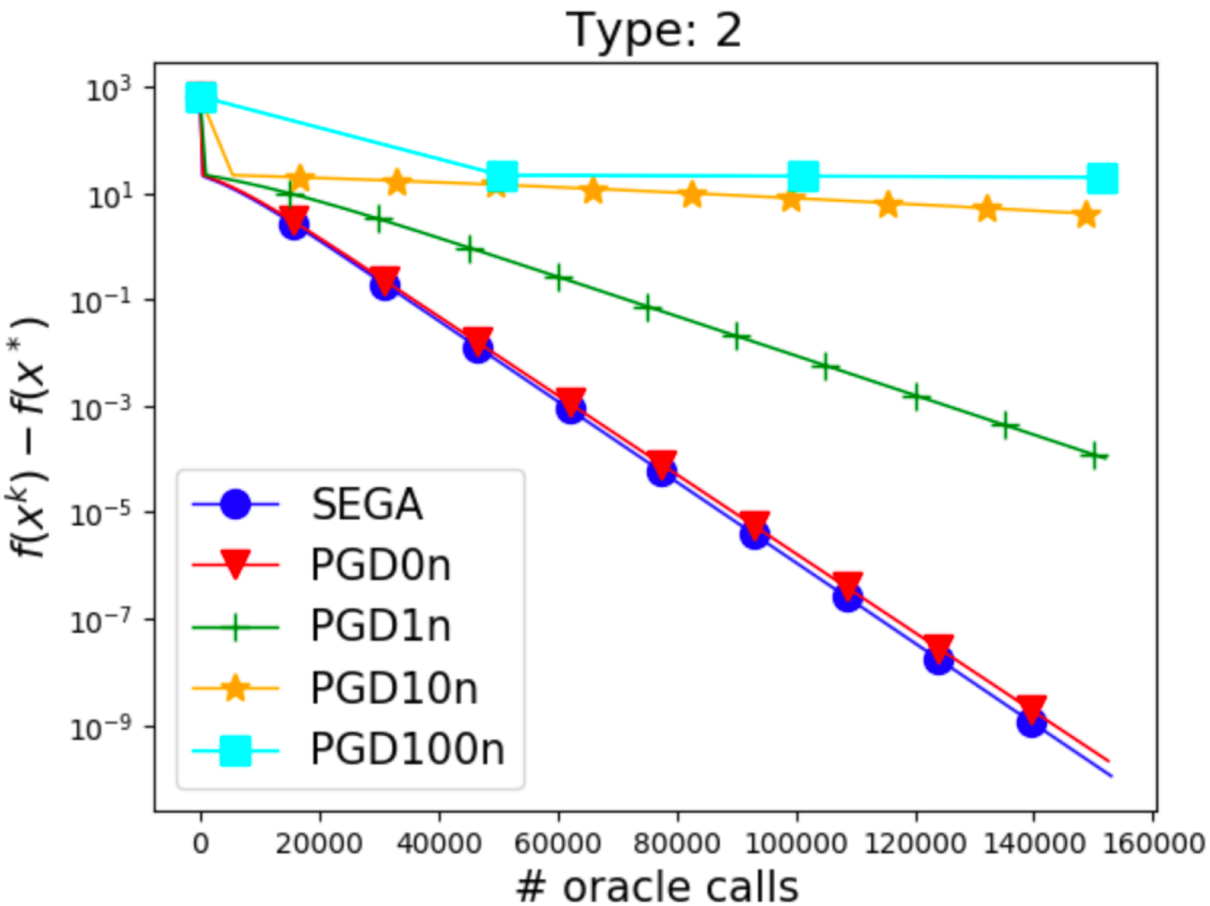

SEGA
CD
biasSEGA


SEGA
CD
biasSEGA

SEGA vs Projected Gradient Descent

Gaussian Sketch, Ball Constraint

$\mathbf{S} = \text{Gaussian vector} \qquad R(x) = 1_{\mathcal{B}(0,1)}(x)$

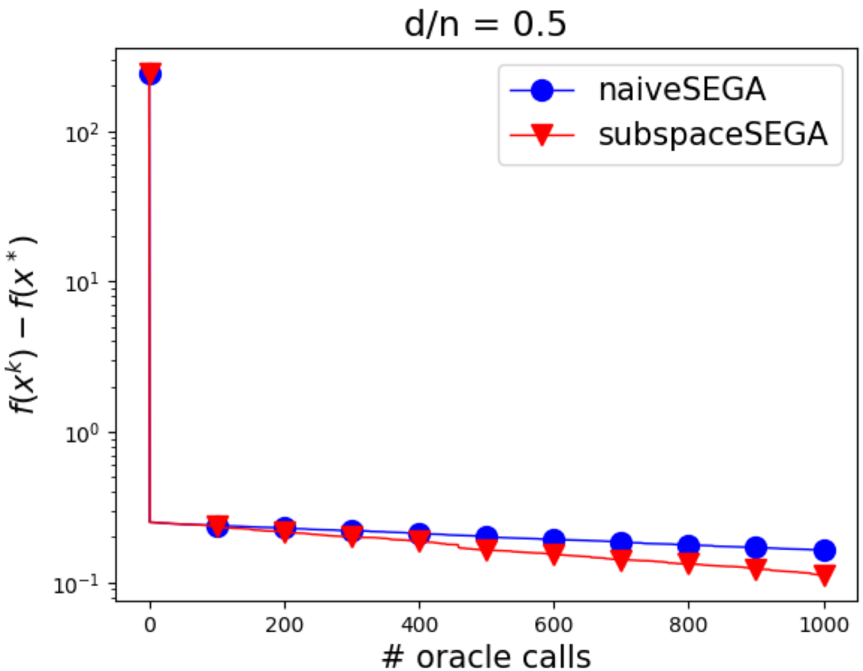


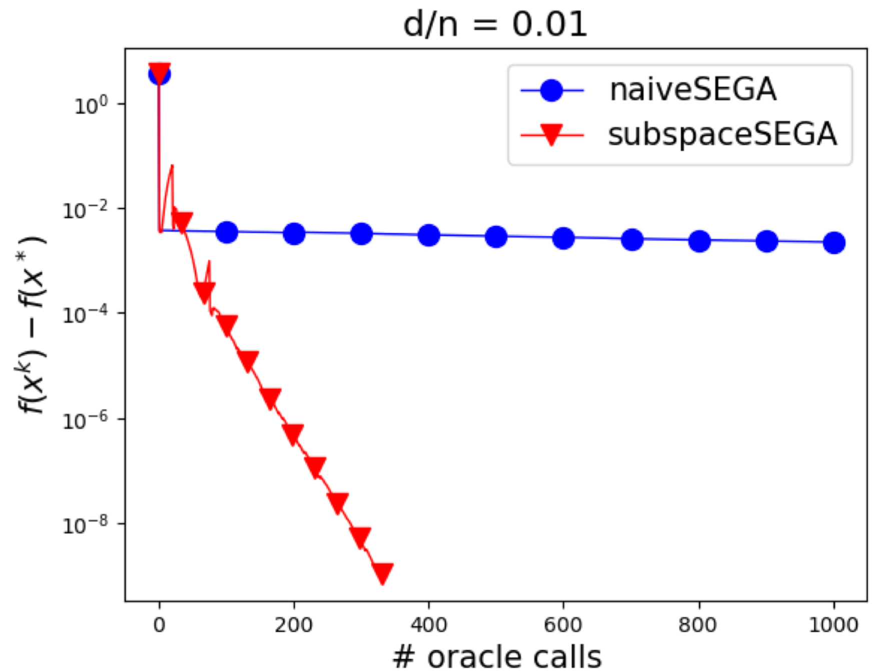
n = 500

SEGA vs Subspace SEGA

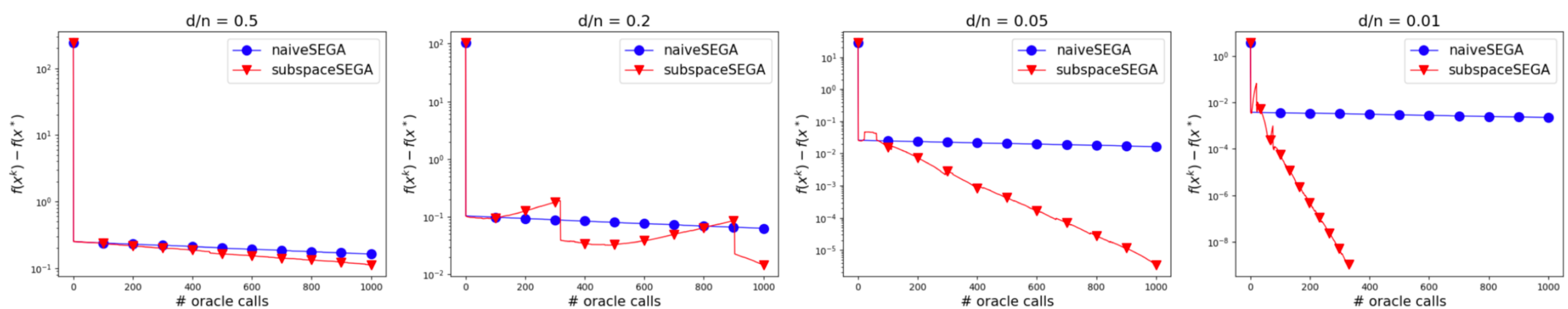
SEGA vs Subspace SEGA

$$f(x) = \phi(Ax) \Rightarrow \nabla f(x) \in \mathrm{Range}(A^\top)$$

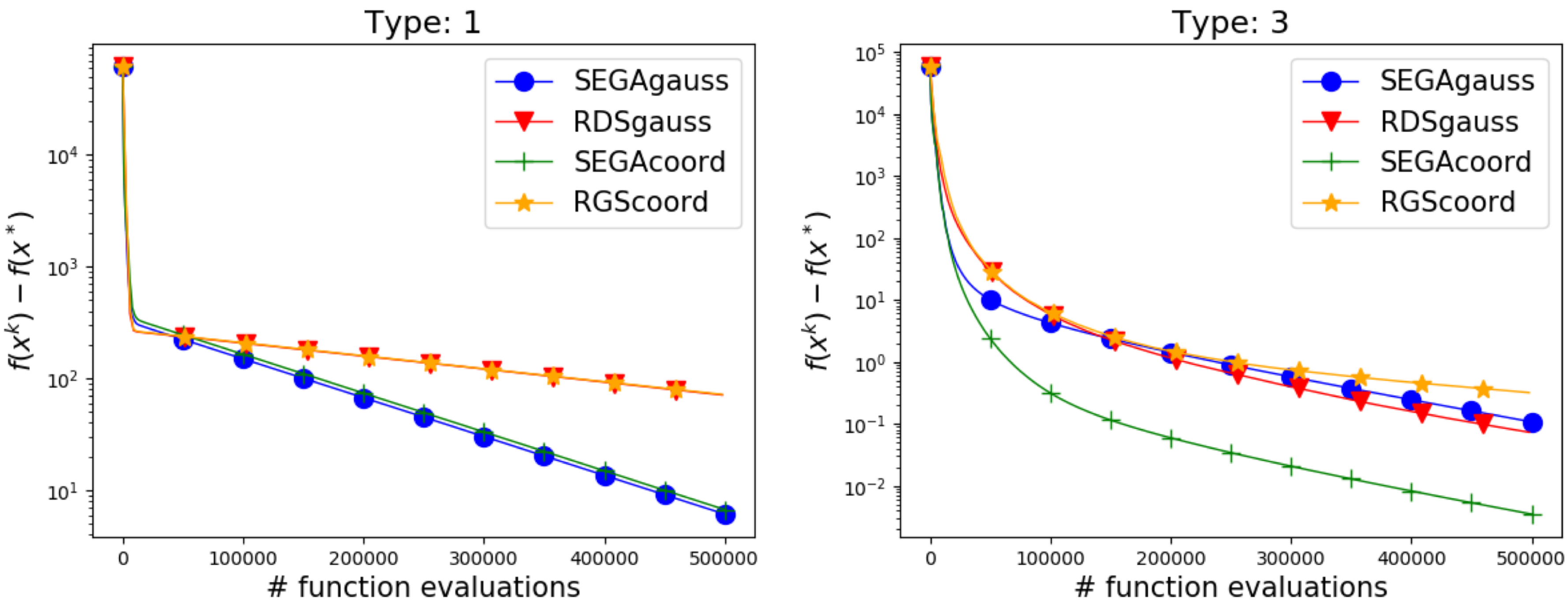




n = 1,000



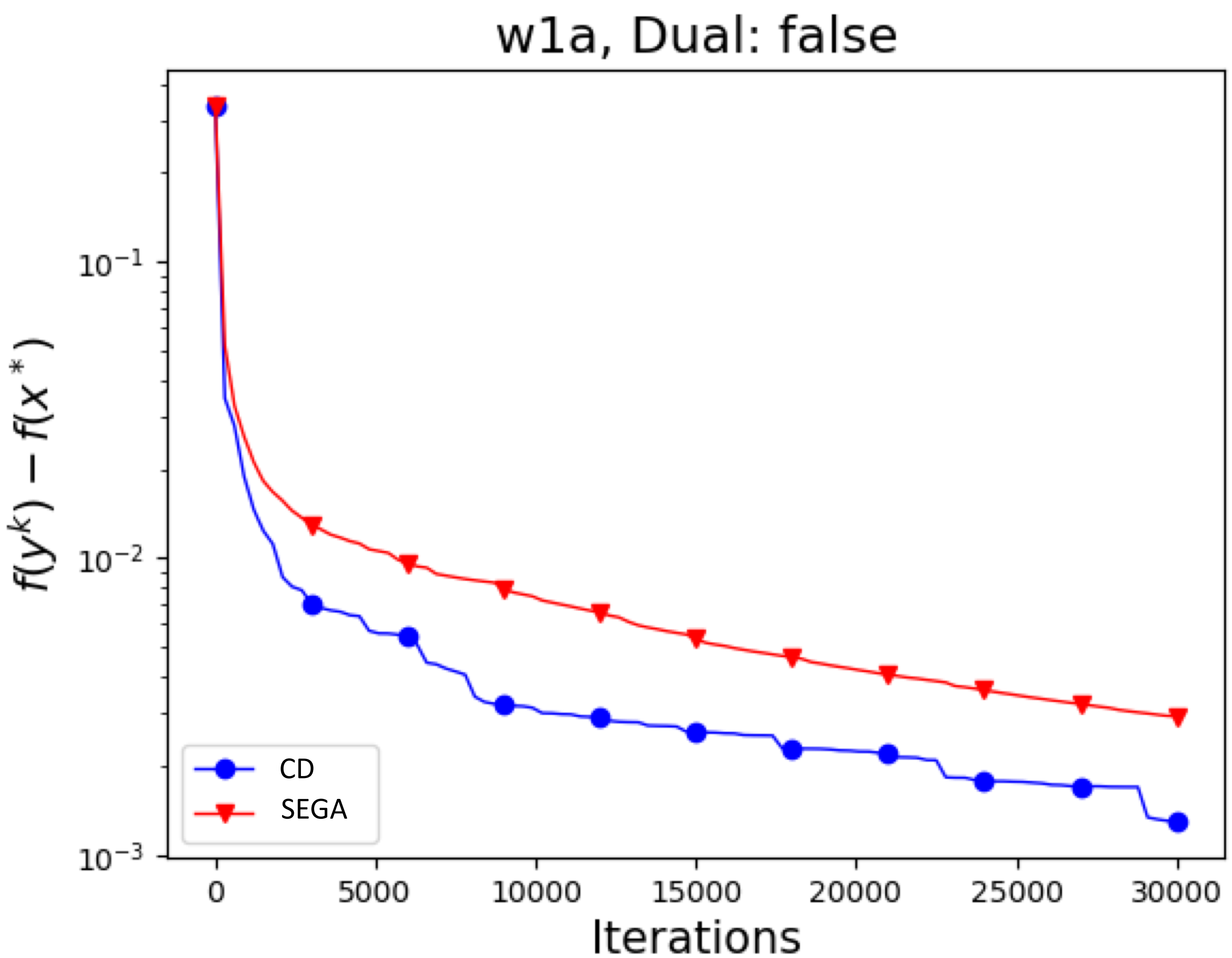
SEGA vs Random Direct Search



El Houcine Bergou, Eduard Gorbunov and P.R.
Stochastic three points method for unconstrained smooth minimization
arXiv:1902.03591, 2019

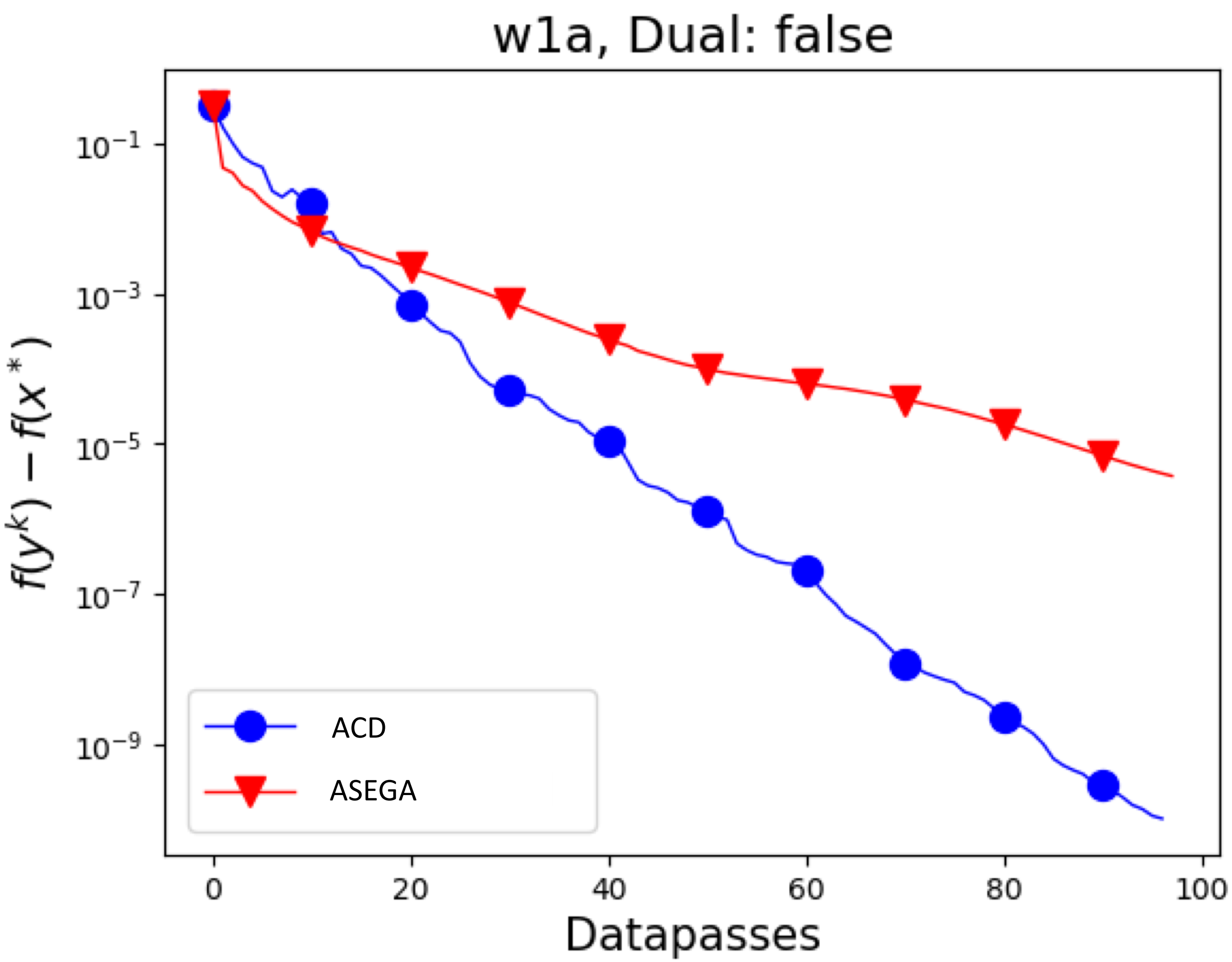
SEGA vs Coordinate Descent

SEGA vs CD



w1a, Dual: false
$f(y^k) - f(x^*)$
10^{-1}
10^{-2}
10^{-3}
0
5000
10000
15000
20000
25000
30000
Iterations
CD
SEGA

Accelerated SEGA vs Accelerated CD



5. Summary

Summary

- New Stochastic First-Order Oracle: SkEtched GrAdient (SEGA)
- New Stochastic Proximal SGD method. Comes in several variants:
  - SEGA (based on the SEGA Estimator)
  - Biased SEGA
  - Subspace SEGA
  - Accelerated SEGA
- Coordinate sketches:
  - Same complexity as state-of-the art CD methods
  - Can handle non-separable regularizer R

The End