

Variance Reduction for Gradient Compression

Peter Richtárik



King Abdullah University
of Science and Technology

MIPT

September 27, 2019

Abstract

Over the past few years, various **randomized gradient compression techniques** (e.g., quantization, sparsification, sketching) have been proposed for **reducing communication in distributed training of very large machine learning models**.

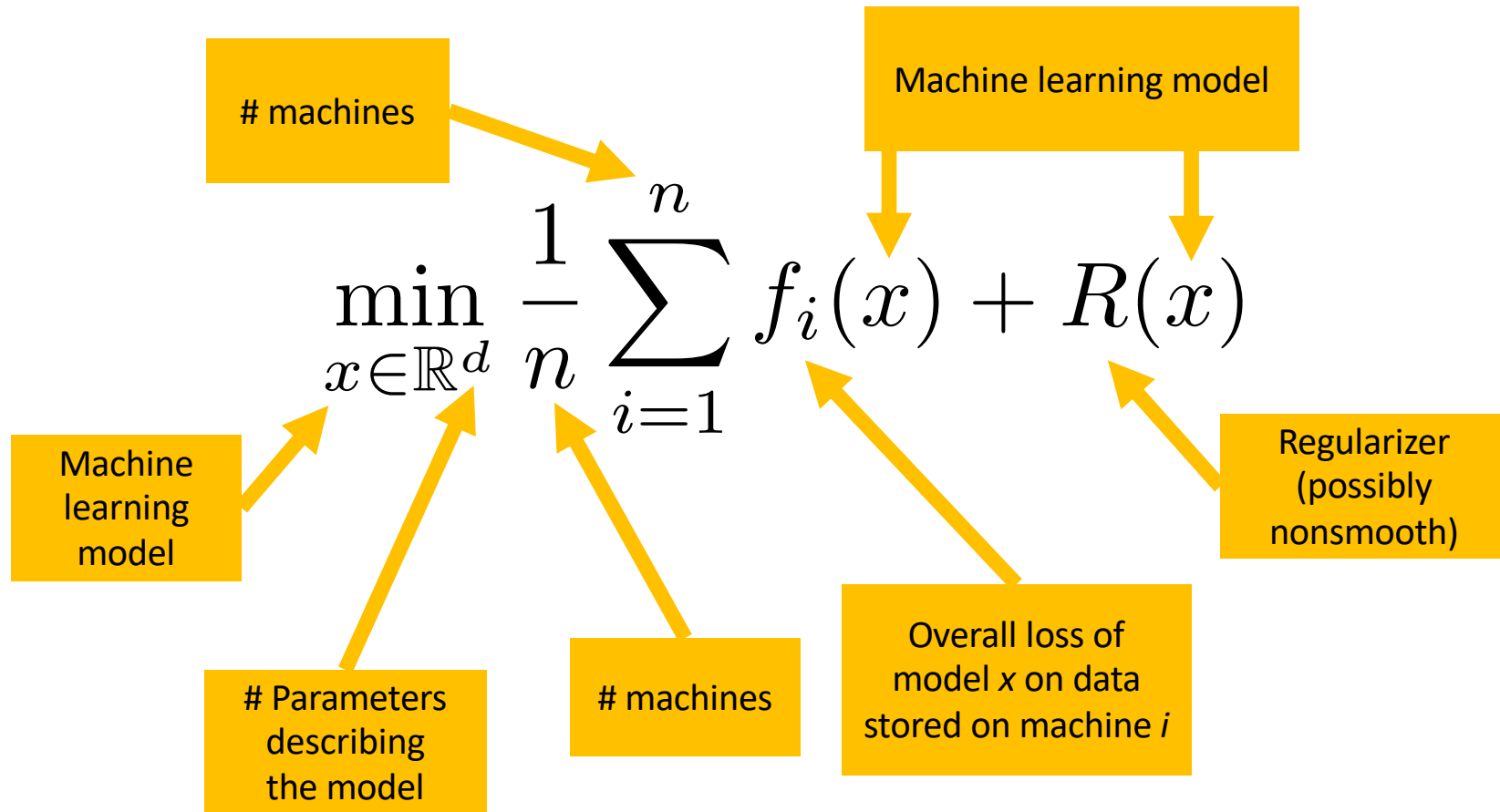
However, despite high level of research activity in this area, **surprisingly little is known about how such compression techniques should properly interact with first order optimization algorithms**. For instance, randomized compression increases the variance of the stochastic gradient estimator, and this has an adverse effect on convergence speed. While a number of variance-reduction techniques exists for taming the variance of stochastic gradients arising from sub-sampling in finite-sum optimization problems, no variance reduction techniques exist for taming the variance introduced by gradient compression. Further, gradient compression techniques are invariably applied to unconstrained problems, and it is not known whether and how they could be applied to solve constrained or proximal problems.

In this talk I will give positive resolutions to both of these problems. In particular, **I will show how one can design fast variance-reduced proximal stochastic gradient descent methods in settings where stochasticity comes from gradient compression**.

1. Motivation

The Problem

The Problem



Distributed Gradient Descent

Distributed Gradient Descent

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



$$x^{k+1} = \text{prox}_{\gamma R} \left(x^k - \underbrace{\gamma \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k)} \right)$$

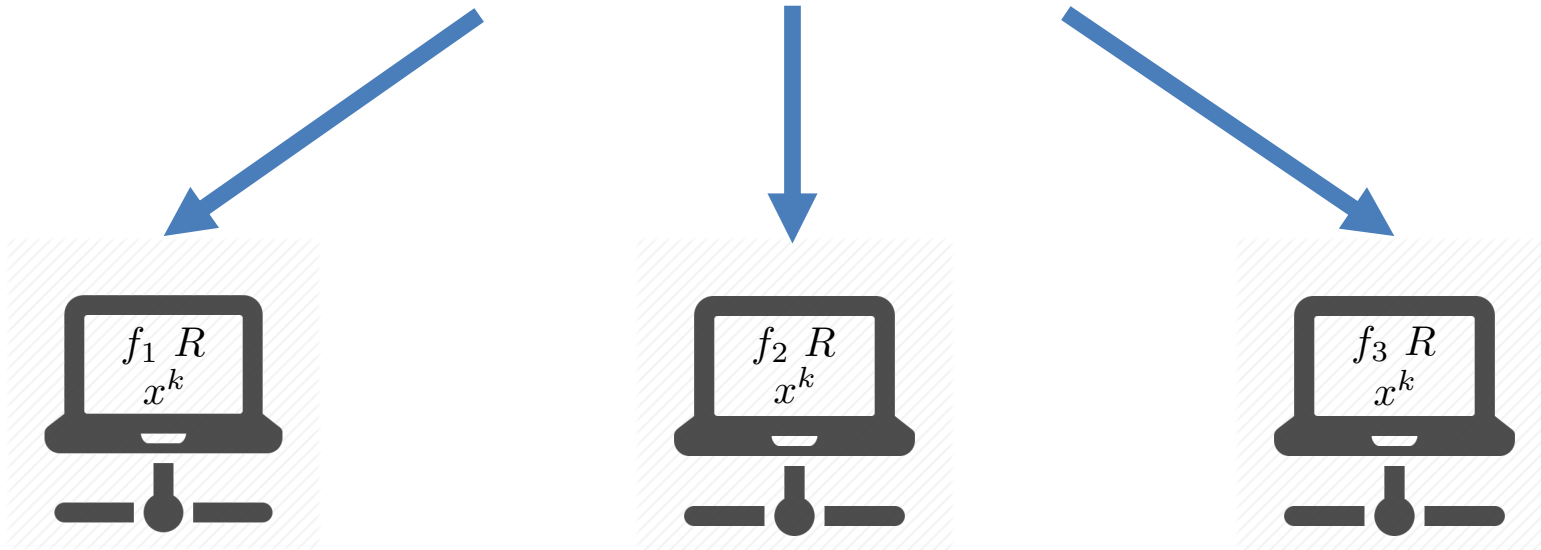

$$\text{prox}_{\gamma R}(x) \stackrel{\text{def}}{=} \arg \min_{y \in \mathbb{R}^d} \left(\gamma R(y) + \frac{1}{2} \|y - x\|^2 \right)$$

Gradient is computed in a distributed fashion

Distributing the Data

Example: $n = 3$

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$



Distributed Gradient Descent

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

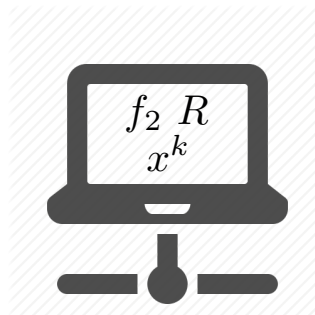
Parameter server



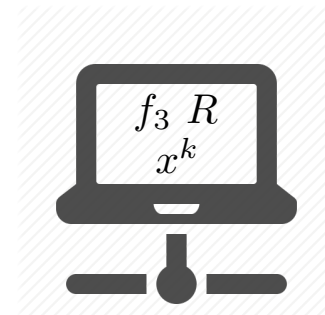
$$g^k = \frac{\quad + \quad +}{3}$$



$$\nabla f_1(x^k)$$



$$\nabla f_2(x^k)$$



$$\nabla f_3(x^k)$$

Distributed Gradient Descent

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

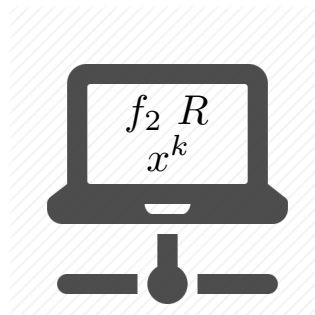
Parameter server



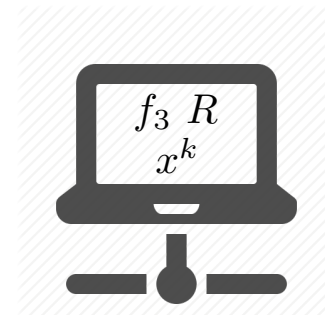
g^k



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



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

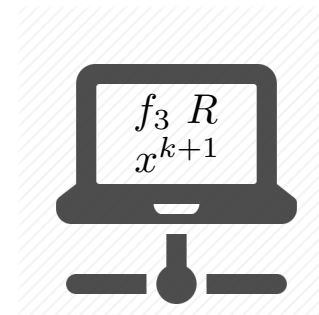
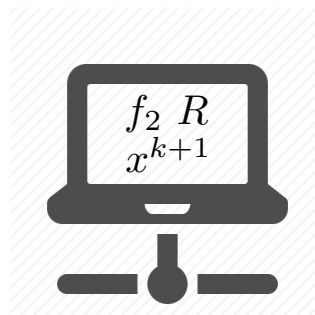


$$x^{k+1} = \text{prox}_{\gamma R} (x^k - \gamma g^k)$$

Distributed Gradient Descent

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

Parameter server



Gradient Compression

Key Issue: Communication is Expensive

- Strategies to deal with this:
 - Do **more computation** on each node before communication
 - CoCoA, CoCoA+ [Ma et al 2017]
 - Local GD, Local SGD, Federated Averaging [Khaled 2019a, Khaled 2019b]
 - Do **less communication by compressing communicated messages (e.g., gradients)**

Ma, Konečný, Jaggi, Smith, Jordan, R and Takáč
Distributed optimization with arbitrary local solvers
Optimization Methods and Software 32(4):813-848, 2017
OMS Most Read Paper, 2017

Compression Operators

Random operator
(e.g., sparsification, dithering,
quantization, sketching)

$$\mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

“Good” Properties:

1 $\mathcal{C}$ is “easy to communicate”

2 $\mathbb{E}_{\mathcal{C}} [\mathcal{C}(x)] = x$ ← Unbiasedness

3 $\mathbb{E}_{\mathcal{C}} [\|\mathcal{C}(x) - x\|^2] \leq \omega \|x\|^2$ ← Bounded variance

GD with Compressed Gradients

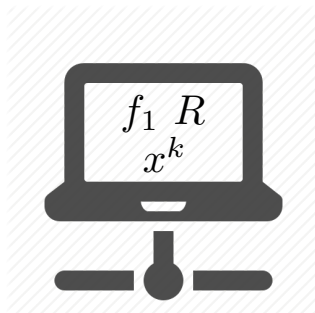
$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

[Alistarh et al, 2017]

Parameter server

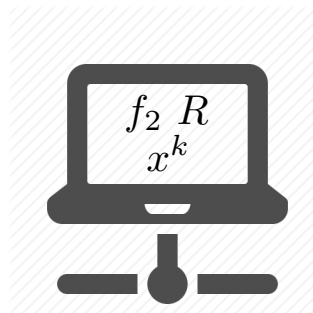


$$g^k = \frac{\quad + \quad +}{3}$$



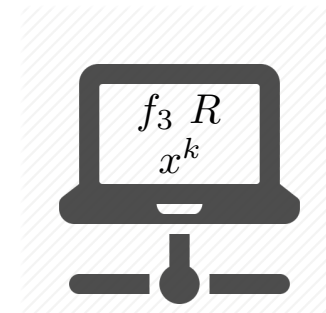
$$\cancel{\nabla f_1(x^k)}$$

$$\mathcal{C}(\nabla f_1(x^k))$$



$$\cancel{\nabla f_2(x^k)}$$

$$\mathcal{C}(\nabla f_2(x^k))$$



$$\cancel{\nabla f_3(x^k)}$$

$$\mathcal{C}(\nabla f_3(x^k))$$

GD with Compressed Gradients

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

[Alistarh et al, 2017]

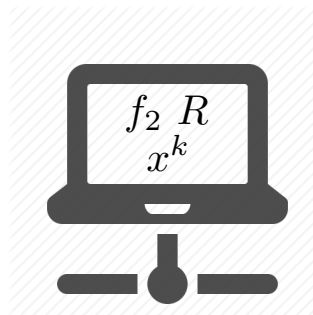
Parameter server



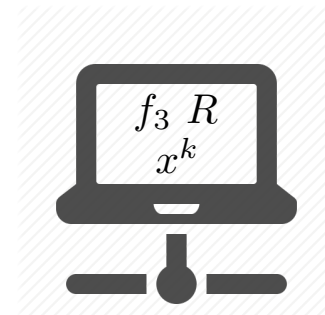
g^k



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



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



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

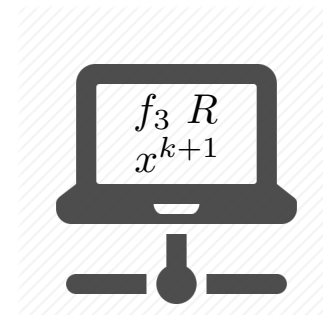
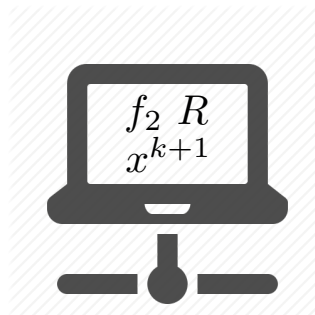
GD with Compressed Gradients

$$\min_{x \in \mathbb{R}^d} \frac{1}{3} \sum_{i=1}^3 f_i(x) + R(x)$$

Parameter server



[Alistarh et al, 2017]



GD with Compressed Gradients

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n f_i(x) + R(x) \quad \Rightarrow \quad x^{k+1} = \text{prox}_{\gamma R} \left(x^k - \underbrace{\gamma \frac{1}{n} \sum_{i=1}^n \mathcal{C}(\nabla f_i(x^k))}_{g^k} \right)$$

$R \equiv 0$
special $\mathcal{C}$ } [QSGD: Alistarh et al, 2017]

The Good: unbiasedness

$$\mathbb{E}_{\mathcal{C}} [g^k] = \mathbb{E}_{\mathcal{C}} \left[\frac{1}{n} \sum_{i=1}^n \mathcal{C}(\nabla f_i(x^k)) \right] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathcal{C}} [\mathcal{C}(\nabla f_i(x^k))] = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k)$$

$\mathbb{E}_{\mathcal{C}} [\mathcal{C}(x)] = x$

The Bad: introduction of variance!

- Does not converge linearly to the solution even if f is smooth & strongly convex
- Does not work for regularized problems!

Variance Reduction for Gradient Compression

Solution: VR for Gradient Compression

Learning the gradients at the optimum: $h_i^k \rightarrow \nabla f_i(x^*)$

$$g^k = \frac{1}{n} \sum_{i=1}^n \mathcal{C}(\nabla f_i(x^k) - h_i^k) + h_i^k$$

First VR method
for gradient
compression

Paper	Problem	Compression $\mathcal{C}$
SEGA [Hanzely, Mishchenko, R, NeurIPS 2018]	$f(x)$	sketching $\mathcal{C}(x) = M S (S^\top S)^\dagger S^\top x$
DIANA [Mishchenko, Gorbunov, Takáč, R, 2019]	$\frac{1}{n} \sum_{i=1}^n f_i(x)$	ternary quantization $M = (\mathbb{E} [S(S^\top S)^\dagger S^\top])^{-1}$
99% [Mishchenko, Hanzely, R, 2019]	$\frac{1}{n} \sum_{i=1}^n f_i(x)$	sparsification / coordinate sketching
DIANA [Horváth, Kovalev, Mishchenko, R, Stich 2019]	$\frac{1}{n} \sum_{i=1}^n \left(\frac{1}{m} \sum_{j=1}^m f_{ij}(x) \right)$	general $\mathbb{E}_{\mathcal{C}} [\mathcal{C}(x)] = x \quad \mathbb{E}_{\mathcal{C}} [\ \mathcal{C}(x) - x\ ^2] \leq \omega \ x\ ^2$

First VR method
for gradient
compression
in $n > 1$ case

First VR method
for **general**
gradient
compression

References



Filip Hanzely, Konstantin Mishchenko and P. R.
SEGA: Variance reduction via gradient sketching
NeurIPS, 2018

SEGA



Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč and P. R.
Distributed learning with compressed gradient differences
arXiv:1901.09269, 2019

DIANA



Konstantin Mishchenko, Filip Hanzely and Peter Richtárik
99% of distributed optimization is a waste of time: the issue and how to fix it
arXiv:1901.09437, 2019

99%



Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, P. R. and Sebastian Stich
Stochastic distributed learning with gradient quantization and variance reduction
arXiv:1904.05115, 2019

DIANA

2. SEGA: Introduction

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 Lab



King Abdullah University
of Science and Technology



Composite Minimization

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

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

$$\min_{x \in \mathbb{R}^d} F(x) \stackrel{\text{def}}{=} f(x) + R(x)$$

Dimension d :
very large

convex & closed
(and not necessarily separable)

Gradient Sketch

New Stochastic First Order Oracle

Sketched Gradient (SEGA) Oracle

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

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


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

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

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

Examples

1

Gaussian sketch

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

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

2

Coordinate sketch

$$\mathbf{S} = \mathbf{e}_i \text{ with probability } p_i > 0$$

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

Why Bother?

- Useful for understanding **gradient compression in distributed training**
 - the first variance reduction strategy in this setup
- Optimization under a **new oracle**
 - worthy of study on its own
- **Extending coordinate descent** (subspace descent) methods to **non-separable R**
 - we get the same theory as state-of-the-art RCD methods in special case

Proximal Stochastic Gradient Descent

$$\text{prox}_{\gamma R}(x) \stackrel{\text{def}}{=} \arg \min_{y \in \mathbb{R}^d} \left(\gamma R(y) + \frac{1}{2} \|y - x\|^2 \right)$$

stepsize

$$x^{k+1} = \text{prox}_{\gamma R} \left(x^k - \gamma g^k \right)$$

Key question:

How to construct a “good” estimator
using the **SkEtched GrAdient (SEGA)**
oracle?

“Good” estimator
of the gradient

3. 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**

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

3. **Diminishing variance**

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

Sketch & Project

Sketch & Project

New estimator of the gradient

Previous estimator of the gradient

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

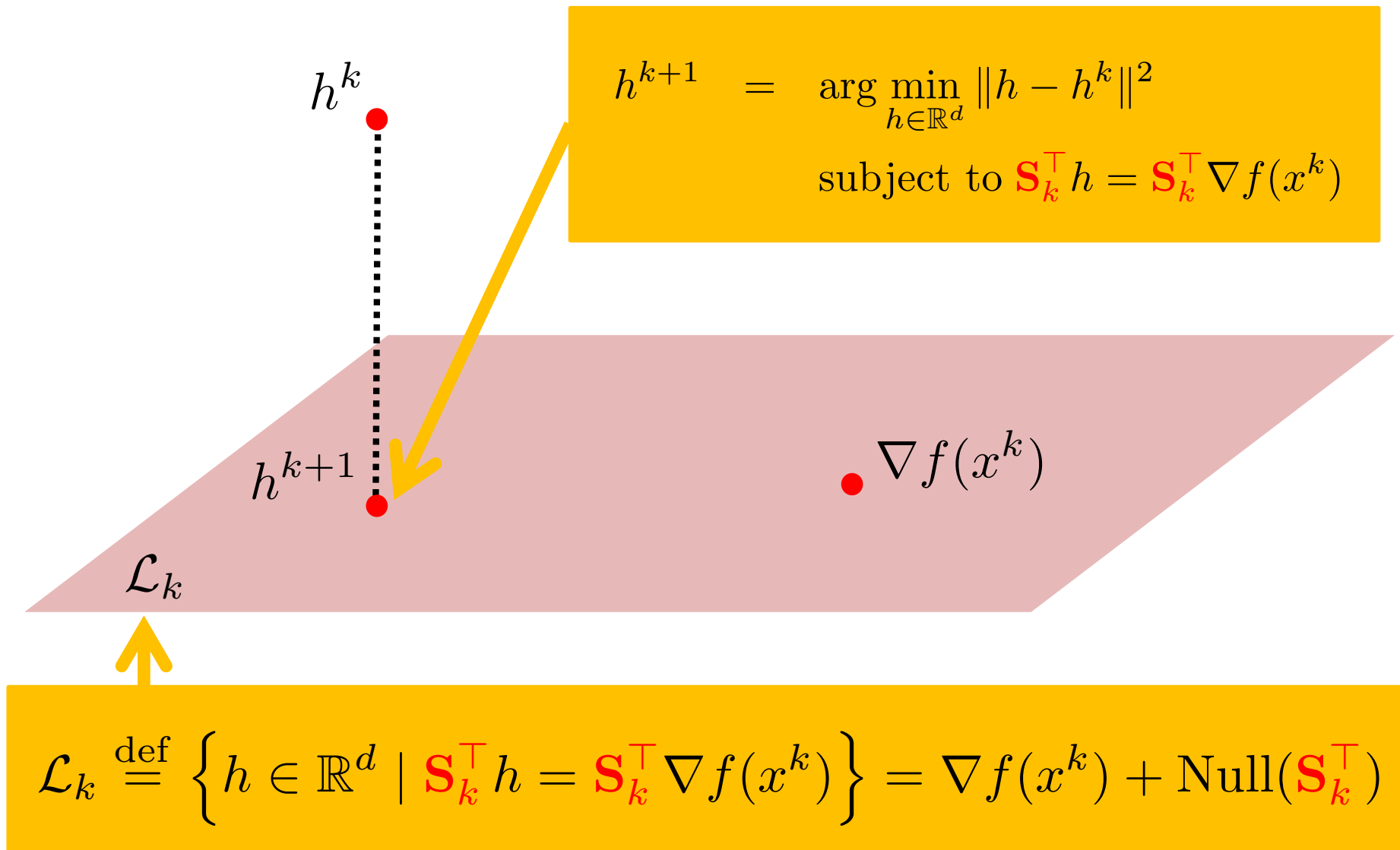
$$\text{subject to } \mathbf{S}_k^\top h = \underbrace{\mathbf{S}_k^\top \nabla f(x^k)}_{\text{Sketched Gradient}}$$

Closed-form solution:

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

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

Sketch & Project: Visualization



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



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

New understanding
of Quasi-Newton
Rules

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

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



Ion Necoara, Andrei Patrascu and P.R.

Randomized Projection Methods for Convex Feasibility Problems: Conditioning and Convergence Rates

arXiv:1801.04873, 2018

Extension to
Convex
Feasibility

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

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

Acceleration

Unbiasedness: SEGA for Coordinate Sketches

$$d = 2$$

2D Example

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

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

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

Case 1

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

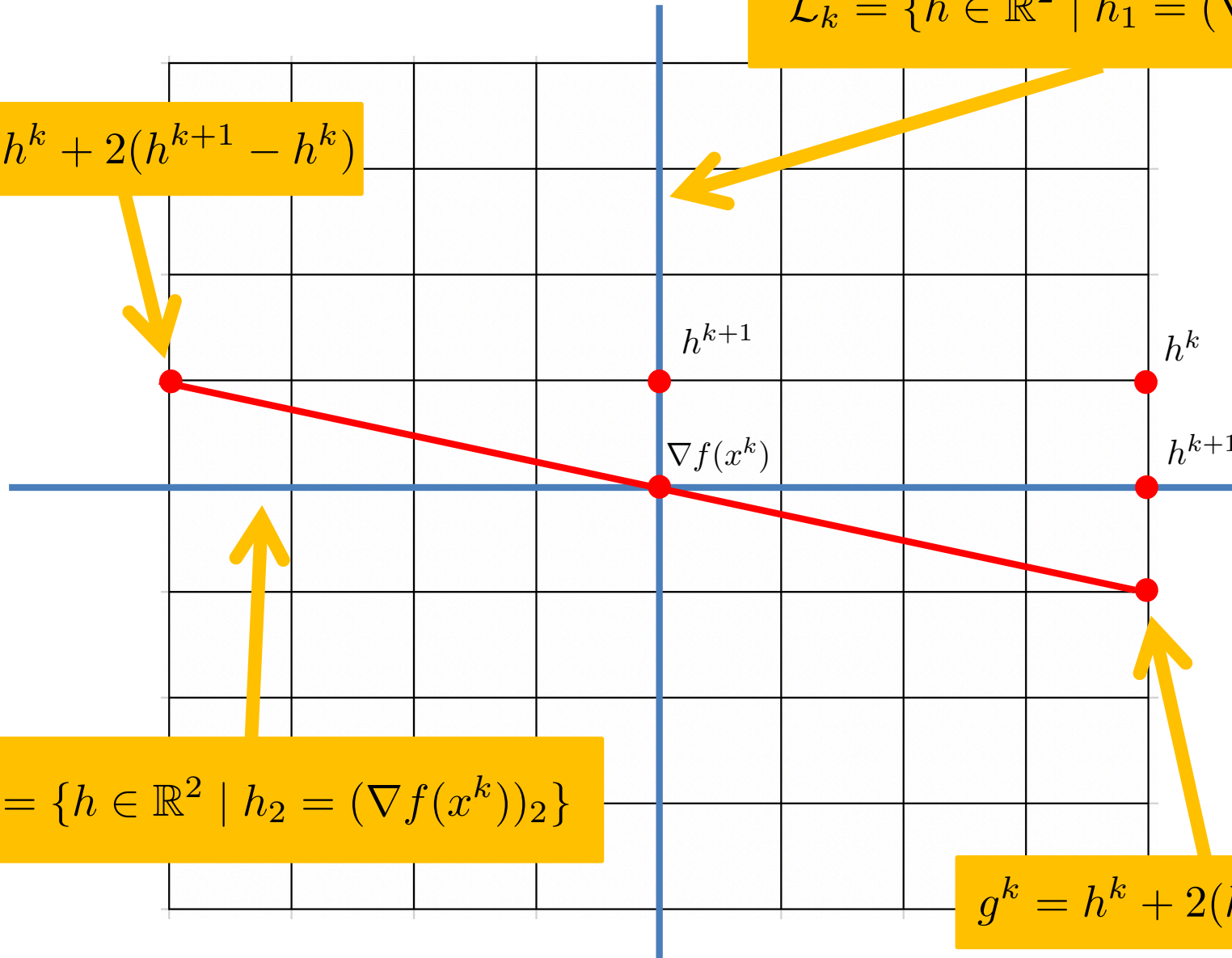
SEGA Estimator

$d = 2$

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

$$\mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_1 = (\nabla f(x^k))_1\}$$

$$g^k = h^k + 2(h^{k+1} - h^k)$$



$$\mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_2 = (\nabla f(x^k))_2\}$$

$$g^k = h^k + 2(h^{k+1} - h^k)$$

Case 2

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

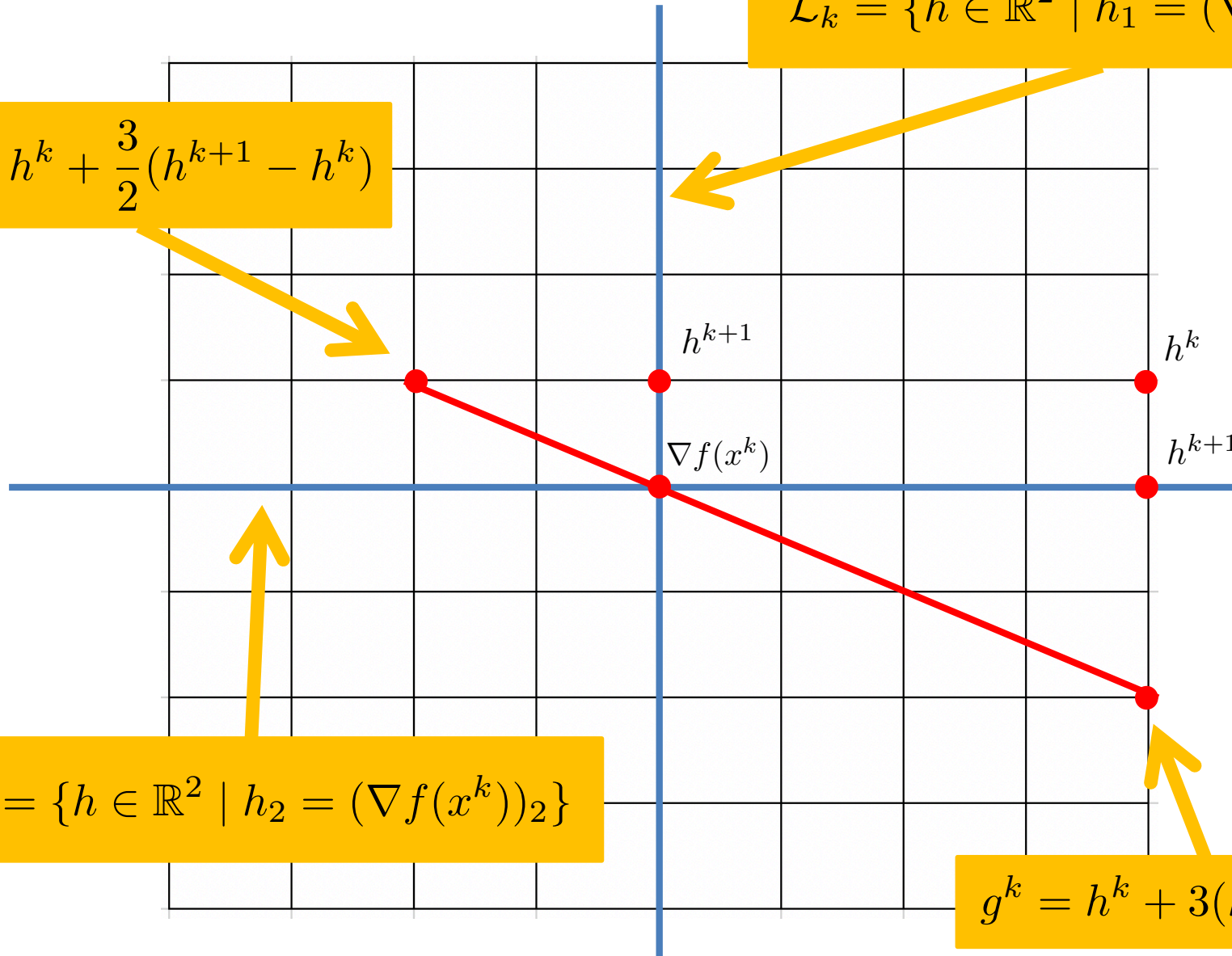
SEGA Estimator

$d = 2$

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

$$\mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_1 = (\nabla f(x^k))_1\}$$

$$g^k = h^k + \frac{3}{2}(h^{k+1} - h^k)$$



$$\mathcal{L}_k = \{h \in \mathbb{R}^2 \mid h_2 = (\nabla f(x^k))_2\}$$

$$g^k = h^k + 3(h^{k+1} - h^k)$$

SEGA for General Sketches

SEGA Estimator

SEGA estimator

Bias correcting random variable

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

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

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

4. SEGA: The Algorithm

The Algorithm

The SEGA Algorithm

$$\min_{x \in \mathbb{R}^d} F(x) \stackrel{\text{def}}{=} f(x) + R(x)$$

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

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

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

Perform **Sketch & Project**

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

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

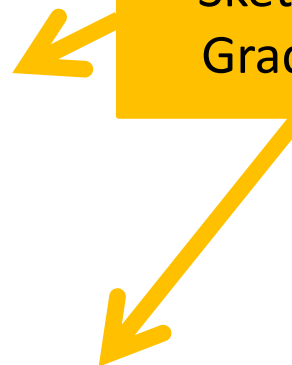
2 Compute the **SEGA Estimator**

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

3 Perform **Proximal SGD** step

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

Sketched
Gradient



Variants of SEGA

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

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

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

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

3. Subspace SEGA

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

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

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

$$h \in \text{Range}(A^\top)$$

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

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

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

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

Stepsize can't be too large:

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

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

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

Complexity:

Coordinate Sketch

Coordinate Sketch: Arbitrary Sampling Setup

Random subset of $\{1, \dots, d\}$

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

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

Hadamard product

Complexity Results

$$R \equiv 0$$

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

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

Method	Complexity
SEGA importance sampling	$8.55 \cdot \frac{\text{Tr}(\mathbf{L})}{\mu} \log \frac{1}{\epsilon}$
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



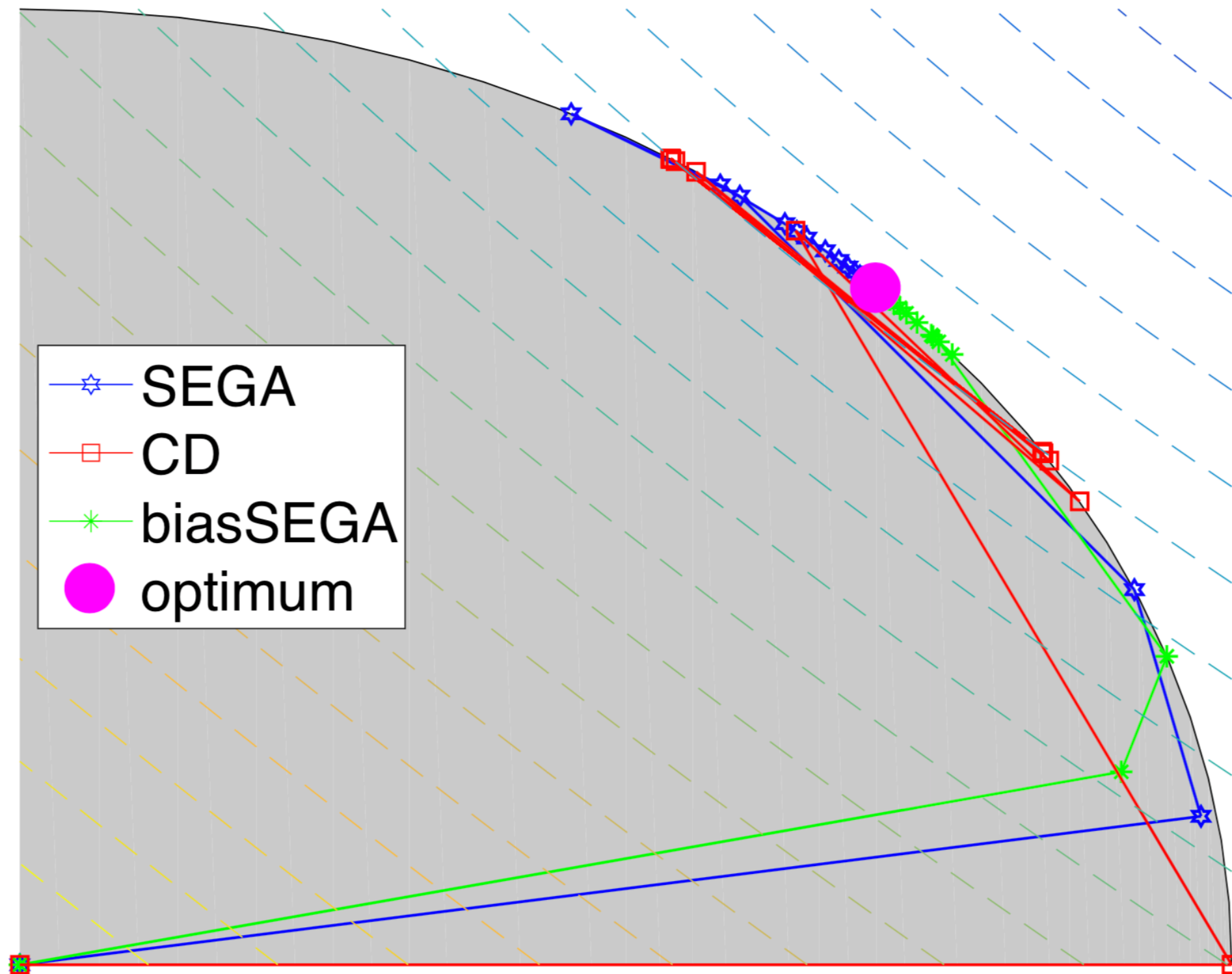
Filip Hanzely and P.R.

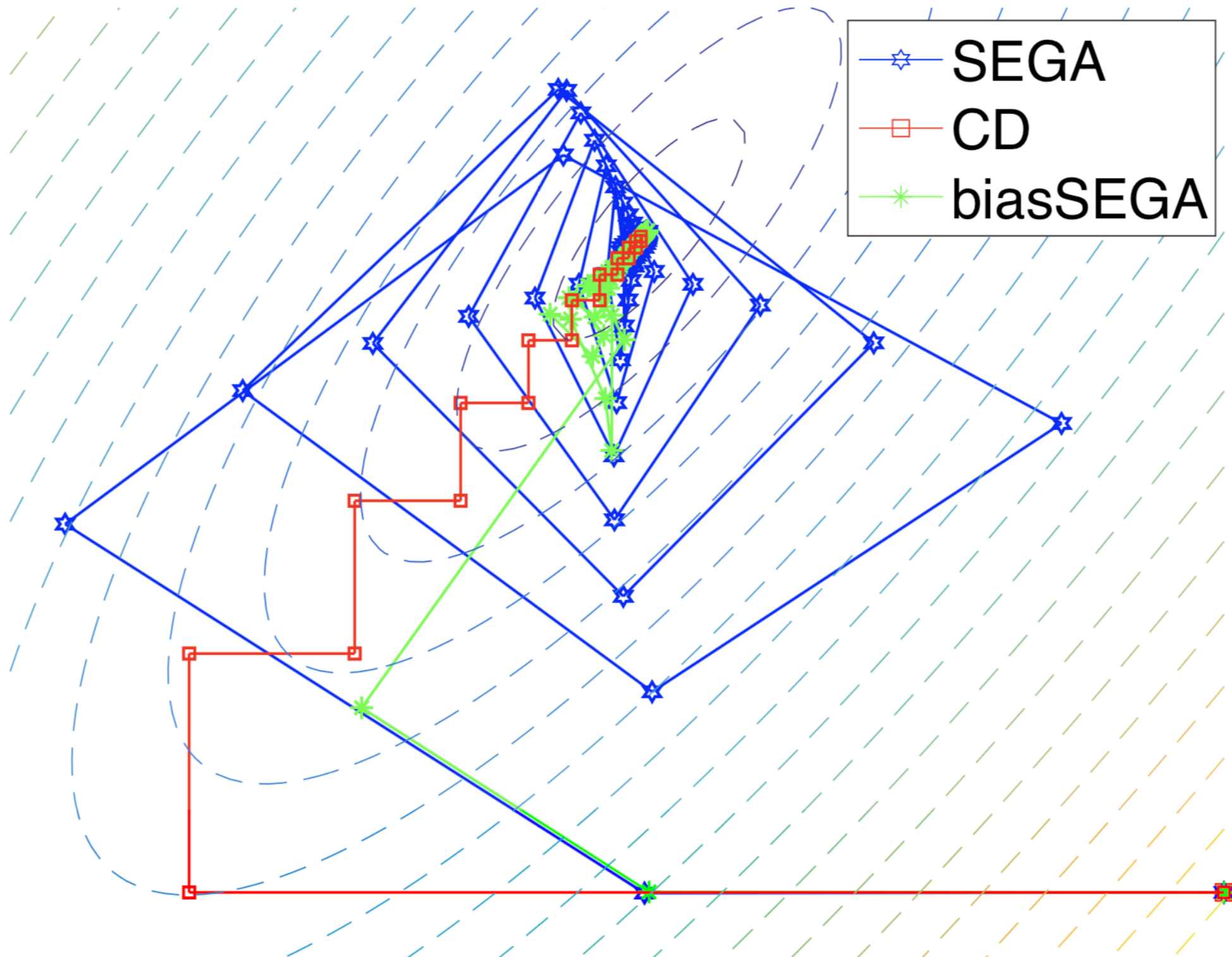
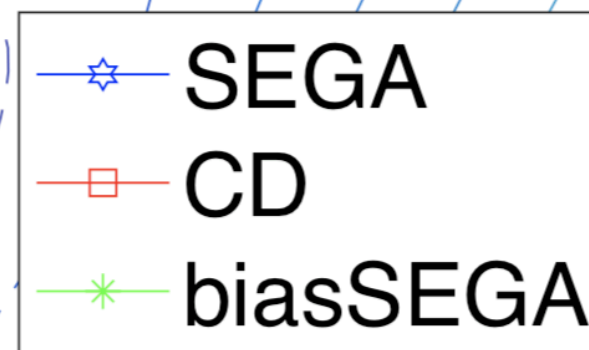
Accelerated coordinate descent with arbitrary sampling and best rates for minibatches

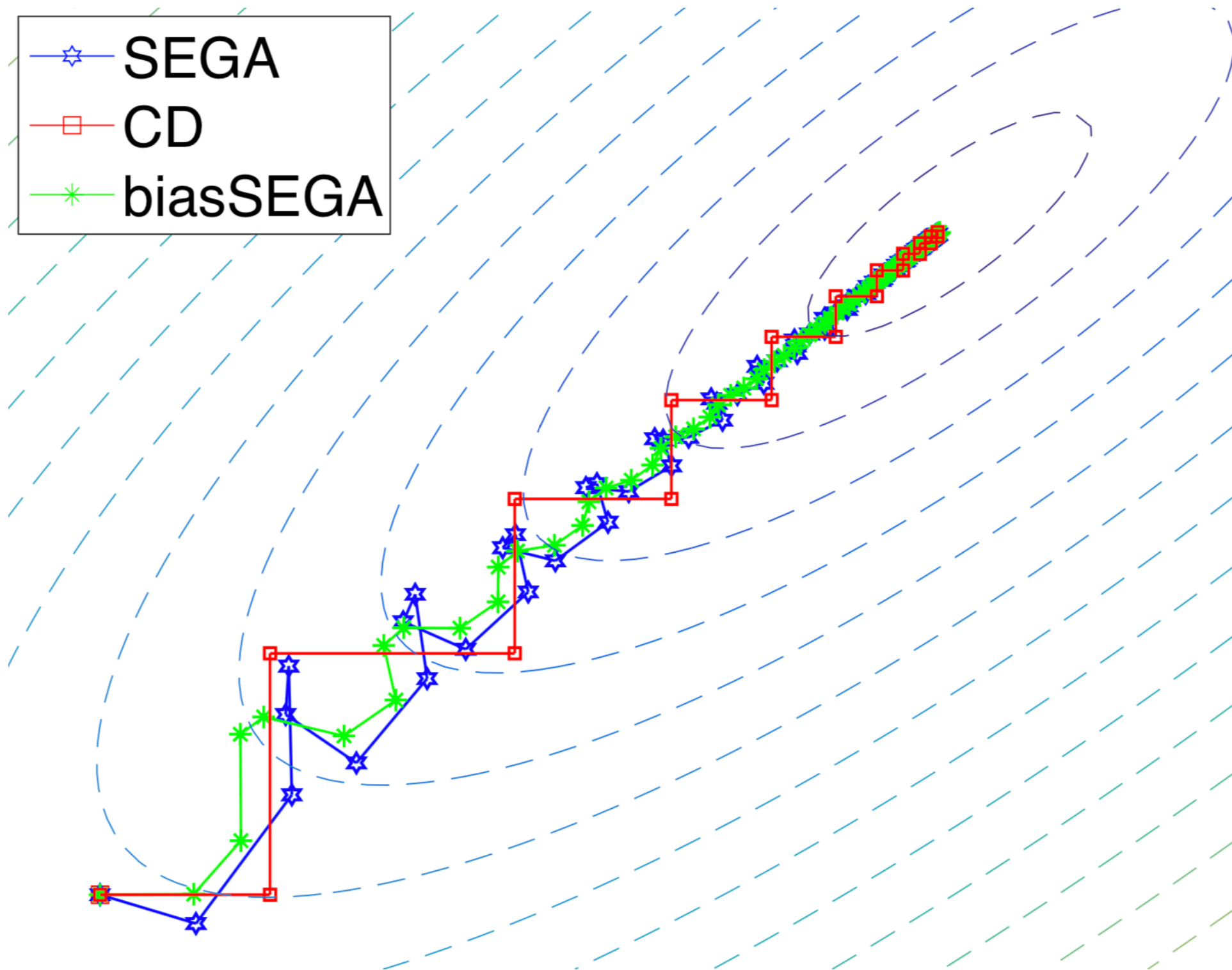
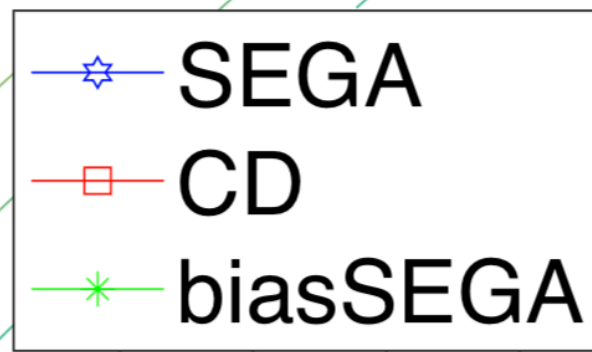
AISTATS 2019

5. Experiments

Illustration in 2D



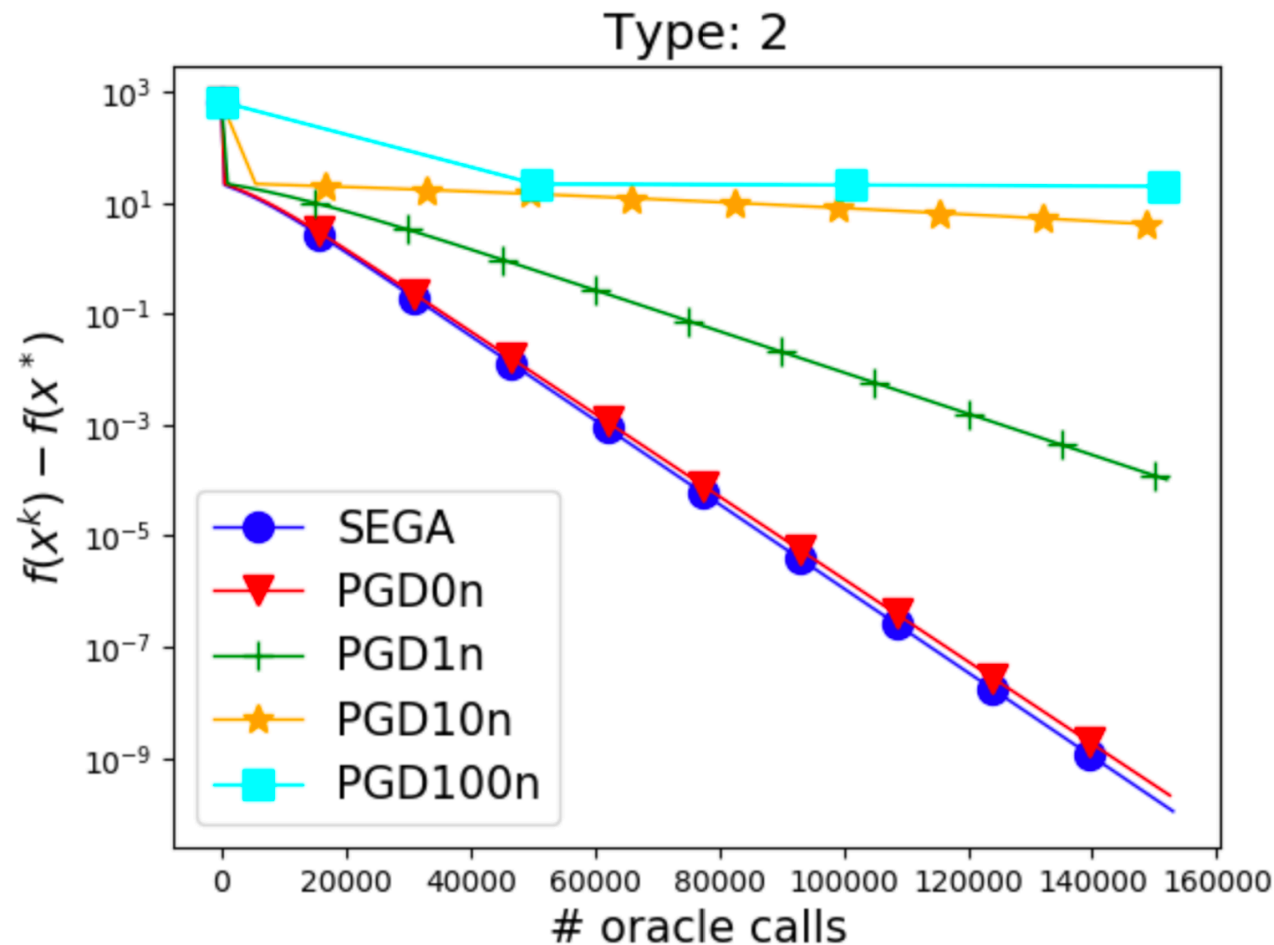




SEGA vs Projected Gradient Descent

Gaussian Sketch, Ball Constraint

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



$d = 500$

SEGA vs
Subspace SEGA

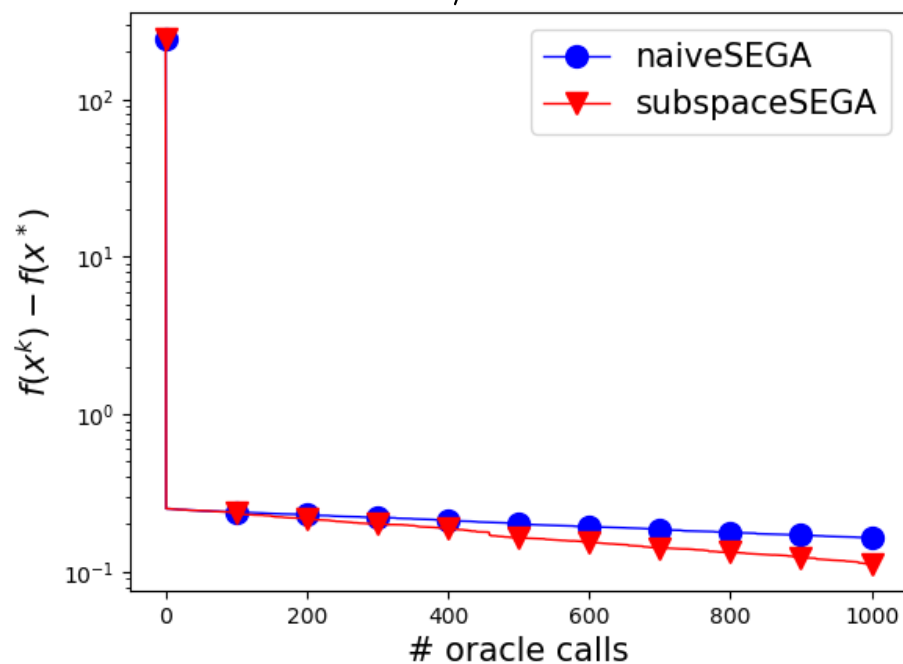
SEGA vs Subspace SEGA

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

$$\dim \text{Range}(\mathbf{A}^\top) \leq n \ll d$$

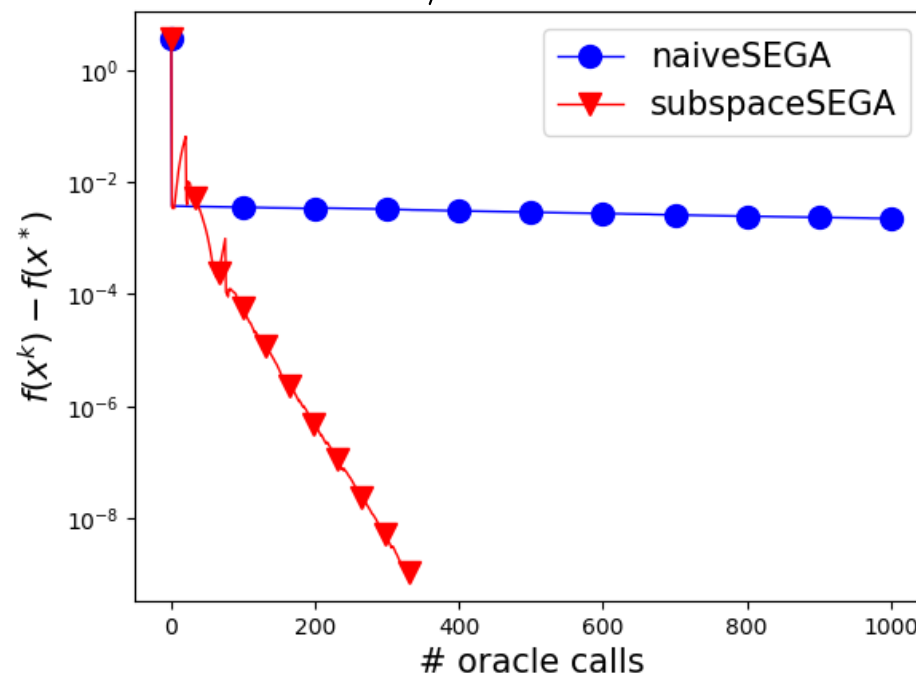
$$f(x) = \|\mathbf{A}x - b\|^2 \quad \Rightarrow \quad \nabla f(x) \in \text{Range}(\mathbf{A}^\top)$$

$$d/n = 2$$



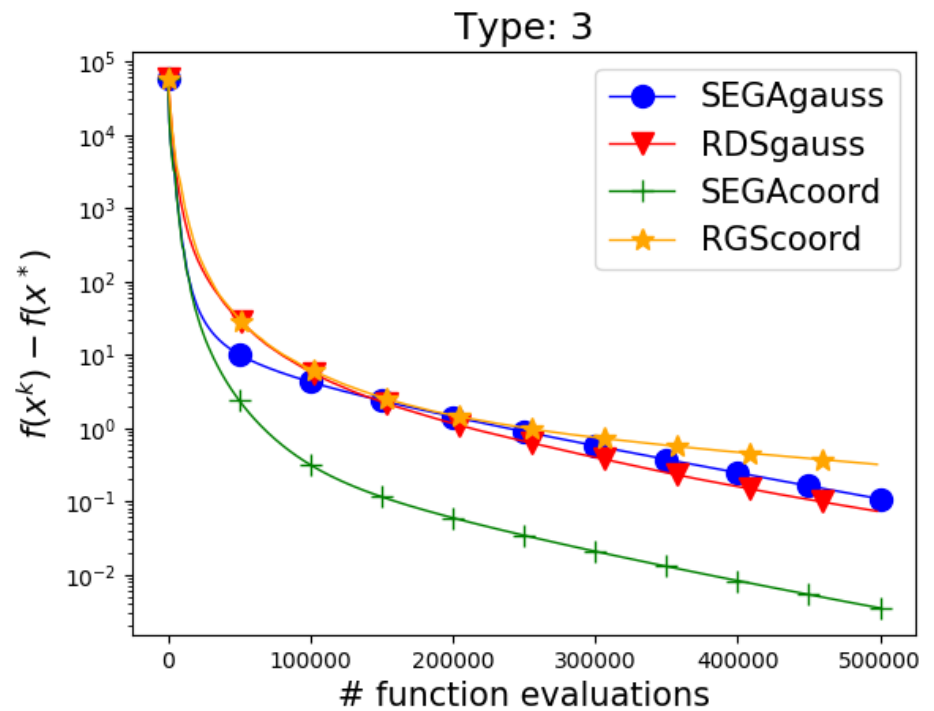
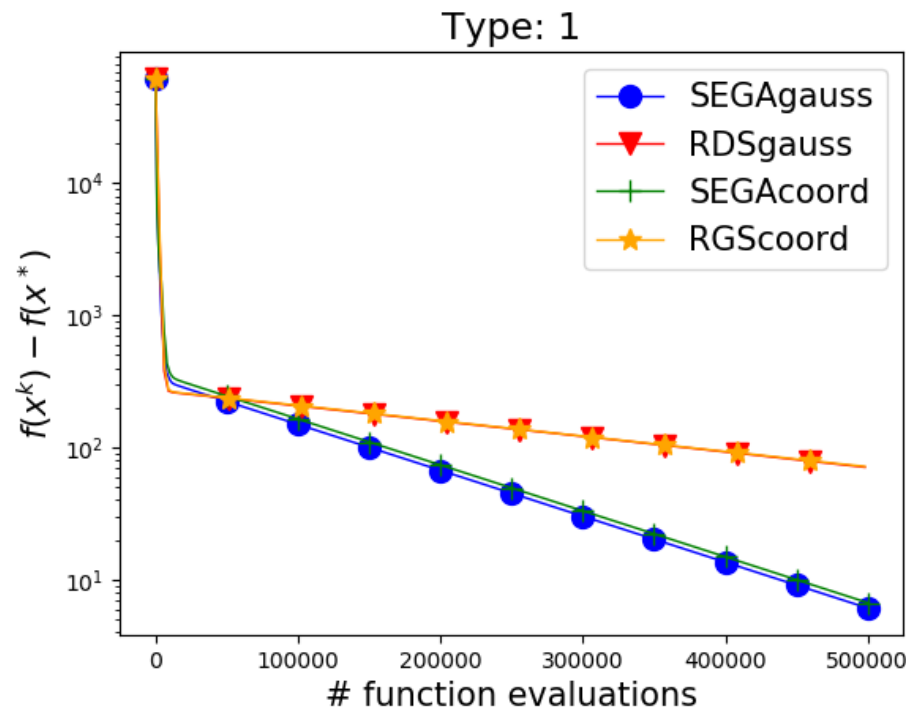
$\approx 2 \times$ speedup

$$d/n = 100$$



$\approx 100 \times$ speedup

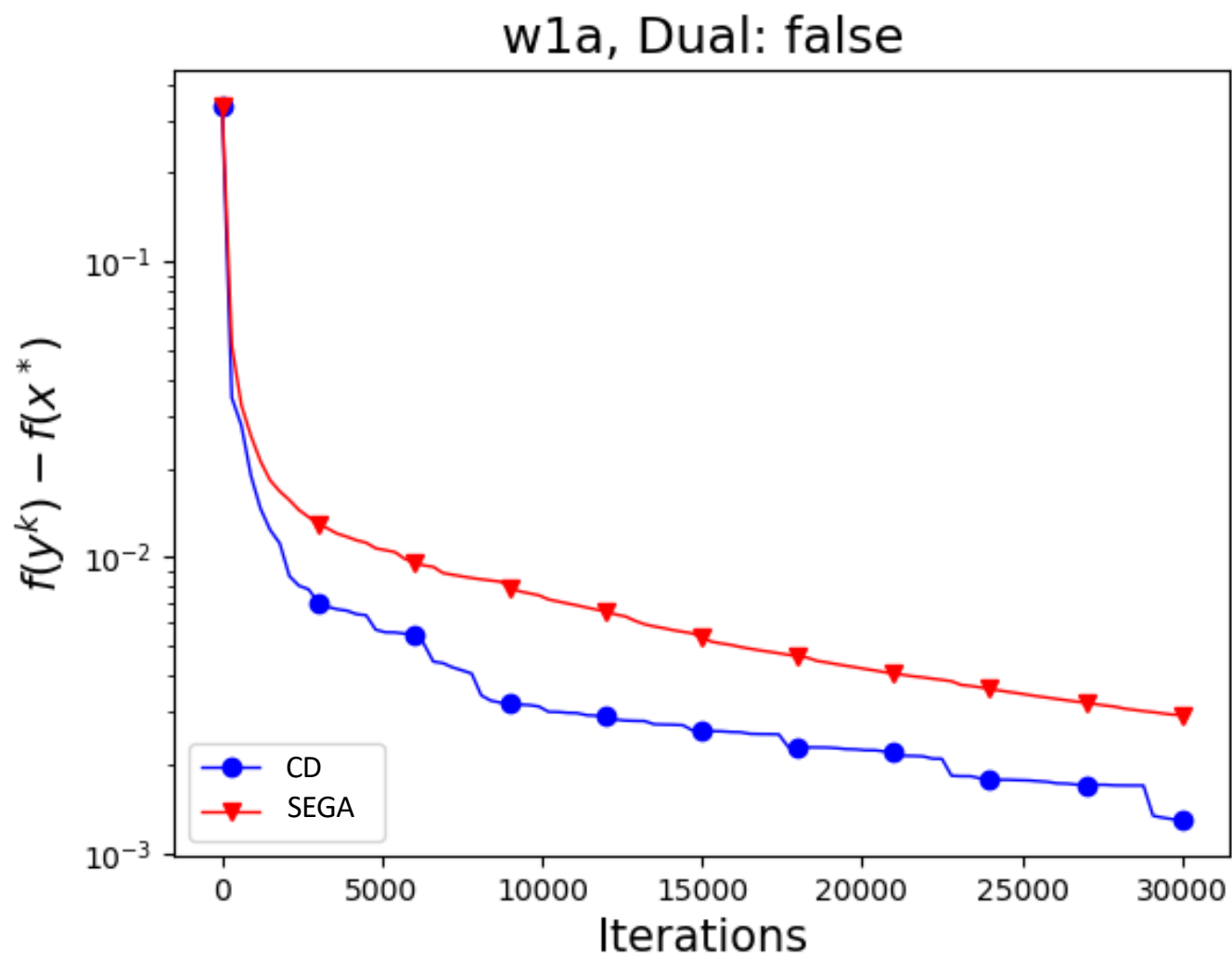
SEGA vs Random Direct Search



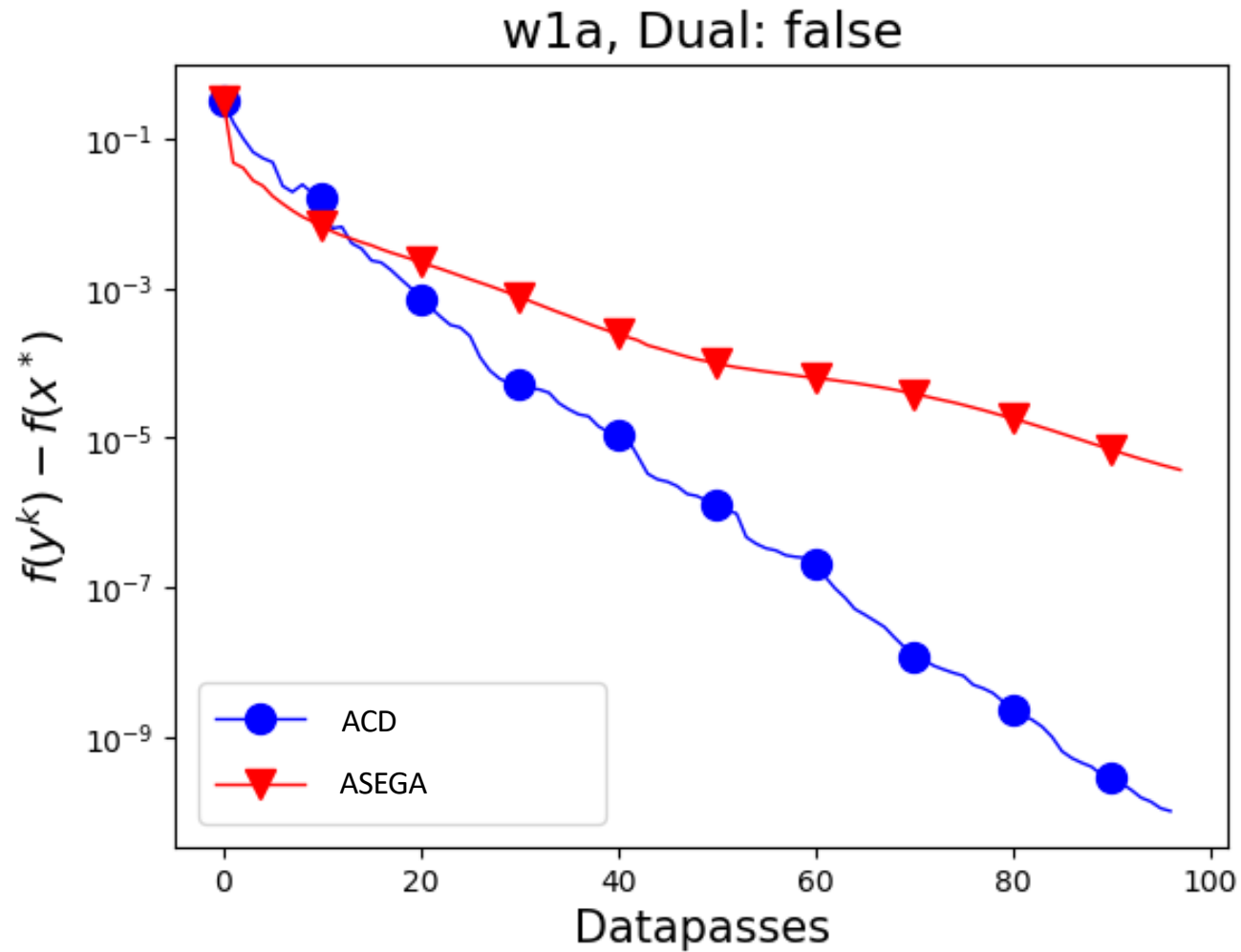
El Houcine Bergou, Eduard Gorbunov and P.R.
Stochastic three points method for unconstrained smooth minimization
arXiv:1902.03591, 2019

SEGA vs Coordinate Descent

SEGA vs CD



Accelerated SEGA vs Accelerated CD



6. 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