SEGA: Variance Reduction via Gradient Sketchin

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

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

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

THE UNIVERSITY OF EDINBURGH

MIPT MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

Problem setup

Optimization Problem:

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

Oracle: sketched gradient for random matrix S

$$\zeta(S,x) \overset{\text{def}}{=} S^\top \nabla f(x) \in \mathbb{R}^b$$

Proximal operator of R is cheap.
R is not separable $\rightarrow$ Coordinate descent fails.
How to do subspace descent with nonseparable prox function R?

Assumptions

M-smoothness:

$$f(x) \le f(y) + \langle \nabla f(y), x-y\rangle + \tfrac{1}{2}(x-y)^\top M(x-y)$$

μ-strong convexity:

$$f(x) \ge f(y) + \langle \nabla f(y), x-y\rangle + \tfrac{\mu}{2}\|x-y\|^2$$

$\rightarrow$ natural for ERM with linear predictors

SEGA: Variance reduced coordinate descent

Learning gradient from sketches:

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

Step in the unbiased direction ($\mathbf{E}\left[g^k\right] = \nabla f(x^k)$):

$$g^k = h^k + \theta_k Z_k(\nabla f(x^k) - h^k),$$

$$Z_k \overset{\text{def}}{=} S_k\left(S_k^\top S_k\right)^\dagger S_k^\top.$$

As $x_k \to x^*$, we have $g^k \to 0$, which is not the case for subspace descent.

Main Contributions

- **SEGA with general analysis.**
- **Subspace SEGA.**
- **(Almost) recovered rates of CD.**

Algorithm

Algorithm 1 SEGA: SkEtched GrAdient Method

1: **Initialize** $x^0, h^0 \in \mathbb{R}^n$; $B \succ 0$; **distribution** $\mathcal{D}$; **stepsize** $\alpha > 0$
2: **for** $k = 0, 1, 2, \dots$ **do**
3: **Sample** $S_k \sim \mathcal{D}$
4: $g^k = h^k + \theta_k B^{-1} Z_k(\nabla f(x^k) - h^k)$
5: $x^{k+1} = \mathrm{prox}_{\alpha R}(x^k - \alpha g^k)$
6: $h^{k+1} = h^k + B^{-1} Z_k(\nabla f(x^k) - h^k)$
7: **end for**

Convergence result

Suppose that S is uniform distribution of standard coordinate basis vectors. Then with stepsize $\alpha = \Omega(\frac{1}{n\lambda_{\max}(M)})$ we have

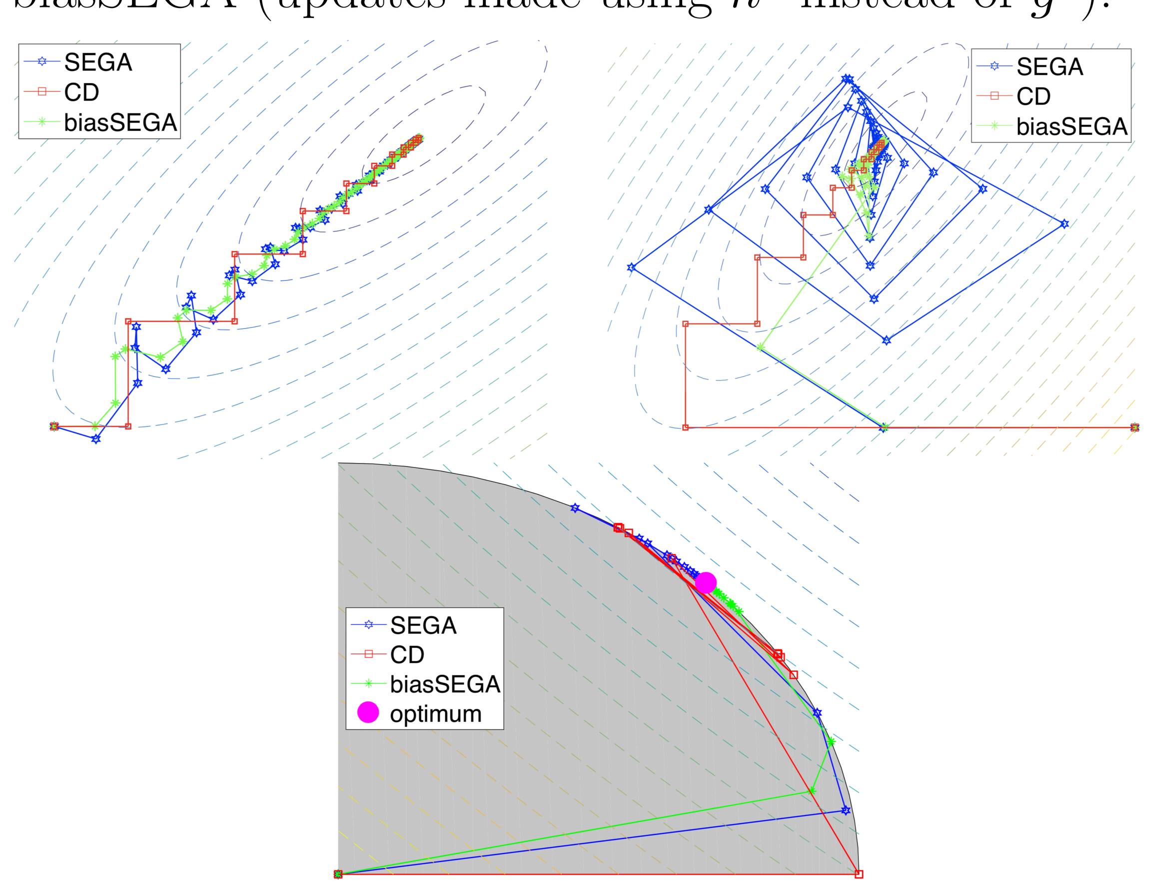
$$\mathbb{E}\Phi^{k+1} \le (1-\alpha\mu)\Phi^k$$

for $\Phi^k = \|x^k - x^*\|^2 + \sigma\alpha\|h^k - \nabla f(x^*)\|^2$.

General convergence with arbitrary weighted norm and arbitrary sketching distribution is provided.

Algorithm behavior

Iterates evolution of SEGA, coordinate descent and biasSEGA (updates made using h^k instead of g^k).



Here R is the indicator function of the unit ball $\Rightarrow$ CD does not converge!

Subspace SEGA

Suppose $f(x) = \phi(Ax)$.
Idea: Exploit that ∇f lies in known subspace.
New update for h (and g):

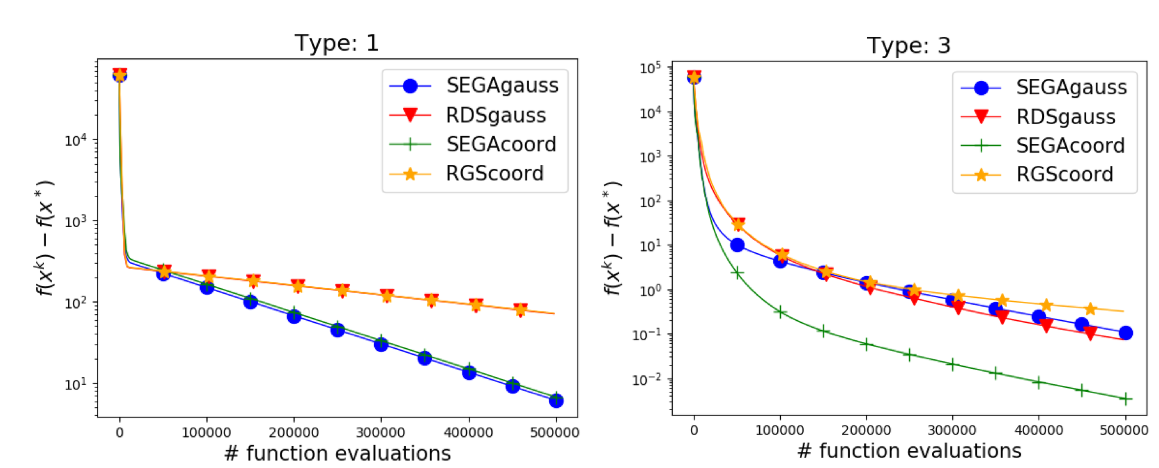
$$h^{k+1} = \arg\min_{h\in\mathbb{R}^n} \|h-h^k\|^2 \quad \text{subject to} \quad S_k^\top h = S_k^\top \nabla f(x^k), \quad h \in \mathbf{Range}\left(A^\top\right)$$

If S_k is sampled from columns of $A^\top$, we might achieve $\Omega(\frac{n}{d})$ speedup over the naive version of SEGA. $(A \in \mathbb{R}^{d\times n})$

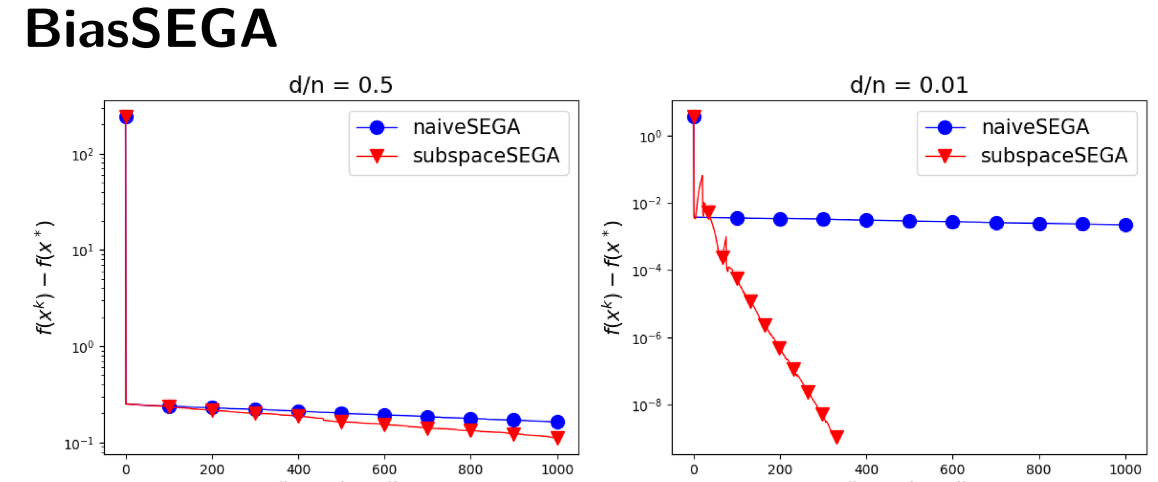
Experiments

Zeroth-order setting

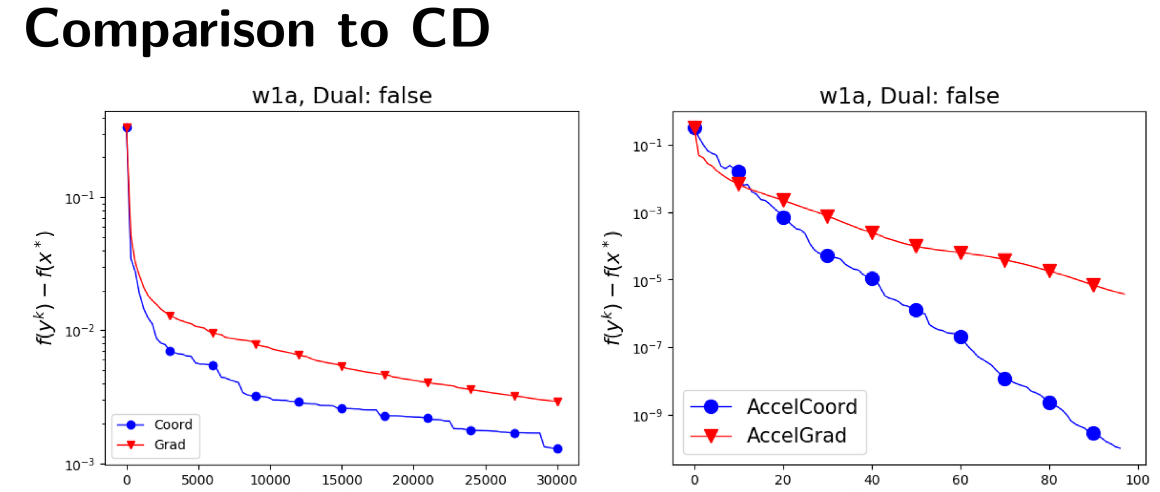
- Directional gradient is $\Omega(n)$ times cheaper to full with forward diff
- Comparison with Random Direct Search [1]



BiasSEGA



Comparison to CD



SEGA at coordinate descent setup

- Sketches S are column submatrices of identity
- Probability vector p: $\mathbb{P}(e_i \in S) = p_i$
- Probability matrix P: $\mathbb{P}(e_i \in S, e_j \in S) = P_{i,j}$
- ESO vector v (for minibatching):

$$P \circ M \preceq \mathbf{Diag}\,(p \circ v)$$

Accelerated SEGA

Algorithm 2 ASEGA: Accelerated SEGA

1: $x^0 = y^0 = z^0 \in \mathbb{R}^n$; $h^0 \in \mathbb{R}^n$; S; parameters $\alpha, \beta, \tau, \mu > 0$
2: **for** $k = 1, 2, \dots$ **do**
3: $x^k = (1-\tau)y^{k-1} + \tau z^{k-1}$
4: Sample S_k, and compute g^k, h^{k+1}
5: $y^k = x^k - \alpha p^{-1} \circ g^k$
6: $z^k = \frac{1}{1+\beta\mu}(z^k + \beta\mu x^k - \beta g^k)$
7: **end for**

Rates

Method	Complexity
Nonaccelerated, importance sampling,	$8.55 \cdot \frac{\mathbf{Tr}(M)}{\mu}\log\frac{1}{\epsilon}$
Nonaccelerated, arbitrary sampling	$8.55 \cdot \left(\max_i \frac{v_i}{p_i\mu}\right)\log\frac{1}{\epsilon}$
Accelerated, importance sampling,	$9.8 \cdot \frac{\sum_i \sqrt{M_{ii}}}{\sqrt{\mu}}\log\frac{1}{\epsilon}$
Accelerated, arbitrary sampling	$9.8 \cdot \sqrt{\max_i \frac{v_i}{p_i^2\mu}}\log\frac{1}{\epsilon}$

Up to constant factor same rates as CD [2, 3]

References

[1] El Houcine Bergou, Peter Richtárik, and Eduard Gorbunov. Random direct search method for minimizing nonconvex, convex and strongly convex functions. *Manuscript*, 2018.

[2] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1110–1119, 2016.

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