



One Method to Rule Them All: Variance Reduction for Data, Parameters and Many New Methods

Filip Hanzely, Peter Richtarik



Problem and Introduction

Problem

strongly convex

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

convex, proximable

(specified later)

M_i smooth and convex

matrix smoothness
(specified later)

Oracle: random linear measurements of (stochastic) gradient
(specified later)

Main result: Very general algorithm

Problem

strongly convex

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

convex, proximable

\mathbf{M}_i smooth and convex

$$f_i(x) \leq f_i(y) + \langle \nabla f_i(y), x - y \rangle + \frac{1}{2}(x - y)^\top \mathbf{M}_i(x - y)$$

ϕ_i is L_i smooth

$$f_i(x) = \phi_i(A_i x)$$

$$\mathbf{M}_i = L_i A_i^\top A_i$$

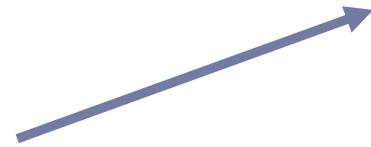
Gradient Descent, SGD and Variance reduction

let $\psi = 0$

GD: $x^+ = x - \alpha \nabla f(x)$

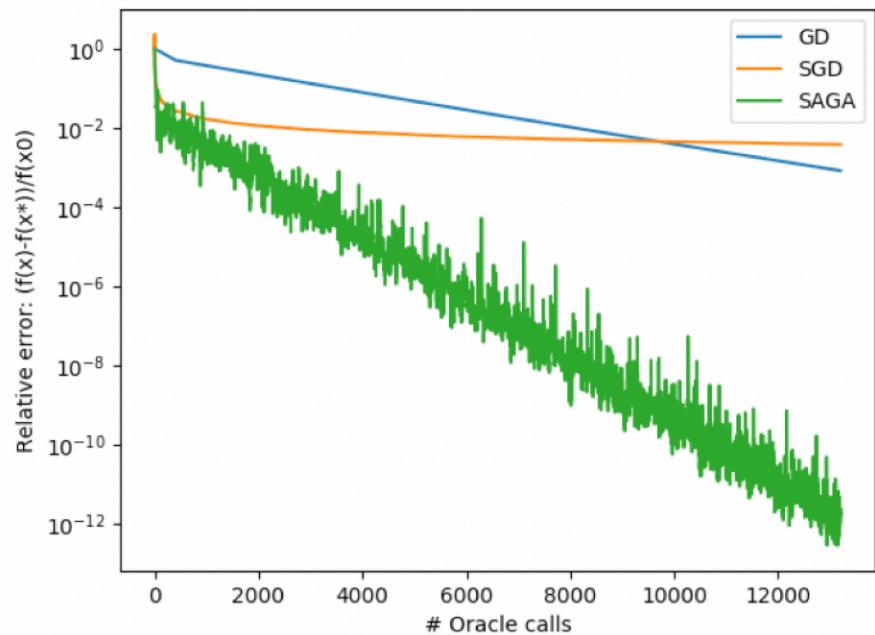
SGD: $x^+ = x - \alpha \nabla f_i(x)$

SAGA: $x^+ = x - \alpha g$



Variance reduced stochastic gradient

$g \rightarrow 0$ as $x \rightarrow x^*$, $E[g] = \nabla f(x)$





Variance Reduction in Optimization: Classical Approach

Control Variates in optimization [Johnson and Zhang 2013, Defazio et al. 2014]

Trick to reduce variance of stochastic gradient

Standard in statistics, relatively new in optimization

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

(no ψ for simplicity)

$$E[g^k] = \nabla f(x^k)$$

$$x^{k+1} = x^k - \alpha g^k \quad g^k = \nabla f_i(x^k) - Y^k + E[Y^k]$$

Goal: High $\text{Cov}(Y^k, \nabla f_i(x^k))$ (to decrease variance of g^k)

Solution: Set $Y^k = \nabla f_i(\phi_i^k)$ for $\phi_i^k = x^l$, where l is index corresponding to the last evaluation of $\nabla f_i(\cdot)$

As $x^k \rightarrow x^*$, we will have both $\nabla f_i(x^k) \rightarrow \nabla f_i(x^*)$ and $Y^k \rightarrow \nabla f_i(x^*)$ and thus $Y^k \rightarrow \nabla f_i(x^k)$, which means $\text{Var}(g^k) \rightarrow 0$

SAGA [Defazio et. al. 2014]

Construct control variates from freshest information

Algorithm 2 SAGA [3]

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$

Set $\psi_j^0 = x^0$ for each $j \in \{1, 2, \dots, n\}$

for $k = 0, 1, 2, \dots$ **do**

 Sample $j \in [n]$ uniformly at random

 Set $\phi_j^{k+1} = x^k$ and $\phi_i^{k+1} = \phi_i^k$ for $i \neq j$

$$g^k = \nabla f_j(\phi_j^{k+1}) - \nabla f_j(\phi_j^k) + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\phi_i^k)$$

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

end for

stochastic gradient via control variates



Variance Reduction in Optimization: SEGA

Regularization and Proximal operator

$$\text{minimize } f(x) + \psi(x)$$

(no finite sum)

convex, non-smooth, simple

Proximal operator of ψ is computable

$$\text{prox}_{\alpha\psi}(x) \stackrel{\text{def}}{=} \underset{y \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \psi(y) + \frac{1}{2\alpha} \|y - x\|^2 \right\}$$

If ψ is indicator function, then prox is projection

If ψ is L1 loss, then prox is soft thresholding

Proximal gradient: $x^+ = \text{prox}_{\alpha\psi}(x - \alpha \nabla f(x))$

gradient step

$$= \underset{y}{\operatorname{argmin}} \left\{ \psi(y) + \frac{1}{2\alpha} \|y - (x - \alpha \nabla f(x))\|^2 \right\}$$

Nearly gradient step, but considers ψ as well

Coordinate descent and proximal regularizer

CD: Compute partial derivative, take step

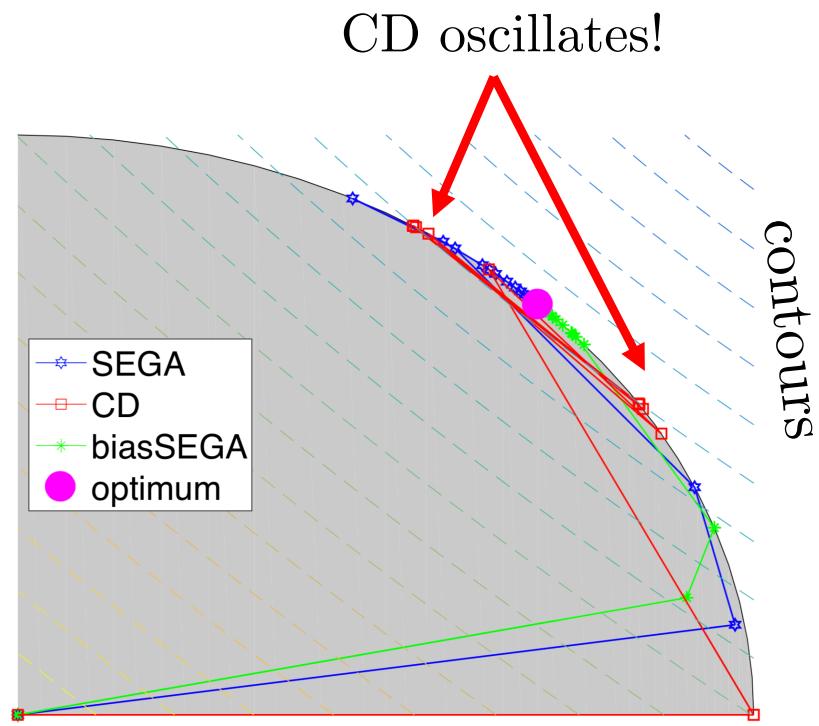
Example

$$\text{minimize } f(x) + \psi(x)$$
$$f(x) = \|x - z\|^2$$

$\|z\|$ is big enough

$$\psi(x) = \begin{cases} 0 & \|x\| \leq 1 \\ \infty & \|x\| > 1 \end{cases}$$

Proximal CD can jump away from optimum



Solution: CD with variance reduction
= SEGA

SEGA [Hanzely et. al. 2018]

Coordinate descent with variance reduction

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

$$E[g^k] = \nabla f(x^k)$$



$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k) \quad g^k = d\nabla_j f(x^k) - Y^k + E[Y^k]$$

Goal: High Cov($Y^k, d\nabla_j f(x^k)$) (to decrease variance of g^k)

Solution: Set $Y^k = d\nabla_j f(\phi_j^k)$ for $\phi_j^k = x^l$, where l is index corresponding to the last evaluation of $\nabla_j f(\cdot)$

As $x^k \rightarrow x^*$, we will have both $\nabla_j f(x^k) \rightarrow \nabla_j f(x^*)$ and $Y^k \rightarrow d\nabla_j f(x^*)$ and thus $\text{Var}(g^k) \rightarrow 0$

SEGA [Hanzely et. al. 2018]

Coordinate descent with variance reduction

Construct control variates from freshest information

Algorithm 5 SEGA

Require: Stepsize $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$

Set $h^0 = 0$

for $k = 0, 1, 2, \dots$ **do**

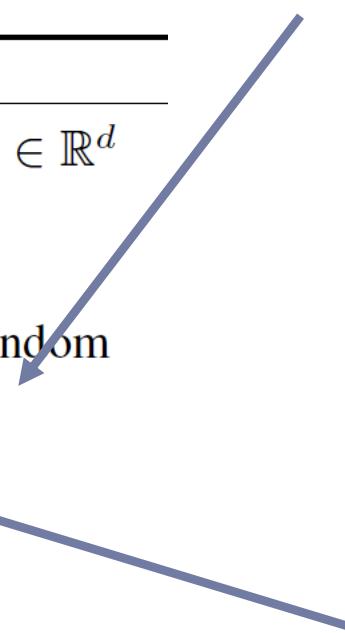
 Sample $i \in \{1, \dots, d\}$ uniformly at random

 Set $h^{k+1} = h^k + e_i(\nabla_i f(x^k) - h_i^k)$

$g^k = h^k + de_i(\nabla_i f(x^k) - h_i^k)$

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

end for



stochastic gradient via control variates

SEGA [Hanzely et. al. 2018]

Algorithm 5 SEGA

Require: Stepsize $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$

Set $h^0 = 0$

for $k = 0, 1, 2, \dots$ **do**

 Sample $i \in \{1, \dots, d\}$ uniformly at random

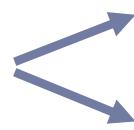
 Set $h^{k+1} = h^k + e_i(\nabla_i f(x^k) - h_i^k)$

$g^k = h^k + de_i(\nabla_i f(x^k) - h_i^k)$

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

end for

$\psi = 0 :$



Up to constant factor rate of coordinate descent

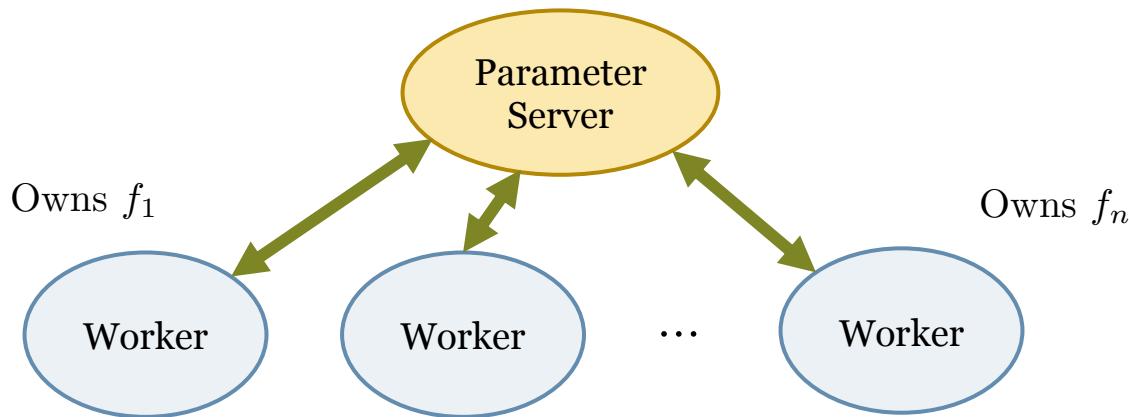
Up to constant factor rate of accelerated coordinate descent

$\psi \neq 0:$  No importance sampling



Variance Reduction in Optimization: Parallel Coordinate Descent

Parallel Coordinate Descent



Workers: Compute the local gradient update and send it to PS

Server: Average updates and send them back to workers

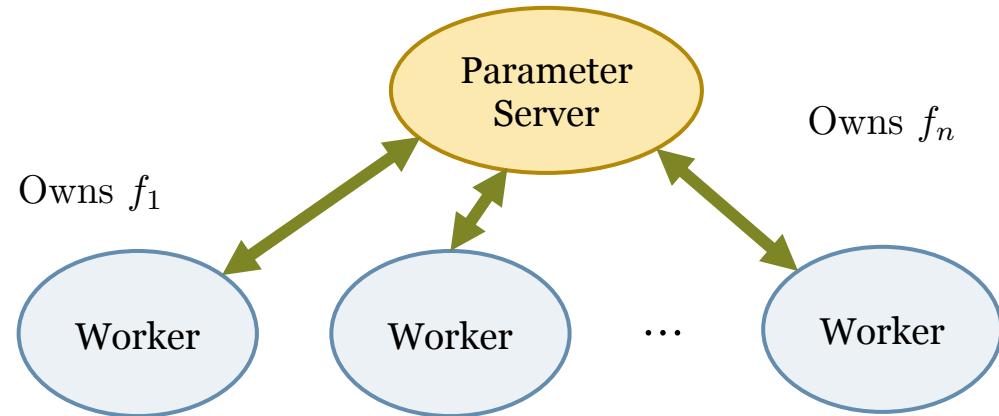
Bottleneck: Communication from workers to server

Parallel Coordinate Descent

Holds for GD, SGD, SAGA, ASGD

$$\text{if } \forall i : \nabla f_i(x^*) = 0$$

Holds for GD with SEGA trick
(no extra assumptions)



[Mishchenko, Hanzely and Richtarik, 2018]

All workers might throw away as much as $\frac{n-1}{n}$ of total number of coordinates independently at random without hurting the convergence

Constant factor at most

Reduced communication to the server for no cost!

Natural Extension: Can we do SAGA and SEGA at the same time?



Unification of the Methods

Goals

Unification of broad range of variance reduced algorithms in single method and single analysis

Analysis that include best bounds for recovered algorithms

Understanding relations between algorithms

New algorithms

Arbitrary sampling results
(tight rate for given smoothness structure of objective and algorithm)

Proximal methods

High level idea

Jacobian

$$\mathbf{G}(x) = [\nabla f_1(x), \dots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}$$

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

random linear operator
(fixed distribution)

Oracle: random linear measurement of Jacobian $\mathcal{S}\mathbf{G}(x)$

$$(0, 0, \dots, 1, \dots, 0, 0)^\top$$

Left multiplication with e_j^\top : observing j -th coordinate

Right multiplication with e_i : observing ∇f_i

Goal: Develop efficient variance reduction
Provide as fast rate as possible

3 key ingredients

- 1) Construct Jacobian estimator for control variates (sketch and project)

$$\mathbf{J}^{k+1} = \operatorname{argmin}_{\mathbf{J}} \|\mathbf{J} - \mathbf{J}^k\|^2 \quad \text{s. t. } \mathcal{S}\mathbf{J} = \mathcal{S}\mathbf{G}(x^k)$$

random linear operator  $\mathbf{G}(x) = [\nabla f_1(x), \dots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}$ 

W.L.O.G. $\mathcal{S} = \operatorname{Proj}(\operatorname{Range}(\mathcal{S}))$

- 2) Construct stochastic gradient using control variates

$$g^k = \frac{1}{n} \mathbf{J}^k e + \frac{1}{n} \mathcal{U} (\mathbf{G}(x^k) - \mathbf{J}^k) e$$

random linear operator; $E[\mathcal{U}(\mathbf{X})] = \mathbf{X}$  vector of ones 

\mathcal{S}, \mathcal{U} might be correlated

- 3) Take proximal SGD step

Algorithm

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

Algorithm 1 Generalized JacSketch (GJS)

- ```

1: Parameters: Stepsize $\alpha > 0$, random projector \mathcal{S} and unbiased sketch \mathcal{U}
2: Initialization: Choose solution estimate $x^0 \in \mathbb{R}^d$ and Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$
3: for $k = 0, 1, \dots$ do
4: Sample realizations of \mathcal{S} and \mathcal{U} , and perform sketches $\mathcal{S}\mathbf{G}(x^k)$ and $\mathcal{U}\mathbf{G}(x^k)$
5: $\mathbf{J}^{k+1} = \mathbf{J}^k - \mathcal{S}(\mathbf{J}^k - \mathbf{G}(x^k))$ update the Jacobian estimate
6: $g^k = \frac{1}{n}\mathbf{J}^k \textcolor{blue}{e} + \frac{1}{n}\mathcal{U}(\mathbf{G}(x^k) - \mathbf{J}^k) \textcolor{blue}{e}$ construct the gradient estimator
7: $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$ perform the proximal SGD step
8: end for

```

# Convergence (main theorem)

Property of randomness and smoothness

Smoothness operator (notation)

$$\mathcal{M}^{\frac{1}{2}} : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}; \mathcal{M}^{\frac{1}{2}}(\mathbf{X}) = \left[ \mathbf{M}_1^{\frac{1}{2}}(\mathbf{X}_{:1}), \mathbf{M}_2^{\frac{1}{2}}(\mathbf{X}_{:2}), \dots, \mathbf{M}_n^{\frac{1}{2}}(\mathbf{X}_{:n}) \right]$$

Condition on the stepsize  $\alpha$

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}\mathbf{e}\|^2] + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\frac{1}{2}} \mathbf{X} \right\|^2 \leq \frac{1}{n} \left\| \mathcal{M}^{\frac{1}{2}} \mathbf{X} \right\|^2$$

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}\mathbf{e}\|^2] + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\frac{1}{2}} \mathbf{X} \right\|^2 \leq (1 - \alpha\sigma) \left\| \mathcal{B} \mathcal{M}^{\frac{1}{2}} \mathbf{X} \right\|^2$$

Satisfied for small  $\alpha$

For all  $\mathbf{X}$

$$\mathbb{E} [\Psi^k] \leq (1 - \alpha\sigma)^k \Psi^0$$

strong convexity

$$\Psi^k := \|x^k - x^*\|^2 + \alpha \left\| \mathcal{B} \mathcal{M}^{\frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2$$

Maximize  $\alpha$  s. t. the bounds hold



## Special cases

# Saga with arbitrary sampling

---

**Algorithm 3** SAGA with arbitrary sampling (a variant of (Qian et al., 2019))

---

**Require:** learning rate  $\alpha > 0$ , starting point  $x^0 \in \mathbb{R}^d$ , random sampling  $R \subseteq \{1, 2, \dots, n\}$

Set  $\phi_j^0 = x^0$  for each  $j \in [n]$

**for**  $k = 0, 1, 2, \dots$  **do**

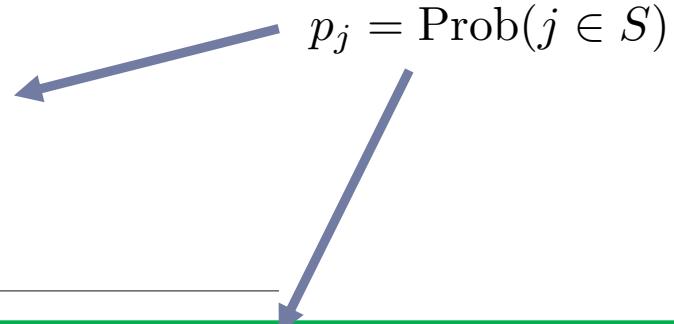
    Sample random  $R^k \subseteq \{1, 2, \dots, n\}$

    Set  $\phi_j^{k+1} = \begin{cases} x^k & j \in R^k \\ \phi_j^k & j \notin R^k \end{cases}$

$$g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi_j^k) + \sum_{j \in R^k} \frac{1}{n p_j} (\nabla f_j(\phi_j^{k+1}) - \nabla f_j(\phi_j^k))$$

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

**end for**



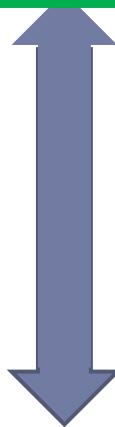
**Corollary 13 (Convergence rate of SAGA)** Let  $\alpha = \min_j \frac{n p_j}{4 v_j + n \sigma}$ . Then the iteration complexity of Algorithm 3 is  $\max_j \left( \frac{4 v_j + n \sigma}{n \sigma p_j} \right) \log \frac{1}{\epsilon}$ .

$$\forall h_i \in \mathbb{R}^d : \mathbb{E} \left[ \left\| \sum_{i \in S} \mathbf{M}_i^{\frac{1}{2}} h_i \right\|^2 \right] \leq \sum_{i=1}^n p_i v_i \|h_i\|^2$$

Recovered rate from Qian et. al. (2019) and Defazio et. al. (2014)

# Saga with arbitrary sampling

$$\mathbb{E} \left[ \left\| \sum_{i \in S} \mathbf{M}_i^{\frac{1}{2}} h_i \right\|^2 \right] \leq \sum_{i=1}^n p_i v_i \|h_i\|^2 \quad \longleftarrow \quad \text{What is this??}$$



$$|S| = 1 \text{ w. p. 1} \longrightarrow v_i = \lambda_{\max} \mathbf{M}_i$$

$$|S| = n \text{ w. p. 1} \longrightarrow v_i = \lambda_{\max} \left( \sum_{i=1}^n \mathbf{M}_i \right)$$

$$\mathbf{P}_{i,j} = \text{Prob}(i \in S, j \in S) ee^\top; \mathbf{P} \in \mathbb{R}^{nd \times nd}$$

$$\mathbf{P} \circ \left( \left( \mathbf{M}_1^{\frac{1}{2}}, \dots, \mathbf{M}_n^{\frac{1}{2}} \right) \left( \mathbf{M}_1^{\frac{1}{2}}, \dots, \mathbf{M}_n^{\frac{1}{2}} \right)^\top \right) \preceq \text{Diag}(p \circ v) \otimes I$$

Same story as for coordinate descent

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

Use minibatch samplings for CD

# Saga with arbitrary sampling

**Corollary 13 (Convergence rate of SAGA)** Let  $\alpha = \min_j \frac{n\mathbf{p}_j}{4v_j + n\sigma}$ . Then the iteration complexity of Algorithm 3 is  $\max_j \left( \frac{4v_j + n\sigma}{n\sigma\mathbf{p}_j} \right) \log \frac{1}{\epsilon}$ .

$$\mathbb{E} \left[ \left\| \sum_{i \in S} \mathbf{M}_i^{\frac{1}{2}} h_i \right\|^2 \right] \leq \sum_{i=1}^n p_i v_i \|h_i\|^2$$

$$L_j = \lambda_{\max} \mathbf{M}_j$$

$$\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i$$

$f_j$  is  $L_j$  smooth; single element with  $p \propto L$   $\Rightarrow \frac{4\bar{L}}{\mu} + \max_j p_j^{-1}$

$f$  is  $L$  smooth; sample all w.p. 1  $\Rightarrow \frac{4L}{\mu}$

$$L = \lambda_{\max} \left( \frac{1}{n} \sum_{i=1}^n \mathbf{M}_i \right)$$

$$L \leq \bar{L} \leq nL$$

+ everything in between

# Loopless SVRG with arbitrary sampling

---

**Algorithm 8** LSVRG (LSVRG (Kovalev et al., 2019) with arbitrary sampling) [NEW METHOD]

---

**Require:** learning rate  $\alpha > 0$ , starting point  $x^0 \in \mathbb{R}^d$ , random sampling  $R \subseteq \{1, 2, \dots, n\}$

Set  $\phi = x^0$

**for**  $k = 0, 1, 2, \dots$  **do**

    Sample a random subset  $R^k \subseteq \{1, 2, \dots, n\}$

$$g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi^k) + \sum_{j \in R^k} \frac{1}{n p_j} (\nabla f_j(x^k) - \nabla f_j(\phi^k))$$

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

$$\text{Set } \phi^{k+1} = \begin{cases} x^k & \text{with probability } \rho \\ \phi^k & \text{with probability } 1 - \rho \end{cases}$$

**end for**

---

**Corollary 22 (Convergence rate of LSVRG)** Let  $\alpha = \min_j \frac{n}{4 \frac{v_j}{p_j} + \frac{\sigma n}{\rho}}$ . Then, iteration complexity of Algorithm 8 is  $\max_j \left( 4 \frac{v_j}{n \sigma p_j} + \frac{1}{\rho} \right) \log \frac{1}{\epsilon}$ .

$f_j$  is  $L_j$  smooth; single element with  $p \propto L \Rightarrow \frac{4\bar{L}}{\mu} + \rho^{-1}$

$$\mathbb{E} \left[ \left\| \sum_{i \in S} \mathbf{M}_i^{\frac{1}{2}} h_i \right\|^2 \right] \leq \sum_{i=1}^n p_i v_i \|h_i\|^2$$

New results!

# Shifted SGD with arbitrary sampling

If  $\mathbf{G}(\mathbf{x}^*)$  is known – no need for learning Jacobian

---

## Algorithm 7 SGD-shift [NEW METHOD]

---

**Require:** learning rate  $\alpha > 0$ , starting point  $x^0 \in \mathbb{R}^d$ , random sampling  $R \subseteq \{1, 2, \dots, n\}$

**for**  $k = 0, 1, 2, \dots$  **do**

    Sample random  $R^k \subseteq \{1, 2, \dots, n\}$

$$g^k = \frac{1}{n} \mathbf{G}(x^*) \textcolor{blue}{e} + \sum_{j \in R^k} \frac{1}{n \textcolor{blue}{p}_j} (\nabla f_j(x^k) - \nabla f_j(x^*))$$

$$x^{k+1} = \text{prox}_{\alpha \psi}(\textcolor{brown}{x}^k - \alpha g^k)$$

**end for**

---

**Corollary I.1** (Convergence rate of SGD-AS-shift). *Suppose that  $f_j$  is  $\mathbf{M}_j$ -smooth for all  $j$  and suppose that  $v$  satisfies (32). Let  $\alpha = n \min_j \frac{\textcolor{blue}{p}_j}{v_j}$ . Then, iteration complexity of Algorithm 7 is*

$$\max_j \left( \frac{v_j}{n \textcolor{blue}{p}_j \sigma} \right) \log \frac{1}{\epsilon}.$$

# SEGA with arbitrary sampling

---

**Algorithm 5** SEGA with arbitrary sampling

---

**Require:** Stepsize  $\alpha > 0$ , starting point  $x^0 \in \mathbb{R}^d$ , random sampling  $L \subseteq \{1, 2, \dots, d\}$

Set  $h^0 = 0$

**for**  $k = 0, 1, 2, \dots$  **do**

    Sample random  $L^k \subseteq \{1, 2, \dots, d\}$

    Set  $h^{k+1} = h^k + \sum_{i \in L^k} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$

$g^k = h^k + \sum_{i \in L^k} \frac{1}{p_i} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$

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

**end for**

---

**Corollary 16 (Convergence rate of SEGA)** *Iteration complexity of Algorithm 5 with  $\alpha = \min_i \frac{p_i}{4m_i + \sigma}$  is  $\max_i \left( \frac{4m_i + \sigma}{p_i \sigma} \right) \log \frac{1}{\epsilon}$ .*

$\mathbf{M} = \text{Diag}(m)$ .

Tighter rate than from original SEGA paper

# SVRCD with arbitrary sampling

---

**Algorithm 6** SVRCD [NEW METHOD]

**Require:** starting point  $x^0 \in \mathbb{R}^d$ , random sampling  $L \subseteq \{1, 2, \dots, d\}$ , probability  $\rho$ , stepsize  $\alpha > 0$

Set  $h^0 = 0$

**for**  $k = 0, 1, 2, \dots$  **do**

- Sample random  $L^k \subseteq \{1, 2, \dots, d\}$
- $$g^k = h^k + \sum_{i \in L^k} \frac{1}{p_i} (\nabla_i f(x^k) - h_i^k) e_i$$
- $$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$
- Set 
$$h^{k+1} = \begin{cases} h^k & \text{with probability } 1 - \rho \\ \nabla f(x^k) & \text{with probability } \rho \end{cases}$$

**end for**

---

**Corollary 18** Iteration complexity of Algorithm 6 with  $\alpha = \min_i \frac{1}{4m_i/p_i + \sigma/\rho}$  is  $\left(\frac{1}{\rho} + \max_i \frac{4m_i}{p_i\sigma}\right) \log \frac{1}{\epsilon}$

New Algorithm!

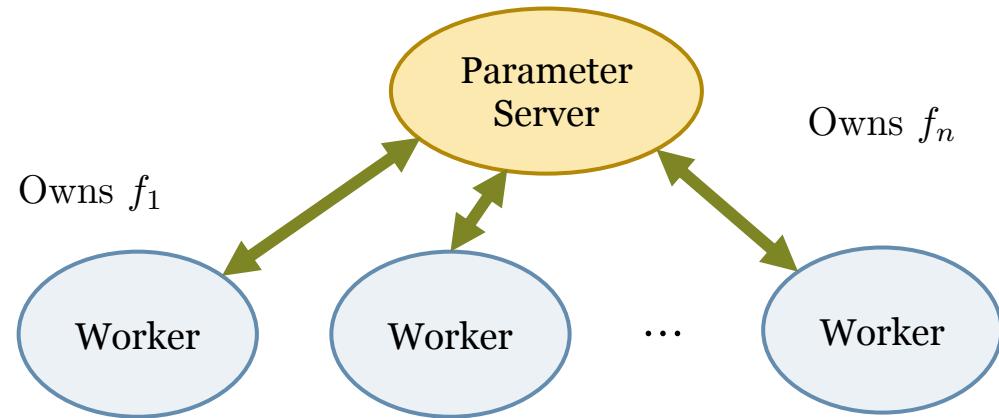
$\mathbf{M} = \text{Diag}(m)$ .

# ISAEGA

Holds for GD, SGD, SAGA, ASGD

$$\text{if } \forall i : \nabla f_i(x^*) = 0$$

Holds for GD with SEGA trick  
(no extra assumptions)



[Mishchenko, Hanzely and Richtarik, 2018]

All workers might throw away as much as  $\frac{n-1}{n}$  of total number of coordinates independently at random without hurting the convergence

Reduced communication to the server for no cost!      Constant factor at most

Natural Extension: Can we do SAGA and SEGA at the same time?

**YES! With arbitrary sampling**

# ISAEGA

**Algorithm 16 ISAEGA [NEW METHOD]**

**Input:**  $x^0 \in \mathbb{R}^d$ , # parallel units  $T$ , each owning set of indices  $N_t$  (for  $1 \leq t \leq T$ ), distributions  $\mathcal{D}_t$  over subsets of  $N_t$ , distributions  $\mathcal{D}_t$  over subsets coordinates  $[d]$ , stepsize  $\alpha$   
 $\mathbf{J}^0 = 0$

**for**  $k = 0, 1, \dots$  **do**

**for**  $t = 1, \dots, T$  in parallel **do**

Sample  $R_t \sim \mathcal{D}_t$ ;  $R_t \subseteq N_t$  (independently on each machine)

Sample  $L_t \sim \mathcal{D}_t$ ;  $L_t \subseteq [d]$  (independently on each machine)

Observe  $\nabla_{L_t} f_j(x^k)$  for  $j \in R_t$

For  $i \in [d], j \in N_t$  set  $\mathbf{J}_{i,j}^{k+1} = \begin{cases} \nabla_i f_j(x^k) & \text{if } i \in [d], j \in R_t, i \in L_t \\ \mathbf{J}_{i,j}^k & \text{otherwise} \end{cases}$

Send  $\mathbf{J}_{: \mathcal{N}_t}^{k+1} - \mathbf{J}_{: \mathcal{N}_t}^k$  to master

▷ Sparse; low communication

**end for**

$$g^k = \left( \mathbf{J}^k + \sum_{t=1}^T \left( \textcolor{red}{p}^{t-1} \textcolor{blue}{p}^{t-1}^\top \right) \circ \left( (\sum_{i \in L_t} \textcolor{red}{e}_i \textcolor{red}{e}_i^\top) (\mathbf{J}^{k+1} - \mathbf{J}^k)_{:\textcolor{blue}{N}_t} \left( \sum_{j \in R_t} \textcolor{blue}{e}_j \textcolor{blue}{e}_j^\top \right) \right) \right) e$$

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

**end for**



## **Other results**

# SAGA is a special case of SEGA

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^n f_j(x)$$

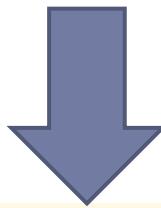
Equivalent problems

$$\min_{x' \in \mathbb{R}^{nd}} f'(x') + \psi'(x')$$

$$f'(x') = \frac{1}{n} \sum_{j=1}^n f_j(x_j)$$

$$\psi'(x') = I_{x_1=\dots=x_n}(x)$$

Indicator function of set  $x_1 = \dots = x_n$



Apply SAGA

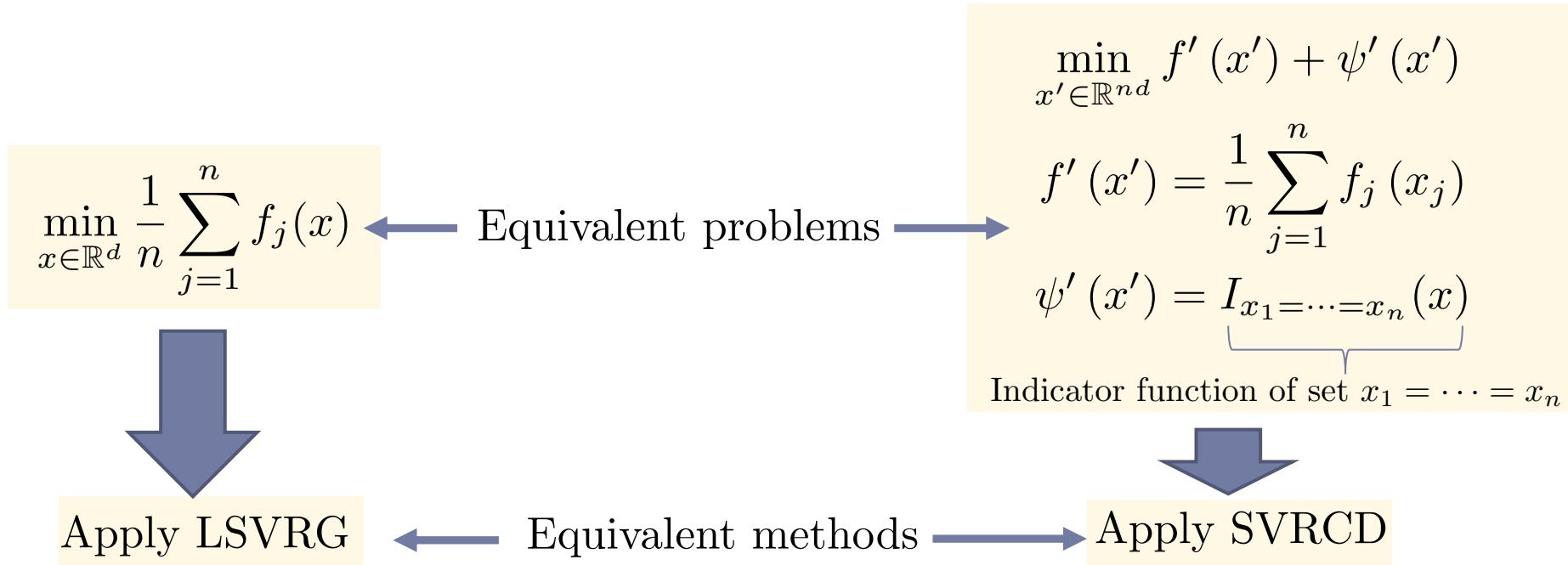
Equivalent methods



Apply SEGA

Theory of SEGA (with a twist) generalizes theory of SAGA

# SVRG is a special case of SVRCD



Theory of SVRCD (+ small twist) generalizes theory of LSVRG

Same result for Accelerated SVRCD and Katyusha

new