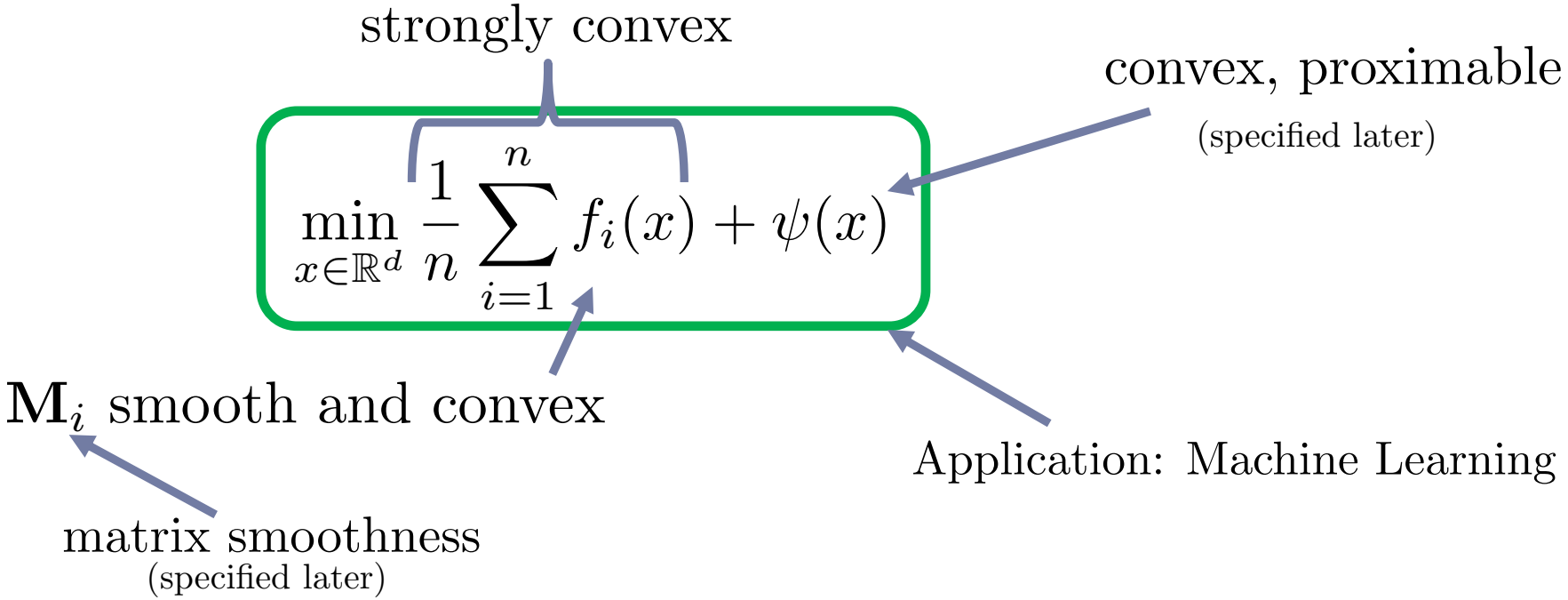
One Method Rules Them All: Variance Reduction for Data, Parameters and Many New Methods

Filip Hanzely, Peter Richtarik

Problem and Intoduction

Problem



Oracle: random linear measurments of (stochastic) gradient

(specified later)

Main result: Very general algorithm

Problem

strongly convex

convex, proximable

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

$\mathbf{M}_i$ smooth and convex

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

ϕ_i is L_i smooth

$$f_i(x) = \phi_i(A_i x) \quad \rightarrow \quad \mathbf{M}_i = L_i A_i^\top A_i$$

Gradient Descent, SGD and Variance reduction

let $\psi = 0$

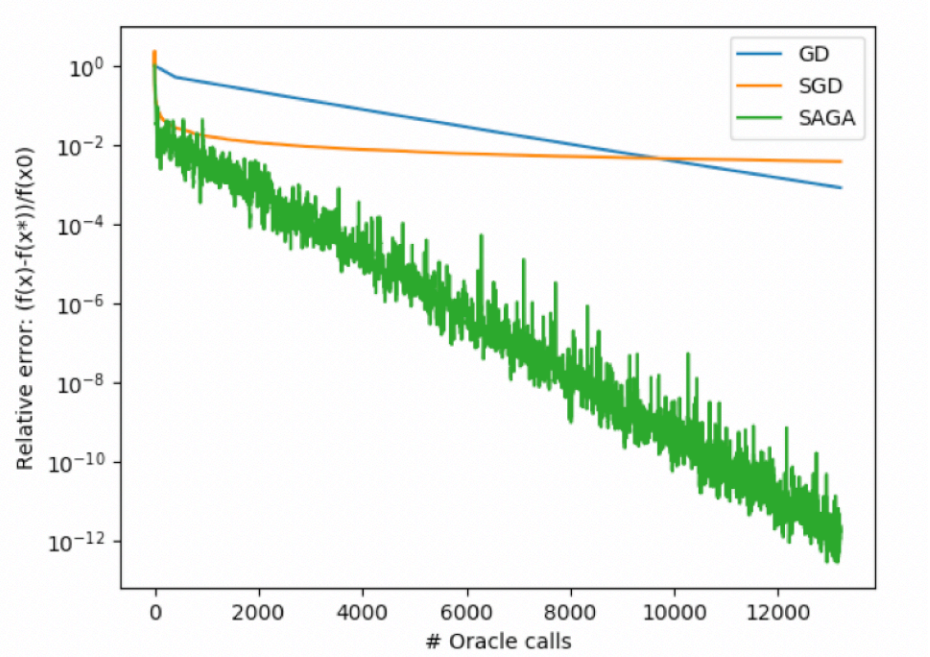
GD: $x^+ = x - \alpha \nabla f(x)$

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

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

Variance reduced stochastic gradient

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



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)

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

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

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

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

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

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 $j \in \{1, 2, \ldots, n\}$

for $k = 0, 1, 2, \ldots$ **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

(no finite sum)

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

convex, non-smooth, simple

Proximal operator of ψ is computable

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

If ψ is indicator function, then prox is projection
If ψ is L1 loss, then prox is soft tresholding

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

gradient step

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

Nearly gradient step, but considers ψ as well

Coordinate descent and proximal regularizer

CD: Compute partial derivative, take step

Example

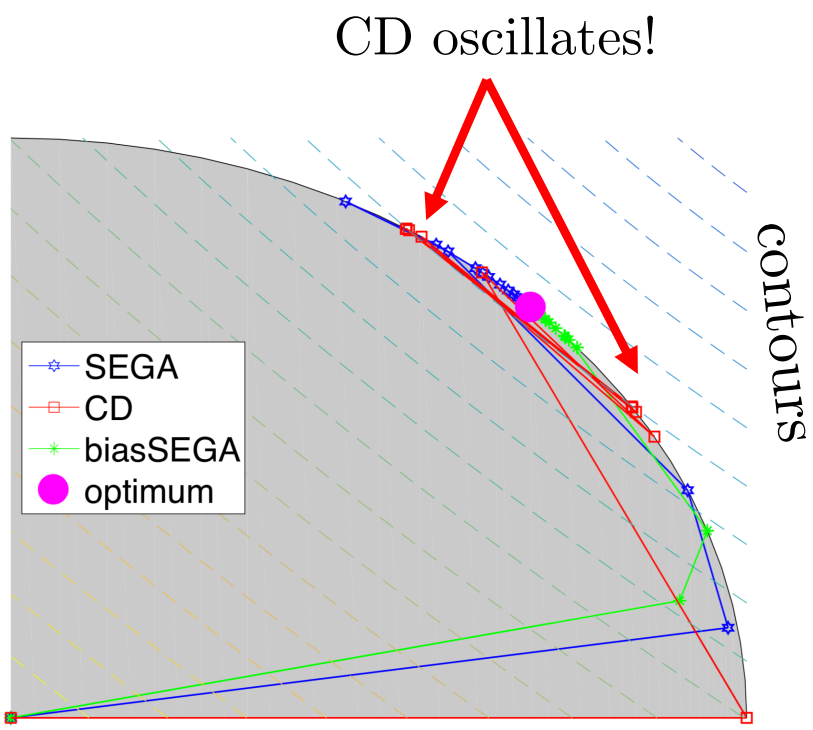
$$\text{minimize } f(x) + \psi(x)$$

$$f(x) = ||x - z||^2$$

$\|z\|$ 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

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

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

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

Goal: High $\text{Cov}\left(Y^k, \nabla_j f\left(x^k\right)\right)$ (to decrease variance of g^k)

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

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

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 $k = 0, 1, 2, \ldots$ **do**

Sample $i \in \{1, \ldots d\}$ 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 $k = 0, 1, 2, \ldots$ **do**

Sample $i \in \{1, \ldots d\}$ 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

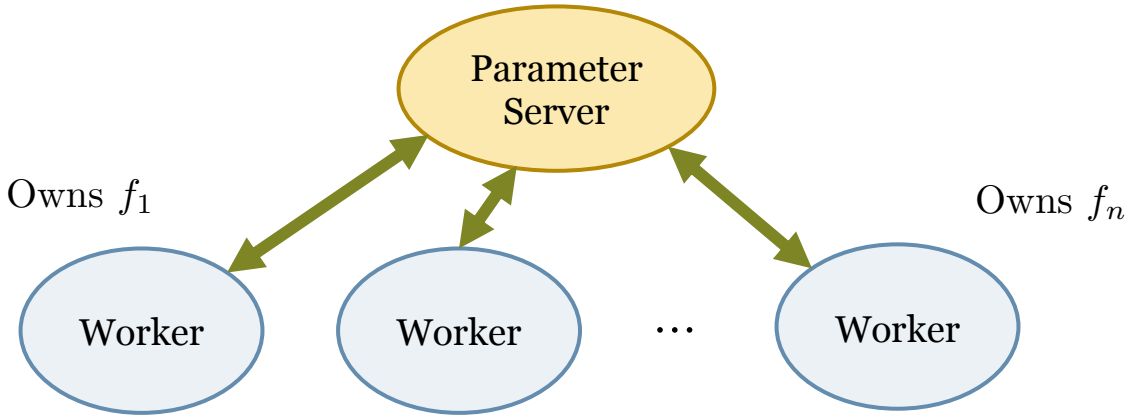
$\psi = 0$: → Up to constant factor rate of coordinate descent

→ 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

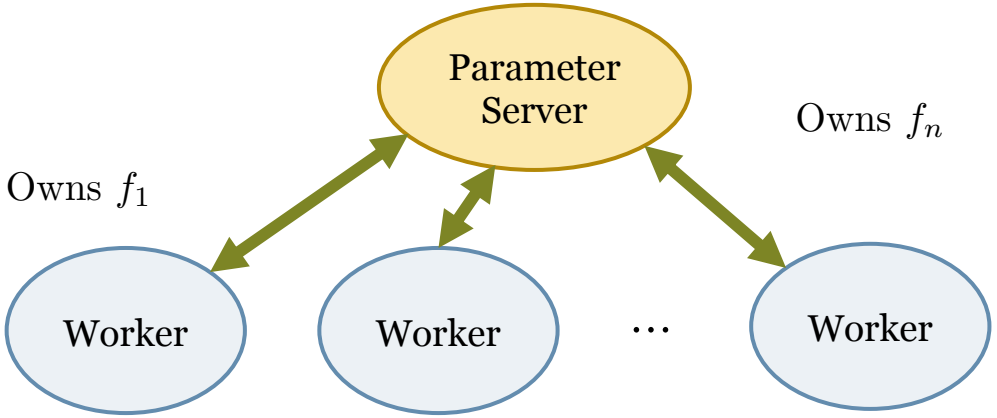
Server: Average updates and send them back to workers

Bottleneck: Communication from workers to server

Parallel Coordinate Descent

Holds for GD, SGD, SAGA, ASGD
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

Proximal methods

High level idea

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

Jacobian

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

random linear operator
(fixed distribution)

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

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

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)

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

random linear operator

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

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

2) Construct stochastic gradient using control variates

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

vector of ones

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

$\mathcal{S}$,$\mathcal{U}$ 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`)

Parameters: Stepsize $\alpha > 0$, random projector $\mathcal{S}$ and unbiased sketch $\mathcal{U}$

Initialization: Choose solution estimate $x^0 \in \mathbb{R}^d$ and Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d\times n}$

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

  Sample realizations of $\mathcal{S}$ and $\mathcal{U}$, and perform sketches $\mathcal{S}\mathbf{G}(x^k)$ and $\mathcal{U}\mathbf{G}(x^k)$

  $\mathbf{J}^{k+1} = \mathbf{J}^k - \mathcal{S}(\mathbf{J}^k - \mathbf{G}(x^k))$ update the Jacobian estimate

  $g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}\mathcal{U}\left(\mathbf{G}(x^k) - \mathbf{J}^k\right) e$ construct the gradient estimator

  $x^{k+1} = \mathrm{prox}_{\alpha\psi}(x^k - \alpha g^k)$ perform the proximal `SGD` step

end for

Convergence

Main theorem:

$$\mathcal{M}^{\dagger\frac{1}{2}}:\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n};\mathcal{M}^{\dagger\frac{1}{2}}(\mathbf{X})=\left[\mathbf{M}^{\dagger\frac{1}{2}}_{1}(\mathbf{X}_{:1}),\mathbf{M}^{\dagger\frac{1}{2}}_{2}(\mathbf{X}_{:2}),\cdots,\mathbf{M}^{\dagger\frac{1}{2}}_{n}(\mathbf{X}_{:n})\right]$$

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

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

Satisfied for small α

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

Faster for larger α

strong convexity

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

Maximize α 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 $R \subseteq \{1, 2, \ldots, n\}$

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

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

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

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

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

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

end for

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

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

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

Saga with arbitrary sampling

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

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

$L_j = \lambda_{\max}\mathbf{M_j}$

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

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. $1 \Rightarrow \frac{4L}{\mu}$

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

+ 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 $R \subseteq \{1, 2, \ldots, n\}$

Set $\phi = x^0$

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

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

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

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

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

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 $p \propto L \Rightarrow \dfrac{4\bar{L}}{\mu} + p_0^{-1}$

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

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 $R \subseteq \{1, 2, \ldots, n\}$

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

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

$$g^k = \frac{1}{n}\mathbf{G}(x^*)e + \sum_{j \in R^k} \frac{1}{np_j}\left(\nabla f_j(x^k) - \nabla f_j(x^*)\right)$$

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

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 $\alpha = n \min_j \frac{p_j}{v_j}$. Then, iteration complexity of Algorithm 7 is* $\max_j \left(\frac{v_j}{np_j\sigma}\right) \log \frac{1}{\epsilon}$.

SEGA with arbitrary sampling

Algorithm 5 SEGA with arbitrary sampling

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

Set $h^0 = 0$

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

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

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

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

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

end for

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

f is Diag(m) smooth

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 ρ, stepsize $\alpha > 0$

Set $h^0 = 0$

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

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

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

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

$$\text{Set } h^{k+1} = \begin{cases} h^k & \text{with probability } 1-\rho \\ \nabla f(x^k) & \text{with probability } \rho \end{cases}$$

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!

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

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

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

$$x' = [x_1^\top, \ldots, x_n^\top]^\top$$

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

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

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

Apply SAGA

Equivalent methods

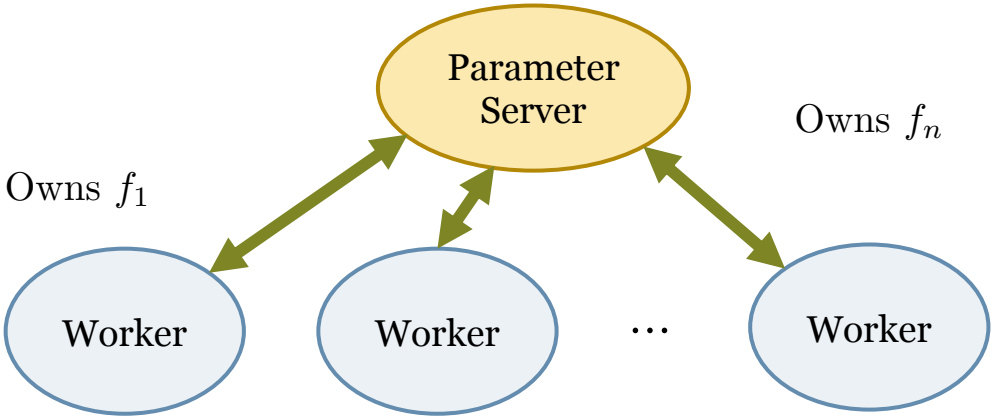
Apply SEGA

Theory of SEGA (+ small twist) generalizes theory of SAGA

ISAEGA

Holds for GD, SGD, SAGA, ASGD
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?

YES! With arbitrary sampling