

OPERATOR SPLITTING METHODS IN DATA ANALYSIS

Communication Efficient SGD for Distributed Optimization

Sebastian U. Stich

MLO Lab, EPFL, Switzerland
`sebastian.stich@epfl.ch`

March 20, 2019

Optimization Problem:

machine learning

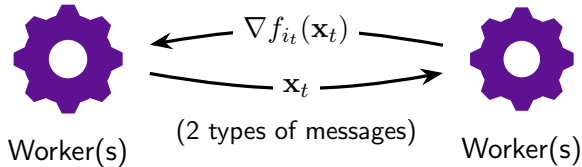
$$f^* := \min_{\mathbf{x} \in \mathbb{R}^d} \mathbb{E}_{\xi} F(\mathbf{x}, \xi)$$

Goal/Setting: Make best use of available computing devices!

Special Case:

finite support

$$f_n^* := \min_{\mathbf{x} \in \mathbb{R}^d} \left[f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) \right]$$



Motivation:

Communication between machines is much slower than computation.

Goal:

Communicate as little as possible!

Part 0 (baselines):

Distributed SGD

Part I (cool stuff):

SGD with Error Feedback & Local SGD

Part II:

Open Problems

Stochastic Gradient Descent (SGD)

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_{i_t}(\mathbf{x}_t)$$

$f: \mathbb{R}^d \rightarrow \mathbb{R}$ *L-smooth* (possibly non-convex)

Special case: μ -strongly convex and L -smooth.

$$\kappa := \frac{L}{\mu}$$

f strongly convex, $\mathbb{E} \|\nabla f_{i_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2 \leq \sigma^2$ (Assumption)

Stochastic Gradient Descent (SGD):

$$f(\bar{\mathbf{x}}_T) - f^* = O\left(\frac{\sigma^2}{T}\right)$$

This rate is optimal, in the sense $f_n^* - f^* = O\left(\frac{1}{n}\right)$.

f strongly convex, $\mathbb{E} \|\nabla f_{i_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2 \leq \sigma^2$ (Assumption)

Stochastic Gradient Descent (SGD):

$$f(\bar{\mathbf{x}}_T) - f^* = O\left(\frac{\sigma^2}{T}\right)$$

This rate is optimal, in the sense $f_n^* - f^* = O\left(\frac{1}{n}\right)$.

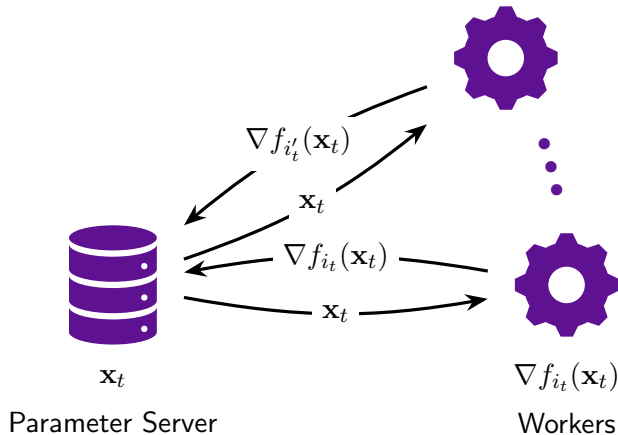
Mini-batch SGD:

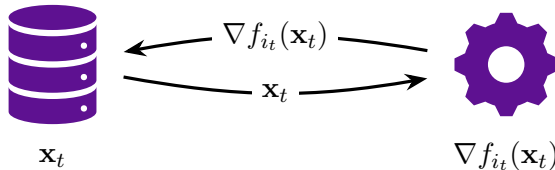
$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{W} \sum_{w=1}^W \nabla f_{i_t^w}(\mathbf{x}_t)$$

$$f(\bar{\mathbf{x}}_T) - f^* = O\left(\frac{\sigma^2}{WT}\right)$$

Can evaluate W gradients in parallel. (Assumes $W \ll T$).

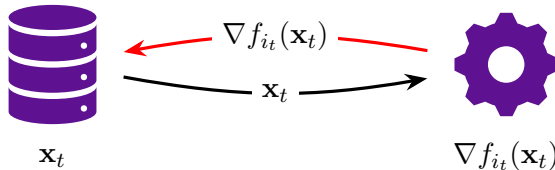
Distributed SGD



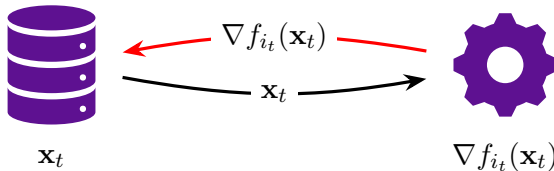


- Assume 1 Worker, 1 Parameter Server

for now



- Assume 1 Worker, 1 Parameter Server for now
- Each iteration requires communication of $\nabla f_{i_t}(\mathbf{x}_t) \in \mathbb{R}^d$.



- Assume 1 Worker, 1 Parameter Server for now
- Each iteration requires communication of $\nabla f_{i_t}(\mathbf{x}_t) \in \mathbb{R}^d$.

Can we do better?



Quantization

Quantization:

see e.g. [AGL⁺ 16] QSGD

- Idea: send sparse vector $Q(\nabla f_{i_t}(\mathbf{x}_t))$ instead of $\nabla f_{i_t}(\mathbf{x}_t)$
- Problem: Proof requires $\mathbb{E}Q(\nabla f_{i_t}(\mathbf{x}_t)) = \nabla f(\mathbf{x})$

Example 1: Random Sparsification

$$Q(\mathbf{x}) = \frac{d}{k} \text{rand}_k(\mathbf{x}), \quad \text{rand}_k(\mathbf{x}) = \begin{cases} \mathbf{x}_i & \text{with prob. } \frac{k}{d} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{E} \|Q(\mathbf{x})\|^2 \leq \frac{d}{k} \|\mathbf{x}\|^2, \text{ sparsity } \mathbb{E} \|Q(\mathbf{x})\|_0 = k$$

Example 1: Random Sparsification

$$Q(\mathbf{x}) = \frac{d}{k} \text{rand}_k(\mathbf{x}), \quad \text{rand}_k(\mathbf{x}) = \begin{cases} \mathbf{x}_i & \text{with prob. } \frac{k}{d} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{E} \|Q(\mathbf{x})\|^2 \leq \frac{d}{k} \|\mathbf{x}\|^2, \text{ sparsity } \mathbb{E} \|Q(\mathbf{x})\|_0 = k$$

Example 2: Random Quantization [WXY⁺ 16] [AGL⁺ 16]

$$Q(\mathbf{x}) = \|\mathbf{x}\| \cdot \text{"round to } s \text{ bits"} \left(\frac{\mathbf{x}}{\|\mathbf{x}\|} \right)$$

$$\mathbb{E} \|Q(\mathbf{x})\|^2 \leq \frac{\sqrt{d}}{s} \|\mathbf{x}\|^2, \text{ sparsity } \mathbb{E} \|Q(\mathbf{x})\|_0 \leq s(s + \sqrt{d})$$

Quantization:see e.g. [AGL⁺ 16] QSGD

- Idea: send sparse vector $Q(\nabla f_{i_t}(\mathbf{x}_t))$ instead of $\nabla f_{i_t}(\mathbf{x}_t)$
- Problem: Proof requires $\mathbb{E}Q(\nabla f_{i_t}(\mathbf{x}_t)) = \nabla f(\mathbf{x})$

Examples:

	$Q(\mathbf{y})$	sparsity	σ^2
rand- k	$\frac{d}{k} \text{rand}_k(\mathbf{y})$	k	$\leq \frac{d}{k} \sigma^2$
round to s bits	$\ \mathbf{y}\ \text{round}_s\left(\frac{\mathbf{y}}{\ \mathbf{y}\ }\right)$	$s(s + \sqrt{d})$	$\leq \frac{\sqrt{d}}{s} \sigma^2$

Quantization:see e.g. [AGL⁺ 16] QSGD

- Idea: send sparse vector $Q(\nabla f_{i_t}(\mathbf{x}_t))$ instead of $\nabla f_{i_t}(\mathbf{x}_t)$
- Problem: Proof requires $\mathbb{E}Q(\nabla f_{i_t}(\mathbf{x}_t)) = \nabla f(\mathbf{x})$

Examples:

	$Q(\mathbf{y})$	sparsity	σ^2
rand- k	$\frac{d}{k} \text{rand}_k(\mathbf{y})$	k	$\leq \frac{d}{k} \sigma^2$
round to s bits	$\ \mathbf{y}\ \text{round}_s\left(\frac{\mathbf{y}}{\ \mathbf{y}\ }\right)$	$s(s + \sqrt{d})$	$\leq \frac{\sqrt{d}}{s} \sigma^2$

Consequence: SGD convergence rate

$$\begin{array}{cc} \text{rand-}k & \text{round-}1 \\ O\left(\frac{d/k \cdot \sigma^2}{T}\right) & O\left(\frac{\sqrt{d} \cdot \sigma^2}{T}\right) \end{array}$$

Quantization:see e.g. [AGL⁺ 16] QSGD

- Idea: send sparse vector $Q(\nabla f_{i_t}(\mathbf{x}_t))$ instead of $\nabla f_{i_t}(\mathbf{x}_t)$
- Problem: Proof requires $\mathbb{E}Q(\nabla f_{i_t}(\mathbf{x}_t)) = \nabla f(\mathbf{x})$

Examples:

	$Q(\mathbf{y})$	sparsity	σ^2
rand- k	$\frac{d}{k} \text{rand}_k(\mathbf{y})$	k	$\leq \frac{d}{k} \sigma^2$
round to s bits	$\ \mathbf{y}\ \text{round}_s\left(\frac{\mathbf{y}}{\ \mathbf{y}\ }\right)$	$s(s + \sqrt{d})$	$\leq \frac{\sqrt{d}}{s} \sigma^2$

Consequence: SGD convergence rate

$$\begin{array}{ccc} \text{rand-}k & \text{round-}1 & \text{round-}\sqrt{d} \\ O\left(\frac{d/k \cdot \sigma^2}{T}\right) & O\left(\frac{\sqrt{d} \cdot \sigma^2}{T}\right) & \underbrace{O\left(\frac{\sigma^2}{T}\right)}_{\text{Not sparse}} \end{array}$$

Quantization:see e.g. [AGL⁺ 16] QSGD

- Idea: send sparse vector $Q(\nabla f_{i_t}(\mathbf{x}_t))$ instead of $\nabla f_{i_t}(\mathbf{x}_t)$
- Problem: Proof requires $\mathbb{E}Q(\nabla f_{i_t}(\mathbf{x}_t)) = \nabla f(\mathbf{x})$

Examples:

	$Q(\mathbf{y})$	sparsity	σ^2
rand- k	$\frac{d}{k} \text{rand}_k(\mathbf{y})$	k	$\leq \frac{d}{k} \sigma^2$
round to s bits	$\ \mathbf{y}\ \text{round}_s(\frac{\mathbf{y}}{\ \mathbf{y}\ })$	$s(s + \sqrt{d})$	$\leq \frac{\sqrt{d}}{s} \sigma^2$

Consequence: SGD convergence rate

rand- k	round-1	round- $\sqrt{d}$	rand- k /top- k
$O\left(\frac{d/k \cdot \sigma^2}{T}\right)$	$O\left(\frac{\sqrt{d} \cdot \sigma^2}{T}\right)$	$O\left(\frac{\sigma^2}{T}\right)$	$O\left(\frac{\sigma^2}{T} + \frac{(d/k)^2}{T^2}\right)$
		Not sparse	We can fix this!

Error Feedback

1 Use **memory** $\mathbf{m}_t \in \mathbb{R}^d$ to keep track of the errors:

$$\begin{aligned}\mathbf{q}_t &= Q(\mathbf{m}_t + \gamma_t \nabla f_{i_t}(\mathbf{x}_t)) \\ \mathbf{m}_{t+1} &= \mathbf{m}_t + \gamma_t \nabla f_{i_t}(\mathbf{x}_t) - \mathbf{q}_t \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \mathbf{q}_t\end{aligned}\quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} \text{worker} \\ \\ \text{server} \end{array}$$

1 Use **memory** $\mathbf{m}_t \in \mathbb{R}^d$ to keep track of the errors:

$$\begin{aligned} \mathbf{q}_t &= Q(\mathbf{m}_t + \gamma_t \nabla f_{i_t}(\mathbf{x}_t)) \\ \mathbf{m}_{t+1} &= \mathbf{m}_t + \gamma_t \nabla f_{i_t}(\mathbf{x}_t) - \mathbf{q}_t \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \mathbf{q}_t \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} \text{worker} \\ \\ \text{server} \end{array}$$

2 Quantization operator does not need to be unbiased.

Assumption:

$$\mathbb{E} \|\mathbf{y} - Q(\mathbf{y})\|^2 \leq (1 - \delta) \|\mathbf{y}\|^2 \quad \forall \mathbf{y} \in \mathbb{R}^d \quad (*)$$

Examples:

+ block-variants

rand- k , top- k , (scaled) round $_s$, and combinations thereof

Theorem 1 [SCJ]: For Q as in (*), $\gamma_t = \frac{8}{\mu(a+t)}$, $a = 5\frac{d}{k}$:

$$\mathbb{E}f(\bar{\bar{\mathbf{x}}}_T) - f^* = O\left(\frac{G^2}{\mu T} + \frac{\delta^2 \kappa G^2}{\mu T^2}\right)$$

with $\bar{\bar{\mathbf{x}}}_T = \frac{1}{\sum_{t=0}^{T-1} w_t} \sum_{i=0}^{T-1} w_t \mathbf{x}_t$, $w_t = (a+t)^2$, $\kappa = \frac{L}{\mu}$.

Theorem 2 [KRSJ]: For Q as in (*), $\gamma = \frac{1}{\sqrt{T+1}}$,

$$\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\mathbf{x}_t)\|^2 = O\left(\frac{f(\mathbf{x}_0) - f^* + LG^2}{\sqrt{T+1}} + \frac{\delta^2 L^2 G^2}{T+1}\right)$$

$$\mathbb{E} \|\nabla f_{i_t}(\mathbf{x}_t)\|^2 \leq G^2$$

(Assumption)

Proof (Sketch):

Main idea: Study a ‘virtual’ sequence

$$\tilde{\mathbf{x}}_{t+1} := \tilde{\mathbf{x}}_t - \gamma_t \nabla f_{i_t}(\mathbf{x}_t)$$

Observe

$$\mathbf{m}_t = \mathbf{x}_t - \tilde{\mathbf{x}}_t \stackrel{(\text{def})}{=} \sum_{j=0}^{t-1} \gamma_j \nabla f_{i_j}(\mathbf{x}_j) - \sum_{j=0}^{t-1} \mathbf{q}_j$$

Key Lemma: $\mathbb{E} \|\mathbf{m}_t\|^2 \leq 20 \gamma_t^2 \frac{d^2}{k^2} G^2$

using (*)

Perturbed SGD: $\mathbf{x}_t$ behaves as normal SGD

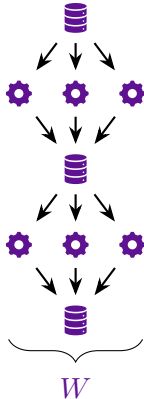


Goal: Reduce communication even more!

$$|\text{communication}| = \frac{|\text{total computation}|}{|\text{local task}|} \times \underbrace{|\text{message}|}_{\text{Strategy I}}$$



Strategy II: Increase $|\text{local task}|$!

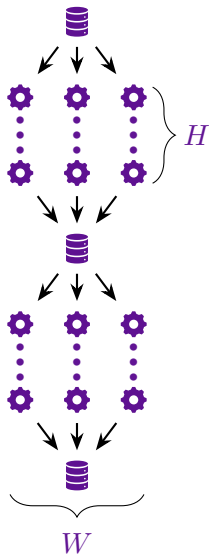


Mini-batch SGD:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma_t}{W} \sum_{w=1}^W \nabla f_{i_t^w}(\mathbf{x}_t)$$

- compute W gradients in parallel
- at $\mathbf{x}_t$
- communication every round

Local SGD



Local SGD:

$$\mathbf{x}_{t+1}^w := \begin{cases} \mathbf{x}_t^w - \gamma_t \nabla f_{i_t^w}(\mathbf{x}_t^w), & \text{if } H \nmid t+1 \\ \frac{1}{W} \sum_{w=1}^W (\mathbf{x}_t^w - \gamma_t \nabla f_{i_t^w}(\mathbf{x}_t^w)) & \text{if } H \mid t+1 \end{cases}$$

- W runs of SGD in parallel
- for different $\mathbf{x}_t^w$
- communication every H rounds

Local SGD

			rate	comm.
$H = 1$	mini-batch SGD	$O\left(\frac{\sigma^2}{WT}\right)$	+	-
$H = T$	'one-shot averaging'	$O\left(\frac{\sigma^2}{T}\right)$	-	+
$H = O(\sqrt{T/W})$		$O\left(\frac{\sigma^2}{WT}\right)$	+	+

Theorem 3 [S]:

$$\mathbb{E}f(\bar{\mathbf{x}}_T) - f^* = O\left(\frac{\sigma^2}{\mu TW}\right) \quad \text{for } T = \Omega(H^2W)$$

$$\text{with } \bar{\mathbf{x}}_T := \frac{1}{W \sum_{t=0}^{T-1} w_t} \sum_{w=1}^W \sum_{t=0}^{T-1} w_t \mathbf{x}_t^w.$$

Proof (Sketch): Study 'virtual sequence'

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \frac{\eta_t}{W} \sum_{w=1}^W \nabla f_{i_t^w}(\mathbf{x}_t^w)$$

that behaves as mini-batch SGD. □

Discussion

- Only for small H , first step towards better understanding.
- Improve (in communication) over mini-batch SGD with batch size W .
- However, if we compare the our bound to mini-batch SGD with the same communication pattern as local SGD (batch size HW), our bounds are worse!

Counter-intuitive: assume $H \rightarrow \infty$

- local SGD converges to $\mathbf{x}^*$ (consider one local sequence)
- mini-batch does 'one step of GD', computed at $\mathbf{x}_0$

Open Problems

Goal: Reduce communication even more!

$$|\text{communication}| = \frac{|\text{total computation}|}{|\text{local task}|} \times \underbrace{|\text{message}|}_{\text{Strategy I}}$$



Strategy II: Increase $|\text{local task}|$!



Strategy III: Increase $|\text{local task}|$!

$$\min \left[f(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}) \right]$$



$$\min \left[f_1(\mathbf{x}) := \sum_{i=1}^{n/2} f_i(\mathbf{x}) \right]$$

$$\min \left[f_2(\mathbf{x}) := \sum_{i=n/2+1}^n f_i(\mathbf{x}) \right]$$

$\Downarrow$ (any algorithm we like) $\Downarrow$

$$(\mathbf{x}_1^*, f_1^*, \dots)$$

$$(\mathbf{x}_2^*, f_2^*, \dots)$$



Can we get $\mathbf{x}^*$?

Given: optimization problem $\mathcal{F}(\mathbf{x}, \mathcal{I})$

$$\min \sum_{i=1}^n f_i(\mathbf{x})$$

Question: Can we design subproblems

$$\mathcal{F}_1(\mathbf{x}, \mathcal{I}_1), \dots, \mathcal{F}_W(\mathbf{x}, \mathcal{I}_W) \quad \bigcup_{w=1}^W \mathcal{I}_w = \mathcal{I}, \quad \mathcal{I}_i \cap \mathcal{I}_j = \emptyset, i \neq j$$

that are easy to solve in parallel

$$\min \sum_{i \in \mathcal{I}_w} f_i(\mathbf{x}) + \mathcal{R}_{\mathcal{I}_w}(\mathbf{x})$$

$$|(\mathbf{x}_w^*, \mathbf{m}_w) = \mathcal{F}_w(\mathbf{x})| = O(d)$$

and that allow **cheap** approximation of $\mathbf{x}^*$

$$\mathbf{x}^* \approx \frac{1}{W} \sum_{w=1}^W \mathbf{x}_w^*$$

$$\mathbf{x}^* \approx \mathcal{A}(\mathbf{x}_1^*, \mathbf{m}_1, \dots, \mathbf{x}_W^*, \mathbf{m}_W)$$

Applications:

- $\mathcal{F}(\mathbf{x}, \mathcal{I} \cup \{i\})$
- $\mathcal{F}(\mathbf{x}, \mathcal{I} \setminus \{i\})$

Distributed/Federated Optimization

Privacy/Opt-out

Q I: $\mathcal{F}(\mathbf{x}, \mathcal{I} \cup \{i\} \setminus \cup\{j\}) \stackrel{?}{=} \mathcal{F}(\mathbf{x}, \mathcal{I}) + \mathcal{F}(\mathbf{x}, \{i\}) - \mathcal{F}(\mathbf{x}, \{j\})$

Q II: The proofs of **mem SGD** and **local SGD** used that

$$\{\text{mem, local}\} \text{ SGD} = \text{SGD} + o\left(\frac{1}{T}\right)$$

(similarly for **momentum SGD**). But this technique will never allow to **surpass SGD**. Is there something better we can do?

Q III: Does **local SGD** (not) beat **mini-batch SGD**?

OPERATOR SPLITTING METHODS IN DATA ANALYSIS

Communication Efficient SGD for Distributed Optimization

Sebastian U. Stich

MLO Lab, EPFL, Switzerland
`sebastian.stich@epfl.ch`

March 20, 2019

References

- SCJ 18 S.U. Stich, J-B. Cordonnier, M. Jaggi. Sparsified SGD with Memory, NeurIPS 2018.
- S 18 S.U. Stich. Local SGD Converges Fast and Communicates Little, 2018.
- KQSJ 19 S.P. Karimireddy, Q. Rebjock, S.U. Stich, M. Jaggi. Error Feedback Fixes SignSGD and other Gradient Compression Schemes, manuscript, 2019.
- KSJ 19 A. Koloskova, S.U. Stich, M. Jaggi. Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication, manuscript, 2019.