A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik
https://richtarik.org

ICCOPT 2019 Summer School

August 3, 2019

Contents I

Contents II

Contents III

Contents IV

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 1: Introduction

What is This Course About?

- A short introductory course (4×90 mins)
- Focus on the problem

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

  where f has a complicated structure related to **training supervised machine learning models.**
- **We will cover a bit deeply** rather than a lot superficially
  - Should feel like a normal PhD level course
  - This is not a tutorial; we'll do proofs
  - Slow pace
  - **I expect interruptions/questions!**
  - There are no stupid questions; except for those you do not ask!
- Prepares you for **further study and research**
- **Slides are on** `https://richtarik.org`

Key Principles Covered

We will focus on **understanding selected key principles** related to stochastic gradient descent (SGD)

- convergence
- importance sampling
- minibatching
- variance reduction
- compression/quantization

using a **novel unified view** [5] which arose as a synthesis of a large body of research over the last few years

What is Excluded?

- Non-convex problems (e.g., neural networks)
- Non-smooth problems
- Acceleration / momentum (i.e., optimal methods)
- Higher order and zero-order methods
- Non-Euclidean structure (e.g., methods based on Bregman divergence)
- Dual problem and dual methods

Structure of Optimization Problems Arising in Training Supervised Machine Learning Models

Optimization Problems Arising in Machine Learning

In this course, we are interested in the optimization problem

$$\min_{x \in \mathbb{R}^d} f(x) + R(x). \tag{1}$$

Typical structure of f:

- **Infinite sum**

$$f(x) = \mathbf{E}_{\xi \sim \mathcal{D}} \left[f_\xi(x) \right], \tag{2}$$

- **Finite sum:**

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{3}$$

- **Finite Sum of Finite Sums:**

$$f_i(x) = \frac{1}{m} \sum_{j=1}^{m} f_{ij}(x), \tag{4}$$

These problems are of key importance in **supervised learning theory and practice.**

Common feature: It is prohibitively expensive to compute the gradient of f, while an unbiased estimator of the gradient can be computed efficiently/cheaply.

Stochastic Optimization and Machine Learning

In the stochastic optimization problem

$$f(x) = \mathbf{E}_{\xi \sim \mathcal{D}} \left[f_\xi(x) \right]$$

- x represents a **machine learning model** described by d parameters/features (e.g., logistic regression or a deep neural network),
- $\mathcal{D}$ is an unknown **distribution of labelled examples**,
- $f_\xi(x)$ represents the **loss** of model x on data point ξ, and
- f is the **generalization error.**

Problem (1) seeks to find the model x minimizing the generalization error.

- In statistical learning theory one assumes that while $\mathcal{D}$ is not known, samples $\xi \sim \mathcal{D}$ are available.
- In such a case, $\nabla f(x)$ is not computable, while $\nabla f_\xi(x)$, which is an unbiased estimator of the gradient of f at x, is easily computable.

Finite Sum Problems

In this course we will focus on functions f which arise as averages of a very large number of (smooth) functions:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x).$$

- This problem often arises by approximation of the stochastic optimization loss function (2) via **Monte Carlo integration.**
- Known as the **empirical risk minimization (ERM)** problem.
- ERM is currently the **dominant paradigm for solving supervised learning problems** [25].
- If index i is chosen uniformly at random from $[n] \overset{\text{def}}{=} \{1, 2, \ldots, n\}$, $\nabla f_i(x)$ is an **unbiased estimator of** $\nabla f(x)$**.**
- Typically, $\nabla f_i(x)$ **is about** n **times less expensive** to compute than $\nabla f(x)$.

Distributed Training

In distributed training of supervised models, one considers the finite sum problem (3), with n being the number of machines, and each f_i

- also having a **finite sum structure,** i.e.,

$$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x), \tag{5}$$

  where m corresponds to the number of training examples stored on machine i.
- or an **infinite-sum structure,** i.e.,

$$f_i(x) = \mathbf{E}_{\xi_i \sim \mathcal{D}_i}\left[f_{i\xi_i}(x)\right], \tag{6}$$

  where $\mathcal{D}_i$ is the distribution of data stored on machine i.

Stochastic Gradient Descent (SGD)

SGD

Stochastic gradient descent (SGD) [23, 18, 27] is a state-of-the-art algorithmic paradigm for solving optimization problems (1) in situations when f is either of structure (2) or (3).

In its generic form, (proximal) SGD defines the new iterate by subtracting a multiple of a stochastic gradient from the current iterate, and subsequently applying the proximal operator of R:

$$\boxed{x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)} \tag{7}$$

- g^k is an **unbiased estimator of the gradient** (i.e., a "stochastic gradient"):

$$\mathbf{E}\left[g^k \mid x^k\right] = \nabla f(x^k). \tag{8}$$

- $$\mathrm{prox}_R(x) \stackrel{\text{def}}{=} \operatorname*{argmin}_u \left\{ R(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

The Prox Operator

Some facts about the prox operator[1]:

- **single-valuedness:** $x \mapsto \mathrm{prox}_R(x)$ is a function
- **non-expansiveness:**

$$\|\mathrm{prox}_R(x) - \mathrm{prox}_R(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^d$$

- **Moreau decomposition:**

$$\mathrm{prox}_R(x) + \mathrm{prox}_{R^*}(x) = x, \quad \forall x \in \mathbb{R}^d$$

Here R^* is the **Fenchel conjugate**[2] of R.

[1] Assume $R : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is proper, closed and convex.

[2] $R^*(x) \stackrel{\text{def}}{=} \sup_{y \in \mathbb{R}^d} \{\langle x, y \rangle - R(y)\}$

Prox Calculus (Beck's 2017 book [3])

$f(\mathbf{x})$	$\mathrm{prox}_f(\mathbf{x})$	Assumptions	Reference
$\sum_{i=1}^m f_i(\mathbf{x}_i)$	$\mathrm{prox}_{f_1}(\mathbf{x}_1) \times \cdots \times \mathrm{prox}_{f_m}(\mathbf{x}_m)$		Theorem 6.6
$g(\lambda \mathbf{x} + \mathbf{a})$	$\frac{1}{\lambda}\left[\mathrm{prox}_{\lambda^2 g}(\lambda \mathbf{x} + \mathbf{a}) - \mathbf{a}\right]$	$\lambda \neq 0, \mathbf{a} \in \mathbb{E}$, g proper	Theorem 6.11
$\lambda g(\mathbf{x}/\lambda)$	$\lambda \mathrm{prox}_{g/\lambda}(\mathbf{x}/\lambda)$	$\lambda \neq 0$, g proper	Theorem 6.12
$g(\mathbf{x}) + \frac{c}{2}\|\mathbf{x}\|^2 + \langle \mathbf{a}, \mathbf{x} \rangle + \gamma$	$\mathrm{prox}_{\frac{1}{c+1} g}(\frac{\mathbf{x}-\mathbf{a}}{c+1})$	$\mathbf{a} \in \mathbb{E}$, $c > 0$, $\gamma \in \mathbb{R}$, g proper	Theorem 6.13
$g(\mathcal{A}(\mathbf{x}) + \mathbf{b})$	$\mathbf{x} + \frac{1}{\alpha}\mathcal{A}^T(\mathrm{prox}_{\alpha g}(\mathcal{A}(\mathbf{x}) + \mathbf{b}) - \mathcal{A}(\mathbf{x}) - \mathbf{b})$	$\mathbf{b} \in \mathbb{R}^m$, $\mathcal{A} : \mathbb{V} \to \mathbb{R}^m$, g proper closed convex, $\mathcal{A} \circ \mathcal{A}^T = \alpha I$, $\alpha > 0$	Theorem 6.15
$g(\|\mathbf{x}\|)$	$\mathrm{prox}_g(\|\mathbf{x}\|)\frac{\mathbf{x}}{\|\mathbf{x}\|}$, $\mathbf{x} \neq \mathbf{0}$ $\{\mathbf{u} : \|\mathbf{u}\| = \mathrm{prox}_g(0)\}$, $\mathbf{x} = \mathbf{0}$	g proper closed convex, $\mathrm{dom}(g) \subseteq [0, \infty)$	Theorem 6.18

Prox Computations (Beck's 2017 book [3])

$f(\mathbf{x})$	$\mathrm{dom}(f)$	$\mathrm{prox}_f(\mathbf{x})$	Assumptions	Reference
$\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}+\mathbf{b}^T\mathbf{x}+c$	$\mathbb{R}^n$	$(\mathbf{A}+\mathbf{I})^{-1}(\mathbf{x}-\mathbf{b})$	$\mathbf{A}\in\mathbb{S}^n_+$, $\mathbf{b}\in\mathbb{R}^n$, $c\in\mathbb{R}$	Section 6.2.3
λx^3	$\mathbb{R}_+$	$\frac{-1+\sqrt{1+12\lambda[x]_+}}{6\lambda}$	$\lambda>0$	Lemma 6.5
μx	$[0,\alpha]\cap\mathbb{R}$	$\min\{\max\{x-\mu,0\},\alpha\}$	$\mu\in\mathbb{R}$, $\alpha\in[0,\infty]$	Example 6.14
$\lambda\|\mathbf{x}\|$	$\mathbb{E}$	$\left(1-\frac{\lambda}{\max\{\|\mathbf{x}\|,\lambda\}}\right)\mathbf{x}$	$\|\cdot\|$—Euclidean norm, $\lambda>0$	Example 6.19
$-\lambda\|\mathbf{x}\|$	$\mathbb{E}$	$\left(1+\frac{\lambda}{\|\mathbf{x}\|}\right)\mathbf{x}$, $\mathbf{x}\neq\mathbf{0}$; $\{\mathbf{u}:\|\mathbf{u}\|=\lambda\}$, $\mathbf{x}=\mathbf{0}$.	$\|\cdot\|$—Euclidean norm, $\lambda>0$	Example 6.21
$\lambda\|\mathbf{x}\|_1$	$\mathbb{R}^n$	$\mathcal{T}_\lambda(\mathbf{x})=[\|\mathbf{x}\|-\lambda\mathbf{e}]_+\odot\mathrm{sgn}(\mathbf{x})$	$\lambda>0$	Example 6.8
$\|\boldsymbol{\omega}\odot\mathbf{x}\|_1$	$\mathrm{Box}[-\boldsymbol{\alpha},\boldsymbol{\alpha}]$	$\mathcal{S}_{\boldsymbol{\omega},\boldsymbol{\alpha}}(\mathbf{x})$	$\boldsymbol{\alpha}\in[0,\infty]^n$, $\boldsymbol{\omega}\in\mathbb{R}^n_+$	Example 6.23
$\lambda\|\mathbf{x}\|_\infty$	$\mathbb{R}^n$	$\mathbf{x}-\lambda P_{B_{\|\cdot\|_1}[\mathbf{0},1]}(\mathbf{x}/\lambda)$	$\lambda>0$	Example 6.48
$\lambda\|\mathbf{x}\|_a$	$\mathbb{E}$	$\mathbf{x}-\lambda P_{B_{\|\cdot\|_{a,*}}[\mathbf{0},1]}(\mathbf{x}/\lambda)$	$\|\mathbf{x}\|_a$—arbitrary norm, $\lambda>0$	Example 6.47
$\lambda\|\mathbf{x}\|_0$	$\mathbb{R}^n$	$\mathcal{H}_{\sqrt{2\lambda}}(x_1)\times\cdots\times\mathcal{H}_{\sqrt{2\lambda}}(x_n)$	$\lambda>0$	Example 6.10
$\lambda\|\mathbf{x}\|^3$	$\mathbb{E}$	$\frac{2}{1+\sqrt{1+12\lambda\|\mathbf{x}\|}}\mathbf{x}$	$\|\cdot\|$—Euclidean norm, $\lambda>0$,	Example 6.20
$-\lambda\sum_{j=1}^n\log x_j$	$\mathbb{R}^n_{++}$	$\left(\frac{x_j+\sqrt{x_j^2+4\lambda}}{2}\right)_{j=1}^n$	$\lambda>0$	Example 6.9
$\delta_C(\mathbf{x})$	$\mathbb{E}$	$P_C(\mathbf{x})$	$\emptyset\neq C\subseteq\mathbb{E}$	Theorem 6.24
$\lambda\sigma_C(\mathbf{x})$	$\mathbb{E}$	$\mathbf{x}-\lambda P_C(\mathbf{x}/\lambda)$	$\lambda>0$, $C\neq\emptyset$ closed convex	Theorem 6.46
$\lambda\max\{x_i\}$	$\mathbb{R}^n$	$\mathbf{x}-\lambda P_{\Delta_n}(\mathbf{x}/\lambda)$	$\lambda>0$	Example 6.49
$\lambda\sum_{i=1}^k x_{[i]}$	$\mathbb{R}^n$	$\mathbf{x}-\lambda P_C(\mathbf{x}/\lambda)$, $C=H_{\mathbf{e},k}\cap\mathrm{Box}[\mathbf{0},\mathbf{e}]$	$\lambda>0$	Example 6.50
$\lambda\sum_{i=1}^k\|x_{(i)}\|$	$\mathbb{R}^n$	$\mathbf{x}-\lambda P_C(\mathbf{x}/\lambda)$, $C=B_{\|\cdot\|_1}[\mathbf{0},k]\cap\mathrm{Box}[-\mathbf{e},\mathbf{e}]$	$\lambda>0$	Example 6.51
$\lambda M_f^\mu(\mathbf{x})$	$\mathbb{E}$	$\mathbf{x}+\frac{\lambda}{\mu+\lambda}\left(\mathrm{prox}_{(\mu+\lambda)f}(\mathbf{x})-\mathbf{x}\right)$	$\lambda,\mu>0$, f proper closed convex	Corollary 6.64
$\lambda d_C(\mathbf{x})$	$\mathbb{E}$	$\mathbf{x}+\min\left\{\frac{\lambda}{d_C(\mathbf{x})},1\right\}(P_C(\mathbf{x})-\mathbf{x})$	$\emptyset\neq C$ closed convex, $\lambda>0$	Lemma 6.43
$\frac{\lambda}{2}d_C^2(\mathbf{x})$	$\mathbb{E}$	$\frac{\lambda}{\lambda+1}P_C(\mathbf{x})+\frac{1}{\lambda+1}\mathbf{x}$	$\emptyset\neq C$ closed convex, $\lambda>0$	Example 6.65
$\lambda H_\mu(\mathbf{x})$	$\mathbb{E}$	$\left(1-\frac{\lambda}{\max\{\|\mathbf{x}\|,\mu+\lambda\}}\right)\mathbf{x}$	$\lambda,\mu>0$	Example 6.66
$\rho\|\mathbf{x}\|_1^2$	$\mathbb{R}^n$	$\left(\frac{v_i x_i}{v_i+2\rho}\right)_{i=1}^n$, $\mathbf{v}=\left[\sqrt{\frac{\rho}{\mu}}\|\mathbf{x}\|-2\rho\right]_+\mathbf{e}^T\mathbf{v}=1$ ($\mathbf{0}$ when $\mathbf{x}=\mathbf{0}$)	$\rho>0$	Lemma 6.70
$\lambda\|\mathbf{A}\mathbf{x}\|_2$	$\mathbb{R}^n$	$\mathbf{x}-\mathbf{A}^T(\mathbf{A}\mathbf{A}^T+\alpha^*\mathbf{I})^{-1}\mathbf{A}\mathbf{x}$, $\alpha^*=0$ if $\|\mathbf{v}_0\|_2\leq\lambda$; otherwise, $\|\mathbf{v}_{\alpha^*}\|_2=\lambda$; $\mathbf{v}_\alpha\equiv(\mathbf{A}\mathbf{A}^T+\alpha\mathbf{I})^{-1}\mathbf{A}\mathbf{x}$	$\mathbf{A}\in\mathbb{R}^{m\times n}$ with full row rank, $\lambda>0$	Lemma 6.68

Stochastic Gradient

There are **infinitely many** ways of obtaining a random vector g^k satisfying (8).

- **Pros: flexibility** to construct stochastic gradients in various ways **based on problem structure,** and in order to **target desirable properties** such as
  - convergence speed,
  - iteration cost,
  - overall complexity,
  - parallelizability,
  - suitability for given computing architecture,
  - communication cost,
  - generalization properties.
- **Cons: A crazy ZOO of methods.**
  - Hard to get into the field, hard to keep up with new results
  - Considerable **challenges in terms of convergence analysis.** Indeed, if one aims to, as one should, obtain the sharpest bounds possible, dedicated analyses are needed to handle each of the particular variants of SGD.

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 3: General Analysis of SGD

Based on:

[5] E. Gorbunov, F. Hanzely and P.R., **A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent**, arXiv:1905.11261, 2019

The Plan

- We first introduce the **key (weak and powerful) assumption on the stochastic gradients** g^k enabling our general analysis (Assumption 1),
- then state our **assumptions on** f (Assumption 4),
- and finally state and comment on a **unified convergence result** (Theorem 2).

Key Assumption on the Stochastic Gradients

Notation

- $\langle x, y \rangle \stackrel{\text{def}}{=} \sum_i x_i y_i$ is the **standard Euclidean inner product**
- $\|x\| \stackrel{\text{def}}{=} \langle x, x \rangle^{1/2}$ is the **Euclidean norm**
- By $D_f(x, y)$ we denote the **Bregman divergence** associated with f:

$$D_f(x, y) \stackrel{\text{def}}{=} f(x) - f(y) - \langle \nabla f(y), x - y \rangle$$

- $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$.

Modelling the Evolution of the Stochastic Gradients

We now introduce an assumption on the stochastic gradients $\{g^k\}_{k\geq 0}$ generated by SGD.

Assumption 1 (Gorbunov, Hanzely & R 2019 [5])

Let $\{x^k\}_{k\geq 0}$ be the random iterates produced by SGD.

1. **Unbiasedness:** *We first assume that the stochastic gradients g^k are unbiased, i.e.,*

$$\mathbf{E}\left[g^k \mid x^k\right] = \nabla f(x^k), \quad \forall k \geq 0. \tag{9}$$

2. **Two recursions bounding evolution of the stochastic gradients:** *Further, we assume that there exist non-negative constants A, B, C, D_1, D_2, ρ and a (possibly) random sequence $\{\sigma_k^2\}_{k\geq 0}$ such that the following two relations hold:*

$$\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \leq 2AD_f(x^k, x^*) + B\sigma_k^2 + D_1, \tag{10}$$

$$\mathbf{E}\left[\sigma_{k+1}^2 \mid \sigma_k^2\right] \leq (1-\rho)\sigma_k^2 + 2CD_f(x^k, x^*) + D_2, \tag{11}$$

First Comments About Assumption 1

- The **expectation** is with respect to the randomness of the algorithm (note that x^k, g^k and possibly σ_k^2 are random).
- The **parameters** in the two recursions offer the **flexibility to model many** SGD **methods.**
- We will only consider **unbiased estimators;** see (9).
  - Virtually all known methods utilize unbiased estimators.
  - Convergence analysis is easier.
- The world of SGD methods utilizing **biased** estimators (i.e., g^k not satisfying (9)) is small at the moment; and not very well understood. Examples:
  - the SAG method of Schmidt, Le Roux and Bach [24].
  - the SARAH method of Nguyen et al [20].
  - the STP method of Bergou, Gorbunov and R [4]
- **Optimistic view:** The ideal situation is to have $x^k = x^*$ and $g^k = \nabla f(x^*)$. In this case, we can choose $A = B = D_1 = 0$ in the first recursion. Note that then

$$x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma \nabla f(x^k)) = x^k$$

Variance-reduced Methods

Definition 1 (Variance-reduced methods)

We say that SGD with stochastic gradients satisfying Assumption 1 is **variance reduced** if $D_1 = D_2 = 0$.

- This is the **first quantitative definition of a what it means for an SGD method to be variance-reduced.**
- We shall see (Theorem 2) that **variance-reduced methods converge to the solution, i.e., $x^k \to x^*$,** whereas **non-variance reduced methods only converge to a neighbourhood of the solution.**
- Meaning of $\{\sigma_k^2\}_{k\geq 0}$: the sequence encodes the progress of the **variance-reduction process** employed.
  - If no variance reduction process is employed, then one has $\sigma_k \equiv 0$. In this case, one can set the parameters B, ρ, C and D_2 as follows:

$$B = 0, \quad \rho = 1, \quad C = 0, \quad D_2 = 0. \tag{12}$$

  This eliminates the second recursion, which is not necessary.
  - For instance, GD and simple variants of SGD (with uniform sampling, importance sampling, and various minibatching techniques) are in this category.

GD Satisfies Assumption 1

Assumption 2

f is **convex,** *i.e.,*

$$f(x) \geq D_f(x,y) \stackrel{def}{=} f(y) + \langle \nabla f(y), x - y\rangle, \quad \forall x, y \in \mathbb{R}^d$$

and **L-smooth,** *i.e.,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in \mathbb{R}^d.$$

These conditions imply that[3] (see Nesterov's book [19])

$$\|\nabla f(x) - \nabla f(y)\|^2 \leq 2L D_f(x,y), \quad \forall x, y \in \mathbb{R}^d. \tag{13}$$

Hence, if f satisfies Assumption 2, then **gradient descent satisfies Assumption 1** with

$$A = L, \ B = 0, \ D_1 = 0, \ \sigma_k = 0, \ \rho = 1, \ C = 0, \ D_2 = 0. \tag{14}$$

[3] Hence, D_f can be used as a measure of proximity for the gradients.

Analysis of SGD under Assumption 1

Generic SGD Method

Having introduced Assumption 1, let us again write down the format of a generic SGD method for solving problem (1) (i.e., $\min f + R$) satisfying this assumption.

Algorithm 1 Generic SGD method

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$
for $k = 0, 1, 2, \ldots$ **do**
    Construct stochastic gradient g^k satisfying Assumption 1
    $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$

Strong Convexity of f (in a relaxed form)

Assumption 3 (Unique solution)

For simplicity we assume that (1) *has a unique minimizer, which we denote* x^*.

Remark: The uniqueness assumption can be lifted, but we will not do this.

Assumption 4 (μ-strong quasi-convexity)

There exists $\mu > 0$ *such that* $f : \mathbb{R}^d \to \mathbb{R}$ *is* μ*-strongly quasi-convex. That is, the following inequality holds:*

$$f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle + \frac{\mu}{2} \|x^* - x\|^2, \qquad \forall x \in \mathbb{R}^d. \tag{15}$$

Main Result

Theorem 2 (Gorbunov, Hanzely & R 2019 [5])

Let Assumptions 1, 3 and 4 be satisfied. Choose constant M *such that* $M > \frac{B}{\rho}$. *Choose a stepsize satisfying*

$$0 < \gamma \leq \min\left\{\frac{1}{\mu}, \frac{1}{A + CM}\right\}. \tag{16}$$

Then the iterates $\{x^k\}_{k \geq 0}$ *of* SGD *(Algorithm 1; see also* (7)*) satisfy*

$$\mathbf{E}\left[V^k\right] \leq \max\left\{(1 - \gamma\mu)^k, \left(1 + \frac{B}{M} - \rho\right)^k\right\} V^0 + \frac{(D_1 + MD_2)\gamma^2}{\min\left\{\gamma\mu, \rho - \frac{B}{M}\right\}}, \tag{17}$$

where the **Lyapunov function** V^k *is defined by*

$$V^k \overset{def}{=} \left\|x^k - x^*\right\|^2 + M\gamma^2\sigma_k^2. \tag{18}$$

Meaning: **Linear convergence rate for a wide range of SGD methods up to a certain oscillation radius**, controlled by the additive term in (17), and namely, by parameters D_1 and D_2.

Table: Special Cases

Problem	Method	Alg #	Citation	VR?	AS?	Quant?	RCD?	Section	Result
(1)+(2)	SGD	Alg 1	[26]	✗	✗	✗	✗	A.1	Cor A.1
(1)+(3)	SGD-SR	Alg 2	[6]	✗	✓	✗	✗	A.2	Cor A.2
(1)+(3)	SGD-MB	Alg 3	**NEW**	✗	✗	✗	✗	A.3	Cor A.3
(1)+(3)	SGD-star	Alg 4	**NEW**	✓	✓	✗	✗	A.4	Cor A.4
(1)+(3)	SAGA	Alg 5	[5]	✓	✗	✗	✗	A.5	Cor A.5
(1)+(3)	N-SAGA	Alg 6	**NEW**	✗	✗	✗	✗	A.6	Cor A.6
(1)	SEGA	Alg 7	[11]	✓	✗	✗	✓	A.7	Cor A.7
(1)	N-SEGA	Alg 8	**NEW**	✗	✗	✗	✓	A.8	Cor A.8
(1)+(3)	SVRG[a]	Alg 9	[15]	✓	✗	✗	✗	A.9	Cor A.9
(1)+(3)	L-SVRG	Alg 10	[13, 18]	✓	✗	✗	✗	A.10	Cor A.10
(1)+(3)	DIANA	Alg 11	[20, 14]	✗	✗	✓	✗	A.11	Cor A.11
(1)+(3)	DIANA[b]	Alg 12	[20, 14]	✓	✗	✓	✗	A.11	Cor A.12
(1)+(3)	Q-SGD-SR	Alg 13	**NEW**	✗	✓	✓	✗	A.12	Cor A.13
(1)+(3)+(4)	VR-DIANA	Alg 14	[14]	✓	✗	✓	✗	A.13	Cor A.15
(1)+(3)	JacSketch	Alg 15	[9]	✓	✓✗	✗	✗	A.14	Cor A.16

Table 1: List of specific existing (in some cases generalized) and new methods which fit our general analysis framework. VR = variance reduced method, AS = arbitrary sampling, Quant = supports gradient quantization, RCD = randomized coordinate descent type method. [a] Special case of SVRG with 1 outer loop only; [b] Special case of DIANA with 1 node and quantization of exact gradient.

Table: Parameters

Method	A	B	ρ	C	D_1	D_2
SGD	$2L$	0	1	0	$2\sigma^2$	0
SGD-SR	$2\mathcal{L}$	0	1	0	$2\sigma^2$	0
SGD-MB	$\frac{A'+L(\tau-1)}{\tau}$	0	1	0	$\frac{D'}{\tau}$	0
SGD-star	$2\mathcal{L}$	0	1	0	0	0
SAGA	$2L$	2	$1/n$	L/n	0	0
N-SAGA	$2L$	2	$1/n$	L/n	$2\sigma^2$	$\frac{\sigma^2}{n}$
SEGA	$2dL$	$2d$	$1/d$	L/d	0	0
N-SEGA	$2dL$	$2d$	$1/d$	L/d	$2d\sigma^2$	$\frac{\sigma^2}{d}$
SVRG[a]	$2L$	2	0	0	0	0
L-SVRG	$2L$	2	p	Lp	0	0
DIANA	$\left(1+\frac{2\omega}{n}\right)L$	$\frac{2\omega}{n}$	α	$L\alpha$	$\frac{(1+\omega)\sigma^2}{n}$	$\alpha\sigma^2$
DIANA[b]	$(1+2\omega)L$	2ω	α	$L\alpha$	0	0
Q-SGD-SR	$2(1+\omega)\mathcal{L}$	0	1	0	$2(1+\omega)\sigma^2$	0
VR-DIANA	$\left(1+\frac{4\omega+2}{n}\right)L$	$\frac{2(\omega+1)}{n}$	α	$\left(\frac{1}{m}+4\alpha\right)L$	0	0
JacSketch	$2\mathcal{L}_1$	$\frac{2\lambda_{\max}}{n}$	$\lambda_{\min}$	$\frac{\mathcal{L}_2}{n}$	0	0

Sanity Check: Gradient Descent

Proximal Gradient Descent: The Algorithm

Algorithm 2 GD

1: **Parameters:** learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Set $g^k = \nabla f(x^k)$
4: $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$

Recovering the Rate of Gradient Descent from Theorem 2

In view of (14) which says that GD satisfies Assumption 1 with

$$A = L,\ B = 0,\ D_1 = 0,\ \sigma_k = 0,\ \rho = 1,\ C = 0,\ D_2 = 0,$$

Theorem 2 (we can choose $M = 1$) yields the following for **gradient descent:**

- Stepsize is restricted to $0 < \gamma \leq \frac{1}{L}$ (since $\mu \leq L$ and $C = 0$)
- The Lyapunov function is $V^k = \left\|x^k - x^*\right\|^2$ (since $\sigma_k^2 \equiv 0$)
- The rate is $\left\|x^k - x^*\right\|^2 \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2$.
- So, for the largest allowable stepsize $\gamma = \frac{1}{L}$, **we recover the typical linear rate $(1 - \frac{\mu}{L})^k$ of gradient descent.** This can be alternatively written as follows:

$$k \geq \frac{L}{\mu} \log \frac{1}{\varepsilon} \quad \Rightarrow \quad \left\|x^k - x^*\right\|^2 \leq \varepsilon \left\|x^0 - x^*\right\|^2.$$

Complexity of GD

Let us formalize the above observations.

Corollary 3

Assume that

- *f is convex and L-smooth (Assumption 2)*
- *The problem $\min f + R$ has a unique solution (Assumption 3)*
- *f is μ-quasi strongly convex (Assumption 4)*

Then GD *with stepsize $0 < \gamma \leq \frac{1}{L}$ satisfies*

$$\left\|x^k - x^*\right\|^2 \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2. \tag{19}$$

Proof of Theorem 2

Basic Facts and Inequalities I

For all $a, b \in \mathbb{R}^d$ and $t > 0$ the following inequalities holds:

$$\langle a, b \rangle \leq \frac{\|a\|^2}{2t} + \frac{t\|b\|^2}{2}, \tag{20}$$

$$\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2, \tag{21}$$

and

$$\frac{1}{2}\|a\|^2 - \|b\|^2 \leq \|a+b\|^2. \tag{22}$$

For a random vector $\xi \in \mathbb{R}^d$ and any $x \in \mathbb{R}^d$ the variance can be decomposed as

$$\mathbf{E}\left[\|\xi - \mathbf{E}[\xi]\|^2\right] = \mathbf{E}\left[\|\xi - x\|^2\right] - \mathbf{E}\left[\|\mathbf{E}[\xi] - x\|^2\right]. \tag{23}$$

A Key Lemma I

The following lemma will be used in the proof of our main theorem.

Lemma 4 (Key single iteration recurrence)

Let Assumptions 1–4 be satisfied. Then the following inequality holds for all $k \geq 0$:

$$\begin{aligned}
&\mathbf{E}\left[\left\|x^{k+1}-x^*\right\|^2\right] + M\gamma^2\mathbf{E}\left[\sigma_{k+1}^2\right] + 2\gamma\left(1-\gamma(A+CM)\right)\mathbf{E}\left[D_f(x^k,x^*)\right] \\
&\leq (1-\gamma\mu)\mathbf{E}\left[\left\|x^k-x^*\right\|^2\right] + (1-\rho)\,M\gamma^2\mathbf{E}\left[\sigma_k^2\right] \\
&\quad + B\gamma^2\mathbf{E}\left[\sigma_k^2\right] + (D_1+MD_2)\gamma^2.
\end{aligned}$$

Proof of Theorem 2

Note first that due to the stepsize restriction (16) we have

$$2\gamma\left(1-\gamma(A+CM)\right)\mathbf{E}\left[D_f(x^k,x^*)\right] > 0,$$

thus we can omit the term.

Unrolling the recurrence from Lemma 4 and using the Lyapunov function notation gives us

$$\begin{aligned}
\mathbf{E}\left[V^k\right] &\leq \max\left\{(1-\gamma\mu)^k, \left(1+\frac{B}{M}-\rho\right)^k\right\}V^0 \\
&\quad + (D_1+MD_2)\gamma^2\sum_{l=0}^{k-1}\max\left\{(1-\gamma\mu)^l, \left(1+\frac{B}{M}-\rho\right)^l\right\} \\
&\leq \max\left\{(1-\gamma\mu)^k, \left(1+\frac{B}{M}-\rho\right)^k\right\}V^0 \\
&\quad + (D_1+MD_2)\gamma^2\sum_{l=0}^{\infty}\max\left\{(1-\gamma\mu)^l, \left(1+\frac{B}{M}-\rho\right)^l\right\} \\
&\leq \max\left\{(1-\gamma\mu)^k, \left(1+\frac{B}{M}-\rho\right)^k\right\}V^0 + \frac{(D_1+MD_2)\gamma^2}{\min\left\{\gamma\mu, \rho-\frac{B}{M}\right\}}.
\end{aligned}$$

Proof of Lemma 4 I

We start with estimating the first term of the Lyapunov function. Let $r^k \overset{\text{def}}{=} x^k - x^*$. Then

$$\begin{aligned}\left\|r^{k+1}\right\|^2 &= \left\|\text{prox}_{\gamma R}(x^k - \gamma g^k) - \text{prox}_{\gamma R}(x^* - \gamma \nabla f(x^*))\right\|^2 \\ &\leq \left\|x^k - x^* - \gamma(g^k - \nabla f(x^*))\right\|^2 \\ &= \left\|r^k\right\|^2 - 2\gamma\langle r^k, g^k - \nabla f(x^*)\rangle + \gamma^2\left\|g^k - \nabla f(x^*)\right\|^2,\end{aligned}$$

where in the inequality we used non-expansiveness of the prox. Taking expectation conditioned on x^k we get

$$\begin{aligned}\mathbf{E}\left[\left\|r^{k+1}\right\|^2 \mid x^k\right] &= \left\|r^k\right\|^2 - 2\gamma\langle r^k, \nabla f(x^k) - \nabla f(x^*)\rangle + \gamma^2\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \\ &\overset{(15)}{\leq} (1-\gamma\mu)\left\|r^k\right\|^2 - 2\gamma D_f(x^k, x^*) + \gamma^2\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \\ &\overset{(9)+(10)}{\leq} (1-\gamma\mu)\left\|r^k\right\|^2 + 2\gamma\left(A\gamma - 1\right)D_f(x^k, x^*) + B\gamma^2\sigma_k^2 + \gamma^2 D_1,\end{aligned}$$

where (15) is μ-strong quasi convexity of f, (9) is unbiasedness of g^k and (10) is the first recursion assumed to hold for g^k.

Proof of Lemma 4 II

Using this we estimate the full expectation of V^{k+1} by applying recursion (11) involving σ_k^2:

$$\begin{aligned}&\mathbf{E}\left[\left\|x^{k+1} - x^*\right\|^2\right] + M\gamma^2\mathbf{E}\left[\sigma_{k+1}^2\right] \\ \overset{(11)}{\leq}\ &(1-\gamma\mu)\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] + 2\gamma\left(A\gamma - 1\right)D_f(x^k, x^*) + B\gamma^2\mathbf{E}\left[\sigma_k^2\right] \\ &+(1-\rho)M\gamma^2\mathbf{E}\left[\sigma_k^2\right] + 2CM\gamma^2\mathbf{E}\left[D_f(x^k, x^*)\right] + (D_1 + MD_2)\gamma^2 \\ =\ &(1-\gamma\mu)\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] + \left(1 + \frac{B}{M} - \rho\right)M\gamma^2\mathbf{E}\left[\sigma_k^2\right] \\ &+2\gamma\left(\gamma(A + CM) - 1\right)\mathbf{E}\left[D_f(x^k, x^*)\right] + (D_1 + MD_2)\gamma^2.\end{aligned}$$

It remains to rearrange the terms.

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 4: SGD with Uniform Sampling

Based on:

[7]

The Plan

We now consider the regularized **finite-sum** problem

$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} f_i(x)}_{f(x)} + R(x). \tag{24}$$

That is, we consider problem (1) with f being of structure (3) .

- We consider the **simplest variant of SGD,** i.e., one using the estimator
$$g^k = \nabla f_i(x^k),$$
where i is chosen uniformly at random at iteration k.
- So, $\mathcal{D}$ is the uniform distribution over $\{1, 2, \ldots, n\}$.
- We will **analyze it using Theorem 2.**

SGD-US: the Algorithm

For the record, here is the formal algorithm:

Algorithm 3 SGD-US

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$
for $k = 0, 1, 2, \ldots$ **do**
  Sample $i^k = i \in \{1, 2, \ldots, n\}$ with probability $\frac{1}{n}$
  $g^k = \nabla f_i(x^k)$ obtain a stochastic gradient
  $x^{k+1} = \text{prox}_{\gamma R}(x^k - \gamma g^k)$

Average Smoothness: The Right Smoothness Notion for SGD-US

- Recall that in the case of GD, verification of the key assumption (Assumption 1) on stochastic gradients relied on inequality (13), i.e.,

$$\|\nabla f(x) - \nabla f(y)\|^2 \leq 2LD_f(x, y), \quad \forall x, y \in \mathbb{R}^d,$$

  which was simply a consequence of convexity and L-smoothness of f.
- In the proof of the main convergence result (Theorem 2) we only used this inequality for $y = x^*$, i.e., in the form

$$\|\nabla f(x) - \nabla f(x^*)\|^2 \leq 2LD_f(x, x^*), \quad \forall x \in \mathbb{R}^d,$$

- **What is the correct notion of smoothness for SGD-US?**

Definition 5 (Average smoothness)

We say that $f = \frac{1}{n}\sum_i f_i$ is **$\mathcal{L}$-smooth on average** (i.e., with respect to $\mathcal{D}$) if there exists $\mathcal{L} > 0$ such that

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f_i(x^*)\|^2 \leq 2\mathcal{L}D_f(x, x^*), \quad \forall x \in \mathbb{R}^d. \tag{25}$$

For simplicity, we will write $(f, \mathcal{D}) \sim AS(\mathcal{L})$ to say that (36) holds.

Computing the Average Smoothness Constant I

Assumption 5

Each f_i is convex and L_i-smooth.

Let $L_{\max} \stackrel{\text{def}}{=} \max_i L_i$.

Under Assumption 6, the following hold:

- $f = \frac{1}{n}\sum_i f_i$ is L-smooth, and $L \leq \frac{1}{n}\sum_i L_i$.
- $$0 \leq D_{f_i}(x,y) \leq \frac{L_i}{2}\|x-y\|^2, \quad \forall x,y \in \mathbb{R}^d. \tag{26}$$
- $$\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leq 2L_i D_{f_i}(x,y), \quad \forall x,y \in \mathbb{R}^d. \tag{27}$$

  (compare this with (13))

Computing the Average Smoothness Constant II

We will now use (27) to compute the **average smoothness constant $\mathcal{L}$.**

Lemma 6

Let Assumption 6 hold (convexity and L_i smoothness of f_i). The average smoothness constant of $f = \frac{1}{n}\sum_i f_i$ is $\mathcal{L} = L_{\max}$.

Proof.

Fix any $x, y \in \mathbb{R}^d$. Then

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f_i(y)\|^2 &\overset{(27)}{\leq} \frac{1}{n}\sum_{i=1}^n 2L_i D_{f_i}(x,y) \\
&\leq 2\left(\max_i L_i\right)\frac{1}{n}\sum_{i=1}^n D_{f_i}(x,y) \\
&= 2L_{\max} D_f(x,y). \qquad (28)
\end{aligned}$$

It remains to compare this with (36). □

Satisfying Assumption 1

We now show that SGD-US satisfies Assumption 1

$$\mathrm{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \leq 2AD_f(x^k, x^*) + B\sigma_k^2 + D_1,$$

$$\mathrm{E}\left[\sigma_{k+1}^2 \mid \sigma_k^2\right] \leq (1-\rho)\sigma_k^2 + 2CD_f(x^k, x^*) + D_2,$$

for the following choice of parameters:

$$A = 2L_{\max},\ B = 0,\ D_1 = \sigma^2,\ \sigma_k^2 = 0,\ \rho = 1,\ C = 0,\ D_2 = 0. \tag{29}$$

Lemma 7 (Gorbunov, Hanzely and R 2019 [5])

Let Assumption 6 hold. Then

$$\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f(x^*)\|^2 \leq 4L_{\max} D_f(x, x^*) + 2\sigma^2. \tag{30}$$

where

$$\sigma^2 \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*) - \nabla f(x^*)\|^2 \tag{31}$$

is the variance of the stochastic gradients at the optimum.

Remark: Lemma 7 is a generalization of in Gower et al 2019 [7, Lemma 2.4], who considered the $R \equiv 0$ case.

Proof of Lemma 7

Expectations are with respect to the random choice of i: chosen uniformly at random from $\{1, 2, \ldots, n\}$. Then

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f(x^*)\|^2 &= \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f_i(x^*) + \nabla f_i(x^*) - \nabla f(x^*)\|^2 \\
&\overset{(21)}{\leq} \frac{1}{n}\sum_{i=1}^n \left(2\|\nabla f_i(x) - \nabla f_i(x^*)\|^2 + 2\|\nabla f_i(x^*) - \nabla f(x^*)\|^2\right) \\
&\overset{(28)+(31)}{\leq} 4L_{\max} D_f(x, x^*) + 2\sigma^2.
\end{aligned}$$

Complexity of SGD-US

A direct consequence of Theorem 2:

Corollary 8

Assume that

- *The problem* $\min f + R$ *has a unique solution (Assumption 3)*
- f *is* μ*-quasi strongly convex (Assumption 4)*
- f_i *are convex and* L_i*-smooth (Assumption 6)*

Then SGD-US *with stepsize* $0 < \gamma \leq \frac{1}{2 \max_i L_i}$ *satisfies*

$$\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2 + \frac{2\gamma\sigma^2}{\mu}. \tag{32}$$

Exercises

Both exercises below consider the $\sigma^2 = 0$ case. In this case, SGD-US is a variance-reduced method (see Def 1).

Exercise 1

Prove that if $\sigma^2 = 0$ *(i.e., if* $\nabla f_i(x^*) = \nabla f_j(x^*)$ *for all* i, j*), then the upper bound in* (30) *can be improved by a factor of 2:*

$$\frac{1}{n}\sum_{i=1}^{n} \left\|\nabla f_i(x) - \nabla f(x^*)\right\|^2 \leq 2L_{\max} D_f(x, x^*). \tag{33}$$

Exercise 2

Show that if $\sigma^2 = 0$*, then Corollary 8 has the following stronger form:* SGD-US *allows for the larger stepsize* $0 < \gamma \leq \frac{1}{\max_i L_i}$*, and satisfies*

$$\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2. \tag{34}$$

Comparing GD and SGD-US in the $\sigma^2 = 0$ Case

Let us compare the convergence of GD and SGD-US:

- GD
  - By Corollary 3, using stepsize $\gamma = \frac{1}{L}$, GD achieves iteration complexity

$$k \geq \frac{L}{\mu} \log \frac{1}{\varepsilon} \quad \Rightarrow \quad \left\|x^k - x^*\right\|^2 \leq \varepsilon \left\|x^0 - x^*\right\|^2$$

  - Cost of 1 iteration: n gradient evaluations
  - **Total complexity:** $\tilde{\mathcal{O}}\left(\frac{nL}{\mu}\right)$
- SGD-US
  - By Corollary 8, using stepsize $\gamma = \frac{1}{L_{\max}}$, SGD-US achieves iteration complexity

$$k \geq \frac{L_{\max}}{\mu} \log \frac{1}{\varepsilon} \quad \Rightarrow \quad \mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq \varepsilon \left\|x^0 - x^*\right\|^2$$

  - Cost of 1 iteration: 1 gradient evaluation
  - **Total complexity:** $\tilde{\mathcal{O}}\left(\frac{L_{\max}}{\mu}\right)$

Conclusion: Note that $nL \leq \sum_{i=1}^n L_i \leq nL_{\max}$, and this inequality can be tight, so that $L = L_{\max}$. So, SGD-US **can be as much as n times faster than GD! Howeevr, it could also be worse since it's possible for $nL \leq L_{\max}$ to hold.**

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

Part 5: SGD with Nonuniform Sampling

Based on:

[7] R.M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P.R. **SGD: General Analysis and Improved Rates**, ICML 2019

[29] P. Zhao and T. Zhang, **Stochastic optimization with importance sampling for regularized loss minimization**, ICML 2015

The Plan

Recall we consider the regularized **finite-sum** problem

$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} f_i(x)}_{f(x)} + R(x). \tag{35}$$

- We now consider the **a non-uniform sampling variant of SGD,** i.e., one using the estimator

$$g^k = \frac{\nabla f_i(x^k)}{np_i},$$

  where i is chosen with probability $p_i > 0$ at iteration k.
- So, $\mathcal{D}$ is this non-uniform distribution over $\{1, 2, \ldots, n\}$.
- As before, we will **analyze it using Theorem 2.**

SGD-NS: the Algorithm

For the record, here is the formal algorithm:

Algorithm 4 SGD-NS

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$, probabilities $p_1, \ldots, p_n$ summing up to one
for $k = 0, 1, 2, \ldots$ **do**
  Sample $i^k = i \in \{1, 2, \ldots, n\}$ with probability $p_i > 0$
  $g^k = \frac{\nabla f_i(x^k)}{np_i}$ obtain a stochastic gradient
  $x^{k+1} = \text{prox}_{\gamma R}(x^k - \gamma g^k)$

Weighted Smoothness: the Right Smoothness Notion for SGD-NS

- Recall that in the case of GD, verification of the key assumption (Assumption 1) on stochastic gradients relied on inequality (13), i.e.,

$$\|\nabla f(x) - \nabla f(y)\|^2 \leq 2LD_f(x,y), \quad \forall x, y \in \mathbb{R}^d,$$

  which was simply a consequence of convexity and L-smoothness of f.
- In the proof of the main convergence result (Theorem 2) we only used this inequality for $y = x^*$, i.e., in the form

$$\|\nabla f(x) - \nabla f(x^*)\|^2 \leq 2LD_f(x,x^*), \quad \forall x \in \mathbb{R}^d,$$

- **What is the correct notion of smoothness for SGD-NS?**

Definition 9 (Weighted smoothness)

We say that $f = \frac{1}{n}\sum_i f_i$ is $\mathcal{L}$-smooth with respect to weights $p_1, \ldots, p_n$ (i.e., with respect to $\mathcal{D}$) if there exists $\mathcal{L} > 0$ such that

$$\sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x)}{np_i} - \frac{\nabla f_i(x^*)}{np_i} \right\|^2 \leq 2\mathcal{L}D_f(x,x^*), \quad \forall x \in \mathbb{R}^d. \tag{36}$$

For simplicity, we will write $(f, \mathcal{D}) \sim WS(\mathcal{L})$ to say that (36) holds.

Computing the Weighted Smoothness Constant I

Assumption 6

Each f_i is convex and L_i-smooth.

Let $L_{\max} \overset{\text{def}}{=} \max_i L_i$.

Under Assumption 6, the following hold:

- $f = \frac{1}{n}\sum_i f_i$ is L-smooth, and $L \leq \frac{1}{n}\sum_i L_i$.
- $$0 \leq D_{f_i}(x,y) \leq \frac{L_i}{2}\|x-y\|^2, \quad \forall x, y \in \mathbb{R}^d. \tag{37}$$
- $$\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leq 2L_i D_{f_i}(x,y), \quad \forall x, y \in \mathbb{R}^d. \tag{38}$$

  (compare this with (13))

Computing the Weighted Smoothness Constant II

We will now use (27) to compute the weighted smoothness constant $\mathcal{L}$.

Lemma 10

Let Assumption 6 hold (convexity and L_i smoothness of f_i). The weighted smoothness constant of $f = \frac{1}{n}\sum_i f_i$ is $\mathcal{L} = \max_i \frac{L_i}{np_i}$.

Proof.

Fix any $x, y \in \mathbb{R}^d$. Then

$$\begin{aligned}
\sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x)}{np_i} - \frac{\nabla f_i(y)}{np_i} \right\|^2 &= \frac{1}{n}\sum_{i=1}^n \frac{1}{np_i}\|\nabla f_i(x) - \nabla f_i(y)\|^2 \qquad (39)\\
&\overset{(27)}{\leq} \frac{1}{n}\sum_{i=1}^n \frac{1}{np_i} 2L_i D_{f_i}(x,y)\\
&\leq 2\left(\max_i \frac{L_i}{np_i}\right)\frac{1}{n}\sum_{i=1}^n D_{f_i}(x,y)\\
&= 2\left(\max_i \frac{L_i}{np_i}\right) D_f(x,y). \qquad (40)
\end{aligned}$$

It remains to compare this with (36). □

Satisfying Assumption 1

We now show that SGD-NS satisfies Assumption 1

$\mathrm{E}\left[\|g^k - \nabla f(x^*)\|^2 \mid x^k\right] \leq 2AD_f(x^k, x^*) + B\sigma_k^2 + D_1,$

$\mathrm{E}\left[\sigma_{k+1}^2 \mid \sigma_k^2\right] \leq (1-\rho)\sigma_k^2 + 2CD_f(x^k, x^*) + D_2,$ for the following choice of parameters:

$$A = 2\mathcal{L},\ B = 0,\ D_1 = \sigma^2,\ \sigma_k^2 = 0,\ \rho = 1,\ C = 0,\ D_2 = 0. \qquad (41)$$

Lemma 11 (Gorbunov, Hanzely and R 2019 [5])

Let Assumption 6 hold. Then

$$\sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x)}{np_i} - \nabla f(x^*) \right\|^2 \leq 4\mathcal{L}D_f(x, x^*) + 2\sigma^2. \qquad (42)$$

where

$$\sigma^2 \overset{def}{=} \sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x^*)}{np_i} - \nabla f(x^*) \right\|^2 \qquad (43)$$

is the weighted variance of the stochastic gradients at the optimum.

Remark: Lemma 11 is a generalization of in Gower et al 2019 [7, Lemma 2.4], who considered the $R \equiv 0$ case.

Proof of Lemma 11

Expectations are with respect to the random choice of i: chosen uniformly at random from $\{1, 2, \ldots, n\}$. Then

$$\begin{aligned}
\sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x)}{np_i} - \nabla f(x^*) \right\|^2 &= \sum_{i=1}^n p_i \left\| \frac{\nabla f_i(x)}{np_i} - \frac{\nabla f_i(x^*)}{np_i} + \frac{\nabla f_i(x^*)}{np_i} - \nabla f(x^*) \right\|^2 \\
&\overset{(21)}{\leq} \sum_{i=1}^n p_i \left(2 \left\| \frac{\nabla f_i(x)}{np_i} - \frac{\nabla f_i(x^*)}{np_i} \right\|^2 + 2 \left\| \frac{\nabla f_i(x^*)}{np_i} - \nabla f(x^*) \right\|^2 \right) \\
&\overset{(40)+(43)}{\leq} 4 \left(\max_i \frac{L_i}{np_i} \right) D_f(x, x^*) + 2\sigma^2.
\end{aligned}$$

Complexity of SGD-NS

A direct consequence of Theorem 2:

Corollary 12

Assume that

- *The problem* $\min f + R$ *has a unique solution (Assumption 3)*
- f *is* μ*-quasi strongly convex (Assumption 4)*
- f_i *are convex and* L_i*-smooth (Assumption 6)*

Then SGD-NS *with stepsize* $0 < \gamma \leq \frac{1}{2\mathcal{L}}$*, where* $\mathcal{L} = \max_i \frac{L_i}{np_i}$*, satisfies*

$$\mathbf{E}\left[\left\| x^k - x^* \right\|^2 \right] \leq (1 - \gamma\mu)^k \left\| x^0 - x^* \right\|^2 + \frac{2\gamma\sigma^2}{\mu}. \tag{44}$$

Exercises

Both exercises below consider the $\sigma^2 = 0$ case. In this case, SGD-NS is a variance-reduced method (see Def 1).

Exercise 3

Prove that if $\sigma^2 = 0$ (i.e., if $\frac{\nabla f_i(x^*)}{p_i} = \frac{\nabla f_j(x^*)}{p_j}$ for all i, j), then the upper bound in (30) can be improved by a factor of 2:

$$\sum_{i=1}^{n} p_i \left\| \frac{\nabla f_i(x)}{np_i} - \nabla f(x^*) \right\|^2 \leq 2\mathcal{L} D_f(x, x^*). \tag{45}$$

Exercise 4

Show that if $\sigma^2 = 0$, then Corollary 12 has the following stronger form: SGD-NS *allows for the larger stepsize $0 < \gamma \leq \frac{1}{\mathcal{L}}$, and satisfies*

$$\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2. \tag{46}$$

Importance Sampling: SGD-IS

In Theorem 12, the fastest rate is obtained when we allow the stepsize to be the largest. We can optimize the stepsize by choosing probabilities which mininimze $\mathcal{L} = \max_i \frac{L_i}{np_i}$:

$$\min_{p_1,\ldots,p_n} \max_i \frac{L_i}{np_i}.$$

This leads to the importance sampling probabilities

$$p_i = \frac{L_i}{\sum_j L_j} \quad \Rightarrow \quad \mathcal{L} = \frac{\sum_i L_i}{n} \overset{\text{def}}{=} \bar{L}.$$

Algorithm 5 SGD-IS

1: **Parameters:** learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Sample $i^k = i \in \{1, 2, \ldots, n\}$ with probability $p_i = \frac{L_i}{\sum_j L_j}$
4: $g^k = \frac{\nabla f_i(x^k)}{np_i}$ obtain a stochastic gradient
5: $x^{k+1} = \text{prox}_{\gamma R}(x^k - \gamma g^k)$

Comparing GD and SGD-IS in the $\sigma^2 = 0$ Case

Let us compare the convergence of GD and SGD-IS:

- GD
  - By Corollary 3, using stepsize $\gamma = \frac{1}{L}$, GD achieves iteration complexity

$$k \geq \frac{L}{\mu} \log \frac{1}{\varepsilon} \quad \Rightarrow \quad \left\|x^k - x^*\right\|^2 \leq \varepsilon \left\|x^0 - x^*\right\|^2$$

  - Cost of 1 iteration: n gradient evaluations
  - **Total complexity:** $\tilde{\mathcal{O}}\left(\frac{nL}{\mu}\right)$
- SGD-IS
  - By Corollary 12, using the importance sampling probabilities $p_i = \frac{L_i}{\sum_j L_j}$, stepsize $\gamma = \frac{1}{\mathcal{L}} = \frac{1}{\bar{L}}$, SGD-IS achieves iteration complexity

$$k \geq \frac{\bar{L}}{\mu} \log \frac{1}{\varepsilon} \quad \Rightarrow \quad \mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq \varepsilon \left\|x^0 - x^*\right\|^2$$

  - Cost of 1 iteration: 1 gradient evaluation
  - **Total complexity:** $\tilde{\mathcal{O}}\left(\frac{\bar{L}}{\mu}\right)$

Conclusion: Note that $nL \leq \sum_{i=1}^n L_i = n\bar{L}$, and this inequality can be tight, so that $L = \bar{L}$. So, SGD-IS **can be as much as n times faster than** GD**!**

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

Part 6: SGD via Stochastic Reformulation

Based on:

[7] R.M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P.R. **SGD: General Analysis and Improved Rates**, ICML 2019

Summary

- SGD-SR: a vast **generalization of the previous methods,** one that includes GD, SGD-US and SGD-NS as special cases.
- Sheds light on **minibatching** and gives a **formula for the optimal minibatch size**
- Smoothness (GD), average smoothness (SGD-US), and weighted smoothness (SGD-NS) get generalized to **expected smoothness.**
  - **See also Robert M. Gower's talk at ICCOPT 2019:** R.M. Gower, Expected smoothness is the key to understanding the mini-batch complexity of stochastic gradient methods, Location: H 0104, Symposium: Recent Advancements in Optimization Methods for Machine Learning (Part IV), **Session: Wednesday 16:00–17:15, Talk time: 16:25–16:50**
- Development based on the idea of **stochastic reformulation** of deterministic problems
  - idea pioneered by Takáč and R [22] in the **context of stochastic quadratic optimization problems / linear feasibility**
  - generalized to **convex feasibility** by Necoara, R and Patrascu [17]
    - solved an open problem related to the efficiency of extrapolated parallel projection methods
  - extended to finite-sum problems in the context of variance reduction (**"controlled" stochastic reformulation**) by Gower, R and Bach [6]
  - **adapted to non-variance reduced SGD methods** by Gower et al [7]

Go to External Slides

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 7: Shifted SGD

Based on:

[5] E. Gorbunov, F. Hanzely and P.R., **A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent**, arXiv1905.11261, 2019

[9] F. Hanzely and P.R., **One method to rule them all: variance reduction for data, parameters and many new methods**, arXiv:1905.11266, 2019

Motivation

As before, we consider the regularized **finite-sum** problem

$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^n f_i(x)}_{f(x)} + R(x). \tag{47}$$

- Recall that SGD does not converge to the solution x^* if $D_1 > 0$ or $D_2 > 0$, i.e., if it is not a **variance reduced method.**
- In particular, SGD-US does not converge to the solution x^* if the variance of the stochastic gradients at the optimum ("gradient noise" for short), i.e.,

$$\sigma^2 \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*) - \nabla f(x^*)\|^2,$$

  is positive.
- **Main idea: Can we force σ^2 to be zero via some simple algebraic trick right at the start?**
  - Answer: yes!
  - This leads to the SGD-SHIFT method [5, 9]
  - The method **can't be implemented!**
  - The method **gives intuition into what variance reduction is about!**

Reformulation by Shifting the Functions: the Idea

1. Define the **shifted functions**

$$\phi_i(x) \stackrel{\text{def}}{=} f_i(x) - \langle a_i, x\rangle, \quad \forall i = 1, 2, \ldots, n,$$

where $a_i \in \mathbb{R}^d$ are some vectors to be determined.
   - If f_i is convex and L_i-smooth, so is ϕ_i
   - The **gradients are shifted:** $\nabla\phi_i(x) = \nabla f_i(x) - a_i$
2. We now want to and **apply SGD-US to a reformulated problem:**

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

3. In order for the new problem to be **equivalent** to the original one, we need to make sure that $f \stackrel{\text{def}}{=} \frac{1}{n}\sum_i f_i = \frac{1}{n}\sum_i \phi_i \stackrel{\text{def}}{=} \phi$. This is achieved if the **aggregate shift is zero:** $\sum_{i=1}^n a_i = 0$.
4. **In order to get $\sigma^2 = 0$,** we want $\nabla\phi_i(x^*) = \nabla\phi(x^*)$ for all i:

$$\nabla f_i(x^*) - a_i = \tfrac{1}{n}\textstyle\sum_j (\nabla f_j(x^*) - a_j), \quad \forall i$$

Solving for a_i, we get

$$a_i = \nabla f_i(x^*) - \tfrac{1}{n}\textstyle\sum_i \nabla f_j(x^*) = \nabla f_i(x^*) - \nabla f(x^*).$$

Reformulation by Shifting the Functions: Summary

So, instead of (60), we are solving the **shifted problem**

$$\boxed{\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \phi_i(x) + R(x)} \tag{48}$$

where the individual functions are

$$\phi_i(x) \stackrel{\text{def}}{=} \nabla f_i(x) - \langle \underbrace{\nabla f_i(x^*) - \nabla f(x^*)}_{a_i}, x\rangle \tag{49}$$

and the **stochastic gradient is given by**

$$\nabla\phi_i(x) = \nabla f_i(x) - \nabla f_i(x^*) + \nabla f(x^*)$$

Shifted SGD

We now apply SGD-US to the shifted problem (48)+(49):

$$\boxed{x^{k+1} = \operatorname{prox}_{\gamma R}\left(x^k - \gamma \nabla \phi_i(x^k)\right)}$$

where i is chosen uniformly at random from $\{1, 2, \ldots, n\}$.

Writing this method in the notation of the original problem (60), we get the SGD-SHIFT method:

Algorithm 6 SGD-SHIFT

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$, gradients at the optimum $\nabla f_1(x^*), \ldots, \nabla f_n(x^*)$
Set $\nabla f(x^*) = \frac{1}{n}\sum_i \nabla f_i(x^*)$
for $k = 0, 1, 2, \ldots$ **do**
    Sample $i^k = i \in \{1, 2, \ldots, n\}$ with probability $\frac{1}{n}$
    $g^k = \nabla f_i(x^k) - \nabla f_i(x^*) + \nabla f(x^*)$
    $x^{k+1} = \operatorname{prox}_{\gamma R}(x^k - \gamma g^k)$

SGD-SHIFT is utterly impractical since we do not know the optimal gradients $\nabla f_1(x^*), \ldots, \nabla f_n(x^*)$!

Shifted SGD: Theory

Since by design $\sigma^2 = 0$ for the reformulated problem, it follows from Theorem 2 that the random iterates of SGD-SHIFT converge to x^*, at a linear rate.

Applying Exercise 2 in combination with Corollary 8, we get the following result:

Corollary 13

Assume that

- *The problem* $\min f + R$ *has a unique solution (Assumption 3)*
- f *is* μ*-quasi strongly convex (Assumption 4)*
- f_i *are convex and* L_i*-smooth (Assumption 6)*

SGD-SHIFT *allows for the larger stepsize* $0 < \gamma \leq \frac{1}{\max_i L_i}$*, and its iterates converge as*

$$\mathbf{E}\left[\left\|x^k - x^*\right\|^2\right] \leq (1 - \gamma\mu)^k \left\|x^0 - x^*\right\|^2. \tag{50}$$

SGD-SHIFT vs GD: Understanding What Variance Reduced Methods Do

- The gradient estimator of SGD-SHIFT utilizes a **fixed and impossible-to-evaluate shift:**

$$\nabla\phi_i(x^k) = \nabla f_i(x^k) - \nabla f_i(x^*) + \nabla f(x^*)$$

- The gradient estimator of GD has a similar structure, but has a **variable and costly-to-evaluate shift:**

$$\nabla f(x^k) = \nabla f_i(x^k) - \nabla f_i(x^k) + \nabla f(x^k) \tag{51}$$

Insight: Variance reduced methods (such as SVRG, S2GD, SAGA, L-SVRG, MISO, Finito, JacSketch) achieve similar effect (convergence to x^* as opposed to just a neighbourhood) by **doing something in between of what GD and SGD-SHIFT do. They use the information accumulated across the iterates to progressively learn the gradients at the optimum!**

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 8: Loopless SVRG

Based on:

[10] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. **Variance reduced stochastic gradient descent with neighbors.** NIPS 2015

[15] D. Kovalev, S. Horváth and P. R. **Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop**, arXiv:1901.08689, 2019

Motivation

- So far we have seen a few variance-reduced SGD methods, **all with serious issues:**
  - GD: this methods needs to perform n gradient computations in each iteration, which is **inefficient.**
  - SGD-US, SGD-SR: in the special **lucky** case when $\sigma^2 = 0$ (e.g., in the "over-parameterized" regime when $\nabla f_i(x^*) = 0$ for all i)
  - SGD-SHIFT: this method is practically **not implementable.**
- We will now describe and analyze the first variance-reduced method which is both **efficient** and **implementable**:

Loopless SVRG (L-SVRG)

- L-SVRG was independently developed by Hofmann et al [10] (earlier) and Kovalev & R [15].
  - The name L-SVRG was coined in [15]
  - Here we follow the general analysis in [5] (Theorem 2) to recover the result from [15]
- L-SVRG is a **"loopless" variant of the famous SVRG method** of Johnson and Zhang [13]. See also S2GD [14] which appeared in the same year (2013).

Loopless SVRG: the Algorithm

Using interpretation (51) of GD, **L-SVRG can be interpreted as "lazy" gradient descent**

Algorithm 7 Loopless SVRG (L-SVRG) [15]

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$, probability $0 < p \leq 1$

$w^0 = x^0$ (initialize the reference point)

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

  Sample $i^k = i \in \{1, 2, \ldots, n\}$ with probability $\frac{1}{n}$

  $g^k = \nabla f_i(x^k) - \nabla f_i(w^k) + \nabla f(w^k)$

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

  Update the reference point:

$$w^{k+1} = \begin{cases} x^k & \text{with probability } p \\ w^k & \text{with probability } 1-p \end{cases}$$

Exercise 5

Show that the estimator is unbiased $\mathbf{E}\left[g^k \mid x^k, w^k\right] = \nabla f(x^k)$.

Cost per Iteration

In each iteration, one has to compute

- 1 gradient: $\nabla f_i(x^k)$ (always)
- n gradients: $\nabla f(w^k) = \frac{1}{n}\sum_i \nabla f_i(w^k)$ (with probability p)
- 1 gradients: $\nabla f_i(w^k)$ (with probability $1-p$)

So, the expected # of gradient evaluations per 1 iteration of L-SVRG is:

$$\begin{aligned} Cost &= 1 + pn + (1-p)1 \\ &= 2 + p(n-1). \end{aligned}$$

- A practical (and theoretically optimal) choice is $p = \frac{1}{n}$, for which

$$Cost \leq 3 = \mathcal{O}(1). \tag{52}$$

- Another good (and theoretically optimal) choice is $p = \frac{\mu}{L_{\max}}$, for which

$$Cost = \mathcal{O}\left(1 + \frac{n\mu}{L_{\max}}\right). \tag{53}$$

Key Lemma

We now show that L-SVRG satisfies Assumption 1.

Lemma 14 (Lemmas 4.2–4.3 from [15] extended to prox setup)

Let Assumptions 6 and 3 be satisfied. Then

$$\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \leq 4L_{\max} D_f(x^k, x^*) + 2\sigma_k^2 \tag{54}$$

and

$$\mathbf{E}\left[\sigma_{k+1}^2 \mid x^k\right] \leq (1-p)\sigma_k^2 + 2L_{\max} p D_f(x^k, x^*), \tag{55}$$

where

$$\sigma_k^2 \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(w^k) - \nabla f_i(x^*)\right\|^2. \tag{56}$$

- So, Assumption 1 is satisfied with

$$A = 2L_{\max},\ B = 2,\ D_1 = 0,\ \sigma_k = (56),\ \rho = p,\ C = Lp,\ D_2 = 0.$$

- Since $D_1 = D_2 = 0$, L-SVRG is a **variance-reduced** variant of SGD

Convergence of L-SVRG

Corollary 15 (Convergence of L-SVRG)

Let the following assumptions be satisfied:

- *Assumptions 3 (unique solution),*
- *Assumption 4 (μ-quasi strong convexity) and*
- *Assumption 6 (convexity and L_i-smoothness of f_i).*

Then L-SVRG *with* $\gamma = \frac{1}{6L_{\max}}$ *satisfies*

$$\mathbf{E}\left[V^k\right] \leq \left(1 - \min\left\{\frac{\mu}{6L_{\max}}, \frac{p}{2}\right\}\right)^k V^0, \tag{57}$$

where

$$V^k \overset{(18)}{=} \left\|x^k - x^*\right\|^2 + M\gamma^2\sigma_k^2 \overset{(56)}{=} \left\|x^k - x^*\right\|^2 + \frac{1}{9L_{\max}^2 pn}\sum_{i=1}^n \left\|\nabla f_i(w^k) - \nabla f_i(x^*)\right\|^2$$

Exercise 6

Formally perform the proof of Corollary 15. Hint: Apply Theorem 2 to Algorithm 7 with $M = \frac{4}{p}$.

Total Complexity

Number of iterations: In view of Corollary 15, we have

$$k \geq \max\left\{\frac{6L_{\max}}{\mu}, \frac{2}{p}\right\}\log\frac{1}{\varepsilon} \quad \Rightarrow \quad \mathbf{E}\left[V^k\right] \leq \varepsilon V^0.$$

Cost of 1 iteration:

- If we choose $p = \frac{1}{n}$, in view of (52) the expected cost of 1 iteration is $Cost \leq 3 = \mathcal{O}(1)$. Hence, the total complexity of L-SVRG is

$$\text{Total complexity} = \mathcal{O}\left(\max\left\{\frac{L_{\max}}{\mu}, n\right\}\log\frac{1}{\varepsilon}\right) \tag{58}$$

- If we choose $p = \frac{\mu}{L_{\max}}$, in view of (53) the expected cost of 1 iteration is $Cost = \mathcal{O}\left(1 + \frac{n\mu}{L_{\max}}\right)$. Hence, the total complexity of L-SVRG is

$$\text{Total complexity} = \mathcal{O}\left(\max\left\{\frac{L_{\max}}{\mu}, n\right\}\log\frac{1}{\varepsilon}\right) \tag{59}$$

Comparison of Total Complexities

Let us compare the complexities of the variance-reduced methods L-SVRG with SGD-US:

Method	GD	SGD-US for $\sigma^2 = 0$	L-SVRG $p = \mathcal{O}(\frac{1}{n})$ or $p = \mathcal{O}(\frac{\mu}{L_{\max}})$
Total complexity	$\frac{nL}{\mu} \log \frac{1}{\varepsilon}$	$\frac{L_{\max}}{\mu} \log \frac{1}{\varepsilon}$	$\max\left\{\frac{L_{\max}}{\mu}, n\right\} \log \frac{1}{\varepsilon}$

- **Big data regime:** $n \geq \frac{L_{\max}}{\mu}$
  - L-SVRG is slower than SGD-US
  - But one needs to be very lucky for $\sigma^2 = 0$ to be the case!
  - If $\sigma^2 > 0$, then SGD-US converges to a neighbourhood of the solution only!
- **Small data regime:** $n \leq \frac{L_{\max}}{\mu}$
  - L-SVRG and SGD-US are equally fast

Insight

Insight: When we are (very) lucky to be in the $\sigma^2 = 0$ regime, variance reduction is not needed in the first place, and SGD-US is better than L-SVRG in the big data regime. They are the same in the small data regime. However, **when variance reduction is desirable ($\sigma^2 > 0$), then L-SVRG shines: it delivers on its promise and is an efficient method which converges as fast (in the small data regime) or almost as fast (in the big data regime) to x^* as if variance reduction as needed at all.**

Proof of Lemma 14

All expectations in the first part of the proof are conditioned on x^k and w^k. Using the definition of g^k, we get:

$$\begin{aligned}
\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2\right] &\overset{\text{L-SVRG}}{=} \mathbf{E}\left[\left\|\nabla f_i(x^k) - \nabla f_i(x^*) + \nabla f_i(x^*) - \nabla f_i(w^k) + \nabla f(w^k) - \nabla f(x^*)\right\|^2\right] \\
&\overset{(21)}{\leq} 2\mathbf{E}\left[\left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2\right] \\
&\quad + 2\mathbf{E}\left[\left\|\nabla f_i(x^*) - \nabla f_i(w^k) - \mathbf{E}\left[\nabla f_i(x^*) - \nabla f_i(w^k)\right]\right\|^2\right] \\
&\overset{(27)+(23)}{\leq} 2\mathbf{E}\left[2L_i D_{f_i}(x^k, x^*)\right] \\
&\quad + 2\mathbf{E}\left[\left\|\nabla f_i(w^k) - \nabla f_i(x^*)\right\|^2\right] \\
&\leq 4L_{\max} D_f(x^k, x^*) + 2\sigma_k^2.
\end{aligned}$$

The second recursion can be obtained directly from the definition (56) of σ_k^2 and the update rule for w^k:

$$\begin{aligned}
\mathbf{E}\left[\sigma_{k+1}^2 \mid x^k\right] &= (1-p)\sigma_k^2 + p\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2 \\
&\overset{(27)}{\leq} (1-p)\sigma_k^2 + p\frac{1}{n}\sum_{i=1}^n 2L_i D_{f_i}(x^k, x^*) \\
&= (1-p)\sigma_k^2 + 2L_{\max} p D_f(x^k, x^*).
\end{aligned}$$

Extensions

- **Arbitrary sampling:** L-SVRG with arbitrary sampling (i.e., a subset of $\{1, 2, \ldots, n\}$ can be sampled to form g^k in each step, following an arbitrary distribution) was developed and analyzed by Qu, Qian and R [21].
- **Acceleration:** accelerated variant (L-Katyusha) was also developed in [21]. Katyusha was developed by Alen-Zhu [2].
- **Beyond strong convexity:** An analysis in the convex and smooth nonconvex can be found in [21].

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 10: Distributed Optimization with Compressed Communication via DIANA

Based on:

[16] K. Mishchenko, E. Gorbunov, M. Takáč and P. R. **Distributed Learning with Compressed Gradient Differences**, arXiv:1901.09269, 2019

[12] S. Horváth, D. Kovalev, K. Mishchenko, P. R. and S. Stich, **Stochastic Distributed Learning with Gradient Quantization and Variance Reduction**, arXiv:1904.05115, 2019

Motivation

We are still considering the regularized **finite-sum** problem

$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} f_i(x)}_{f(x)} + R(x). \tag{60}$$

We now move onto **distributed optimization:**

- We utilize a distributed compute system because:
  - f is described by **too much data to be stored on a singe computer,** and hence the data (i.e., functions) need to be stored on different computers of distributed system (e.g., cluster, supercomputer)
  - a **single computer is not powerful enough** for the task at hand and we have access to multiple computers.
- There are n machines which can work in parallel.
- Machine i contains information about:
  - f_i (this function can in turn be very complex)
  - regularizer R

Distributed Gradient Descent

- We **want all machines to be utilized**, so in each iteration we want to access all n functions $f_1, \dots, f_n$
- Ideally we would want to implement GD:
  1. Each machine computes $\nabla f_i(x^k)$
  2. These gradients are sent to a **parameter server**, which aggregates (i.e., averages) them and broadcasts the average

$$\nabla f(x^k) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x^k)$$

     back to the nodes
  3. Each node performs the prox gradient step to obtain x^{k+1}:

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

  4. The process is repeated
- **Key Issue: Communication of the (often very high dimensional) gradient $\nabla f_i(x^k) \in \mathbb{R}^d$ from node i to the parameter server over a network is very slow, and forms the bottleneck.**

Gradient Compression

A popular solution to the communication bottleneck is **gradient compression:**

- The idea is for each node/machine to replace $\nabla f_i(x^k) \in \mathbb{R}^d$ by $\mathcal{C}(\nabla f_i(x^k)) \in \mathbb{R}^d$, where

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

  is some (often randomized) **compression operator.**
- **Trade-off between accuracy and compression:** We want the compressed gradient to be small in size, so that it is easy to communicate. But the more we compress, the less accurate it will be, which will adversely affect the algorithm.
- **Intuition behind compression vs convergence:**
  - **No compression:** recovers gradient decent (GD)
  - **A bit of compression:** may lead to improvement in the overall complexity
  - **A lot of compression:** will lead to a worse performance than no compression.
  - **Too much compression:** will break the method altogether (divergence).

Compression Operators

Definition 16 (Compression Operator)

We say that a randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is an ω-compression operator ($\omega > 0$) if for all $x \in \mathbb{R}^d$ it satisfies

$$\mathbf{E}\left[\mathcal{C}(x)\right] = x, \qquad \mathbf{E}\left[\left\|\mathcal{C}(x) - x\right\|^2\right] \leq \omega \left\|x\right\|^2. \tag{61}$$

We write $\mathcal{C} \in \mathbb{B}(\omega)$.

Examples:

- Random sparsification
- Random dithering
- Natural compression [11]
- Natural dithering [11] (exponentially better than random dithering)

A Naive Variant of Gradient Descent with Compression

Algorithm 8 GD with Compression (`GD-compress`)

Parameters: learning rate $\gamma > 0$, starting point $x^0 \in \mathbb{R}^d$, compression operator $\mathcal{C}$
for $k = 0, 1, 2, \ldots$ **do**
  for all nodes $i \in \{1, 2, \ldots, n\}$ **in parallel do**
    Compute local gradient $\nabla f_i(x^k)$
    Compress local gradient $g_i^k = \mathcal{C}(\nabla f_i(x^k))$
    Send g_i^k to parameter server
    Receive the aggregate $g^k = \frac{1}{n}\sum_{i=1}^n g_i^k$
    $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma g^k)$

This method is **naive** since:

- It is **not known to converge in the $R \neq 0$ case**
- We have turned `GD` into `SGD`, and hence **have to live with convergence to a neighbourhood of the solution**

The DIANA Algorithm

DIANA: A Method That Fixes These Issues

DIANA (Mishchenko et al [16]):

- is the first method which **fixes the issues** of the naive method.
- is a **variance-reduction strategy for tackling the variance introduced by the compression operators**
- maintains local estimates $\{h_i^k\}$ of the gradients at the optimum $(\nabla f_i(x^*))$ and **compresses the difference** between the local gradient and h_i^k
  - Since $h_i^k \to \nabla f_i(x^*)$ and $\nabla f_i(x^k) \to \nabla f_i(x^*)$ (which needs to be proved), then **the gradient differences converge to zero**

$$\nabla f_i(x^k) - h_i^k \to 0.$$

  - As a result, the **compression introduces less and less variance,**
  - And hence DIANA is able to reduce the variance introduced by compression.

The Variance Reduction Technique Behind DIANA

Method	Alg #	Gradient Estimator $g^k =$
GD		$\nabla f_i(x^k) - \nabla f_i(x^k) + \nabla f(x^k)$
SGD-SHIFT	Alg 6	$\nabla f_i(x^k) - \nabla f_i(x^*) + \nabla f(x^*)$
L-SVRG	Alg 7	$\nabla f_i(x^k) - \nabla f_i(w^k) + \nabla f(w^k)$
DIANA (1 node)		$\mathcal{C}(\nabla f(x^k) - h^k) + h^k$
DIANA (n nodes)	Alg 9	$\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^k) - h_i^k) + h_i^k$

DIANA: the Algorithm

Algorithm 9 DIANA [16, 12]

Parameters: learning rates $\alpha > 0$ and $\gamma > 0$, initial iterate $x^0 \in \mathbb{R}^d$; initial vectors $h_1^0, \ldots, h_n^0 \in \mathbb{R}^d$ (stored on the nodes)
Initialize: $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$ (stored on the master)
for $k = 0, 1, \ldots$ **do**
  Broadcast x^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    Compute $\nabla f_i(x^k)$
    $\Delta_i^k = \nabla f_i(x^k) - h_i^k$
    Compress $\hat{\Delta}_i^k = \mathcal{C}(\Delta_i^k)$
    $h_i^{k+1} = h_i^k + \alpha\hat{\Delta}_i^k$
    $\hat{g}_i^k = h_i^k + \hat{\Delta}_i^k$
  Aggregate received messages $\hat{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \hat{\Delta}_i^k$
  Compute gradient estimator $g^k = \frac{1}{n}\sum_{i=1}^n \hat{g}_i^k = h^k + \hat{\Delta}^k$
  Take proximal SGD step $x^{k+1} = \mathrm{prox}_{\gamma R}\left(x^k - \gamma g^k\right)$
  $h^{k+1} = \frac{1}{n}\sum_{i=1}^n h_i^{k+1} = h^k + \alpha\hat{\Delta}^k$

Key Lemma

Lemma 17 (Lemma 1 and consequence of Lemma 2 from [12])

Suppose that $\alpha \leq \frac{1}{1+\omega}$. For all iterations $k \geq 0$ of Algorithm 9 it holds

$$\mathbf{E}\left[g^k \mid x^k\right] = \nabla f(x^k), \tag{62}$$

$$\mathbf{E}\left[\left\|g^k - \nabla f(x^*)\right\|^2 \mid x^k\right] \leq \left(1 + \frac{2\omega}{n}\right)\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2 \tag{63}$$

$$+\frac{2\omega\sigma_k^2}{n}, \tag{64}$$

$$\mathbf{E}\left[\sigma_{k+1}^2 \mid x^k\right] \leq (1-\alpha)\sigma_k^2 + \frac{\alpha}{n}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2. \tag{65}$$

where $\sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \left\|h_i^k - \nabla f_i(x^*)\right\|^2$.

Bounding further $\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|^2 \leq 2L_{\max} D_f(x^k, x^*)$ in the above lemma, we see that Assumption 1 holds. So, Theorem 2 applies.

Convergence of DIANA

Corollary 18

Assume that

- *f_i is convex and L_i-smooth for all $i \in [n]$*
- *and f is μ-quasi strongly convex,*

If the stepsizes satisfy $\alpha \leq \frac{1}{\omega+1}$, $\gamma \leq \frac{1}{\left(1+\frac{2\omega}{n}\right)L_{\max} + ML_{\max}\alpha}$, where $M > \frac{2\omega}{n\alpha}$, then the iterates of DIANA *satisfy*

$$\mathbf{E}\left[V^k\right] \leq \max\left\{(1-\gamma\mu)^k, \left(1 + \frac{2\omega}{nM} - \alpha\right)^k\right\} V^0, \tag{66}$$

where the Lyapunov function V^k is defined by

$$V^k \overset{def}{=} \left\|x^k - x^*\right\|^2 + M\gamma^2\sigma_k^2.$$

Understanding the Rate

For the particular choice

$$\alpha = \frac{1}{\omega + 1}, \qquad M = \frac{4\omega(\omega+1)}{n}, \qquad \gamma = \frac{1}{\left(1 + \frac{6\omega}{n}\right) L_{\max}},$$

the iteration complexity of DIANA is

$$\max\left\{\frac{1}{\gamma\mu}, \frac{1}{\alpha - \frac{2\omega}{nM}}\right\} \log\frac{1}{\varepsilon} = \boxed{\max\left\{\kappa + \kappa\frac{6\omega}{n}, 2(\omega+1)\right\} \log\frac{1}{\varepsilon}} \qquad (67)$$

where $\kappa \overset{\text{def}}{=} \frac{L_{\max}}{\mu}$.

Key Insight: As long as $\omega = \mathcal{O}(\min\{n, \kappa\})$, the iteration complexity of DIANA is the same as that of GD: $\mathcal{O}\left(\kappa \log\frac{1}{\varepsilon}\right)$. However, by allowing for compression, we communicate less, and so the overall complexity can improve.

Examples of Compression Operators

Random Sparsification

Definition 19 (Random sparsification)

Fix $r \in \{1, 2, \ldots, d\}$ and let $\xi \in \mathbb{R}^d$ be a (uniformly distributed) random binary vector with r nonzero entries. The random sparsification operator is given by

$$\mathcal{C}(x) = \frac{d}{r}(\xi \odot x),$$

where $\odot$ denotes the Hadamard (entry-wise) product.

Remarks:

- $\mathcal{C}(x)$ has at most r nonzero entries (if x was dense, then exactly r).
- Since $\xi_i = 1$ with probability $\frac{r}{d}$ and zero otherwise,

$$\mathbf{E}[\mathcal{C}(x)_i] = \frac{d}{r}\mathbf{E}[(\xi \odot x)_i] = \frac{d}{r}\left(\frac{r}{d}1 \cdot x_i + \left(1 - \frac{r}{d}\right)0 \cdot x_i\right) = x_i,$$

  and hence $\mathcal{C}$ is **unbiased.**
- It can be shown that $\mathcal{C} \in \mathbb{B}(\omega)$ for $\omega = \frac{d}{r} - 1$ (Stich et al [26]).

Random Dithering

Definition 20 (Random dithering)

Fix $p \in [1, \infty]$ and number of levels $s \in \{1, 2, \ldots\}$. The random dithering operator $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is defined as

$$\mathcal{C}(x) = \text{sign}(x) \cdot \|x\|_p \cdot \frac{1}{s} \cdot \left\lfloor s\frac{|x|}{\|x\|_p} + \xi \right\rfloor,$$

where $\text{sign}(x) \in \mathbb{R}^d$ and $|x| \in \mathbb{R}^d$ are applied entry-wise, $\xi \in \mathbb{R}^d$ is a random vector with independent entries uniformly distributed on $[0, 1]$, and $\|x\|_p \overset{\text{def}}{=} (\sum_i |x_i|^p)^{1/p}$ is the L_p norm.

Remarks:

- For $p = 2$ used in the `QSGD` method of Alistarh et al [1]
- For $p = \infty$ and $s = 1$ used (no theory) in the `Terngrad` method of Wen et al [28]
- For $p \geq 1$ and $s = 1$ studied in [16]

Properties of Random Dithering

Lemma 21 (Alistarh et al [1] for $p = 2$; Horváth et al [12] for general p)

Random dithering operator $\mathcal{C}$ is unbiased (i.e., $\mathbf{E}[\mathcal{C}(x)] = x$ for all x), and for any fixed $x \in \mathbb{R}^d$, it satisfies (61) with

$$\omega(x) \overset{def}{=} 2 + \frac{\|x\|_1 \|x\|_\infty}{s\|x\|_2^2},$$

which is a decreasing function in p. Moreover, $\mathcal{C} \in \mathbb{B}(\omega)$ for

$$\omega = \mathcal{O}\left(\frac{d^{1/p} + d^{1/2}}{s}\right)$$

for all $p \geq 1$ and $s \geq 1$.

Communication:

- For $p = 2$, the **expected density** of the compressed vector is [1]

$$\mathbf{E}[\|\mathcal{C}(x)\|_0] = \mathcal{O}(s(s + \sqrt{d}))$$

- **Encoding** a nonzero coordinate of $\mathcal{C}(x)$ requires $\mathcal{O}(\log s)$ bits.

Natural Compression

- **Natural compression** $\mathcal{C}_{\text{nat}}$ will be applied element-wise to x: $(\mathcal{C}_{\text{nat}}(x))_i = \mathcal{C}_{\text{nat}}(x_i)$.
- Natural compression performs a **randomized rounding of its input $t \in \mathbb{R}$ to one of the two closest integer powers of 2.**
- Given nonzero t, let $\alpha \in \mathbb{R}$ be such that $|t| = 2^\alpha$ (i.e., $\alpha = \log_2 |t|$). Then

$$2^{\lfloor \alpha \rfloor} \leq |t| = 2^\alpha \leq 2^{\lceil \alpha \rceil} \tag{68}$$

  and we round t to either $\text{sign}(t)2^{\lfloor \alpha \rfloor}$, or to $\text{sign}(t)2^{\lceil \alpha \rceil}$.
- When $t = 0$, we set $\mathcal{C}_{\text{nat}}(0) = 0$.
- **The probabilities are chosen so that $\mathcal{C}_{\text{nat}}(t)$ is an unbiased estimator of t,** i.e., $\mathbf{E}[\mathcal{C}_{\text{nat}}(t)] = t$ for all t.
- If t is an integer power of 2, then $\mathcal{C}_{\text{nat}}$ will leave t unchanged.

Lemma 22 (Horváth et al [11])

$\mathcal{C}_{\text{nat}} \in \mathbb{B}(\frac{1}{8})$.

Natural Compression: Examples

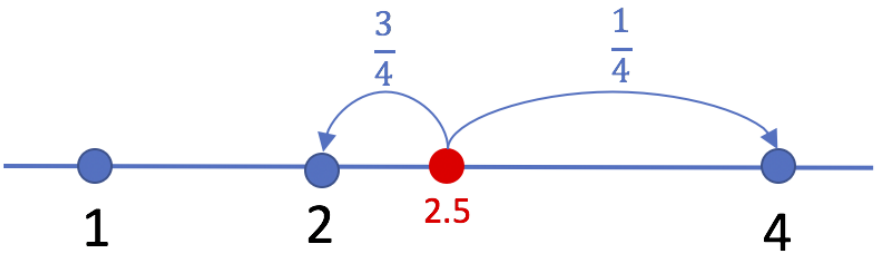

Figure: An illustration of natural compression applied to $t = 2.5$: $\mathcal{C}_{\rm nat}(2.5) = 2$ with probability $\frac{4-2.5}{2} = 0.75$, and $\mathcal{C}_{\rm nat}(2.5) = 4$ with probability $\frac{2.5-2}{2} = 0.25$. This choice of probabilities ensures that the compression operator is unbiased, i.e., $\mathbf{E}\left[\mathcal{C}_{\rm nat}(t)\right] = t$ for all t.

Example 23

- $t = -2.75$ will be rounded to either -4 or -2 (since $-2^2 \leq -2.75 \leq -2^1$)
- $t = 0.75$ will be rounded to either $\frac{1}{2}$ or 1 (since $2^{-1} \leq 0.75 \leq 2^0$)

Implementation of Natural Compression I

- Performing natural compression of a real number in a binary floating point format is computationally cheap.
- Excluding the randomization step, $\mathcal{C}_{\rm nat}$ amounts to simply **dispensing off the mantissa in the binary representation.**

Implementation of Natural Compression II

Binary32

- The most common computer format for real numbers, **binary**32 (resp. *binary*64) **of the IEEE 754 standard, represents each number with** 32 (resp. 64) **bits,** where the first bit represents the sign, 8 (resp. 11) bits are used for the exponent, and the remaining 23 (resp. 52) bits are used for the mantissa.
- A scalar $t \in \mathbb{R}$ is represented in the form $(s, e_7, e_6, \ldots, e_0, m_1, m_2, \ldots, m_{23})$, where $s, e_i, m_j \in \{0, 1\}$ are bits, via the relationship

$$t = (-1)^s \times 2^{e-127} \times (1+m), \quad e = \sum_{i=0}^{7} e_i 2^i, \quad m = \sum_{j=1}^{23} m_j 2^{-j}, \quad (69)$$

where s is the **sign**, e is the **exponent** and m is the **mantissa**.

Implementation of Natural Compression III

Example 24

A *binary*32 representation of $t = -2.75$ is visualized below. In this case, $s = 1$, $e_7 = 1$, $m_2 = m_3 = 1$ and hence

$$t = (-1)^s \times 2^{e-127} \times (1 + m) = -1 \times 2 \times (1 + 2^{-2} + 2^{-3}) = -2.75.$$

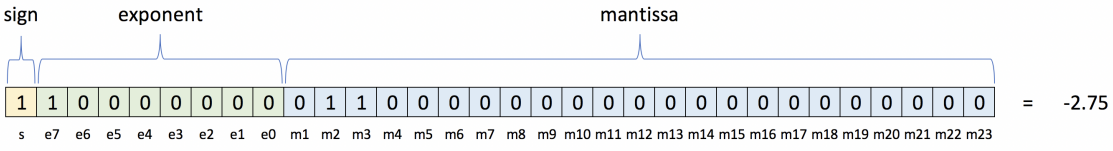

Figure: IEEE 754 single-precision binary floating-point format: *binary*32.

- It is clear from (69) that $0 \leq m < 1$, and hence $2^{e-127} \leq |t| < 2^{e-126}$ (compare this with (68)). Moreover, $p(t) = \frac{2^{e-126} - |t|}{2^{e-127}} = 2 - |t|2^{127-e} = 1 - m$.

Implementation of Natural Compression IV

- Hence, **natural compression of t represented as binary32 is given as follows:**

$$\mathcal{C}_{\rm nat}(t) = \begin{cases} (-1)^s \times 2^{e-127}, & \text{with probability } 1-m, \\ (-1)^s \times 2^{e-126}, & \text{with probability } m. \end{cases}$$

- Observe that $(-1)^s \times 2^{e-127}$ is **obtained from t by setting the mantissa m to zero, and keeping both the sign s and exponent e unchanged.**
- Similarly, $(-1)^s \times 2^{e-126}$ is **obtained from t by setting the mantissa m to zero, keeping the sign s, and increasing the exponent by one, which amounts to a simple shift of the bits forming the exponent to the left by one spot.**
- Hence, **both values can be computed from t essentially without any computation.**

Implementation of Natural Compression V

Communication savings

- In case of $\mathrm{binary32}$, the output $\mathcal{C}_{\rm nat}(t)$ of natural compression is encoded using the 8 bits in the exponent and an extra bit for the sign. **This is $3.56\times$ less communication.**
- In case of $\mathrm{binary64}$, we only need 11 bits for the exponent and 1 bit for the sign, and **this is $5.82\times$ less communication.**

A Guided Walk Through the ZOO of Stochastic Gradient Descent Methods

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

Part 9: SEGA

Based on:

[8] F. Hanzely, K. Mishchenko and P. R.
SEGA: Variance reduction via gradient sketching, NeurIPS 2018

[5] E. Gorbunov, F. Hanzely and P.R., **A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent**, arXiv1905.11261, 2019

The Setup

Consider solving the problem

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

in situations when

- the dimension d **is very large,** and
- we only have access to **random linear transformations of the gradient of f:**

$$S^\top \nabla f(x^k), \qquad S \quad \text{is a random matrix}$$

About SEGA

- The **first method designed to work in this regime**
- SEGA is a **variance reduction strategy** for the variance implied by the random measurement process.
- As other variance-reduced methods we have seen, SEGA also maintains a sequence of auxiliary iterates h^k converging to $\nabla f(x^*)$
- Under the typical assumptions of this course (e.g., smoothness and strong convexity), SEGA **can solve the problem to optimality, at a linear rate.**
- When specialized to **coordinate sketches**, SEGA enjoys the same complexity bounds (up to small constants) as state-of-the-art **coordinate descent methods**
- It's the first **coordinate descent type method** which works with any R, and not just (block) separable R
- Developed by Hanzely, Mishchenko and R [8]

Go to External Slides
(but there won't be time for this)

Bibliography I

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic.
QSGD: Communication-efficient SGD via gradient quantization and encoding.
In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] Zeyuan Allen-Zhu.
Katyusha: The first direct acceleration of stochastic gradient methods.
In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.

[3] Amir Beck.
First order methods in optimization.
MOS-SIAM Series on Optimization, 2017.

[4] El Houcine Bergou, Eduard Gorbunov, and Peter Richtárik.
Stochastic three points method for unconstrained smooth minimization.
arXiv preprint arXiv:1902.03591, 2019.

[5] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik.
A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent.
arXiv preprint arXiv1905.11261, 2019.

Bibliography II

[6] Robert M Gower, Peter Richtárik, and Francis Bach.
Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching.
arXiv preprint arXiv:1805.02632, 2018.

[7] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik.
SGD: General analysis and improved rates.
In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[8] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik.
SEGA: Variance reduction via gradient sketching.
In *Advances in Neural Information Processing Systems 31*, pages 2082–2093, 2018.

[9] Filip Hanzely and Peter Richtárik.
One method to rule them all: variance reduction for data, parameters and many new methods.
arXiv preprint arXiv:1905.11266, 2019.

Bibliography III

[10] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams.
Variance reduced stochastic gradient descent with neighbors.
In *Advances in Neural Information Processing Systems*, pages 2305–2313, 2015.

[11] Samuel Horváth, Chen-Yu Ho, Ľudovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik.
Natural compression for distributed deep learning.
arXiv preprint arXiv:1905.10988, 2019.

[12] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Peter Richtárik, and Sebastian Stich.
Stochastic distributed learning with gradient quantization and variance reduction.

arXiv preprint arXiv:1904.05115, 2019.

[13] Rie Johnson and Tong Zhang.
Accelerating stochastic gradient descent using predictive variance reduction.
In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

Bibliography IV

[14] Jakub Konečný and Peter Richtárik.
Semi-stochastic gradient descent methods.
Frontiers in Applied Mathematics and Statistics, pages 1–14, 2017.

[15] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik.
Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop.
arXiv preprint arXiv:1901.08689, 2019.

[16] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik.
Distributed learning with compressed gradient differences.
arXiv preprint arXiv:1901.09269, 2019.

[17] Ion Necoara, Peter Richtárik, and Andrei Patrascu.
Randomized projection methods for convex feasibility problems: conditioning and convergence rates.
SIAM Journal on Optimization, 2019.

[18] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro.
Robust stochastic approximation approach to stochastic programming.
SIAM Journal on Optimization, 19(4):1574–1609, 2009.

Bibliography V

[19] Yurii Nesterov.
Introductory lectures on convex optimization: a basic course (Applied Optimization).
Kluwer Academic Publishers, 2004.

[20] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč.
SARAH: A novel method for machine learning problems using stochastic recursive gradient.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 2017.

[21] Xun Qian, Zheng Qu, and Peter Richtárik.
L-svrg and l-katyusha with arbitrary sampling.
arXiv preprint arXiv:1906.01481, 2019.

[22] Peter Richtárik and Martin Takáč.
Stochastic reformulations of linear systems: algorithms and convergence theory.
arXiv:1706.01108, 2017.

Bibliography VI

[23] H. Robbins and S. Monro.
A stochastic approximation method.
Annals of Mathematical Statistics, 22:400–407, 1951.

[24] Mark Schmidt, Nicolas Le Roux, and Francis Bach.
Minimizing finite sums with the stochastic average gradient.
Mathematical Programming, 162(1–2):83–112, 2017.

[25] Shai Shalev-Shwartz and Shai Ben-David.
Understanding machine learning: from theory to algorithms.
Cambridge University Press, 2014.

[26] S. U. Stich, J.-B. Cordonnier, and M. Jaggi.
Sparsified SGD with memory.
In *Advances in Neural Information Processing Systems*, 2018.

[27] Sharan Vaswani, Francis Bach, and Mark Schmidt.
Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron.
In *22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *PMLR*, pages 1195–1204, 2019.

Bibliography VII

[28] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li.

Terngrad: Ternary gradients to reduce communication in distributed deep learning.

In *Advances in Neural Information Processing Systems*, pages 1509–1519, 2017.

[29] Peilin Zhao and Tong Zhang.

Stochastic optimization with importance sampling for regularized loss minimization.

In *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, volume 37, pages 1–9, 2015.