SGD with Coordinate Sampling: Theory and Practice

SIERRA Seminar

Joint work with François Portier, arXiv 2105.11818

Rémi Leluc

July 6, 2022

Outline

Introduction

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- ∇f hard to compute (large-scale problems) or even intractable !

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- ∇f hard to compute (large-scale problems) or even intractable !
- (ERM example) data $z_1, \ldots, z_n \subset \mathcal{Z}$ and a differentiable loss function

$\ell : \mathbb{R}^p \times \mathcal{Z} \to \mathbb{R}$, the objective function f is the so-called empirical risk

$$\forall \theta \in \mathbb{R}^p, \quad f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, z_i).$$

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- ∇f hard to compute (large-scale problems) or even intractable !
- (ERM example) data $z_1, \ldots, z_n \subset \mathcal{Z}$ and a differentiable loss function $\ell : \mathbb{R}^p \times \mathcal{Z} \to \mathbb{R}$, the objective function f is the so-called empirical risk

$$\forall \theta \in \mathbb{R}^p, \quad f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, z_i).$$

- The true gradient is given by $\nabla f(\theta) = n^{-1} \sum_{i=1}^{n} \nabla \ell(\theta, z_i)$ and requires to see all the samples in the dataset !

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- ∇f hard to compute (large-scale problems) or even intractable !
- (ERM example) data $z_1, \ldots, z_n \subset \mathcal{Z}$ and a differentiable loss function $\ell : \mathbb{R}^p \times \mathcal{Z} \to \mathbb{R}$, the objective function f is the so-called empirical risk

$$\forall \theta \in \mathbb{R}^p, \quad f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, z_i).$$

- The true gradient is given by $\nabla f(\theta) = n^{-1} \sum_{i=1}^{n} \nabla \ell(\theta, z_i)$ and requires to see all the samples in the dataset !
- The gradient estimates at θ_t are given by

$$g(\theta_t, \xi_{t+1}) = \nabla_\theta \ell(\theta_t, z_j)$$

where $j = \xi_{t+1} \sim \mathcal{U}(\llbracket 1, n \rrbracket)$ is uniformly distributed.

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (Zeroth-Order) Finite differences estimate $g(\theta)$

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (Zeroth-Order) Finite differences estimate $g(\theta)$

$$\forall k \in [\![1, p]\!], \quad \frac{f(\theta + he_k) - f(\theta)}{h} \underset{h \to 0}{\approx} \partial_k f(\theta)$$

The (full) deterministic gradient estimate

$$g(\theta) = \sum_{k=1}^{p} \frac{f(\theta + he_k) - f(\theta)}{h} e_k \underset{h \to 0}{\approx} \nabla f(\theta)$$

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (Zeroth-Order) Finite differences estimate $g(\theta)$

$$\forall k \in [\![1, p]\!], \quad \frac{f(\theta + he_k) - f(\theta)}{h} \underset{h \to 0}{\approx} \partial_k f(\theta)$$

The (full) deterministic gradient estimate

$$g(\theta) = \sum_{k=1}^{p} \frac{f(\theta + he_k) - f(\theta)}{h} e_k \underset{h \to 0}{\approx} \nabla f(\theta)$$

- (1st order Stochastic) Unbiased estimate $g(\cdot, \xi)$ s.t.

$$\mathbb{E}_\xi[g(\theta, \xi)] = \nabla f(\theta)$$

Introduction

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (Zeroth-Order) Finite differences estimate $g(\theta)$

$$\forall k \in [\![1, p]\!], \quad \frac{f(\theta + he_k) - f(\theta)}{h} \underset{h \to 0}{\approx} \partial_k f(\theta)$$

The (full) deterministic gradient estimate

$$g(\theta) = \sum_{k=1}^{p} \frac{f(\theta + he_k) - f(\theta)}{h} e_k \underset{h \to 0}{\approx} \nabla f(\theta)$$

- (1st order Stochastic) Unbiased estimate $g(\cdot, \xi)$ s.t.

$$\mathbb{E}_\xi[g(\theta, \xi)] = \nabla f(\theta)$$

- (SGD): Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \gamma_{t+1} g(\theta_t, \xi_{t+1})$$

Introduction: Bridging the gap

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (SGD): Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \gamma_{t+1} g(\theta_t, \xi_{t+1})$$

Introduction: Bridging the gap

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (SGD): Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \gamma_{t+1} g(\theta_t, \xi_{t+1})$$

- (SCGD): Stochastic Coordinate Gradient Descent

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

ζ_{t+1} is a random variable valued in $[\![1, p]\!]$.

Introduction: Bridging the gap

We consider the following type of optimization problem:

$$\min_{\theta \in \mathbb{R}^p} f(\theta)$$

- (SGD): Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \gamma_{t+1} g(\theta_t, \xi_{t+1})$$

- (SCGD): Stochastic Coordinate Gradient Descent

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

ζ_{t+1} is a random variable valued in $[\![1, p]\!]$.

- Covers many approaches : generate gradient estimate g and coordinate ζ_{t+1}.

Contributions

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

$$(SCGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1})$$

with $D(k) = e_k e_k^T = Diag(0, \ldots, 0, 1, 0, \ldots, 0)$.

Contributions

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

$$(SCGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1})$$

with $D(k) = e_k e_k^T = Diag(0, \ldots, 0, 1, 0, \ldots, 0)$.

- How to update the selecting matrix $D(\zeta_{t+1})$?

Contributions

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

$$(SCGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1})$$

with $D(k) = e_k e_k^T = Diag(0, \ldots, 0, 1, 0, \ldots, 0)$.

- How to update the selecting matrix $D(\zeta_{t+1})$?
  $\rightarrow$ We develop an algorithm MUSKETEER to leverage the data structure and move along relevant directions.

Contributions

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g(\theta_t, \xi_{t+1})^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

$$(SCGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1})$$

with $D(k) = e_k e_k^T = Diag(0, \ldots, 0, 1, 0, \ldots, 0)$.

- How to update the selecting matrix $D(\zeta_{t+1})$?
  $\rightarrow$ We develop an algorithm MUSKETEER to leverage the data structure and move along relevant directions.
- What condition on ζ_{t+1} for convergence ? $\rightarrow$ We analyze the properties of SCGD algorithms (convergence of the iterates, convergence of the policy, non-asymptotic bounds)

Related work

- CD with true gradient ∇f (Loshchilov et al., 2011; Richtárik and Takáč, 2013; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017)

Related work

- CD with true gradient ∇f (Loshchilov et al., 2011; Richtárik and Takáč, 2013; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017)
- Nutini et al. (2015) $\rightarrow$ **Gauss-Southwell rule** with ∇f, here we have stochastic g and ζ_{t+1}.

Related work

- CD with true gradient ∇f (Loshchilov et al., 2011; Richtárik and Takáč, 2013; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017)
- Nutini et al. (2015) $\rightarrow$ **Gauss-Southwell rule** with ∇f, here we have stochastic g and ζ_{t+1}.
- **Sparsification methods** (Alistarh et al., 2017; Wangni et al., 2018), unbiased estimate of the gradient and no theoretical results $\rightarrow$ MUSKETEER performs no reweighting (biased) and theoretical results (convergence).

General framework and notation

- Denote by $g_t = g(\theta_t, \xi_{t+1})$, only one coordinate is selected at random according to ζ_{t+1} valued in $[\![1, p]\!]$

General framework and notation

- Denote by $g_t = g(\theta_t, \xi_{t+1})$, only one coordinate is selected at random according to ζ_{t+1} valued in $[\![1, p]\!]$

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

General framework and notation

- Denote by $g_t = g(\theta_t, \xi_{t+1})$, only one coordinate is selected at random according to ζ_{t+1} valued in $[\![1, p]\!]$

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

- The distribution of ζ_{t+1}, noted $\zeta_{t+1} \sim Q(d_t)$ is the **coordinate sampling policy** and is characterized by the probability weights vector $d_t = (d_t^{(1)}, \ldots, d_t^{(p)})$

General framework and notation

- Denote by $g_t = g(\theta_t, \xi_{t+1})$, only one coordinate is selected at random according to ζ_{t+1} valued in $[\![1, p]\!]$

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

- The distribution of ζ_{t+1}, noted $\zeta_{t+1} \sim Q(d_t)$ is the **coordinate sampling policy** and is characterized by the probability weights vector $d_t = (d_t^{(1)}, \ldots, d_t^{(p)})$

$$d_t^{(k)} = \mathbb{P}(\zeta_{t+1} = k | \mathcal{F}_t), \quad k \in [\![1, p]\!].$$

General framework and notation

- Denote by $g_t = g(\theta_t, \xi_{t+1})$, only one coordinate is selected at random according to ζ_{t+1} valued in $[\![1, p]\!]$

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

- The distribution of ζ_{t+1}, noted $\zeta_{t+1} \sim Q(d_t)$ is the **coordinate sampling policy** and is characterized by the probability weights vector $d_t = (d_t^{(1)}, \ldots, d_t^{(p)})$

$$d_t^{(k)} = \mathbb{P}(\zeta_{t+1} = k | \mathcal{F}_t), \quad k \in [\![1, p]\!].$$

- The distribution of the random matrix $D(\zeta_{t+1})$ is fully characterized by the matrix

$$D_t = \mathbb{E}[D(\zeta_{t+1}) | \mathcal{F}_t] = Diag(d_t^{(1)}, \ldots, d_t^{(p)}).$$

General view: Unbiased and Adaptive policies

General update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1} h(\theta_t, \omega_{t+1})$$

where h is a gradient generator and $(\omega_t)_{t \geq 1}$ is a sequence of random variables, $\omega_t = (\xi_t, \zeta_t)$ for SCGD.

General view: Unbiased and Adaptive policies

General update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1} h(\theta_t, \omega_{t+1})$$

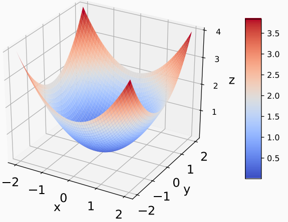
where h is a gradient generator and $(\omega_t)_{t \geq 1}$ is a sequence of random variables, $\omega_t = (\xi_t, \zeta_t)$ for SCGD.

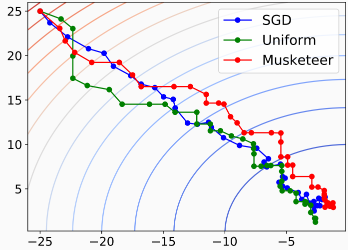
- (SGD) $h(\theta, \omega_{t+1}) = g(\theta, \xi_{t+1})$
- (SCGD) $h(\theta, \omega_{t+1}) = D(\zeta_{t+1}) g(\theta, \xi_{t+1})$
- (Unbiased) $h(\theta, \omega_{t+1}) = D_t^{-1} D(\zeta_{t+1}) g(\theta, \xi_{t+1})$ (Wangni et al., 2018)

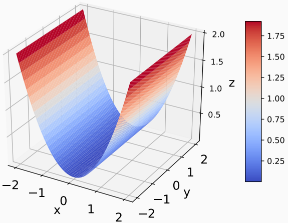
MUSKETEER

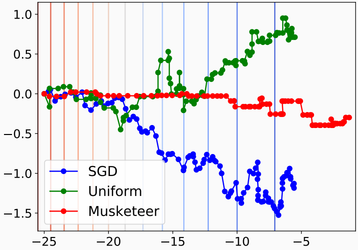
MUSKETEER

MUltivariate
Stochastic
Knowledge
Extraction
Through
Exploration
Exploitation
Reinforcement


x
y
z


SGD
Uniform
Musketeer


x
y
z


SGD
Uniform
Musketeer

MUSKETEER

MUSKETEER may be seen as an adaptive bandit problem with 'arms=coordinates' to draw. It alternates between 2 phases:

MUSKETEER

MUSKETEER may be seen as an adaptive bandit problem with 'arms=coordinates' to draw. It alternates between 2 phases:

- **Exploration phase (one for all).**

$\rightarrow$ fixed d_n, draw random coordinate and move along selected direction

$\rightarrow$ cumulative gains for the visited coordinates

MUSKETEER

MUSKETEER may be seen as an adaptive bandit problem with 'arms=coordinates' to draw. It alternates between 2 phases:

- **Exploration phase (one for all).**

$\rightarrow$ fixed d_n, draw random coordinate and move along selected direction
$\rightarrow$ cumulative gains for the visited coordinates

MUSKETEER

MUSKETEER may be seen as an adaptive bandit problem with 'arms=coordinates' to draw. It alternates between 2 phases:

- **Exploration phase (one for all).**

$\rightarrow$ fixed d_n, draw random coordinate and move along selected direction

$\rightarrow$ cumulative gains for the visited coordinates

- **Exploitation phase. (all for one)**

MUSKETEER

MUSKETEER may be seen as an adaptive bandit problem with 'arms=coordinates' to draw. It alternates between 2 phases:

- **Exploration phase (one for all).**

$\rightarrow$ fixed d_n, draw random coordinate and move along selected direction
$\rightarrow$ cumulative gains for the visited coordinates

- **Exploitation phase. (all for one)**

$\rightarrow$ share knowledge of the cumulative gains
$\rightarrow$ update the coordinate sampling probability vector d_n (EXP3)

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

• The *exploration* phase may be seen as a **multi-armed bandit problem** where the arms are the gradient coordinates for $k \in [\![1, p]\!]$.

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

• The *exploration* phase may be seen as a **multi-armed bandit problem** where the arms are the gradient coordinates for $k \in [\![1, p]\!]$.

• For $t \in [\![1, T]\!]$, $\zeta \sim Q(d_n)$ and the relative gradient $g_t^{(\zeta)}/d_n^{(\zeta)}$, representing the reward, is observed.

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

• The *exploration* phase may be seen as a **multi-armed bandit problem** where the arms are the gradient coordinates for $k \in [\![1, p]\!]$.

• For $t \in [\![1, T]\!]$, $\zeta \sim Q(d_n)$ and the relative gradient $g_t^{(\zeta)}/d_n^{(\zeta)}$, representing the reward, is observed.

$\rightarrow$ **1) Update the iterate** $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1} g_t^{(\zeta)}$

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

• The *exploration* phase may be seen as a **multi-armed bandit problem** where the arms are the gradient coordinates for $k \in [\![1, p]\!]$.

• For $t \in [\![1, T]\!]$, $\zeta \sim Q(d_n)$ and the relative gradient $g_t^{(\zeta)}/d_n^{(\zeta)}$, representing the reward, is observed.

$\rightarrow$ **1) Update the iterate** $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1} g_t^{(\zeta)}$

$\rightarrow$ **2) Build cumulative gains**

$$\widetilde{G}_{t+1}^{(\zeta)} = \widetilde{G}_t^{(\zeta)} + g^{(\zeta)}(\theta_t, \xi)/d_n^{(\zeta)}$$

MUSKETEER: Exploration phase

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop.

• The *exploration* phase may be seen as a **multi-armed bandit problem** where the arms are the gradient coordinates for $k \in [\![1, p]\!]$.

• For $t \in [\![1, T]\!]$, $\zeta \sim Q(d_n)$ and the relative gradient $g_t^{(\zeta)}/d_n^{(\zeta)}$, representing the reward, is observed.

$\rightarrow$ **1) Update the iterate** $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1} g_t^{(\zeta)}$

$\rightarrow$ **2) Build cumulative gains**

$$\widetilde{G}_{t+1}^{(\zeta)} = \widetilde{G}_t^{(\zeta)} + g^{(\zeta)}(\theta_t, \xi)/d_n^{(\zeta)}$$

$\rightarrow$ (Variants with abs or square) $|g^{(\zeta)}(\theta_t, \xi)|$ or $g^{(\zeta)}(\theta_t, \xi)^2$

$\rightarrow$ Value of T ? RL trade-off

MUSKETEER: Exploitation phase

- Starting from $G_0 = (0, \dots, 0)$, the total gain G_n is updated in a online manner during the *exploitation* phase using the update rule

MUSKETEER: Exploitation phase

- Starting from $G_0 = (0, \ldots, 0)$, the total gain G_n is updated in a online manner during the *exploitation* phase using the update rule

$$G_{n+1} = G_n + \widetilde{G}_T, \quad \widetilde{G}_T = \sum_{t=1}^{T} D_n^{-1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1}). \tag{1}$$

MUSKETEER: Exploitation phase

- Starting from $G_0 = (0, \ldots, 0)$, the total gain G_n is updated in a online manner during the *exploitation* phase using the update rule

$$G_{n+1} = G_n + \widetilde{G}_T, \quad \widetilde{G}_T = \sum_{t=1}^{T} D_n^{-1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1}). \tag{1}$$

- EXP3 algorithm (Auer et al., 2002) to update the probability weights through a mixture.

MUSKETEER: Exploitation phase

- Starting from $G_0 = (0, \ldots, 0)$, the total gain G_n is updated in a online manner during the *exploitation* phase using the update rule

$$G_{n+1} = G_n + \widetilde{G}_T, \quad \widetilde{G}_T = \sum_{t=1}^{T} D_n^{-1} D(\zeta_{t+1}) g(\theta_t, \xi_{t+1}). \tag{1}$$

- EXP3 algorithm (Auer et al., 2002) to update the probability weights through a mixture. Given $\eta > 0$ and $\lambda \in [0, 1]$, we have for all $k \in [\![1, p]\!]$,

$$d_{n+1}^{(k)} = (1 - \lambda) \frac{\exp(\eta G_n^{(k)} / (nT))}{\sum_{j=1}^{d} \exp(\eta G_n^{(j)} / (nT))} + \lambda \frac{1}{p}. \tag{2}$$

MUSKETEER: Exploration/Exploitation phases

Explore(T, d_n)

for $t = 1, \ldots, T$ **do**
  Sample coordinate $\zeta \sim Q(d_n)$ and data ξ
  Update iterate: $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1} g^{(\zeta)}(\theta_t, \xi)$
  Update gain: $\widetilde{G}_{t+1}^{(\zeta)} = \widetilde{G}_t^{(\zeta)} + g^{(\zeta)}(\theta_t, \xi)/d_n^{(\zeta)}$
end for
Return vector of gains $\widetilde{G}_T$

Exploit($G_n, \widetilde{G}_T, \lambda, \eta$)

Update total gain G_n using (1)
Update probability weights d_{n+1} using (2)

Numerical Experiments

Numerical Experiments

- We apply ERM to regularized **regression** and **classification** problems.

Numerical Experiments

- We apply ERM to regularized **regression** and **classification** problems.
- Given a data matrix $X = (x_{i,j}) \in \mathbb{R}^{n\times p}$ with labels $y \in \mathbb{R}^n$ and a regularization parameter $\mu > 0$, the *Ridge regression* is

$$\min_{\theta\in\mathbb{R}^p} f(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{i,j}\theta_j)^2 + \frac{\mu}{2}\|\theta\|_2^2$$

Numerical Experiments

- We apply ERM to regularized **regression** and **classification** problems.
- Given a data matrix $X = (x_{i,j}) \in \mathbb{R}^{n \times p}$ with labels $y \in \mathbb{R}^n$ and a regularization parameter $\mu > 0$, the *Ridge regression* is

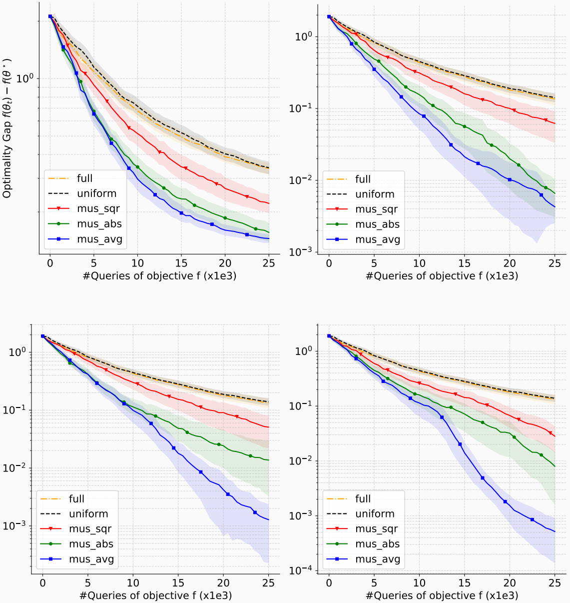
$$\min_{\theta \in \mathbb{R}^p} f(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{i,j}\theta_j)^2 + \frac{\mu}{2} \|\theta\|_2^2$$

and the ℓ_2*-regularized logistic regression* is defined by

$$\min_{\theta \in \mathbb{R}^p} f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \sum_{j=1}^{p} x_{i,j}\theta_j)) + \mu \|\theta\|_2^2$$

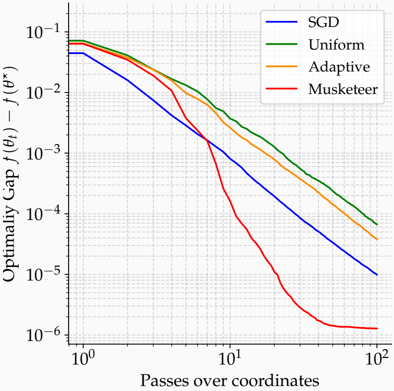
where μ is set to the classical value $\mu = 1/n$.
- The columns are drawn as $X[:, k] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for $k \in [\![1, p]\!]$.
- Setting $\gamma_t = 1/t$, $n = 10,000$, $p = 250$, $T = \lfloor \sqrt{p} \rfloor = 15$.

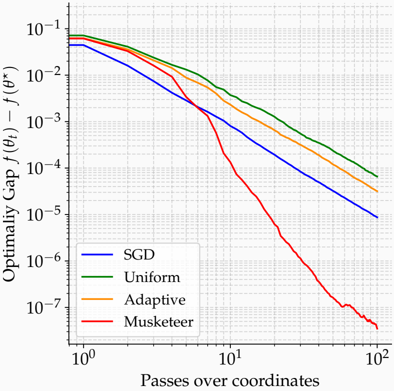
Zero-Order Ridge Regression $\alpha = 2, 5, 7, 10$



Numerical Experiments: Ridge Regression

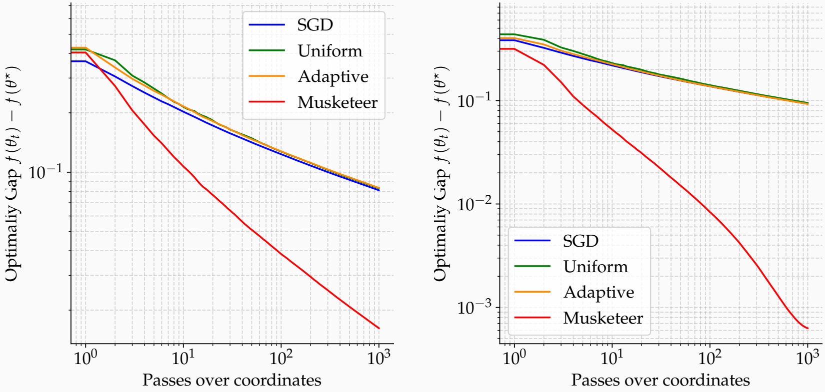
- $\alpha = 5$ and $\alpha = 10$





Numerical Experiments: Logistic Regression

- $\alpha = 2$ and $\alpha = 5$

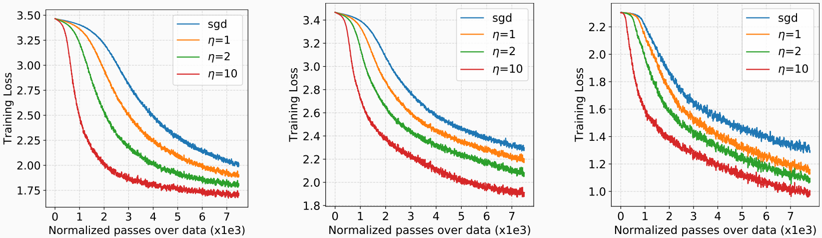


Numerical Experiments

- **MNIST, Fashion-MNIST, CIFAR10**
- linear layers for MNIST and Fashion-MNIST ($p = 55,050$ and $T = 234$), convolutional layers for CIFAR10 ($p = 64,862$ and $T = 254$).

Numerical Experiments

- **MNIST, Fashion-MNIST, CIFAR10**
- linear layers for MNIST and Fashion-MNIST ($p = 55,050$ and $T = 234$), convolutional layers for CIFAR10 ($p = 64,862$ and $T = 254$).

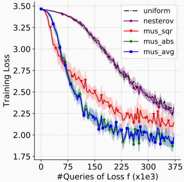


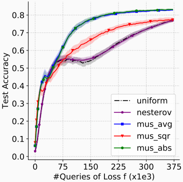
Numerical Experiments

- **MNIST and Fashion-MNIST (ZO)** ($p = 55,050$ and $T = 234$)

Numerical Experiments

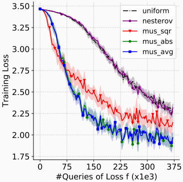
- **MNIST and Fashion-MNIST (ZO)** ($p = 55,050$ and $T = 234$)

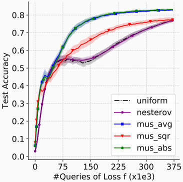


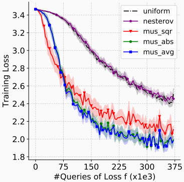


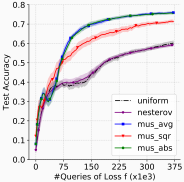
Numerical Experiments

- **MNIST and Fashion-MNIST (ZO)** ($p = 55,050$ and $T = 234$)









Main results

Main results

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

- The gradient generator is $g_h(\cdot, \xi)$ where $h \geq 0$ is the underlying bias.

Main results

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

- The gradient generator is $g_h(\cdot, \xi)$ where $h \geq 0$ is the underlying bias.

Biased Gradient

There exists constant $c \geq 0$ such that

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \| \mathbb{E}_\xi[g_h(\theta, \xi)] - \nabla f(\theta)\| \leq ch.$$

- $c = 0$ recovers 1st-order gradient estimates.

Main results

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

- The gradient generator is $g_h(\cdot, \xi)$ where $h \geq 0$ is the underlying bias.

Biased Gradient

There exists constant $c \geq 0$ such that

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \| \mathbb{E}_\xi[g_h(\theta, \xi)] - \nabla f(\theta)\| \leq ch.$$

- $c = 0$ recovers 1st-order gradient estimates.
- This assumption covers general zeroth-order estimates.

ZO gradient estimates

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

ZO gradient estimates

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

Example 1 (smoothing). (Nesterov and Spokoiny, 2017). The smoothed gradient estimate is given by

$$\forall \theta \in \mathbb{R}^p, g_h(\theta, \xi) = h^{-1}[f(\theta + hU, \xi) - f(\theta, \xi)]U$$

where $U \sim \mathcal{N}(0, I)$. (Alternative version with $U \sim Unif(\mathbb{S})$)

ZO gradient estimates

$$\min_{\theta \in \mathbb{R}^p} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$$

Example 1 (smoothing). (Nesterov and Spokoiny, 2017). The smoothed gradient estimate is given by

$$\forall \theta \in \mathbb{R}^p, g_h(\theta, \xi) = h^{-1}[f(\theta + hU, \xi) - f(\theta, \xi)]U$$

where $U \sim \mathcal{N}(0, I)$. (Alternative version with $U \sim Unif(\mathbb{S})$)
Example 2 (finite differences). The finite differences gradient estimate is given by

$$\forall \theta \in \mathbb{R}^p, g_h(\theta, \xi) = \sum_{k=1}^{p} g_h(\theta, \xi)^{(k)} e_k$$

where for all $k = 1, \ldots, p$ the coordinates are

$$g_h(\theta, \xi)^{(k)} = h^{-1}[f(\theta + he_k, \xi) - f(\theta, \xi)]$$

General form

There exists probability measure ν satisfying $\int_{\mathbb{R}^p} xx^\top \nu(\mathrm{d}x) = I_p$,

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^p} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x).$$

General form

There exists probability measure ν satisfying $\int_{\mathbb{R}^p} xx^\top \nu(\mathrm{d}x) = I_p$,

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^p} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x).$$

- Satisfies biased gradient assumption with $c = \sqrt{C}L/2$ where $C = \int_{\mathbb{R}^p} \|x\|_2^6 \nu(\mathrm{d}x) < \infty$.

General form

There exists probability measure ν satisfying $\int_{\mathbb{R}^p} xx^\top \nu(\mathrm{d}x) = I_p$,

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^p} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x).$$

• Satisfies biased gradient assumption with $c = \sqrt{C}L/2$ where $C = \int_{\mathbb{R}^p} \|x\|_2^6 \nu(\mathrm{d}x) < \infty$.

• The smoothed gradient estimate is recovered when ν is the standard Gaussian measure.

General form

There exists probability measure ν satisfying $\int_{\mathbb{R}^p} xx^\top \nu(\mathrm{d}x) = I_p$,

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^p} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x).$$

- Satisfies biased gradient assumption with $c = \sqrt{C}L/2$ where $C = \int_{\mathbb{R}^p} \|x\|_2^6 \nu(\mathrm{d}x) < \infty$.
- The smoothed gradient estimate is recovered when ν is the standard Gaussian measure.
- Take $\nu = \sum_{k=1}^p \delta_{e_k}/p$ covers the finite differences estimate.

General form

There exists probability measure ν satisfying $\int_{\mathbb{R}^p} xx^\top \nu(\mathrm{d}x) = I_p$,

$$\forall h > 0, \theta \in \mathbb{R}^p, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^p} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x).$$

• Satisfies biased gradient assumption with $c = \sqrt{C}L/2$ where $C = \int_{\mathbb{R}^p} \|x\|_2^6 \nu(\mathrm{d}x) < \infty$.

• The smoothed gradient estimate is recovered when ν is the standard Gaussian measure.

• Take $\nu = \sum_{k=1}^p \delta_{e_k}/p$ covers the finite differences estimate.

• (MUSKETEER) Use a measure ν that evolves through time and put different weights on the different directions !

Assumption

- f is L-smooth and lower bounded by $f^\star$.

Assumption

- f is L-smooth and lower bounded by $f^\star$.
- (Growth condition) With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$

$$\forall h > 0, \theta \in \mathbb{R}^p \quad \mathbb{E}\left[\|g_h(\theta,\xi)\|_\infty^2\right] \leq 2\mathcal{L}\left(f(\theta) - f^\star\right) + \sigma^2.$$

Assumption

- f is L-smooth and lower bounded by $f^\star$.
- (Growth condition) With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$

$$\forall h > 0, \theta \in \mathbb{R}^p \quad \mathbb{E}\left[\|g_h(\theta, \xi)\|_\infty^2\right] \leq 2\mathcal{L}\left(f(\theta) - f^\star\right) + \sigma^2.$$

Gradient generator $g_t = g_{h_{t+1}}(\theta_t, \xi_{t+1})$

Assumption

- f is L-smooth and lower bounded by $f^\star$.
- (Growth condition) With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$

$$\forall h > 0, \theta \in \mathbb{R}^p \quad \mathbb{E}\left[\|g_h(\theta, \xi)\|_\infty^2\right] \leq 2\mathcal{L}\left(f(\theta) - f^\star\right) + \sigma^2.$$

Gradient generator $g_t = g_{h_{t+1}}(\theta_t, \xi_{t+1})$

$$(SGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

Theorem (Almost sure convergence of (biased) SGD)

Let (θ_t) obtained by SGD and assume that the learning rates satisfy the Robbins-Monro condition and $h_t^2 = O(\gamma_t)$ then $\nabla f(\theta_t) \to 0$ a.s. when $t \to \infty$.

Assumption

- f is L-smooth and lower bounded by $f^\star$.
- (Growth condition) With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$

$$\forall h > 0, \theta \in \mathbb{R}^p \quad \mathbb{E}\left[\|g_h(\theta,\xi)\|_\infty^2\right] \leq 2\mathcal{L}\left(f(\theta) - f^\star\right) + \sigma^2.$$

Gradient generator $g_t = g_{h_{t+1}}(\theta_t, \xi_{t+1})$

$$(SGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

Theorem (Almost sure convergence of (biased) SGD)

Let (θ_t) obtained by SGD and assume that the learning rates satisfy the Robbins-Monro condition and $h_t^2 = O(\gamma_t)$ then $\nabla f(\theta_t) \to 0$ a.s. when $t \to \infty$.

- When f coercive and unique solution $\{\theta : \nabla f(\theta) = 0\} = \{\theta^\star\}$ then almost sure convergence towards minimizer $\theta_t \to \theta^\star$.

Main results: particular SCGD

$$(SGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

$$(SCGD) \quad \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g_t$$

Main results: particular SCGD

$$
\begin{aligned}
(SGD) \quad & \theta_{t+1} = \theta_t - \gamma_{t+1} g_t \\
(SCGD) \quad & \theta_{t+1} = \theta_t - \gamma_{t+1} D(\zeta_{t+1}) g_t
\end{aligned}
$$

Theorem (Almost sure convergence of particular SCGD)

Let (θ_t) obtained by SCGD and assume that the learning rates satisfy the Robbins-Monro condition and $h_t^2 = O(\gamma_t)$:
(i) (max gradient) if $\zeta_{t+1} = \arg\max_{k=1,\dots,p} |\partial_k f(\theta_t)|$ then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.
(ii) (gradient weights) if $D_t \propto (|\nabla_k f(\theta_t)|^q)_{1 \leq k \leq p}$ with $q > 0$ then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.

Main results: general SCGD

Extended Robbins-Monro

$\beta_{t+1} = \min_{1\leq k\leq p} d_t^{(k)}$. The sequences $(\gamma_t)_{t\geq 1}$ and $(\beta_t)_{t\geq 1}$ are positive with $\sum_{t\geq 1}\gamma_t\beta_t = +\infty, \sum_{t\geq 1}\gamma_t^2 < +\infty$.

Main results: general SCGD

Extended Robbins-Monro

$\beta_{t+1} = \min_{1 \le k \le p} d_t^{(k)}$. The sequences $(\gamma_t)_{t \ge 1}$ and $(\beta_t)_{t \ge 1}$ are positive with $\sum_{t \ge 1} \gamma_t \beta_t = +\infty, \sum_{t \ge 1} \gamma_t^2 < +\infty$.

- Note that $D_t = I_p$ recovers SGD with standard Robbins-Monro.

Main results: general SCGD

Extended Robbins-Monro

$\beta_{t+1} = \min_{1 \leq k \leq p} d_t^{(k)}$. The sequences $(\gamma_t)_{t\geq 1}$ and $(\beta_t)_{t\geq 1}$ are positive with $\sum_{t\geq 1} \gamma_t \beta_t = +\infty, \sum_{t\geq 1} \gamma_t^2 < +\infty$.

- Note that $D_t = I_p$ recovers SGD with standard Robbins-Monro.

Theorem (Almost sure convergence general SCGD)

Let (θ_t) obtained by SCGD and assume that the learning rates satisfy the extended Robbins-Monro condition and $h_t^2 = O(\gamma_t)$. If (β_t) is lower bounded then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.

Main results: general SCGD

Extended Robbins-Monro

$\beta_{t+1} = \min_{1 \leq k \leq p} d_t^{(k)}$. The sequences $(\gamma_t)_{t \geq 1}$ and $(\beta_t)_{t \geq 1}$ are positive with $\sum_{t \geq 1} \gamma_t \beta_t = +\infty, \sum_{t \geq 1} \gamma_t^2 < +\infty$.

- Note that $D_t = I_p$ recovers SGD with standard Robbins-Monro.

Theorem (Almost sure convergence general SCGD)

Let (θ_t) obtained by SCGD and assume that the learning rates satisfy the extended Robbins-Monro condition and $h_t^2 = O(\gamma_t)$. If (β_t) is lower bounded then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.

- Similarly with stronger assumptions (f coercive, unique critical point), there is convergence of the iterates towards minimizer $\theta^\star$.

Main results: MUSKETEER

Theorem (Almost sure convergence)

The sequence of iterates $(\theta_t)_{t\geq 0}$ *obtained by the MUSKETEER satisfies* $\nabla f(\theta_t) \to 0$ *almost surely as* $t \to +\infty$.

Main results: MUSKETEER

Theorem (Almost sure convergence)

The sequence of iterates $(\theta_t)_{t\geq 0}$ *obtained by the MUSKETEER satisfies* $\nabla f(\theta_t) \to 0$ *almost surely as* $t \to +\infty$.

Theorem (Weak convergence)

The MUSKETEER's coordinate policy $(Q(d_n))_{n\in\mathbb{N}}$ *converges weakly to the uniform distribution, i.e.,* $Q(d_n) \rightsquigarrow \mathcal{U}([\![1, p]\!])$ *as* $n \to +\infty$.

Main results: MUSKETEER

Theorem (Almost sure convergence)

The sequence of iterates $(\theta_t)_{t\geq 0}$ *obtained by the MUSKETEER satisfies* $\nabla f(\theta_t) \to 0$ *almost surely as* $t \to +\infty$.

Theorem (Weak convergence)

The MUSKETEER's coordinate policy $(Q(d_n))_{n\in\mathbb{N}}$ *converges weakly to the uniform distribution, i.e.,* $Q(d_n) \rightsquigarrow \mathcal{U}([\![1, p]\!])$ *as* $n \to +\infty$.

Theorem (Non-asymptotic bounds, (Moulines and Bach, 2011))

Let $(\theta_t)_{t\in\mathbb{N}}$ *obtained by MUSKETEER with* $\gamma_t = \gamma t^{-\alpha}$ *then*

$$\mathbb{E}\left[f(\theta_t) - f^\star\right] = O(1/t), \quad (\alpha = 1)$$

Future work

- Study the behavior of the rescaled sequence $(\theta_t - \theta^\star)/\sqrt{\gamma_t}$ for MUSKETEER and general SCGD methods.
- Study the asymptotic behavior of other adaptive sampling strategies
- Study the extensions with Nesterov acceleration schemes and momentum

Q & A

Thank you

References

Agarwal, A., P. L. Bartlett, P. Ravikumar, and M. J. Wainwright (2012). Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory 58*(5), 3235–3249.

Agarwal, A., M. J. Wainwright, P. L. Bartlett, and P. K. Ravikumar (2009). Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pp. 1–9.

Agarwal, N., B. Bullins, and E. Hazan (2016). Second-order stochastic optimization in linear time. *stat 1050*, 15.

Alain, G., A. Lamb, C. Sankar, A. Courville, and Y. Bengio (2015). Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*.

Alber, Y. I., A. N. Iusem, and M. V. Solodov (1998). On the projected subgradient method for nonsmooth convex optimization in a hilbert space. *Mathematical Programming 81*(1), 23–35.

Bibliography ii

Alistarh, D., D. Grubic, J. Li, R. Tomioka, and M. Vojnovic (2017). Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720.

Alistarh, D., T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli (2018). The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, pp. 5973–5983.

Allen-Zhu, Z., Z. Qu, P. Richtárik, and Y. Yuan (2016). Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pp. 1110–1119.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation 10*(2), 251–276.

Aude, G., M. Cuturi, G. Peyré, and F. Bach (2016). Stochastic optimization for large-scale optimal transport. *arXiv preprint arXiv:1605.08527*.

Auer, P., N. Cesa-Bianchi, and P. Fischer (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning 47*(2-3), 235–256.

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing 32*(1), 48–77.

Barakat, A. and P. Bianchi (2018). Convergence and dynamical behavior of the adam algorithm for non convex stochastic optimization. *arXiv preprint arXiv:1810.02263*.

Baxter, J. and P. L. Bartlett (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research 15*, 319–350.

Beck, A. and L. Tetruashvili (2013). On the convergence of block coordinate descent type methods. *SIAM journal on Optimization 23*(4), 2037–2060.

Bellman, R. and R. Kalaba (1957). Dynamic programming and statistical communication theory. *Proceedings of the National Academy of Sciences of the United States of America 43*(8), 749.

Benveniste, A., M. Métivier, and P. Priouret (2012). *Adaptive algorithms and stochastic approximations*, Volume 22. Springer Science & Business Media.

Bercu, B., B. Delyon, and E. Rio (2015). *Concentration inequalities for sums and martingales*. Springer.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-dynamic programming*, Volume 5. Athena Scientific Belmont, MA.

Bertsekas, D. P. and J. N. Tsitsiklis (2000). Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization 10*(3), 627–642.

Borel, M. É. (1909). Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940) 27*(1), 247–271.

Bottou, L. and O. Bousquet (2008). The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pp. 161–168.

Bottou, L., F. E. Curtis, and J. Nocedal (2018). Optimization methods for large-scale machine learning. *Siam Review 60*(2), 223–311.

Bottou, L. and C.-J. Lin (2007). Support vector machine solvers. *Large scale kernel machines 3*(1), 301–320.

Boucheron, S., G. Lugosi, and P. Massart (2013). *Concentration Inequalities*. Oxford University Press.

Boyer, C. and A. Godichon-Baggioni (2020). On the asymptotic rate of convergence of stochastic newton algorithms and their weighted averaged versions. *arXiv preprint arXiv:2011.09706*.

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics 6*(1), 76–90.

Bubeck, S. et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning 8*(3-4), 231–357.

Byrd, R. H., G. M. Chin, W. Neveitt, and J. Nocedal (2011). On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization 21*(3), 977–995.

Byrd, R. H., P. Lu, J. Nocedal, and C. Zhu (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing 16*(5), 1190–1208.

Cardoso, J.-F. (1998). Blind signal separation: statistical principles. *Proceedings of the IEEE 86*(10), 2009–2025.

Chen, H.-F., L. Guo, and A.-J. Gao (1987). Convergence and robustness of the robbins-monro algorithm truncated at randomly varying bounds. *Stochastic Processes and their Applications 27*, 217–231.

Clémençon, S., P. Bertail, E. Chautru, and G. Papa (2019). Optimal survey schemes for stochastic gradient descent with applications to m-estimation. *ESAIM: Probability and Statistics 23*, 310–337.

Csiba, D., Z. Qu, and P. Richtárik (2015). Stochastic dual coordinate ascent with adaptive probabilities. In *International Conference on Machine Learning*, pp. 674–683.

Defazio, A., F. Bach, and S. Lacoste-Julien (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654.

Dekel, O., R. Gilad-Bachrach, O. Shamir, and L. Xiao (2012). Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research 13*(Jan), 165–202.

Delyon, B. (1996). General results on the convergence of stochastic algorithms. *IEEE Transactions on Automatic Control 41*(9), 1245–1255.

Delyon, B. (2000). Stochastic approximation with decreasing gain: Convergence and asymptotic theory. *Unpublished lecture notes, Université de Rennes*, 26.

Delyon, B. and F. Portier (2018). Asymptotic optimality of adaptive importance sampling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3138–3148. Curran Associates Inc.

Delyon, B. and F. Portier (2019). Adaptive importance sampling by kernel smoothing. *arXiv preprint arXiv:1903.08507*.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine 29*(6), 141–142.

Dodu, J., M. Goursat, A. Hertz, J. Quadrat, and M. Viot (1981). Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique. *EDF Bulletin de la Direction des Etudes et Recherches, série C-mathématiques, informatique* (2), 133–164.

Dua, D. and C. Graff (2017). UCI machine learning repository.

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research 12*(Jul), 2121–2159.

Duflo, M. (2013). *Random iterative models*, Volume 34. Springer Science & Business Media.

Fan, R.-E., P.-H. Chen, C.-J. Lin, and T. Joachims (2005). Working set selection using second order information for training support vector machines. *Journal of machine learning research 6*(12).

Fazel, M., R. Ge, S. M. Kakade, and M. Mesbahi (2018). Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*.

Feinberg, V., A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine (2018). Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*.

Fercoq, O., Z. Qu, P. Richtárik, and M. Takáč (2014). Fast distributed coordinate descent for non-strongly convex losses. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE.

Fercoq, O. and P. Richtárik (2015). Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization 25*(4), 1997–2023.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The computer journal 13*(3), 317–322.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software 33*(1), 1.

Gadat, S., F. Panloup, S. Saadane, et al. (2018). Stochastic heavy ball. *Electronic Journal of Statistics 12*(1), 461–529.

Gazagnadou, N., R. M. Gower, and J. Salmon (2019). Optimal mini-batch and step sizes for saga. *arXiv preprint arXiv:1902.00071*.

Glasmachers, T. and U. Dogan (2013). Accelerated coordinate descent with adaptive coordinate frequencies. In *Asian Conference on Machine Learning*, pp. 72–86.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of computation 24*(109), 23–26.

Gopal, S. (2016). Adaptive sampling for sgd by exploiting side information. In *International Conference on Machine Learning*, pp. 364–372.

Gower, R. M., N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik (2019). Sgd: General analysis and improved rates. *arXiv preprint arXiv:1901.09401*.

Gower, R. M., P. Richtárik, and F. Bach (2018). Stochastic quasi-gradient methods: Variance reduction via jacobian sketching. *arXiv preprint arXiv:1805.02632*.

Hall, P. and C. Heyde (1980). *Martingale Limit Theory and Its Application*. Probability and mathematical statistics. Academic Press.

Hall, P. and C. C. Heyde (2014). *Martingale limit theory and its application*. Academic press.

Hanna, J., S. Niekum, and P. Stone (2019). Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pp. 2605–2613. PMLR.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013). Stochastic variational inference. *The Journal of Machine Learning Research 14*(1), 1303–1347.

Howard, R. A. (1960). Dynamic programming and markov processes.

Johnson, R. and T. Zhang (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems*, pp. 1531–1538.

Karimi, H., J. Nutini, and M. Schmidt (2016). Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer.

Kesten, H. et al. (1958). Accelerated stochastic approximation. *The Annals of Mathematical Statistics 29*(1), 41–59.

Khalil, H. K. (2002). *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall.

Kiefer, J., J. Wolfowitz, et al. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics 23*(3), 462–466.

Krizhevsky, A., G. Hinton, et al. (2009). Learning multiple layers of features from tiny images.

Kushner, H. and G. G. Yin (2003). *Stochastic approximation and recursive algorithms and applications*, Volume 35. Springer Science & Business Media.

Kushner, H. J. and D. S. Clark (1978). Stochastic approximation methods for constrained and unconstrained systems.

Kushner, H. J. and H. Huang (1979). Rates of convergence for stochastic approximation type algorithms. *SIAM Journal on Control and Optimization 17*(5), 607–617.

Lai, T. L. et al. (2003). Stochastic approximation. *The annals of Statistics 31*(2), 391–406.

LeCun, Y. A., L. Bottou, G. B. Orr, and K.-R. Müller (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer.

Lee, Y. T. and A. Sidford (2013). Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 147–156. IEEE.

Leluc, R. and F. Portier (2020). Towards asymptotic optimality with conditioned stochastic gradient descent. *arXiv preprint arXiv:2006.02745*.

Levine, S. and V. Koltun (2013). Guided policy search. In *International Conference on Machine Learning*, pp. 1–9.

Loshchilov, I., M. Schoenauer, and M. Sebag (2011). Adaptive coordinate descent. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 885–892.

Lu, Z. and L. Xiao (2015). On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming 152*(1-2), 615–642.

Marceau-Caron, G. and Y. Ollivier (2016). Practical riemannian neural networks. *arXiv preprint arXiv:1602.08007*.

Martens, J. (2014). New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.

Moulines, E. and F. R. Bach (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pp. 451–459.

Murata, N. (1998). A statistical study of on-line learning. *Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK*, 63–92.

Namkoong, H., A. Sinha, S. Yadlowsky, and J. C. Duchi (2017). Adaptive sampling probabilities for non-smooth optimization. In *International Conference on Machine Learning*, pp. 2574–2583.

Necoara, I., Y. Nesterov, and F. Glineur (2014). A random coordinate descent method on large-scale optimization problems with linear constraints. Technical report, Technical Report.

Necoara, I., Y. Nesterov, and F. Glineur (2019). Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming 175*(1-2), 69–107.

Needell, D., R. Ward, and N. Srebro (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*, pp. 1017–1025.

Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization 19*(4), 1574–1609.

Nemirovski, A. S. and D. B. Yudin (1983). Problem complexity and method efficiency in optimization.

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization 22*(2), 341–362.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, Volume 87. Springer Science & Business Media.

Nesterov, Y. and V. Spokoiny (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics 17*(2), 527–566.

Nesterov, Y. and J.-P. Vial (2008). Confidence level solutions for stochastic programming. *Automatica 44*(6), 1559–1568.

Nevelson, M. B. and R. Z. Khas'minskiĭ (1976). *Stochastic approximation and recursive estimation*, Volume 47. American Mathematical Soc.

Nguyen, L. M., P. H. Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč (2018). Sgd and hogwild! convergence without the bounded gradients assumption. *arXiv preprint arXiv:1802.03801*.

Nutini, J., M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke (2015). Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pp. 1632–1641.

Papa, G., P. Bianchi, and S. Clémençon (2015). Adaptive sampling for incremental optimization using stochastic gradient descent. In *International Conference on Algorithmic Learning Theory*, pp. 317–331. Springer.

Papini, M., D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli (2018). Stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1806.05618*.

Papini, M., M. Pirotta, and M. Restelli (2019). Smoothing policies and safe policy gradients. *arXiv preprint arXiv:1905.03231*.

Park, H., S.-I. Amari, and K. Fukumizu (2000). Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks 13*(7), 755–764.

Patel, K. K. and A. Dieuleveut (2019). Communication trade-offs for local-sgd with large step size. *Advances In Neural Information Processing Systems 32 (Nips 2019) 32*(CONF).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research 12*, 2825–2830.

Pelletier, M. (1998a). On the almost sure asymptotic behaviour of stochastic algorithms. *Stochastic processes and their applications 78*(2), 217–244.

Pelletier, M. (1998b). Weak convergence rates for stochastic approximation with application to multiple targets and simulated annealing. *Annals of Applied Probability*, 10–44.

Perekrestenko, D., V. Cevher, and M. Jaggi (2017). Faster coordinate descent via adaptive importance sampling. In *Artificial Intelligence and Statistics*, pp. 869–877. PMLR.

Peters, J. and S. Schaal (2008a). Natural actor-critic. *Neurocomputing 71*(7-9), 1180–1190.

Peters, J. and S. Schaal (2008b). Reinforcement learning of motor skills with policy gradients. *Neural networks 21*(4), 682–697.

Plakhov, A. and P. Cruz (2009). A stochastic approximation algorithm with multiplicative step size modification. *Mathematical Methods of Statistics 18*(2), 185–200.

Polyak, B. T. (1963). Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 3*(4), 643–653.

Polyak, B. T. (1976). Convergence and rate of convergence in iterative stochastic processes. i. the general case. *Avtomatika i telemekhanika* (12), 83–94.

Polyak, B. T. (1990). A new method of stochastic approximation type. *Avtomatika i telemekhanika* (7), 98–107.

Polyak, B. T. and A. B. Juditsky (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization 30*(4), 838–855.

Polyak, B. T. and Y. Z. Tsypkin (1979). Adaptive estimation algorithms: convergence, optimality, stability. *Avtomatika i Telemekhanika* (3), 71–84.

Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qu, Z. and P. Richtárik (2016). Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software 31*(5), 829–857.

Qu, Z., P. Richtárik, and T. Zhang (2015). Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in neural information processing systems*, pp. 865–873.

Richtárik, P. and M. Takáč (2013). On optimal probabilities in stochastic coordinate descent methods. *arXiv preprint arXiv:1310.3438*.

Richtárik, P. and M. Takáč (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming 144*(1-2), 1–38.

Richtárik, P. and M. Takáč (2016). Parallel coordinate descent methods for big data optimization. *Mathematical Programming 156*(1-2), 433–484.

Robbins, H. and S. Monro (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Robbins, H. and D. Siegmund (1971). A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pp. 233–257. Elsevier.

Ruppert, D. (1988). Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering.

Schmidt, M. and N. L. Roux (2013). Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shalev-Shwartz, S., O. Shamir, N. Srebro, and K. Sridharan (2009). Stochastic convex optimization. In *COLT*.

Shalev-Shwartz, S., Y. Singer, N. Srebro, and A. Cotter (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming 127*(1), 3–30.

Shalev-Shwartz, S. and A. Tewari (2011). Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research 12*, 1865–1892.

Shalev-Shwartz, S. and T. Zhang (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research 14*(Feb), 567–599.

Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of computation 24*(111), 647–656.

Sutton, R. S., A. G. Barto, et al. (1998). *Introduction to reinforcement learning*, Volume 2. MIT press Cambridge.

Wangni, J., J. Wang, J. Liu, and T. Zhang (2018). Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 1299–1309.

Welford, B. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics 4*(3), 419–420.

Wen, Z., D. Goldfarb, and K. Scheinberg (2012). Block coordinate descent methods for semidefinite programming. In *Handbook on semidefinite, conic and polynomial optimization*, pp. 533–564. Springer.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning 8*(3-4), 229–256.

Wu, T. T., K. Lange, et al. (2008). Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics 2*(1), 224–244.

Xiao, H., K. Rasul, and R. Vollgraf (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 116.

Zhao, P. and T. Zhang (2015). Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pp. 1–9.