SGD and Friends

How to solve large-scale optimization problems?

Ketan Rajawat

February 24, 2020

Indian Institute of Technology Kanpur

Outline

1. Context
2. Background
3. Vanilla Stochastic Gradient Descent: Large N
4. Variance-Reduced SGD: Moderate N
5. High-dimensional problems: large d
6. Conclusion

Context

Outline

Problem Formulation

Consider the optimization problem:

$$\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i) \tag{$\mathcal{P}$}$$

Problem Formulation

Consider the optimization problem:

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \xi_i) \tag{$\mathcal{P}$}$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ where d is problem dimension

Problem Formulation

Consider the optimization problem:

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \xi_i) \tag{$\mathcal{P}$}$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ where d is problem dimension
- ξ_i indexes the data points/observations/samples

Problem Formulation

Consider the optimization problem:

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \xi_i) \tag{$\mathcal{P}$}$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ where d is problem dimension
- ξ_i indexes the data points/observations/samples
- N is the size of data set

Variants

- Online optimization or $N \to \infty$

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \mathbb{E}_{\xi}\left[f(\mathbf{x}, \xi)\right]$$

Variants

- Online optimization or $N \to \infty$

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \mathbb{E}_{\xi}\left[f(\mathbf{x}, \xi)\right]$$

- Use a regularizer h

$$\min_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x}) := F(\mathbf{x}) + h(\mathbf{x})$$

Variants

- Online optimization or $N \to \infty$

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) := \mathbb{E}_{\xi}\left[f(\mathbf{x}, \xi)\right]$$

- Use a regularizer h

$$\min_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x}) := F(\mathbf{x}) + h(\mathbf{x})$$

- Distributed/decentralized setting with K nodes

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^{K} R_k(\mathbf{x})$$

Challenges of Big Data

- Large dimension d
  - Hessian inverse $[\nabla^2 F(\mathbf{x})]^{-1}$ requires $\mathcal{O}(d^3)$ computations
  - Approximate Hessian inverse still requires $\mathcal{O}(d^2)$ computations, e.g., BFGS
  - Very large d: must store $\mathbf{x}$ on the disk instead of RAM, write operation is bottleneck

Challenges of Big Data

- Large dimension d
    - Hessian inverse $[\nabla^2 F(\mathbf{x})]^{-1}$ requires $\mathcal{O}(d^3)$ computations
    - Approximate Hessian inverse still requires $\mathcal{O}(d^2)$ computations, e.g., BFGS
    - Very large d: must store $\mathbf{x}$ on the disk instead of RAM, write operation is bottleneck

- Large dataset size N
    - Even calculating the gradient $\nabla F(\mathbf{x})$ at every iteration impractical
    - Cannot store entire data on a single machine
    - Read/write operations become the bottleneck

Challenges of Big Data

- Large dimension d
  - Hessian inverse $[\nabla^2 F(\mathbf{x})]^{-1}$ requires $\mathcal{O}(d^3)$ computations
  - Approximate Hessian inverse still requires $\mathcal{O}(d^2)$ computations, e.g., BFGS
  - Very large d: must store $\mathbf{x}$ on the disk instead of RAM, write operation is bottleneck

- Large dataset size N
  - Even calculating the gradient $\nabla F(\mathbf{x})$ at every iteration impractical
  - Cannot store entire data on a single machine
  - Read/write operations become the bottleneck

- Ideally complexity should be $\mathcal{O}(dN)$

Outline

Example: Lasso Regression

Predictors for breast cancer selected via LASSO regression [Wang et al., 2016]

Variables	Coefficient Premenopausal	Coefficient Postmenopausal
Age	0.367	0.346
Body mass index		0.935
Age at menarche		−0.075
Age at 1st give birth		0.141
Number of parity	0.137	−0.184
Breast feeding		−0.110
Oral contraceptive		−0.090
hormone replace treatment		−0.710
Case number of BCFDR	0.855	0.844
Benign breast diseases		0.296
Alcohol drinking	0.631	
LAN	0.264	0.238
Sleep quality	−0.256	−0.122

Age (20, 30, 40, 50, 60, 70, and >70 years old); body mass index (<18.5, 18.5–24, 24–27, and ≥27); age at menarche (<12, 12, 13, 14, 15, and 16~ years old); age at 1st give birth (<20, 20–25, and 25~ years old); number of parity (0, 1, 2, and >2); breast feeding duration (no, <1, 1–3 and, >3 years); LAN (1, dark; 2, few light; and 3, little bright); sleep quality (1, good; 2, common; 3, poor; and 4, poor with sleep pill). BCFDR = breast cancer in first degree-relatives, LAN = light at night, LASSO = least absolute shrinkage and selection operator, SD = standard deviation.

Example: Lasso Regression

- Given feature-label pairs $(\mathbf{a}_i, b_i)$ for each patient $i \in \{1, \ldots, N\}$

Example: Lasso Regression

- Given feature-label pairs $(\mathbf{a}_i, b_i)$ for each patient $i \in \{1, \dots, N\}$
- Optimization problem formulated as

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{a}_i^\top \mathbf{x}, b_i) + \lambda \|\mathbf{x}\|_1$$

Example: Lasso Regression

- Given feature-label pairs $(\mathbf{a}_i, b_i)$ for each patient $i \in \{1, \ldots, N\}$
- Optimization problem formulated as

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{a}_i^\top \mathbf{x}, b_i) + \lambda \|\mathbf{x}\|_1$$

- Loss function ℓ could be least-squares, logistic, hinge loss, etc.

Example: Lasso Regression

- Given feature-label pairs $(\mathbf{a}_i, b_i)$ for each patient $i \in \{1, \ldots, N\}$
- Optimization problem formulated as

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{a}_i^\top \mathbf{x}, b_i) + \lambda \left\| \mathbf{x} \right\|_1$$

- Loss function ℓ could be least-squares, logistic, hinge loss, etc.
- Non-zero entries of $\mathbf{x}$ correspond to features that explain b_i

Example: Lasso Regression

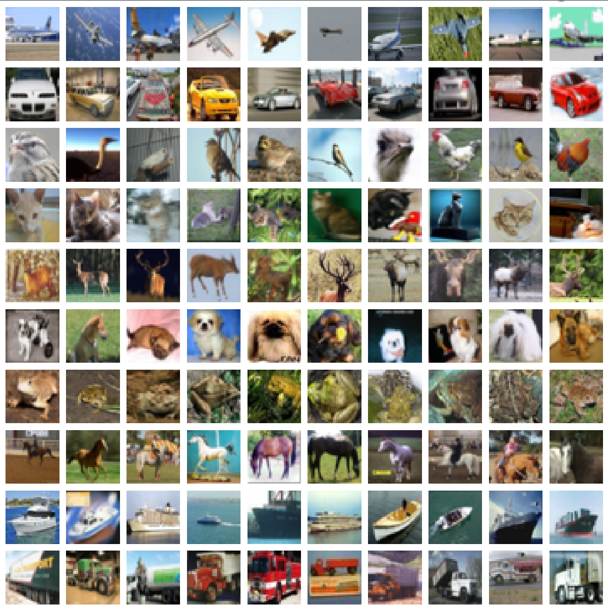
- Given feature-label pairs $(\mathbf{a}_i, b_i)$ for each patient $i \in \{1, \ldots, N\}$
- Optimization problem formulated as

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{a}_i^\top \mathbf{x}, b_i) + \lambda \|\mathbf{x}\|_1$$

- Loss function ℓ could be least-squares, logistic, hinge loss, etc.
- Non-zero entries of $\mathbf{x}$ correspond to features that explain b_i
- ℓ_1-norm penalty "encourages" sparsity

Example: Visual Object Recognition

CIFAR-10 dataset contains 60000 labeled images of 10 objects [Krizhevsky, 2009]



Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and
    - structure of $\mathsf{NN}()$ is defined by the neural network

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and
    - structure of $\mathsf{NN}()$ is defined by the neural network
    - elements of $\mathbf{x}$ are weights/parameters of the network

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and
    - structure of $\mathsf{NN}()$ is defined by the neural network
    - elements of $\mathbf{x}$ are weights/parameters of the network
- $\nabla_{\mathbf{x}} \mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ can be efficiently calculated via *back-propagation*

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and
    - structure of $\mathsf{NN}()$ is defined by the neural network
    - elements of $\mathbf{x}$ are weights/parameters of the network
- $\nabla_{\mathbf{x}} \mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ can be efficiently calculated via *back-propagation*
- Deep Learning community focuses on designing NN

Example: Neural Networks

- Given feature-label pairs $(\mathbf{a}_i, b_i)$, optimization problem is

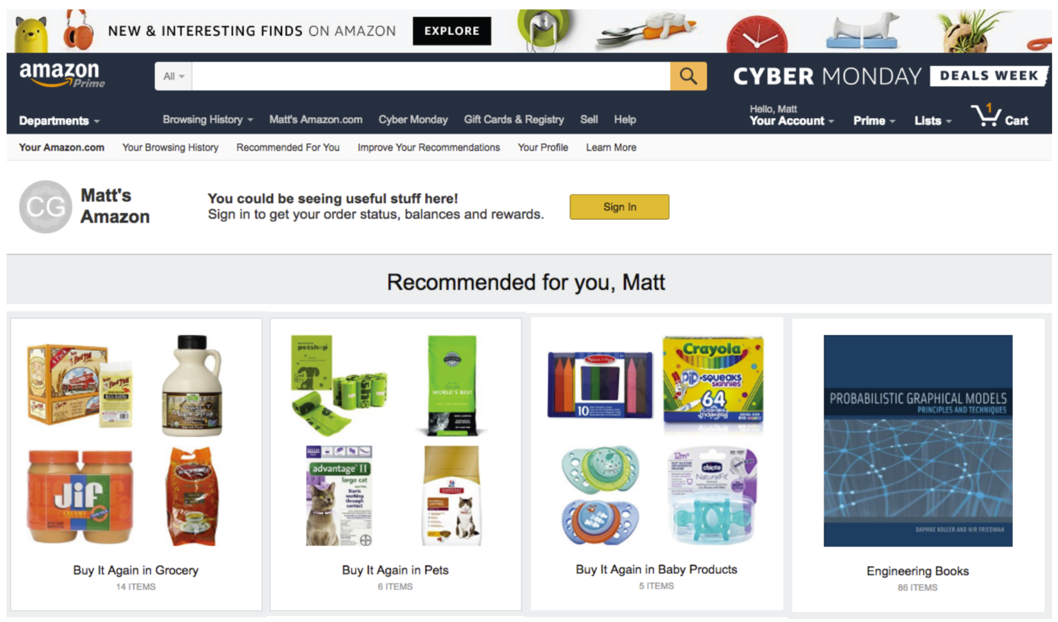
$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, (\mathbf{a}_i, b_i))$$

- Objective f is non-convex and may take the form

$$f(\mathbf{x}, (\mathbf{a}_i, b_i)) = \ell(\mathsf{NN}(\mathbf{a}_i, \mathbf{x}), b_i)$$

- Here, $\mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ is a non-linear function of $\mathbf{x}$, and
  - structure of $\mathsf{NN}()$ is defined by the neural network
  - elements of $\mathbf{x}$ are weights/parameters of the network
- $\nabla_{\mathbf{x}} \mathsf{NN}(\mathbf{a}_i, \mathbf{x})$ can be efficiently calculated via *back-propagation*
- Deep Learning community focuses on designing NN
- Optimization community focuses on solving (GD) for general f

Example: Recommender Systems



Example: Non-negative Matrix Completion

- Given ratings matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$ with observed entries $\{M_{ij}\}_{(i,j) \in \Omega}$

Example: Non-negative Matrix Completion

- Given ratings matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$ with observed entries $\{M_{ij}\}_{(i,j)\in\Omega}$
- Find the complete matrix $\mathbf{X}$

Example: Non-negative Matrix Completion

- Given ratings matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$ with observed entries $\{M_{ij}\}_{(i,j)\in\Omega}$
- Find the complete matrix $\mathbf{X}$
- If $\mathbf{X}$ is suspected to be low-rank, solve [Recht et al., 2011]

$$\min_{\mathbf{X}\in\mathbb{R}_+^{m_1\times m_2}} \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (M_{i,j} - X_{i,j})^2 + \lambda \|\mathbf{X}\|_\star$$

Example: Non-negative Matrix Completion

- Given ratings matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$ with observed entries $\{M_{ij}\}_{(i,j)\in\Omega}$
- Find the complete matrix $\mathbf{X}$
- If $\mathbf{X}$ is suspected to be low-rank, solve [Recht et al., 2011]

$$\min_{\mathbf{X}\in\mathbb{R}_+^{m_1 \times m_2}} \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (M_{i,j} - X_{i,j})^2 + \lambda \|\mathbf{X}\|_\star$$

- Here, $\|\mathbf{X}\|_\star$ encourages $\mathbf{X}$ to be low-rank

Example: Non-negative Matrix Completion

- Given ratings matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$ with observed entries $\{M_{ij}\}_{(i,j)\in\Omega}$
- Find the complete matrix $\mathbf{X}$
- If $\mathbf{X}$ is suspected to be low-rank, solve [Recht et al., 2011]

$$\min_{\mathbf{X}\in\mathbb{R}_{+}^{m_1 \times m_2}} \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (M_{i,j} - X_{i,j})^2 + \lambda \|\mathbf{X}\|_{\star}$$

- Here, $\|\mathbf{X}\|_{\star}$ encourages $\mathbf{X}$ to be low-rank
- High-dimensional problem: since $d = m_1 m_2 \gg |\Omega| = N$

Outline

How to compare?

- Which is better: GD or SGD?

How to compare?

- Which is better: GD or SGD?
- Which variant of SGD should I use for a given problem?

How to compare?

- Which is better: GD or SGD?
- Which variant of SGD should I use for a given problem?
- Such questions arise in any field

How to compare?

- Which is better: GD or SGD?
- Which variant of SGD should I use for a given problem?
- Such questions arise in any field
- Sometimes left unanswered, e.g. in, Deep Learning

How to compare?

- Which is better: GD or SGD?
- Which variant of SGD should I use for a given problem?
- Such questions arise in any field
- Sometimes left unanswered, e.g. in, Deep Learning
- But, the landscape of SGD is much more structured

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit
- So an algorithm that makes fewer calls to the oracle is better!

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit
- So an algorithm that makes fewer calls to the oracle is better!
- Oracle complexity is the cost required to obtain a desired accuracy

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit
- So an algorithm that makes fewer calls to the oracle is better!
- Oracle complexity is the cost required to obtain a desired accuracy

Oracle complexity of SGD: convex objectives

For general convex objective functions, SGD requires $\mathcal{O}(\frac{Ld}{\epsilon^2})$ calls to oracle in order to achieve an optimality gap of ϵ.

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit
- So an algorithm that makes fewer calls to the oracle is better!
- Oracle complexity is the cost required to obtain a desired accuracy

Oracle complexity of SGD: convex objectives

For general convex objective functions, SGD requires $\mathcal{O}(\frac{Ld}{\epsilon^2})$ calls to oracle in order to achieve an optimality gap of ϵ.

- Terms within $\mathcal{O}$ may be initialization dependent
- Notation hides away many complexities

Oracle Complexity

- Given $\mathbf{x}$, an oracle provides us $\nabla f(\mathbf{x}, \xi_i)$
- Call to an oracle costs 1 unit
- So an algorithm that makes fewer calls to the oracle is better!
- Oracle complexity is the cost required to obtain a desired accuracy

Oracle complexity of SGD: convex objectives

For general convex objective functions, SGD requires $\mathcal{O}(\frac{Ld}{\epsilon^2})$ calls to oracle in order to achieve an optimality gap of ϵ.

- Terms within $\mathcal{O}$ may be initialization dependent
- Notation hides away many complexities
- Gap measured by $\|\mathbf{x} - \mathbf{x}^{\star}\|^2$, $\|\nabla F(\mathbf{x})\|^2$, or $F(\mathbf{x}) - F(\mathbf{x}^{\star})$

State-of-the-art in SGD

- New avenues for applying SGD emerge every year

State-of-the-art in SGD

- New avenues for applying SGD emerge every year
- Several variants of SGD are proposed every month

State-of-the-art in SGD

- New avenues for applying SGD emerge every year
- Several variants of SGD are proposed every month
- Papers analyzing performance of these variants come up everyday

State-of-the-art in SGD

- New avenues for applying SGD emerge every year
- Several variants of SGD are proposed every month
- Papers analyzing performance of these variants come up everyday
- Difficult to consolidate and maintain perspective

This Tutorial

- Unified view of many SGD variants

This Tutorial

- Unified view of many SGD variants
- Based on recent results, but readily accessible: “easy” math

This Tutorial

- Unified view of many SGD variants
- Based on recent results, but readily accessible: “easy” math
- First timers: do not try to understand it all, but do ask questions

This Tutorial

- Unified view of many SGD variants
- Based on recent results, but readily accessible: "easy" math
- First timers: do not try to understand it all, but do ask questions
- Up-and-comers: identify gaps and target them, also keep asking questions

This Tutorial

- Unified view of many SGD variants
- Based on recent results, but readily accessible: "easy" math
- First timers: do not try to understand it all, but do ask questions
- Up-and-comers: identify gaps and target them, also keep asking questions
- Experts: what new result am I unaware of?

This Tutorial

- Unified view of many SGD variants
- Based on recent results, but readily accessible: “easy” math
- First timers: do not try to understand it all, but do ask questions
- Up-and-comers: identify gaps and target them, also keep asking questions
- Experts: what new result am I unaware of?
- Later: get slides from my website

References

- Key reference text: [Beck, 2017]
- Introductory (deterministic): [Vandenberghe, 2019]
- [Bubeck et al., 2015] is good introduction to the topic
- Related course lecture notes: [Saunders, 2019, Chen, 2019]
- Sebastien Bubeck's blog: [Bubeck, 2019]
- This tutorial is an amalgamation of [Gorbunov et al., 2019], [Bottou et al., 2018], and [Recht et al., 2011]
- Inspired from the tutorial: `https://www.youtube.com/watch?v=a05S0kL5u30`

Background

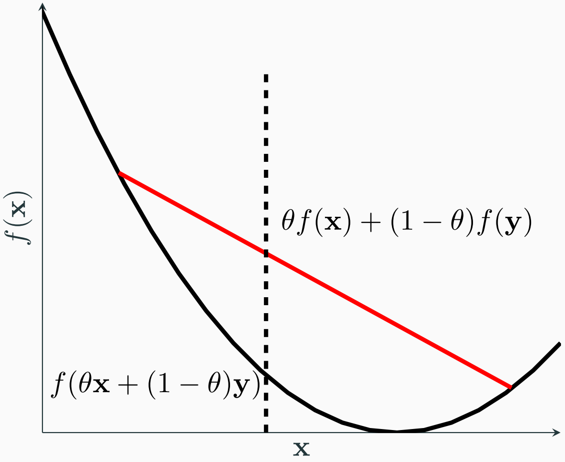
Outline

Convex Functions: Zeroth Order Condition

Definition

A function f is convex if (a) its domain is a convex set; and (b) it satisfies

$$f(\theta \mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y})$$



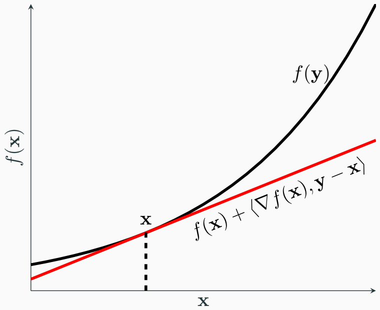
Convex Functions: First and Second Order Conditions

Definition

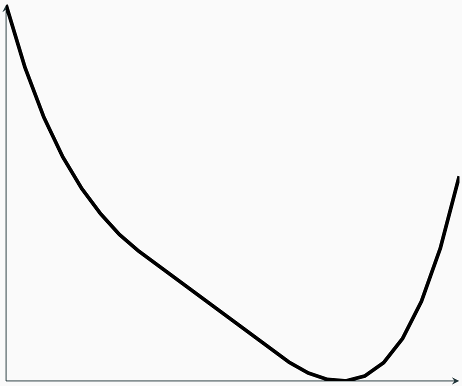
A function f is convex if (a) its domain is a convex set; and (b) it satisfies

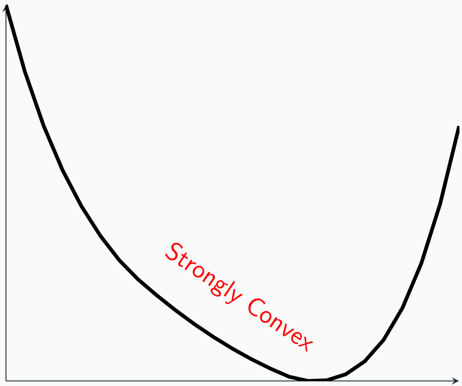
$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

Alternatively: eigenvalues of $(\nabla^2 F(\mathbf{x})) \geq 0$



Strongly Convex Functions





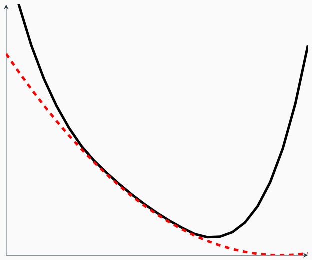
Strongly Convex Functions: Quadratic Lower Bound

Definition

A function F is μ-strongly convex if (a) its domain is a convex set; and (b) it satisfies

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

where $\mu > 0$. Alternatively, eigenvalues of $(\nabla^2 F(\mathbf{x})) \geq \mu$



Strongly Convex Functions: Quadratic Lower Bound

Definition

A function F is μ-strongly convex if (a) its domain is a convex set; and (b) it satisfies

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

where $\mu > 0$. Alternatively, eigenvalues of $(\nabla^2 F(\mathbf{x})) \geq \mu$

ℓ_2-norm square example

The function $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$ is 1-strongly convex

Strongly Convex Functions: Quadratic Lower Bound

Definition

A function F is μ-strongly convex if (a) its domain is a convex set; and (b) it satisfies

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

where $\mu > 0$. Alternatively, eigenvalues of $(\nabla^2 F(\mathbf{x})) \geq \mu$

ℓ_2-norm square example

The function $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$ is 1-strongly convex

Least-squares example

Is the lasso regression objective strongly convex? Recall

$$R(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{a}_i^\top \mathbf{x} - b_i)^2 + \lambda \|\mathbf{x}\|_1.$$

Strongly Convex Functions: Quadratic Lower Bound

Definition

A function F is μ-strongly convex if (a) its domain is a convex set; and (b) it satisfies

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

where $\mu > 0$. Alternatively, eigenvalues of $(\nabla^2 F(\mathbf{x})) \geq \mu$

ℓ_2-norm square example

The function $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$ is 1-strongly convex

Least-squares example

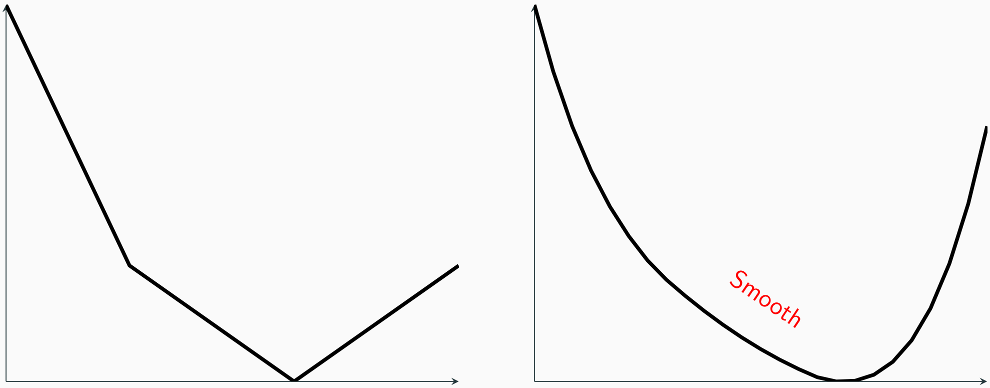
Is the lasso regression objective strongly convex? Recall

$$R(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{a}_i^\top \mathbf{x} - b_i)^2 + \lambda \|\mathbf{x}\|_1 .$$

Show that for this case $\mu =$ smallest eigenvalue of $\frac{1}{N} \sum_{i=1}^{N} \mathbf{a}_i \mathbf{a}_i^\top$

Outline

Smooth Functions



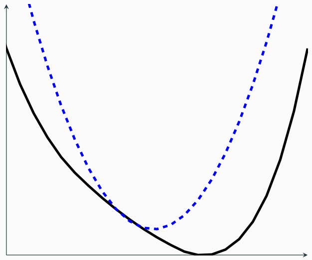
Smooth Functions: Quadratic Upper Bound

Definition

A function F is L-smooth

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Alternatively: eigenvalues of $(\nabla^2 F(\mathbf{x})) \leq L$



Bregman Divergence

- Bregman divergence over a function F is defined as

$$D_F(\mathbf{x}, \mathbf{y}) = F(\mathbf{y}) - F(\mathbf{x}) - \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

Bregman Divergence

- Bregman divergence over a function F is defined as

$$D_F(\mathbf{x}, \mathbf{y}) = F(\mathbf{y}) - F(\mathbf{x}) - \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

- Bregman divergence is not symmetric (and not a metric) but satisfies

$$\frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2 \leq D_F(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Bregman Divergence

- Bregman divergence over a function F is defined as

$$D_F(\mathbf{x}, \mathbf{y}) = F(\mathbf{y}) - F(\mathbf{x}) - \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

- Bregman divergence is not symmetric (and not a metric) but satisfies

$$\begin{aligned} \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2 &\leq D_F(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2 \\ \frac{1}{2L} \|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|^2 &\leq D_F(\mathbf{x}, \mathbf{y}) \leq \frac{1}{2\mu} \|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|^2 \end{aligned}$$

Outline

Non-smooth convex functions

- If h is non-smooth convex, may still define subgradient $\mathbf{v}(\mathbf{x}) \in \partial h(\mathbf{x})$

Non-smooth convex functions

- If h is non-smooth convex, may still define subgradient $\mathbf{v}(\mathbf{x}) \in \partial h(\mathbf{x})$
- Satisfies first order convexity condition as usual

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{v}(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

Non-smooth convex functions

- If h is non-smooth convex, may still define subgradient $\mathbf{v}(\mathbf{x}) \in \partial h(\mathbf{x})$
- Satisfies first order convexity condition as usual

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{v}(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$$

- Optimality condition for $\mathbf{x}^\star = \arg\min_{\mathbf{x}} f(\mathbf{x})$:

$$\mathbf{v}(\mathbf{x}^\star) = 0 \in \partial h(\mathbf{x}^\star)$$

Projection Operator

- Define the projection over a set $\mathcal{X}$ as

$$\mathcal{P}_{\mathcal{X}}\left(\mathbf{x}\right) = \arg\min_{\mathbf{y}\in\mathcal{X}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{x}\right\|^2$$

Projection Operator

- Define the projection over a set $\mathcal{X}$ as

$$\mathcal{P}_{\mathcal{X}}\left(\mathbf{x}\right) = \arg\min_{\mathbf{y}\in\mathcal{X}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{x}\right\|^2$$

- Equivalent formulation

$$\mathcal{P}_{\mathcal{X}}\left(\mathbf{x}\right) = \arg\min_{\mathbf{y}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{x}\right\|^2 + \mathbb{1}_{\mathcal{X}}(\mathbf{x})$$

where the indicator function is defined as

$$\mathbb{1}_{\mathcal{X}}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x}\in\mathcal{X} \\ \infty & \mathbf{x}\notin\mathcal{X} \end{cases}$$

Proximal Operator

- Proximal operator generalizes projection

$$\mathsf{prox}_h(\mathbf{x}) = \mathbf{y}^\star = \arg\min_{\mathbf{y}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + h(\mathbf{x})$$

Proximal Operator

- Proximal operator generalizes projection

$$\text{prox}_h(\mathbf{x}) = \mathbf{y}^\star = \arg\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2 + h(\mathbf{x})$$

- Useful property: differentiate and equate to zero

$$\mathbf{y}^\star - \mathbf{x} + \mathbf{v}(\mathbf{y}^\star) = 0$$

where $\mathbf{y}^\star = \text{prox}_h(\mathbf{x})$ and $\mathbf{v}(\mathbf{y}^\star) \in \partial h(\mathbf{y}^\star)$

Vanilla Stochastic Gradient Descent: Large N

Outline

1. Context
2. Background
3. Vanilla Stochastic Gradient Descent: Large N
   - Gradient Descent vs. Stochastic Gradient Descent
   - Performance of Stochastic Grandient Descent
4. Variance-Reduced SGD: Moderate N
5. High-dimensional problems: large d
6. Conclusion

Gradient Descent vs. Stochastic Gradient Descent

- Gradient descent for solving $(\mathcal{P})$

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_t - \frac{\eta}{N}\sum_{i=1}^{N} \nabla f(\mathbf{x}_t, \xi_i)\right)$$

- N oracle calls per iteration

Gradient Descent vs. Stochastic Gradient Descent

- Gradient descent for solving ($\mathcal{P}$)

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_t - \frac{\eta}{N}\sum_{i=1}^{N}\nabla f(\mathbf{x}_t, \xi_i)\right)$$

- N oracle calls per iteration
- Stochastic gradient descent for solving ($\mathcal{P}$)

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_t - \eta\nabla f(\mathbf{x}_t, \xi_{i_t})\right)$$

where $i_t \in \{1, \ldots, N\}$ is a random number.

Gradient Descent vs. Stochastic Gradient Descent

- Gradient descent for solving ($\mathcal{P}$)

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \left(\mathbf{x}_t - \frac{\eta}{N} \sum_{i=1}^{N} \nabla f(\mathbf{x}_t, \xi_i) \right)$$

- N oracle calls per iteration
- Stochastic gradient descent for solving ($\mathcal{P}$)

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \left(\mathbf{x}_t - \eta \nabla f(\mathbf{x}_t, \xi_{i_t}) \right)$$

  where $i_t \in \{1, \ldots, N\}$ is a random number.
- Descent direction on average: expectation w.r.t. i_t

$$\mathbb{E}_{i_t} \left[\nabla f(\mathbf{x}_t, \xi_{i_t}) \right] = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}_t, \xi_i) = \nabla F(\mathbf{x}_t)$$

Intuition

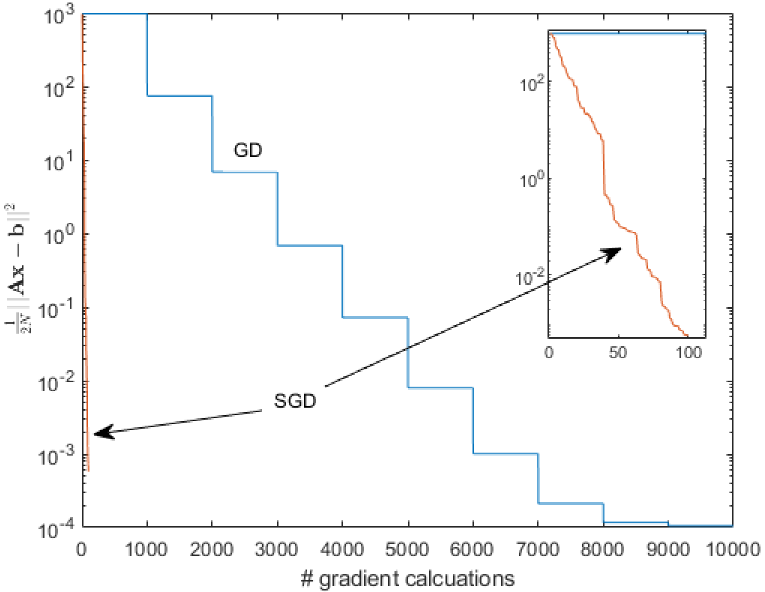
- SGD more efficient at accessing data

Intuition

- SGD more efficient at accessing data
- handles redundancy in dataset better

Intuition

- SGD more efficient at accessing data
- handles redundancy in dataset better
- consider lasso example: features $\mathbf{a}_i \in \text{span}(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)})$



History of SGD

- Given (X, Y) observations, let $\Phi(X)$ be a transformation
- SGD has been applied to specific problems

Algorithm	Loss	Gradient/Subgradient
LMS (Widrow-Hoff'60)	$\frac{1}{2}(Y - \Phi(X)^\top \mathbf{x})^2$	$(\Phi(X)^\top \mathbf{x} - Y)\Phi(X)$

History of SGD

- Given (X, Y) observations, let $\Phi(\mathsf{X})$ be a transformation
- SGD has been applied to specific problems

Algorithm	Loss	Gradient/Subgradient
LMS (Widrow-Hoff'60)	$\frac{1}{2}(\mathsf{Y} - \Phi(\mathsf{X})^\top \mathbf{x})^2$	$(\Phi(\mathsf{X})^\top \mathbf{x} - \mathsf{Y})\Phi(\mathsf{X})$
Perceptron (Rosenblatt'57)	$[-\mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle]_+$	$-\mathsf{Y}\Phi(\mathsf{X})\mathbb{1}_{\mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle \leq 0}$

History of SGD

- Given (X, Y) observations, let $\Phi(\mathsf{X})$ be a transformation
- SGD has been applied to specific problems

Algorithm	Loss	Gradient/Subgradient
LMS (Widrow-Hoff'60)	$\frac{1}{2}(\mathsf{Y} - \Phi(\mathsf{X})^\top \mathbf{x})^2$	$(\Phi(\mathsf{X})^\top \mathbf{x} - \mathsf{Y})\Phi(\mathsf{X})$
Perceptron (Rosenblatt'57)	$[-\mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle]_+$	$-\mathsf{Y}\Phi(\mathsf{X})\mathbb{1}_{\mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle \leq 0}$
SVM (Cortes-Vapnik'95)	$\frac{\lambda}{2}\lVert \mathbf{x} \rVert^2 + [1 - \mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle]_+$	$\lambda \mathbf{x} - \mathsf{Y}\Phi(\mathsf{X})\mathbb{1}_{\mathsf{Y}\langle \Phi(\mathsf{X}), \mathbf{x}\rangle \leq 1}$

Outline

Assumptions

L-smoothness

$$D_F(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Assumptions

L-smoothness

$$D_F(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

μ-convexity

$$D_F(\mathbf{x}, \mathbf{y}) \geq \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Assumptions

L-smoothness

$$D_F(\mathbf{x}, \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

μ-convexity

$$D_F(\mathbf{x}, \mathbf{y}) \geq \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Bounded Variance

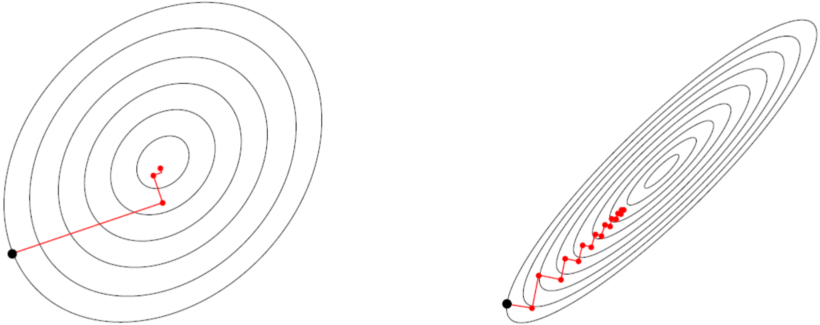
$$\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}, \xi_{i_t})\|^2\right] \leq \sigma^2 + c\,\|\nabla F(\mathbf{x})\|^2$$

$$\Rightarrow \mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] \leq \sigma^2$$

provided $\nabla F(\mathbf{x}^\star) = 0$ and $c \geq 1$.

σ^2 is the inherent data variance

Strong Convexity and Smoothness: Condition Number



(small $\kappa = L/\mu$) (large $\kappa = L/\mu$)

Oracle Complexity for SGD: Strongly Convex + Smooth

Lemma (SGD: Strongly Convex + Smooth [Bottou et al., 2018])

For L-smooth, μ-convex functions, SGD incurs oracle complexity of $\mathcal{O}\left(\frac{L}{\mu\epsilon}\right)$.

Oracle Complexity for SGD: Strongly Convex + Smooth

Lemma (SGD: Strongly Convex + Smooth [Bottou et al., 2018])

For L-smooth, μ-convex functions, SGD incurs oracle complexity of $\mathcal{O}\left(\frac{L}{\mu\epsilon}\right)$.

For simplicity, consider unconstrained version: $\mathbf{x}_{t+1} - \mathbf{x}_t = \eta \nabla f(\mathbf{x}_t, \xi_{i_t})$

Proof: Step 1. Quadratic upper bound (L-smoothness):

$$F(\mathbf{x}_{t+1}) \leq F(\mathbf{x}_t) + \langle \nabla F(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$$

Oracle Complexity for SGD: Strongly Convex + Smooth

Lemma (SGD: Strongly Convex + Smooth [Bottou et al., 2018])

For L-smooth, μ-convex functions, SGD incurs oracle complexity of $\mathcal{O}\left(\frac{L}{\mu\epsilon}\right)$.

For simplicity, consider unconstrained version: $\mathbf{x}_{t+1} - \mathbf{x}_t = \eta \nabla f(\mathbf{x}_t, \xi_{i_t})$

Proof: Step 1. Quadratic upper bound (L-smoothness):

$$\begin{aligned} F(\mathbf{x}_{t+1}) &\leq F(\mathbf{x}_t) + \langle \nabla F(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \\ &= F(\mathbf{x}_t) - \eta \langle \nabla F(\mathbf{x}_t), \nabla f(\mathbf{x}_t, \xi_{i_t}) \rangle + \frac{\eta^2 L}{2} \|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2 \end{aligned}$$

Update Equation

$\mathbf{x}_{t+1} - \mathbf{x}_t = \eta \nabla f(\mathbf{x}_t, \xi_{i_t})$

SGD: Strongly Convex + Smooth

Step 2. Take expectation

$$\mathbb{E}_{i_t}[F(\mathbf{x}_{t+1})] \leq F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right]$$

SGD: Strongly Convex + Smooth

Step 2. Take expectation, use $\mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}_t, \xi_{i_t})\right] = \nabla F(\mathbf{x}_t)$

$$
\begin{aligned}
\mathbb{E}_{i_t}[F(\mathbf{x}_{t+1})] &\le F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&= F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \nabla F(\mathbf{x}_t)\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right]
\end{aligned}
$$

SGD: Strongly Convex + Smooth

Step 2. Take expectation, use $\mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}_t, \xi_{i_t})\right] = \nabla F(\mathbf{x}_t)$

$$\begin{aligned}
\mathbb{E}_{i_t}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&= F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \nabla F(\mathbf{x}_t)\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&\leq F(\mathbf{x}_t) - \eta\left(1 - \frac{\eta L c}{2}\right)\|\nabla F(\mathbf{x}_t)\|_2^2 + \frac{\eta^2\sigma^2 L}{2}
\end{aligned}$$

$$\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}, \xi_{i_t})\|^2\right] \leq \sigma^2 + c\|\nabla F(\mathbf{x})\|^2$$

SGD: Strongly Convex + Smooth

Step 2. Take expectation, use $\mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}_t, \xi_{i_t})\right] = \nabla F(\mathbf{x}_t)$

$$
\begin{aligned}
\mathbb{E}_{i_t}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&= F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \nabla F(\mathbf{x}_t)\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&\leq F(\mathbf{x}_t) - \eta\left(1 - \frac{\eta L c}{2}\right)\|\nabla F(\mathbf{x}_t)\|_2^2 + \frac{\eta^2 \sigma^2 L}{2} \\
&\leq F(\mathbf{x}_t) - \frac{\eta}{2}\|\nabla F(\mathbf{x}_t)\|_2^2 + \frac{\eta^2 \sigma^2 L}{2}
\end{aligned}
$$

$\eta L c < 1$

SGD: Strongly Convex + Smooth

Step 2. Take expectation, use $\mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})] = \nabla F(\mathbf{x}_t)$

$$
\begin{aligned}
\mathbb{E}_{i_t}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&= F(\mathbf{x}_t) - \eta\langle \nabla F(\mathbf{x}_t), \nabla F(\mathbf{x}_t)\rangle + \frac{\eta^2 L}{2}\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t})\|^2\right] \\
&\leq F(\mathbf{x}_t) - \eta\left(1 - \frac{\eta L c}{2}\right)\|\nabla F(\mathbf{x}_t)\|_2^2 + \frac{\eta^2 \sigma^2 L}{2} \\
&\leq F(\mathbf{x}_t) - \frac{\eta}{2}\|\nabla F(\mathbf{x}_t)\|_2^2 + \frac{\eta^2 \sigma^2 L}{2}
\end{aligned}
$$

Function decrement in SGD

Function value decreases (on average) only when the gradient is large!

SGD: Strongly Convex + Smooth

Step 3. Relate $\|\nabla F(\mathbf{x}_t)\|^2$ with optimality gap:
subtract $F(\mathbf{x}^\star)$, and use strong convexity

$$\mathbb{E}_{i_t}\left[F(\mathbf{x}_{t+1})\right] - F(\mathbf{x}^\star) \leq F(\mathbf{x}_t) - F(\mathbf{x}^\star) - \frac{\eta}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta^2\sigma^2 L}{2}$$

SGD: Strongly Convex + Smooth

Step 3. Relate $\|\nabla F(\mathbf{x}_t)\|^2$ with optimality gap:

subtract $F(\mathbf{x}^\star)$, and use strong convexity

$$\begin{aligned}
\mathbb{E}_{i_t}\left[F(\mathbf{x}_{t+1})\right] - F(\mathbf{x}^\star) &\leq F(\mathbf{x}_t) - F(\mathbf{x}^\star) - \frac{\eta}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta^2\sigma^2 L}{2} \\
&\leq (1-\mu\eta)(F(\mathbf{x}_t) - F(\mathbf{x}^\star)) + \frac{\eta^2\sigma^2 L}{2}
\end{aligned}$$

$$\frac{1}{2}\|\nabla F(\mathbf{x}_t)\|^2 \geq \mu(F(\mathbf{x}_t) - F(\mathbf{x}^\star))$$

SGD: Strongly Convex + Smooth

Step 3. Relate $\|\nabla F(\mathbf{x}_t)\|^2$ with optimality gap:
subtract $F(\mathbf{x}^\star)$, and use strong convexity

$$\mathbb{E}_{i_t}\left[F(\mathbf{x}_{t+1})\right] - F(\mathbf{x}^\star) \leq F(\mathbf{x}_t) - F(\mathbf{x}^\star) - \frac{\eta}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{\eta^2\sigma^2 L}{2}$$
$$\leq (1-\mu\eta)(F(\mathbf{x}_t) - F(\mathbf{x}^\star)) + \frac{\eta^2\sigma^2 L}{2}$$

Set $\Delta_t = \mathbb{E}[F(\mathbf{x}_{t+1}) - F(\mathbf{x}^\star)]$

SGD: Strongly Convex + Smooth

Step 3. Relate $\|\nabla F(\mathbf{x}_t)\|^2$ with optimality gap:

subtract $F(\mathbf{x}^\star)$, and use strong convexity

$$\begin{aligned}\mathbb{E}_{i_t}\left[F(\mathbf{x}_{t+1})\right] - F(\mathbf{x}^\star) &\le F(\mathbf{x}_t) - F(\mathbf{x}^\star) - \frac{\eta}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta^2\sigma^2 L}{2} \\ &\le (1-\mu\eta)(F(\mathbf{x}_t) - F(\mathbf{x}^\star)) + \frac{\eta^2\sigma^2 L}{2}\end{aligned}$$

Set $\Delta_t = \mathbb{E}[F(\mathbf{x}_{t+1}) - F(\mathbf{x}^\star)]$

One-step inequality

$$\Delta_{t+1} \le (1-\mu\eta)\Delta_t + \frac{\eta^2\sigma^2 L}{2}$$

SGD: Strongly Convex + Smooth

One-step inequality

$$\Delta_{t+1} \le (1 - \mu\eta)\Delta_t + \frac{\eta^2 \sigma^2 L}{2}$$

Step 4. Obtain final inequality:

One-step inequality

$$\Delta_{t+1} \leq (1 - \mu\eta)\Delta_t + \frac{\eta^2\sigma^2 L}{2}$$

Step 4. Obtain final inequality:

Apply recursively over $t = 1, \ldots, T$:

$$\Delta_{T+1} \leq (1 - \mu\eta)^{T}\Delta_1 + \frac{\eta^2\sigma^2 L}{2}\frac{1}{\mu\eta}$$

SGD: Strongly Convex + Smooth

Final inequality

$$\Delta_{T+1} \leq (1 - \mu\eta)^T \Delta_1 + \frac{\eta\sigma^2 L}{2\mu}$$

Step 5. Pick η:

SGD: Strongly Convex + Smooth

Final inequality

$$\Delta_{T+1} \leq (1 - \mu\eta)^T \Delta_1 + \frac{\eta\sigma^2 L}{2\mu}$$

Step 5. Pick η:

- Equate each term to $\epsilon \Rightarrow \eta = \mathcal{O}(\frac{\mu\epsilon}{\sigma^2 L})$ (ignore unimportant constants)

SGD: Strongly Convex + Smooth

Final inequality

$$\Delta_{T+1} \leq (1-\mu\eta)^T \Delta_1 + \frac{\eta\sigma^2 L}{2\mu}$$

Step 5. Pick η:

- Equate each term to $\epsilon \Rightarrow \eta = \mathcal{O}(\frac{\mu\epsilon}{\sigma^2 L})$ (ignore unimportant constants)
- Solve for T: $(1-\mu\eta)^T = \epsilon$ and use $\log(1-\mu\eta) \approx -\mu\eta$ to obtain

$$T = \mathcal{O}\left(\frac{\sigma^2 L}{\mu\epsilon}\log\left(\frac{1}{\epsilon}\right)\right) \approx \mathcal{O}\left(\frac{\sigma^2 L}{\mu\epsilon}\right)$$

Practical Considerations

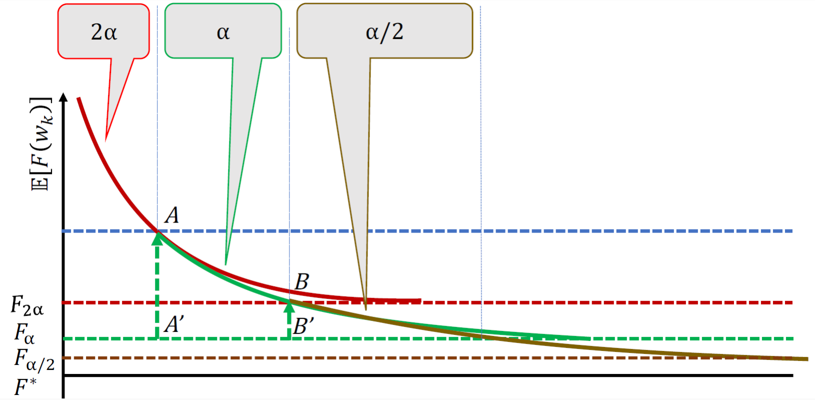
- With fixed η, SGD converges fast, but slows when optimality gap is $\mathcal{O}(\eta)$

Practical Considerations

- With fixed η, SGD converges fast, but slows when optimality gap is $\mathcal{O}(\eta)$
- Can select a diminishing step-size to obtain slight improvement

Practical Considerations

- With fixed η, SGD converges fast, but slows when optimality gap is $\mathcal{O}(\eta)$
- Can select a diminishing step-size to obtain slight improvement
- Other approach: half the step-size when progress stalls [Bottou et al., 2018]



Oracle Complexity for SGD: Smooth

Lemma (SGD: smooth)

For L-smooth functions, SGD incurs oracle complexity of $\mathcal{O}\left(\frac{L}{\epsilon^2}\right)$.

Oracle Complexity for SGD: Smooth

Lemma (SGD: smooth)

For L-smooth functions, SGD incurs oracle complexity of $\mathcal{O}\left(\frac{L}{\epsilon^2}\right)$.

Proof for unconstrained version: $\mathbf{x}_{t+1} - \mathbf{x}_t = \eta \nabla f(\mathbf{x}_t, \xi_{i_t})$.

Recall from L-smoothness and $\eta L c < 1$ (here: $\Delta_t = \mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{x}^\star) \geq 0$):

$$\begin{aligned}
\Delta_{t+1} &\leq \Delta_t - \frac{\eta}{2} \|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta^2 \sigma^2 L}{2} \\
&\leq \Delta_1 - \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla F(\mathbf{x}_t)\|^2 + \frac{T \eta^2 \sigma^2 L}{2}
\end{aligned}$$

□

SGD: Smooth

- Rearrange to obtain:

$$\min_{1\le t\le T} \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|_2^2] \le \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|_2^2] \le \eta\sigma^2 L + \frac{2\Delta_1}{\eta T}$$

SGD: Smooth

- Rearrange to obtain:

$$\min_{1\leq t\leq T}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|_2^2] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|_2^2] \leq \eta\sigma^2 L + \frac{2\Delta_1}{\eta T}$$

- Equate each term to ϵ to obtain $\eta = \dfrac{\epsilon}{\sigma^2 L}$ and

$$T = \mathcal{O}\left(\frac{\sigma^2 L}{\epsilon^2}\right)$$

oracle calls required to reach close to a first order stationary point

Variance-Reduced SGD: Moderate N

Gradient Descent or Stochastic Gradient Descent?

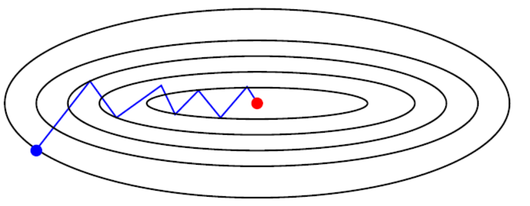


Figure 1: Gradient Descent

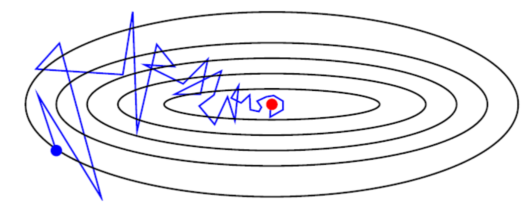


Figure 2: Stochastic Gradient Descent

Gradient Descent or Stochastic Gradient Descent?

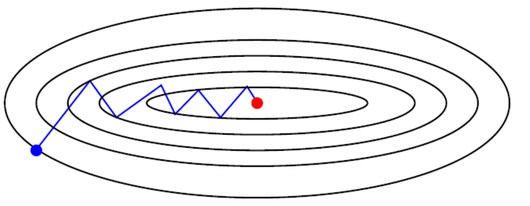


Figure 1: Gradient Descent

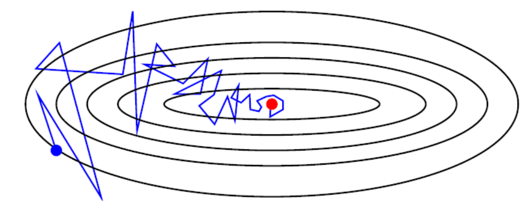


Figure 2: Stochastic Gradient Descent

- Standard gradient descent requires $\mathcal{O}\left(\frac{L}{\mu}\log(\frac{1}{\epsilon})\right)$ iterations

Gradient Descent or Stochastic Gradient Descent?

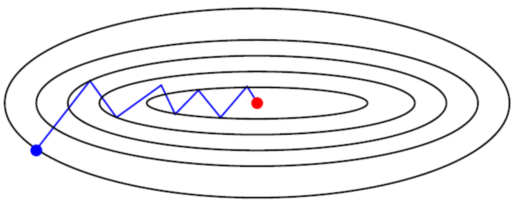


Figure 1: Gradient Descent

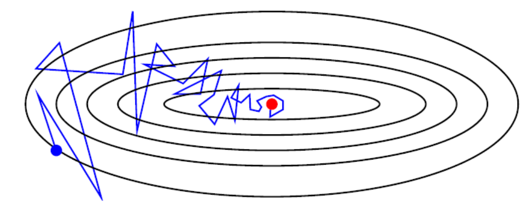


Figure 2: Stochastic Gradient Descent

- Standard gradient descent requires $\mathcal{O}\left(\frac{L}{\mu}\log(\frac{1}{\epsilon})\right)$ iterations
- But each iteration requires N oracle calls: so oracle complexity is $\mathcal{O}\left(\frac{LN}{\mu}\log(\frac{1}{\epsilon})\right)$

Gradient Descent or Stochastic Gradient Descent?

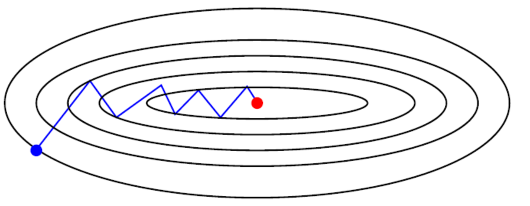


Figure 1: Gradient Descent

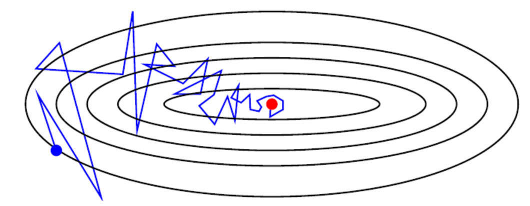
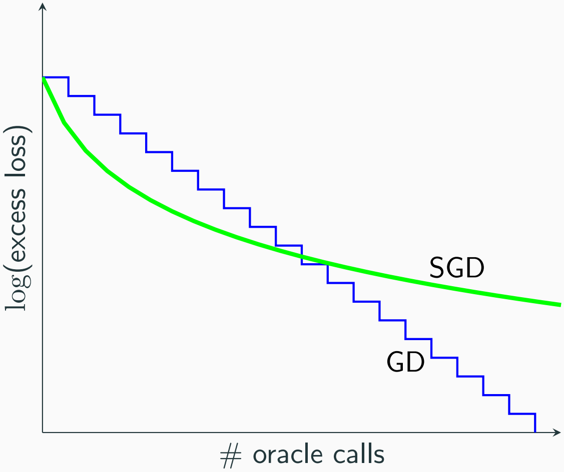


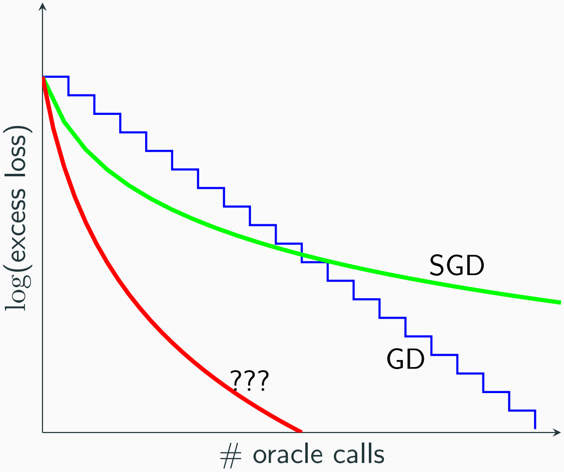
Figure 2: Stochastic Gradient Descent

- Standard gradient descent requires $\mathcal{O}\left(\frac{L}{\mu}\log(\frac{1}{\epsilon})\right)$ iterations
- But each iteration requires N oracle calls: so oracle complexity is $\mathcal{O}\left(\frac{LN}{\mu}\log(\frac{1}{\epsilon})\right)$
- In contrast, SGD requires $\mathcal{O}\left(\frac{L}{\mu\epsilon}\right)$ oracle calls: independent of N

Speeding up SGD?



Speeding up SGD?



Variance Reduction

- We consider the generic SGD algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t$$

where $\mathbf{g}_t$ is an unbiased gradient approximation

- We consider the generic SGD algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t$$

  where $\mathbf{g}_t$ is an unbiased gradient approximation

- Example:

$$\mathbf{g}_t = \frac{1}{N} \sum_{i=1}^{N} \nabla f(\mathbf{x}_t, \xi_i) \quad \text{(GD)}$$

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) \quad \text{(SGD)}$$

(mini-batch)

Variance Reduction

- We consider the generic SGD algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t$$

  where $\mathbf{g}_t$ is an unbiased gradient approximation

- Example:

$$\mathbf{g}_t = \frac{1}{N} \sum_{i=1}^{N} \nabla f(\mathbf{x}_t, \xi_i) \qquad \text{(GD)}$$

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) \qquad \text{(SGD)}$$

$$\mathbf{g}_t = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla f(\mathbf{x}_t, \xi_i) \qquad \text{(mini-batch)}$$

Effect of Mini Batching

- Consider b random variables $\{\mathsf{X}_i\}_{i=1}^{b}$ such that $\mathbb{V}_i(\mathsf{X}_i) = \sigma^2$

Effect of Mini Batching

- Consider b random variables $\{\mathsf{X}_i\}_{i=1}^{b}$ such that $\mathbb{V}_i(\mathsf{X}_i) = \sigma^2$
- Then it holds that $\mathbb{V}_i(\frac{1}{b}\sum_i \mathsf{X}_i) = \frac{\sigma^2}{b}$

Effect of Mini Batching

- Consider b random variables $\{\mathsf{X}_i\}_{i=1}^{b}$ such that $\mathbb{V}_i(\mathsf{X}_i) = \sigma^2$
- Then it holds that $\mathbb{V}_i(\frac{1}{b}\sum_i \mathsf{X}_i) = \frac{\sigma^2}{b}$
- So

$$\#\text{ of iterations} = \mathcal{O}(\frac{L}{\mu b}\log\left(\frac{1}{\epsilon}\right))$$

Effect of Mini Batching

- Consider b random variables $\{\mathsf{X}_i\}_{i=1}^{b}$ such that $\mathbb{V}_i(\mathsf{X}_i) = \sigma^2$
- Then it holds that $\mathbb{V}_i(\frac{1}{b}\sum_i \mathsf{X}_i) = \frac{\sigma^2}{b}$
- So

$$\#\text{ of iterations} = \mathcal{O}(\frac{L}{\mu b}\log\left(\frac{1}{\epsilon}\right))$$

- But each iteration requires b oracle calls: oracle complexity still same

Effect of Mini Batching

- Consider b random variables $\{\mathsf{X}_i\}_{i=1}^{b}$ such that $\mathbb{V}_i(\mathsf{X}_i) = \sigma^2$
- Then it holds that $\mathbb{V}_i(\frac{1}{b}\sum_i \mathsf{X}_i) = \frac{\sigma^2}{b}$
- So

$$\#\text{ of iterations} = \mathcal{O}(\frac{L}{\mu b}\log\left(\frac{1}{\epsilon}\right))$$

- But each iteration requires b oracle calls: oracle complexity still same
- In practice: lesser wall-clock time if gradients can be calculated in parallel

Intuition: Shifted SGD

- Consider the loss functions

$$\phi(\mathbf{x}, \xi_i) = f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x}$$

so that the overall objective remains the same, i.e.,

$$\Phi(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x} = F(\mathbf{x})$$

provided that $\sum_i \mathbf{a}_i = 0$.

Intuition: Shifted SGD

- Consider the loss functions

$$\phi(\mathbf{x}, \xi_i) = f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x}$$

so that the overall objective remains the same, i.e.,

$$\Phi(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x} = F(\mathbf{x})$$

provided that $\sum_i \mathbf{a}_i = 0$.

- Note that $\nabla \phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$

Intuition: Shifted SGD

- Consider the loss functions

$$\phi(\mathbf{x}, \xi_i) = f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x}$$

so that the overall objective remains the same, i.e.,

$$\Phi(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \xi_i) - \mathbf{a}_i^\top \mathbf{x} = F(\mathbf{x})$$

provided that $\sum_i \mathbf{a}_i = 0$.

- Note that $\nabla \phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$
- Recall that SGD performance depends on variance at $\mathbf{x}^\star$

$$\mathbb{V}_{i_t} \left[\| \nabla f(\mathbf{x}^\star, \xi_{i_t}) \| \right] \leq \sigma^2$$

Intuition: Shifted SGD

Shifted gradient

$$\nabla\phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$$

- Goal: select $\mathbf{a}_i$ so that $\mathbb{V}_{i_t}\left[\nabla\phi(\mathbf{x}^\star, \xi_{i_t})\right]$ is small

Intuition: Shifted SGD

Shifted gradient

$$\nabla \phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$$

- Goal: select $\mathbf{a}_i$ so that $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right]$ is small
- Hypothetically, $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right] = 0$ requires

$$\mathbf{a}_i = \nabla f(\mathbf{x}^\star, \xi_i)$$

Intuition: Shifted SGD

Shifted gradient

$$\nabla \phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$$

- Goal: select $\mathbf{a}_i$ so that $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right]$ is small
- Hypothetically, $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right] = 0$ requires

$$\mathbf{a}_i = \nabla f(\mathbf{x}^\star, \xi_i)$$

- Not practical as $\mathbf{x}^\star$ unknown

Intuition: Shifted SGD

Shifted gradient

$$\nabla \phi(\mathbf{x}, \xi_i) = \nabla f(\mathbf{x}, \xi_i) - \mathbf{a}_i$$

- Goal: select $\mathbf{a}_i$ so that $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right]$ is small
- Hypothetically, $\mathbb{V}_{i_t}\left[\nabla \phi(\mathbf{x}^\star, \xi_{i_t})\right] = 0$ requires

$$\mathbf{a}_i = \nabla f(\mathbf{x}^\star, \xi_i)$$

- Not practical as $\mathbf{x}^\star$ unknown
- Clue: availability of estimates of $\nabla f(\mathbf{x}^\star, \xi_i)$ can help!

Unified Theory of Gradient Approximation

- A unified approach to approximating gradients [Gorbunov et al., 2019]

Unified Theory of Gradient Approximation

- A unified approach to approximating gradients [Gorbunov et al., 2019]
- Suppose the unbiased gradient approximation $\mathbf{g}_t$ satisfies:

$$\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$$
$$\mathbb{E}_t[\sigma_{t+1}^2] \leq (1-\rho)\sigma_t^2 + 2CD_F(\mathbf{x}_t, \mathbf{x}^\star)$$

where A, B, C, σ_t^2, and $\rho > 0$ are some constants (depend on L, μ, N) and $\mathbb{E}_t[\cdot]$ is expectation with respect to the random data index at iteration t

Unified Theory of Gradient Approximation

- A unified approach to approximating gradients [Gorbunov et al., 2019]
- Suppose the unbiased gradient approximation $\mathbf{g}_t$ satisfies:

$$\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$$
$$\mathbb{E}_t[\sigma_{t+1}^2] \leq (1-\rho)\sigma_t^2 + 2CD_F(\mathbf{x}_t, \mathbf{x}^\star)$$

where A, B, C, σ_t^2, and $\rho > 0$ are some constants (depend on L, μ, N) and $\mathbb{E}_t[\cdot]$ is expectation with respect to the random data index at iteration t

Lemma (Simplified version of [Gorbunov et al., 2019])

The following rate result holds:

$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^\star\|^2] \leq (1 - \tfrac{\rho}{2}\min\{\tfrac{2\mu}{A\rho+2BC}, 1\})^T B_0$$

where B_0 depends only on the initialization.

Variance Reduced SGD: Proof

Lemma (General result, [Gorbunov et al., 2019])

The following rate result holds:

$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^\star\|^2] \leq (1 - \tfrac{\rho}{2}\min\{\tfrac{2\mu}{A\rho + 2BC}, 1\})^T B_0$$

where B_0 depends only on the initialization.

Variance Reduced SGD: Proof

Lemma (General result, [Gorbunov et al., 2019])

The following rate result holds:

$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^\star\|^2] \leq (1 - \tfrac{\rho}{2}\min\{\tfrac{2\mu}{A\rho+2BC}, 1\})^T B_0$$

where B_0 depends only on the initialization.

Proof: Step 1: Expand the squares :

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 &= \|\mathbf{x}_t - \mathbf{x}^\star - \eta \mathbf{g}_t\|^2 \\ &= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \mathbf{g}_t\rangle + \eta^2 \|\mathbf{g}_t\|^2 \end{aligned}$$

Variance Reduced SGD: Proof

Lemma (General result, [Gorbunov et al., 2019])
The following rate result holds:

$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^\star\|^2] \leq (1 - \tfrac{\rho}{2}\min\{\tfrac{2\mu}{A\rho+2BC}, 1\})^T B_0$$

where B_0 depends only on the initialization.

Proof: Step 1: Expand the squares and use unbiased property $\mathbb{E}_t[\mathbf{g}_t] = \nabla F(\mathbf{x}_t)$:

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 &= \|\mathbf{x}_t - \mathbf{x}^\star - \eta\mathbf{g}_t\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle\mathbf{x}_t - \mathbf{x}^\star, \mathbf{g}_t\rangle + \eta^2\|\mathbf{g}_t\|^2 \\
\Rightarrow \mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] &= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle\mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2]
\end{aligned}$$

Variance Reduced SGD: Proof

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] = \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2]$$

Variance Reduced SGD: Proof

$$\begin{aligned}\mathbb{E}_t[\|\mathbf{x}_{t+1}-\mathbf{x}^\star\|^2] &= \|\mathbf{x}_t-\mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t-\mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2] \\ &\le (1-\eta\mu)\|\mathbf{x}_t-\mathbf{x}^\star\|^2 - 2\eta D_F(\mathbf{x}_t,\mathbf{x}^\star) + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2]\end{aligned}$$

Step 2: Use Strong Convexity

$D_F(\mathbf{x}_t,\mathbf{x}^\star) + D_F(\mathbf{x}^\star,\mathbf{x}_t) =$
$\langle \mathbf{x}_t-\mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle \ge \mu\|\mathbf{x}-\mathbf{y}\|^2$

Variance Reduced SGD: Proof

$$\begin{aligned}\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] &= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2] \\ &\leq (1 - \eta\mu)\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta D_F(\mathbf{x}_t, \mathbf{x}^\star) + \eta^2\mathbb{E}_t[\|\mathbf{g}_t\|^2]\end{aligned}$$

Step 3: Use assumed bounds $\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] \leq (1 - \eta\mu)\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + 2\eta\,(A\eta - 1)\,D_F(\mathbf{x}_t, \mathbf{x}^\star) + B\eta^2\sigma_t^2$$

Variance Reduced SGD: Proof

$$\begin{aligned}\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] &= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2] \\ &\leq (1-\eta\mu)\|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta D_F(\mathbf{x}_t, \mathbf{x}^\star) + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2]\end{aligned}$$

Step 3: Use assumed bounds $\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] \leq (1-\eta\mu)\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + 2\eta\,(A\eta - 1)\,D_F(\mathbf{x}_t, \mathbf{x}^\star) + B\eta^2\sigma_t^2$$

$$\frac{2B\eta^2}{\rho}\mathbb{E}_t[\sigma_{t+1}^2] \leq \frac{2B\eta^2}{\rho}(1-\rho)\sigma_t^2 + \frac{2B\eta^2}{\rho}2CD_F(\mathbf{x}_t, \mathbf{x}^\star)$$

Variance Reduced SGD: Proof

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] = \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2]$$
$$\leq (1 - \eta\mu)\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta D_F(\mathbf{x}_t, \mathbf{x}^\star) + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2]$$

Step 3: Use assumed bounds $\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] \leq (1 - \eta\mu)\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + 2\eta\,(A\eta - 1)\, D_F(\mathbf{x}_t, \mathbf{x}^\star) + B\eta^2\sigma_t^2$$

$$+ \quad \frac{2B\eta^2}{\rho}\mathbb{E}_t[\sigma_{t+1}^2] \leq \frac{2B\eta^2}{\rho}(1 - \rho)\sigma_t^2 + \frac{2B\eta^2}{\rho} 2C D_F(\mathbf{x}_t, \mathbf{x}^\star)$$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 + \frac{2B\eta^2}{\rho}\sigma_{t+1}^2]$$
$$\leq (1 - \mu\eta)\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \left(1 - \frac{\rho}{2}\right)\frac{2B\eta^2}{\rho}\sigma_t^2 + 2\eta^2\left(\frac{A\rho + 2BC}{\rho} - \frac{1}{\eta}\right) D_F(\mathbf{x}_t, \mathbf{x}^\star)$$

Variance Reduced SGD: Proof

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] = \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla F(\mathbf{x}_t)\rangle + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2]$$
$$\leq (1 - \eta\mu)\, \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta D_F(\mathbf{x}_t, \mathbf{x}^\star) + \eta^2 \mathbb{E}_t[\|\mathbf{g}_t\|^2]$$

Step 3: Use assumed bounds $\mathbb{E}_t[\|\mathbf{g}_t\|^2] \leq 2AD_F(\mathbf{x}_t, \mathbf{x}^\star) + B\sigma_t^2$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] \leq (1 - \eta\mu)\, \|\mathbf{x}_t - \mathbf{x}^\star\|^2 + 2\eta\,(A\eta - 1)\, D_F(\mathbf{x}_t, \mathbf{x}^\star) + B\eta^2\sigma_t^2$$

$$+ \qquad \frac{2B\eta^2}{\rho}\mathbb{E}_t[\sigma_{t+1}^2] \leq \frac{2B\eta^2}{\rho}(1-\rho)\sigma_t^2 + \frac{2B\eta^2}{\rho} 2C D_F(\mathbf{x}_t, \mathbf{x}^\star)$$

$$\eta = \frac{\rho}{A\rho + 2BC}$$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 + \frac{2B\eta^2}{\rho}\sigma_{t+1}^2]$$
$$\leq (1 - \mu\eta)\, \|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \left(1 - \frac{\rho}{2}\right) \frac{2B\eta^2}{\rho}\sigma_t^2 + 2\eta^2 \left(\frac{A\rho + 2BC}{\rho} - \frac{1}{\eta}\right) D_F(\mathbf{x}_t, \mathbf{x}^\star)$$

Variance Reduced SGD: Proof

Take full expectation

$$\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_{t+1}^2] \leq \left(1 - \min\{\tfrac{\mu\rho}{A\rho+2BC}, \tfrac{\rho}{2}\}\right) \mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_t^2]$$

Variance Reduced SGD: Proof

Take full expectation and apply recursively

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_{t+1}^2] &\leq \left(1 - \min\{\tfrac{\mu\rho}{A\rho+2BC}, \tfrac{\rho}{2}\}\right) \mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_t^2] \\
&\leq \left(1 - \min\{\tfrac{\mu\rho}{A\rho+2BC}, \tfrac{\rho}{2}\}\right)^t \mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_0^2]
\end{aligned}$$

Variance Reduced SGD: Proof

Take full expectation and apply recursively

$$\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_{t+1}^2] \leq \left(1 - \min\{\tfrac{\mu\rho}{A\rho+2BC}, \tfrac{\rho}{2}\}\right) \mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_t^2]$$

$$\leq \left(1 - \min\{\tfrac{\mu\rho}{A\rho+2BC}, \tfrac{\rho}{2}\}\right)^t \mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 + \tfrac{2B\eta^2}{\rho}\sigma_0^2]$$

Equivalently, to get $\mathbb{E}[\|\mathbf{x}_{T+1} - \mathbf{x}^\star\|^2] \leq \epsilon$ needs

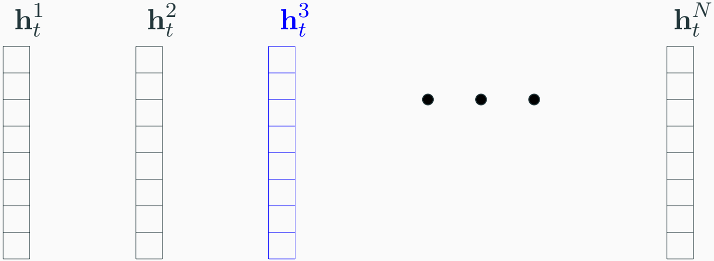
$$T = \frac{\log\left(\frac{1}{\epsilon}\right)}{-\log\left(1 - \min\{\frac{\mu\rho}{A\rho+2BC}, \frac{\rho}{2}\}\right)} \approx \frac{\log\left(\frac{1}{\epsilon}\right)}{\min\{\frac{\mu\rho}{A\rho+2BC}, \frac{\rho}{2}\}}$$

Outline

SAGA

Pick i_t at random from $\{1, 2, \ldots, N\}$

$$\mathbf{h}_{t+1}^{j} = \begin{cases} \mathbf{h}_t^j & j \neq i_t \\ \nabla f(\mathbf{x}_t, \xi_{i_t}) & j = i_t \end{cases}$$

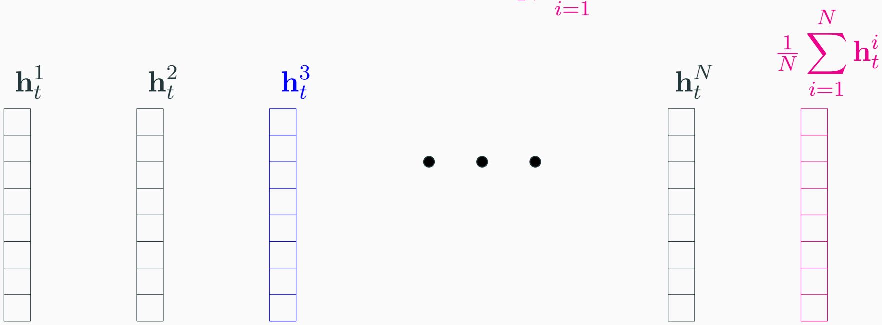


SAGA

Pick i_t at random from $\{1, 2, \ldots, N\}$

$$\mathbf{h}_{t+1}^{j} = \begin{cases} \mathbf{h}_t^j & j \neq i_t \\ \nabla f(\mathbf{x}_t, \xi_{i_t}) & j = i_t \end{cases}$$

$$\mathbf{g}_t = \mathbf{h}_{t+1}^{i_t} - \mathbf{h}_t^{i_t} + \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_t^i$$



SAGA Approximation is Unbiased

Unbiased? $$\mathbb{E}_{i_t}\left[\mathbf{g}_t\right] = \mathbb{E}_{i_t}\left[\mathbf{h}_{t+1}^{i_t}\right] - \mathbb{E}_{i_t}\left[\mathbf{h}_t^{i_t}\right] + \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i$$

SAGA Approximation is Unbiased

$$\mathbb{E}_{i_t}[\mathbf{g}_t] = \mathbb{E}_{i_t}\left[\mathbf{h}_{t+1}^{i_t}\right] - \mathbb{E}_{i_t}\left[\mathbf{h}_t^{i_t}\right] + \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i$$

$$= \nabla F(\mathbf{x}_t)$$

$$\mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})] = \nabla F(\mathbf{x}_t)$$

SAGA Approximation is Unbiased

$$
\begin{aligned}
\mathbb{E}_{i_t}[\mathbf{g}_t] &= \mathbb{E}_{i_t}\left[\mathbf{h}_{t+1}^{i_t}\right] - \mathbb{E}_{i_t}\left[\mathbf{h}_t^{i_t}\right] + \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i \\
&= \nabla F(\mathbf{x}_t) \quad - \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i + \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i
\end{aligned}
$$

$$
\boxed{\mathbb{E}_{i_t}\left[\mathbf{h}_t^{i_t}\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i}
$$

SAGA Approximation is Unbiased

$$\mathbb{E}_{i_t}\left[\mathbf{g}_t\right] = \mathbb{E}_{i_t}\left[\mathbf{h}_{t+1}^{i_t}\right] - \mathbb{E}_{i_t}\left[\mathbf{h}_t^{i_t}\right] + \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_t^i$$

$$= \nabla F(\mathbf{x}_t)$$

.

SAGA Approximation: Variance

Since $\nabla F(\mathbf{x}^\star) = 0$, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t} - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t}\right] \\
&= \mathsf{X} + \mathsf{Y} - \mathbb{E}_{i_t}[\mathsf{Y}]
\end{aligned}$$

SAGA Approximation: Variance

Since $\nabla F(\mathbf{x}^\star) = 0$, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t} - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t}\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}[\mathsf{Y}]
\end{aligned}$$

$$\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] \le 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\left\|\mathbf{h}_t^{i_t} - \nabla f(\mathbf{x}^\star, \xi_{i_t})\right\|^2\right]$$

$$\mathbb{E}[\|\mathsf{X} + \mathsf{Y} - \mathbb{E}[\mathsf{Y}]\|^2] \le 2\mathbb{E}[\|\mathsf{X}\|^2] + 2\mathbb{E}[\|\mathsf{Y}\|^2]$$

SAGA Approximation: Variance

Since $\nabla F(\mathbf{x}^\star) = 0$, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$
\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t} - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t}\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}\left[\mathsf{Y}\right] \\
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\left\|\mathbf{h}_t^{i_t} - \nabla f(\mathbf{x}^\star, \xi_{i_t})\right\|^2\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\left\|\mathbf{h}_t^i - \nabla f(\mathbf{x}^\star, \xi_i)\right\|^2
\end{aligned}
$$

SAGA Approximation: Variance

Since $\nabla F(\mathbf{x}^\star) = 0$, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t} - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t}\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}[\mathsf{Y}]
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\left\|\mathbf{h}_t^{i_t} - \nabla f(\mathbf{x}^\star, \xi_{i_t})\right\|^2\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\left\|\mathbf{h}_t^i - \nabla f(\mathbf{x}^\star, \xi_i)\right\|^2 \\
&\leq \qquad 4LD_F(\mathbf{x}_t, \mathbf{x}^\star) \qquad + \qquad 2\sigma_t^2
\end{aligned}$$

L-smoothness

$\frac{1}{2L}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 \leq$
$f(\mathbf{x}, \xi_i) - f(\mathbf{x}^\star, \xi_i) - \langle \nabla f(\mathbf{x}^\star, \xi_i), \mathbf{x} - \mathbf{x}^\star \rangle$

SAGA Approximation: Variance

Since $\nabla F(\mathbf{x}^\star) = 0$, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$
\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t} - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \mathbf{h}_t^{i_t}\right] \\
&= \quad \mathsf{X} \quad + \quad \mathsf{Y} \quad - \quad \mathbb{E}_{i_t}[\mathsf{Y}] \\
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\left\|\mathbf{h}_t^{i_t} - \nabla f(\mathbf{x}^\star, \xi_{i_t})\right\|^2\right] \\
&= \frac{2}{N}\sum_{i=1}^{N} \|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\left\|\mathbf{h}_t^i - \nabla f(\mathbf{x}^\star, \xi_i)\right\|^2 \\
&\leq \quad 4LD_F(\mathbf{x}_t, \mathbf{x}^\star) \quad + \quad 2\sigma_t^2
\end{aligned}
$$

$$A = 2L, \; B = 2$$

SAGA Approximation: σ_t^2

Recall that

$$\mathbf{h}_{t+1}^{j} = \begin{cases} \mathbf{h}_t^j & j \neq i_t \text{ with prob. } \left(1 - \frac{1}{N}\right) \\ \nabla f(\mathbf{x}_t, \xi_{i_t}) & j = i_t \text{ with prob. } \frac{1}{N} \end{cases}$$

SAGA Approximation: σ_t^2

Recall that

$$\mathbf{h}_{t+1}^j = \begin{cases} \mathbf{h}_t^j & j \neq i_t \text{ with prob. } \left(1 - \frac{1}{N}\right) \\ \nabla f(\mathbf{x}_t, \xi_{i_t}) & j = i_t \text{ with prob. } \frac{1}{N} \end{cases}$$

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}_{i_t}\left[\left\|\mathbf{h}_{t+1}^j - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2\right] \\
&= \frac{1}{N}\sum_{j=1}^{N}\left[\left(1 - \frac{1}{N}\right)\left\|\mathbf{h}_t^j - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2 + \frac{1}{N}\left\|\nabla f(\mathbf{x}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2\right] \\
&\leq \left(1 - \frac{1}{N}\right)\sigma_t^2 + \frac{2L}{N} D_F(\mathbf{x}_t, \mathbf{x}^\star)
\end{aligned}$$

L-smoothness

$\frac{1}{2L}\left\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\right\|^2 \leq$
$f(\mathbf{x}, \xi_i) - f(\mathbf{x}^\star, \xi_i) - \langle \nabla f(\mathbf{x}^\star, \xi_i), \mathbf{x} - \mathbf{x}^\star \rangle$

SAGA Approximation: σ_t^2

Recall that

$$\mathbf{h}_{t+1}^j = \begin{cases} \mathbf{h}_t^j & j \neq i_t \text{ with prob. } \left(1 - \frac{1}{N}\right) \\ \nabla f(\mathbf{x}_t, \xi_{i_t}) & j = i_t \text{ with prob. } \frac{1}{N} \end{cases}$$

$$\begin{aligned} \mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}_{i_t}\left[\left\|\mathbf{h}_{t+1}^j - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2\right] \\ &= \frac{1}{N}\sum_{j=1}^{N}\left[\left(1 - \frac{1}{N}\right)\left\|\mathbf{h}_t^j - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2 + \frac{1}{N}\left\|\nabla f(\mathbf{x}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\right\|^2\right] \\ &\leq \left(1 - \frac{1}{N}\right)\sigma_t^2 + \frac{2L}{N} D_F(\mathbf{x}_t, \mathbf{x}^\star) \end{aligned}$$

$$\rho = \frac{1}{N},\ C = \frac{2L}{N}$$

SAGA: Summary

Plugging in $A = 2L$, $B = 2$, $C = \frac{2L}{N}$, and $\rho = \frac{1}{N}$ (ignoring constants)

$$\mathcal{O}\left(\max\left\{N, \frac{L}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)\right)$$

SAGA: Summary

Plugging in $A = 2L$, $B = 2$, $C = \frac{2L}{N}$, and $\rho = \frac{1}{N}$ (ignoring constants)

$$\mathcal{O}\left(\max\left\{N, \frac{L}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)\right)$$

Algorithm	Oracle Complexity	Storage
GD	$N \quad \times \quad \frac{L}{\mu} \quad \times \quad \log\left(\frac{1}{\epsilon}\right)$	d
SGD	$1 \quad \times \quad \frac{L}{\mu} \quad \times \quad \frac{1}{\epsilon}$	d
SAGA	$\max\left\{N, \frac{L}{\mu}\right\} \quad \times \quad \log\left(\frac{1}{\epsilon}\right)$	dN

SAGA: Summary

Plugging in $A = 2L$, $B = 2$, $C = \frac{2L}{N}$, and $\rho = \frac{1}{N}$ (ignoring constants)

$$\mathcal{O}\left(\max\left\{N, \frac{L}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)\right)$$

Algorithm	Oracle Complexity	Storage
GD	$N \times \frac{L}{\mu} \times \log\left(\frac{1}{\epsilon}\right)$	d
SGD	$1 \times \frac{L}{\mu} \times \frac{1}{\epsilon}$	d
SAGA	$\max\left\{N, \frac{L}{\mu}\right\} \times \log\left(\frac{1}{\epsilon}\right)$	dN

Improves over SGD when N is *not too large* but high storage

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A “loopless” modification of SVRG [Johnson and Zhang, 2013]

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A "loopless" modification of SVRG [Johnson and Zhang, 2013]
- Pick i_t at random from $\{1, 2, \ldots, N\}$ and set

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) + \nabla F(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ and calculate } \nabla F(\mathbf{x}_t) \\ \mathbf{y}_t & \text{with prob. } 1 - \frac{1}{N} \end{cases}$$

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A "loopless" modification of SVRG [Johnson and Zhang, 2013]
- Pick i_t at random from $\{1, 2, \ldots, N\}$ and set

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) + \nabla F(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ and calculate } \nabla F(\mathbf{x}_t) \\ \mathbf{y}_t & \text{with prob. } 1 - \frac{1}{N} \end{cases}$$

- On average, 3 gradients evaluated per iteration

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A "loopless" modification of SVRG [Johnson and Zhang, 2013]
- Pick i_t at random from $\{1, 2, \ldots, N\}$ and set

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) + \nabla F(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ and calculate } \nabla F(\mathbf{x}_t) \\ \mathbf{y}_t & \text{with prob. } 1 - \frac{1}{N} \end{cases}$$

- On average, 3 gradients evaluated per iteration
- Unbiased gradient

$$\mathbb{E}_{i_t}[\mathbf{g}_t] = \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})] - \mathbb{E}_{i_t}[\nabla f(\mathbf{y}_t, \xi_{i_t})] + \nabla F(\mathbf{y}_t)$$

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A "loopless" modification of SVRG [Johnson and Zhang, 2013]
- Pick i_t at random from $\{1, 2, \ldots, N\}$ and set

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) + \nabla F(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ and calculate } \nabla F(\mathbf{x}_t) \\ \mathbf{y}_t & \text{with prob. } 1 - \frac{1}{N} \end{cases}$$

- On average, 3 gradients evaluated per iteration
- Unbiased gradient

$$\begin{aligned} \mathbb{E}_{i_t}[\mathbf{g}_t] &= \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})] - \mathbb{E}_{i_t}[\nabla f(\mathbf{y}_t, \xi_{i_t})] + \nabla F(\mathbf{y}_t) \\ &= \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{y}_t) + \nabla F(\mathbf{y}_t) \end{aligned}$$

Loopless SVRG

- Consider the loopless SVRG proposed in [Kovalev et al., 2019]
- A "loopless" modification of SVRG [Johnson and Zhang, 2013]
- Pick i_t at random from $\{1, 2, \dots, N\}$ and set

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) + \nabla F(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ and calculate } \nabla F(\mathbf{x}_t) \\ \mathbf{y}_t & \text{with prob. } 1 - \frac{1}{N} \end{cases}$$

- On average, 3 gradients evaluated per iteration
- Unbiased gradient

$$\begin{aligned} \mathbb{E}_{i_t}[\mathbf{g}_t] &= \mathbb{E}_{i_t}[\nabla f(\mathbf{x}_t, \xi_{i_t})] - \mathbb{E}_{i_t}[\nabla f(\mathbf{y}_t, \xi_{i_t})] + \nabla F(\mathbf{y}_t) \\ &= \nabla F(\mathbf{x}_t) \end{aligned}$$

Loopless SVRG: Approximation Properties

As in SAGA, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\begin{aligned}\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t})\right] \\ &= \mathsf{X} + \mathsf{Y} - \mathbb{E}_{i_t}[\mathsf{Y}]\end{aligned}$$

Loopless SVRG: Approximation Properties

As in SAGA, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t})\right]$$

$$= \mathsf{X} + \mathsf{Y} - \mathbb{E}_{i_t}[\mathsf{Y}]$$

$$\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] \leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{y}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right]$$

$$\mathbb{E}[\|\mathsf{X} + \mathsf{Y} - \mathbb{E}[\mathsf{Y}]\|^2] \leq 2\mathbb{E}[\|\mathsf{X}\|^2] + 2\mathbb{E}[\|\mathsf{Y}\|^2]$$

Loopless SVRG: Approximation Properties

As in SAGA, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t})\right]$$
$$= \mathsf{X} + \mathsf{Y} - \mathbb{E}_{i_t}[\mathsf{Y}]$$

$$\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] \leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{y}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right]$$
$$= \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{y}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2$$

As in SAGA, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$
\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t})\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}[\mathsf{Y}]
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{y}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{y}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 \\
&\leq \qquad 4LD_F(\mathbf{x}_t, \mathbf{x}^\star) \qquad + \qquad 2\sigma_t^2
\end{aligned}
$$

L-smoothness

$\frac{1}{2L}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 \leq$
$f(\mathbf{x}, \xi_i) - f(\mathbf{x}^\star, \xi_i) - \langle \nabla f(\mathbf{x}^\star, \xi_i), \mathbf{x} - \mathbf{x}^\star \rangle$

Loopless SVRG: Approximation Properties

As in SAGA, add and subtract $\nabla f(\mathbf{x}^\star, \xi_{i_t})$ to write

$$\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t}) + \nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t}) - \mathbb{E}_{i_t}\left[\nabla f(\mathbf{x}^\star, \xi_{i_t}) - \nabla f(\mathbf{y}_t, \xi_{i_t})\right] \\
&= \quad \mathsf{X} \quad + \quad \mathsf{Y} \quad - \quad \mathbb{E}_{i_t}[\mathsf{Y}]
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{x}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] + 2\mathbb{E}_{i_t}\left[\|\nabla f(\mathbf{y}_t, \xi_{i_t}) - \nabla f(\mathbf{x}^\star, \xi_{i_t})\|^2\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 + \frac{2}{N}\sum_{i=1}^{N}\|\nabla f(\mathbf{y}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 \\
&\leq \quad 4LD_F(\mathbf{x}_t, \mathbf{x}^\star) \quad + \quad 2\sigma_t^2
\end{aligned}$$

$$\boxed{A = 2L,\ B = 2}$$

Loopless SVRG: σ_t^2

Recall that

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{y}_t & \text{with prob. } \left(1 - \frac{1}{N}\right) \\ \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ (calculate } \nabla F(\mathbf{x}_t)\text{)} \end{cases}$$

Loopless SVRG: σ_t^2

Recall that

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{y}_t & \text{with prob. } \left(1 - \frac{1}{N}\right) \\ \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ (calculate } \nabla F(\mathbf{x}_t)\text{)} \end{cases}$$

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}[\|\nabla f(\mathbf{y}_{t+1}, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2] \\
&= \frac{1}{N}\sum_{j=1}^{N}\left[\left(1 - \frac{1}{N}\right)\|\nabla f(\mathbf{y}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2 + \frac{1}{N}\|\nabla f(\mathbf{x}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2\right] \\
&\leq \left(1 - \frac{1}{N}\right)\sigma_t^2 + \frac{2L}{N} D_F(\mathbf{x}_t, \mathbf{x}^\star)
\end{aligned}$$

***L*-smoothness**

$\frac{1}{2L}\|\nabla f(\mathbf{x}_t, \xi_i) - \nabla f(\mathbf{x}^\star, \xi_i)\|^2 \leq$
$f(\mathbf{x}, \xi_i) - f(\mathbf{x}^\star, \xi_i) - \langle \nabla f(\mathbf{x}^\star, \xi_i), \mathbf{x} - \mathbf{x}^\star\rangle$

Loopless SVRG: σ_t^2

Recall that

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{y}_t & \text{with prob. } \left(1 - \frac{1}{N}\right) \\ \mathbf{x}_t & \text{with prob. } \frac{1}{N} \text{ (calculate } \nabla F(\mathbf{x}_t)) \end{cases}$$

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}[\|\nabla f(\mathbf{y}_{t+1}, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2] \\
&= \frac{1}{N}\sum_{j=1}^{N} \left[\left(1 - \frac{1}{N}\right)\|\nabla f(\mathbf{y}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2 + \frac{1}{N}\|\nabla f(\mathbf{x}_t, \xi_j) - \nabla f(\mathbf{x}^\star, \xi_j)\|^2\right] \\
&\leq \left(1 - \frac{1}{N}\right)\sigma_t^2 + \frac{2L}{N} D_F(\mathbf{x}_t, \mathbf{x}^\star)
\end{aligned}$$

$$\rho = \frac{1}{N},\ C = \frac{2L}{N}$$

Loopless SVRG: Summary

Algorithm	Oracle Complexity	Storage
GD	$N \quad \times \quad \frac{L}{\mu} \quad \times \quad \log\left(\frac{1}{\epsilon}\right)$	d
SGD	$1 \quad \times \quad \frac{L}{\mu} \quad \times \quad \frac{1}{\epsilon}$	d
SAGA	$\max\left\{N, \frac{L}{\mu}\right\} \quad \times \quad \log\left(\frac{1}{\epsilon}\right)$	dN
L-SVRG	$\max\left\{N, \frac{L}{\mu}\right\} \quad \times \quad \log\left(\frac{1}{\epsilon}\right)$	d

Loopless SVRG: Summary

Algorithm	Oracle Complexity					Storage
GD	N	$\times$	$\frac{L}{\mu}$	$\times$	$\log\left(\frac{1}{\epsilon}\right)$	d
SGD	1	$\times$	$\frac{L}{\mu}$	$\times$	$\frac{1}{\epsilon}$	d
SAGA	$\max\left\{N, \frac{L}{\mu}\right\}$			$\times$	$\log\left(\frac{1}{\epsilon}\right)$	dN
L-SVRG	$\max\left\{N, \frac{L}{\mu}\right\}$			$\times$	$\log\left(\frac{1}{\epsilon}\right)$	d

Loopless SVRG has almost same number of gradient calculations as SAGA but requires same storage as SGD

Outline

1. Context
2. Background
3. Vanilla Stochastic Gradient Descent: Large N
4. Variance-Reduced SGD: Moderate N
   - SAGA and SVRG
   - State-of-the-art and Open Problems
5. High-dimensional problems: large d
6. Conclusion

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD
- Indeed, momentum SGD is prone to error accumulation [Konevcnỳ et al., 2015]

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD
- Indeed, momentum SGD is prone to error accumulation [Konevcnỳ et al., 2015]
- But can it work for variance-reduced algorithms?

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD
- Indeed, momentum SGD is prone to error accumulation [Kone$\mathbf{v}$cnỳ et al., 2015]
- But can it work for variance-reduced algorithms?
- Resolved partially in [Lin et al., 2015] and completely in [Allen-Zhu, 2017]

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD
- Indeed, momentum SGD is prone to error accumulation [Konevcnỳ et al., 2015]
- But can it work for variance-reduced algorithms?
- Resolved partially in [Lin et al., 2015] and completely in [Allen-Zhu, 2017]
- Several variants since then, active area of research

Accelerated Variants

- Accelerated GD proposed by Nesterov in 1983: uses a momentum term
- But acceleration has not been achieved for classical SGD
- Indeed, momentum SGD is prone to error accumulation [Konevcnỳ et al., 2015]
- But can it work for variance-reduced algorithms?
- Resolved partially in [Lin et al., 2015] and completely in [Allen-Zhu, 2017]
- Several variants since then, active area of research

Algorithm	Oracle Complexity	Storage
GD	$N \times \frac{L}{\mu} \times \log\left(\frac{1}{\epsilon}\right)$	d
Accelerated GD	$N \times \sqrt{\frac{L}{\mu}} \times \log\left(\frac{1}{\epsilon}\right)$	d
SGD	$1 \times \frac{L}{\mu} \times \frac{1}{\epsilon}$	d
L-SVRG	$\max\left\{N, \frac{L}{\mu}\right\} \times \log\left(\frac{1}{\epsilon}\right)$	d
Accelerated SVRG	$\left(N + \sqrt{\frac{NL}{\mu}}\right) \times \log\left(\frac{1}{\epsilon}\right)$	d

Accelerated Variants: Smooth + Convex

Algorithm	Oracle Complexity
GD	$N \times L \times \frac{1}{\epsilon}$
Accelerated GD	$N \times \sqrt{L} \times \frac{1}{\sqrt{\epsilon}}$
SGD	$1 \times L \times \frac{1}{\epsilon^2}$
SAGA	$(N + L) \times \frac{1}{\epsilon}$
SVRG+	$N \log\left(\frac{1}{\epsilon}\right) + \frac{L}{\epsilon}$
Accelerated SVRG	$N \log\left(\frac{1}{\epsilon}\right) + \sqrt{\frac{NL}{\epsilon}}$

Non-Convex Finite Sum: SPIDER

- Moderately large $N \leq \epsilon^{-2}$

Non-Convex Finite Sum: SPIDER

- Moderately large $N \leq \epsilon^{-2}$

Algorithm	Oracle Complexity		
GD	N	$\times$	ϵ^{-1}
SGD	1	$\times$	ϵ^{-2}
SVRG/SAGA	$N^{2/3}$	$\times$	ϵ^{-1}
SPIDER/SPIDERBoost	$N^{1/2}$	$\times$	ϵ^{-1}

Non-Convex Finite Sum: SPIDER

- Moderately large $N \leq \epsilon^{-2}$

Algorithm	Oracle Complexity
GD	N $\times$ ϵ^{-1}
SGD	1 $\times$ ϵ^{-2}
SVRG/SAGA	$N^{2/3} \times \epsilon^{-1}$
SPIDER/SPIDERBoost	$N^{1/2} \times \epsilon^{-1}$

- SPIDER [Fang et al., 2018] and SPIDERBoost [Wang et al., 2018] rate optimal in terms of N and ϵ
- Open problem: Adaptive step-size variant of SPIDER?

Non-Convex Online: STORM

- SAGA/SVRG not meant for large N

Non-Convex Online: STORM

- SAGA/SVRG not meant for large N
- SARAH [Nguyen et al., 2017], SPIDER proposed calculating "checkpoint" gradients every ϵ^{-1} samples: mega batches hard to tune

Non-Convex Online: STORM

- SAGA/SVRG not meant for large N
- SARAH [Nguyen et al., 2017], SPIDER proposed calculating "checkpoint" gradients every ϵ^{-1} samples: mega batches hard to tune
- STORM uses momentum + adaptive step-size to achieve optimal rate using single loop

Non-Convex Online: STORM

- SAGA/SVRG not meant for large N
- SARAH [Nguyen et al., 2017], SPIDER proposed calculating "checkpoint" gradients every ϵ^{-1} samples: mega batches hard to tune
- STORM uses momentum + adaptive step-size to achieve optimal rate using single loop

Algorithm	Oracle Complexity
SGD	ϵ^{-2}
SVRG+	$\epsilon^{-5/3}$
SPIDER/SPIDERBoost	$\epsilon^{-3/2}$
STORM	$\epsilon^{-3/2}$

Non-Convex Online: STORM

- SAGA/SVRG not meant for large N
- SARAH [Nguyen et al., 2017], SPIDER proposed calculating "checkpoint" gradients every ϵ^{-1} samples: mega batches hard to tune
- STORM uses momentum + adaptive step-size to achieve optimal rate using single loop

Algorithm	Oracle Complexity
SGD	ϵ^{-2}
SVRG+	$\epsilon^{-5/3}$
SPIDER/SPIDERBoost	$\epsilon^{-3/2}$
STORM	$\epsilon^{-3/2}$

- Open problem: can STORM to handle $\mathcal{X}$, regularizers, etc?

Distributed Setting

- Consider the problem

$$\min_{\mathbf{x}\in\mathcal{X}} \sum_{k\in\mathcal{V}} F_k(\mathbf{x})$$

- Data points $\{\xi_i^k\}_{i=1}^N$ available only at k-th node
- Central server aids in parallelizing: K nodes can offer K-fold speedup in wall-clock time
- State-of-the-art: Parallel Restarted SPIDER matches centralized $\mathcal{O}(\epsilon^{-3/2})$ for online non-convex
- Open problems: Distributed version of STORM? Accelerated variants?

Open Problem: Decentralized Setting

- Again consider the problem

$$\min_{\mathbf{x}\in\mathcal{X}} \sum_{k\in\mathcal{V}} F_k(\mathbf{x})$$

Open Problem: Decentralized Setting

- Again consider the problem

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{k \in \mathcal{V}} F_k(\mathbf{x})$$

- No central server, only communication between peers is allowed

Open Problem: Decentralized Setting

- Again consider the problem

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{k \in \mathcal{V}} F_k(\mathbf{x})$$

- No central server, only communication between peers is allowed
- All existing approaches are either suboptimal or cannot handle $\mathcal{X}$

Open Problem: Decentralized Setting

- Again consider the problem

$$\min_{\mathbf{x}\in\mathcal{X}} \sum_{k\in\mathcal{V}} F_k(\mathbf{x})$$

- No central server, only communication between peers is allowed
- All existing approaches are either suboptimal or cannot handle $\mathcal{X}$
- For non-convex, optimal $\mathcal{O}(\epsilon^{-3/2})$ achieved in [Sun et al., 2019]

Open Problem: Decentralized Setting

- Again consider the problem

$$\min_{\mathbf{x}\in\mathcal{X}} \sum_{k\in\mathcal{V}} F_k(\mathbf{x})$$

- No central server, only communication between peers is allowed
- All existing approaches are either suboptimal or cannot handle $\mathcal{X}$
- For non-convex, optimal $\mathcal{O}(\epsilon^{-3/2})$ achieved in [Sun et al., 2019]
- Open problem: can accelerated rates be obtained for convex decentralized case?

High-dimensional problems: large d

- When d is large, accessing $\nabla F(\mathbf{x})$ becomes difficult

- When d is large, accessing $\nabla F(\mathbf{x})$ becomes difficult
- E.g.: in matrix completion, $\nabla F(\mathbf{X}) \in \mathbb{R}^{m \times n}$ may be unwieldy ($d = mn$)

- When d is large, accessing $\nabla F(\mathbf{x})$ becomes difficult
- E.g.: in matrix completion, $\nabla F(\mathbf{X}) \in \mathbb{R}^{m \times n}$ may be unwieldy ($d = mn$)
- But a few coordinates of $\nabla F(\mathbf{X})$ may be available

- When d is large, accessing $\nabla F(\mathbf{x})$ becomes difficult
- E.g.: in matrix completion, $\nabla F(\mathbf{X}) \in \mathbb{R}^{m \times n}$ may be unwieldy ($d = mn$)
- But a few coordinates of $\nabla F(\mathbf{X})$ may be available
- Motivates coordinate descent and sketched gradient methods

Outline

Sketched Gradient Descent

- Consider recently proposed SEGA [Hanzely et al., 2018]

Sketched Gradient Descent

- Consider recently proposed SEGA [Hanzely et al., 2018]
- Assumes availability of $\mathbf{P}\nabla F(\mathbf{x})$ where $\mathbf{P} \in \mathbb{R}^{p \times d}$ where $p \ll d$

Sketched Gradient Descent

- Consider recently proposed SEGA [Hanzely et al., 2018]
- Assumes availability of $\mathbf{P}\nabla F(\mathbf{x})$ where $\mathbf{P} \in \mathbb{R}^{p \times d}$ where $p \ll d$
- We look at the special case of $p = 1$ and

$$\mathbf{P} = \mathbf{e}_{i_t}^\top = \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 & 0 \end{bmatrix}$$

where i_t is randomly selected from $\{1, \dots, N\}$

Sketched Gradient Descent

- Consider recently proposed SEGA [Hanzely et al., 2018]
- Assumes availability of $\mathbf{P}\nabla F(\mathbf{x})$ where $\mathbf{P} \in \mathbb{R}^{p\times d}$ where $p \ll d$
- We look at the special case of $p = 1$ and

$$\mathbf{P} = \mathbf{e}_{i_t}^\top = \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 & 0 \end{bmatrix}$$

  where i_t is randomly selected from $\{1, \dots, N\}$
- Sketched gradient is not an unbiased estimator!

SEGA: single coordinate update

- Unbiased gradient estimate must be maintained

SEGA: single coordinate update

- Unbiased gradient estimate must be maintained
- Starting with $\mathbf{h}_1 = 0$, we have

$$h_{t+1}^j = \begin{cases} [\nabla F(\mathbf{x}_t)]_j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

$$[\mathbf{g}_t]_j = \begin{cases} d[\nabla F(\mathbf{x}_t)]_j + (1-d)h_t^j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

SEGA: single coordinate update

- Unbiased gradient estimate must be maintained
- Starting with $\mathbf{h}_1 = 0$, we have

$$h_{t+1}^j = \begin{cases} [\nabla F(\mathbf{x}_t)]_j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

$$[\mathbf{g}_t]_j = \begin{cases} d[\nabla F(\mathbf{x}_t)]_j + (1-d)h_t^j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

- Maintain two $d \times 1$ vectors, but update only 1 coordinate at a time

SEGA: single coordinate update

- Unbiased gradient estimate must be maintained
- Starting with $\mathbf{h}_1 = 0$, we have

$$h_{t+1}^j = \begin{cases} [\nabla F(\mathbf{x}_t)]_j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

$$[\mathbf{g}_t]_j = \begin{cases} d[\nabla F(\mathbf{x}_t)]_j + (1-d)h_t^j & j = i_t \\ h_t^j & j \neq i_t \end{cases}$$

- Maintain two $d \times 1$ vectors, but update only 1 coordinate at a time
- Can we get GD-like performance with such sporadic updates?

SEGA: Unbiased Gradient Estimate

- Let us write in compact form:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$$
$$\mathbf{g}_t = \mathbf{h}_t + d\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$$

where $\odot$ denotes element-wise product

SEGA: Unbiased Gradient Estimate

- Let us write in compact form:

$$
\begin{aligned}
\mathbf{h}_{t+1} &= \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t) \\
\mathbf{g}_t &= \mathbf{h}_t + d\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)
\end{aligned}
$$

  where $\odot$ denotes element-wise product

- Note that $\mathbb{E}[\mathbf{e}_{i_t}] = \frac{1}{d}$

SEGA: Unbiased Gradient Estimate

- Let us write in compact form:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$$
$$\mathbf{g}_t = \mathbf{h}_t + d\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$$

  where $\odot$ denotes element-wise product
- Note that $\mathbb{E}[\mathbf{e}_{i_t}] = \frac{1}{d}$
- Unbiased gradient:

$$\mathbb{E}_{i_t}[\mathbf{g}_t] = \mathbf{h}_t + d\mathbb{E}_{i_t}[\mathbf{e}_{i_t}] \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t) = \nabla F(\mathbf{x}_t)$$

SEGA: Approximation Properties

Proceeding as earlier (since $\nabla F(\mathbf{x}^\star) = 0$)

$$
\begin{aligned}
\mathbf{g}_t &= d(\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)) - d\mathbf{e}_{i_t} \odot \mathbf{h}_t + \mathbb{E}_{i_t}\left[d\mathbf{e}_{i_t} \odot \mathbf{h}_t\right] \\
&= \quad \mathsf{X} \quad + \quad \mathsf{Y} \quad - \quad \mathbb{E}_{i_t}\left[\mathsf{Y}\right]
\end{aligned}
$$

SEGA: Approximation Properties

Proceeding as earlier (since $\nabla F(\mathbf{x}^\star) = 0$)

$$\begin{aligned}
\mathbf{g}_t &= d(\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)) - d\mathbf{e}_{i_t} \odot \mathbf{h}_t + \mathbb{E}_{i_t}\left[d\mathbf{e}_{i_t} \odot \mathbf{h}_t\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}\left[\mathsf{Y}\right]
\end{aligned}$$

$$\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] \le 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)\|^2\right] + 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \mathbf{h}_t\|^2\right]$$

$$\mathbb{E}[\|\mathsf{X} + \mathsf{Y} - \mathbb{E}[\mathsf{Y}]\|^2] \le 2\mathbb{E}[\|\mathsf{X}\|^2] + 2\mathbb{E}[\|\mathsf{Y}\|^2]$$

SEGA: Approximation Properties

Proceeding as earlier (since $\nabla F(\mathbf{x}^\star) = 0$)

$$
\begin{aligned}
\mathbf{g}_t &= d(\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)) - d\mathbf{e}_{i_t} \odot \mathbf{h}_t + \mathbb{E}_{i_t}\left[d\mathbf{e}_{i_t} \odot \mathbf{h}_t\right] \\
&= \qquad \mathsf{X} \qquad + \qquad \mathsf{Y} \qquad - \qquad \mathbb{E}_{i_t}\left[\mathsf{Y}\right] \\
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)\|^2\right] + 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \mathbf{h}_t\|^2\right] \\
&= \qquad 2d\left\|\nabla F(\mathbf{x}_t)\right\|^2 \qquad + \qquad 2d\left\|\mathbf{h}_t\right\|^2
\end{aligned}
$$

SEGA: Approximation Properties

Proceeding as earlier (since $\nabla F(\mathbf{x}^\star) = 0$)

$$
\begin{aligned}
\mathbf{g}_t &= d(\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)) - d\mathbf{e}_{i_t} \odot \mathbf{h}_t + \mathbb{E}_{i_t}\left[d\mathbf{e}_{i_t} \odot \mathbf{h}_t\right] \\
&= \quad \text{X} \quad + \quad \text{Y} \quad - \quad \mathbb{E}_{i_t}[\text{Y}] \\
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)\|^2\right] + 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \mathbf{h}_t\|^2\right] \\
&= \quad 2d\left\|\nabla F(\mathbf{x}_t)\right\|^2 \quad + \quad 2d\left\|\mathbf{h}_t\right\|^2 \\
&\leq \quad 4dLD_F(\mathbf{x}_t, \mathbf{x}^\star) \quad + \quad 2d\sigma_t^2
\end{aligned}
$$

L-smoothness

$\frac{1}{2L}\left\|\nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}^\star)\right\|^2 \leq$
$F(\mathbf{x}) - F(\mathbf{x}^\star) = D_F(\mathbf{x}_t, \mathbf{x}^\star)$

SEGA: Approximation Properties

Proceeding as earlier (since $\nabla F(\mathbf{x}^\star) = 0$)

$$
\begin{aligned}
\mathbf{g}_t &= d(\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)) - d\mathbf{e}_{i_t} \odot \mathbf{h}_t + \mathbb{E}_{i_t}\left[d\mathbf{e}_{i_t} \odot \mathbf{h}_t\right] \\
&= \quad \mathsf{X} \quad + \quad \mathsf{Y} \quad - \quad \mathbb{E}_{i_t}\left[\mathsf{Y}\right] \\
\mathbb{E}_{i_t}\left[\|\mathbf{g}_t\|^2\right] &\leq 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \nabla F(\mathbf{x}_t)\|^2\right] + 2d^2\mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot \mathbf{h}_t\|^2\right] \\
&= \quad 2d\left\|\nabla F(\mathbf{x}_t)\right\|^2 \quad + \quad 2d\left\|\mathbf{h}_t\right\|^2 \\
&\leq \quad 4dLD_F(\mathbf{x}_t, \mathbf{x}^\star) \quad + \quad 2d\sigma_t^2
\end{aligned}
$$

$$\boxed{A = 2dL, \ B = 2d}$$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] = \mathbb{E}_{i_t}\left[\left\|\mathbf{h}_{t+1}\right\|^2\right] = \mathbb{E}_{i_t}\left[\left\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\right\|^2\right]$$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$\begin{aligned} \mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \mathbb{E}_{i_t}\left[\|\mathbf{h}_{t+1}\|^2\right] = \mathbb{E}_{i_t}\left[\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\|^2\right] \\ &= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t + \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top \nabla F(\mathbf{x}_t)\right\|^2\right] \end{aligned}$$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] = \mathbb{E}_{i_t}\left[\|\mathbf{h}_{t+1}\|^2\right] &= \mathbb{E}_{i_t}\left[\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t + \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t\right\|^2\right] + \mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t))\|^2\right]
\end{aligned}$$

$$\mathbb{E}_{i_t}\left[(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top\right] =$$
$$\mathbb{E}_{i_t}\left[\mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top\right] - \mathbb{E}_{i_t}\left[\mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top\mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top\right] = 0$$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$
\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] = \mathbb{E}_{i_t}\left[\|\mathbf{h}_{t+1}\|^2\right] &= \mathbb{E}_{i_t}\left[\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t + \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t\right\|^2\right] + \mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t))\|^2\right] \\
&= \left(1 - \frac{1}{d}\right)\mathbb{E}_{i_t}\left[\|\mathbf{h}_t\|^2\right] + \frac{1}{d}\|\nabla F(\mathbf{x}_t)\|^2
\end{aligned}
$$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] &= \mathbb{E}_{i_t}\left[\|\mathbf{h}_{t+1}\|^2\right] = \mathbb{E}_{i_t}\left[\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t + \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t\right\|^2\right] + \mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t))\|^2\right] \\
&= \left(1 - \frac{1}{d}\right)\mathbb{E}_{i_t}\left[\|\mathbf{h}_t\|^2\right] + \frac{1}{d}\|\nabla F(\mathbf{x}_t)\|^2 \\
&\le \left(1 - \frac{1}{d}\right)\sigma_t^2 + \frac{2L}{d} D_F(\mathbf{x}_t, \mathbf{x}^\star)
\end{aligned}$$

L-smoothness

$\frac{1}{2L}\|\nabla F(\mathbf{x}_t)\|^2 \le D_F(\mathbf{x}_t, \mathbf{x}^\star)$

SEGA Approximation: σ_t^2

Recall that $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)$, so

$$
\begin{aligned}
\mathbb{E}_{i_t}\left[\sigma_{t+1}^2\right] = \mathbb{E}_{i_t}\left[\|\mathbf{h}_{t+1}\|^2\right] &= \mathbb{E}_{i_t}\left[\|\mathbf{h}_t + \mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t) - \mathbf{h}_t)\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t + \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&= \mathbb{E}_{i_t}\left[\left\|(\mathbf{I} - \mathbf{e}_{i_t}\mathbf{e}_{i_t}^\top)\mathbf{h}_t\right\|^2\right] + \mathbb{E}_{i_t}\left[\|\mathbf{e}_{i_t} \odot (\nabla F(\mathbf{x}_t))\|^2\right] \\
&= \left(1 - \frac{1}{d}\right)\mathbb{E}_{i_t}\left[\|\mathbf{h}_t\|^2\right] + \frac{1}{d}\|\nabla F(\mathbf{x}_t)\|^2 \\
&\leq \left(1 - \frac{1}{d}\right)\sigma_t^2 + \frac{2L}{d} D_F(\mathbf{x}_t, \mathbf{x}^\star)
\end{aligned}
$$

$$\rho = \frac{1}{d},\ C = \frac{2L}{d}$$

SEGA Summary

- GD uses d gradient entries per iteration

SEGA Summary

- GD uses d gradient entries per iteration
- SEGA uses 1 gradient entry per iteration

SEGA Summary

- GD uses d gradient entries per iteration
- SEGA uses 1 gradient entry per iteration
- Equivalently, GD incurs $d\times$ per iteration cost

SEGA Summary

- GD uses d gradient entries per iteration
- SEGA uses 1 gradient entry per iteration
- Equivalently, GD incurs $d\times$ per iteration cost
- Define oracle complexity $= d\times$ number of gradients required to achieve ϵ-accuracy

SEGA Summary

- GD uses d gradient entries per iteration
- SEGA uses 1 gradient entry per iteration
- Equivalently, GD incurs $d\times$ per iteration cost
- Define oracle complexity $= d\times$ number of gradients required to achieve ϵ-accuracy

Algorithm	Oracle Complexity	Per-iteration cost
GD	$d \times \frac{L}{\mu} \times \log\left(\frac{1}{\epsilon}\right)$	d
SEGA	$d \times \frac{L}{\mu} \times \log\left(\frac{1}{\epsilon}\right)$	1

SEGA is competitive with GD even while looking at one entry at a time!

Outline

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion
  - Observations $\mathbf{Z} \in \mathbb{R}^{N_r \times N_c}$

$$\min_{\mathbf{L},\mathbf{R}} \left\|\mathbf{Z} - \mathbf{L}\mathbf{R}^\top\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{L}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{R}\right\|_F^2$$

where $\mathbf{L} \in \mathbb{R}^{N_r \times r}$, and $\mathbf{R} \in \mathbb{R}^{N_c \times r}$

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion
  - Observations $\mathbf{Z} \in \mathbb{R}^{N_r \times N_c}$

$$\min_{\mathbf{L},\mathbf{R}} \left\|\mathbf{Z} - \mathbf{L}\mathbf{R}^\top\right\|_F^2 + \frac{\mu}{2}\|\mathbf{L}\|_F^2 + \frac{\mu}{2}\|\mathbf{R}\|_F^2$$

    where $\mathbf{L} \in \mathbb{R}^{N_r \times r}$, and $\mathbf{R} \in \mathbb{R}^{N_c \times r}$
  - Low-rank assumption $\Rightarrow r \ll N_c, N_r$

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion
  - Observations $\mathbf{Z} \in \mathbb{R}^{N_r \times N_c}$

$$\min_{\mathbf{L},\mathbf{R}} \left\|\mathbf{Z} - \mathbf{L}\mathbf{R}^\top\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{L}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{R}\right\|_F^2$$

    where $\mathbf{L} \in \mathbb{R}^{N_r \times r}$, and $\mathbf{R} \in \mathbb{R}^{N_c \times r}$
  - Low-rank assumption $\Rightarrow r \ll N_c, N_r$
  - Number of observations $N = N_r N_c$ is extremely large

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion
  - Observations $\mathbf{Z} \in \mathbb{R}^{N_r \times N_c}$

$$\min_{\mathbf{L},\mathbf{R}} \left\|\mathbf{Z} - \mathbf{L}\mathbf{R}^\top\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{L}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{R}\right\|_F^2$$

    where $\mathbf{L} \in \mathbb{R}^{N_r \times r}$, and $\mathbf{R} \in \mathbb{R}^{N_c \times r}$
  - Low-rank assumption $\Rightarrow r \ll N_c, N_r$
  - Number of observations $N = N_r N_c$ is extremely large
  - Number of variables $d = (N_c + N_r)r$ is also very large

Large N and d

- Large $N \Rightarrow$ cannot compute even one entry exactly
- Large $d \Rightarrow$ cannot compute full stochastic gradient
- Large-scale matrix completion
  - Observations $\mathbf{Z} \in \mathbb{R}^{N_r \times N_c}$

$$\min_{\mathbf{L},\mathbf{R}} \left\|\mathbf{Z} - \mathbf{L}\mathbf{R}^\top\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{L}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{R}\right\|_F^2$$

    where $\mathbf{L} \in \mathbb{R}^{N_r \times r}$, and $\mathbf{R} \in \mathbb{R}^{N_c \times r}$
  - Low-rank assumption $\Rightarrow r \ll N_c, N_r$
  - Number of observations $N = N_r N_c$ is extremely large
  - Number of variables $d = (N_c + N_r)r$ is also very large
- Cannot load the variables or observations into the RAM

Curse of Parallelization: Beyond Oracle Complexity

- SGD is inherently serial

Curse of Parallelization: Beyond Oracle Complexity

- SGD is inherently serial
- Consider system with m cores or m distributed servers

Curse of Parallelization: Beyond Oracle Complexity

- SGD is inherently serial
- Consider system with m cores or m distributed servers
- SGD achives ϵ accuracy in $\mathcal{O}(\frac{\sigma^2}{\epsilon})$ oracle calls

Curse of Parallelization: Beyond Oracle Complexity

- SGD is inherently serial
- Consider system with m cores or m distributed servers
- SGD achives ϵ accuracy in $\mathcal{O}(\frac{\sigma^2}{\epsilon})$ oracle calls
- To use multi-core systems, one must parallelize, e.g., using minibatch

$$\text{m-SGD} \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta}{m} \sum_{j \in \mathcal{I}_t} \nabla f(\mathbf{x}_t, \xi_j)$$

where $m = |\mathcal{I}_t|$ stochastic gradients are computed in parallel

Curse of Parallelization: Beyond Oracle Complexity

- SGD is inherently serial
- Consider system with m cores or m distributed servers
- SGD achives ϵ accuracy in $\mathcal{O}(\frac{\sigma^2}{\epsilon})$ oracle calls
- To use multi-core systems, one must parallelize, e.g., using minibatch

  $$\text{m-SGD} \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta}{m} \sum_{j \in \mathcal{I}_t} \nabla f(\mathbf{x}_t, \xi_j)$$

  where $m = |\mathcal{I}_t|$ stochastic gradients are computed in parallel
- What is the wall-clock time?

Curse of Parallelization: Wall Clock Time

- Let t_g = time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and t_r = time to read/write $\mathbf{x}_t$

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

SGD: Total wall-clock time $= t_g \times \sigma^2/\epsilon$

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

$$\begin{aligned} \text{SGD:} \quad & \text{Total wall-clock time} = t_g \times \sigma^2/\epsilon \\ \text{m-SGD:} \quad & \text{Total wall-clock time} = t_g \times \sigma^2/m\epsilon \end{aligned}$$

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

SGD: Total wall-clock time $= t_g \times \sigma^2/\epsilon$

m-SGD: Total wall-clock time $= t_g \times \sigma^2/m\epsilon$

- If $t_r \approx t_g$, writes are not concurrent

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

SGD: Total wall-clock time $= t_g \times \sigma^2/\epsilon$

m-SGD: Total wall-clock time $= t_g \times \sigma^2/m\epsilon$

- If $t_r \approx t_g$, writes are not concurrent

SGD: Total wall-clock time $= (t_g + 2t_r) \times \sigma^2/\epsilon \approx \mathcal{O}(\sigma^2/\epsilon)$

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

$$\begin{aligned} \text{SGD: } & \text{Total wall-clock time} = t_g \times \sigma^2/\epsilon \\ \text{m-SGD: } & \text{Total wall-clock time} = t_g \times \sigma^2/m\epsilon \end{aligned}$$

- If $t_r \approx t_g$, writes are not concurrent

$$\begin{aligned} \text{SGD: } & \text{Total wall-clock time} = (t_g + 2t_r) \times \sigma^2/\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \\ \text{m-SGD: } & \text{Total wall-clock time} = (t_g + (m+1)t_r) \times \sigma^2/m\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \end{aligned}$$

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

$$\begin{aligned} \text{SGD:} &\quad \text{Total wall-clock time} = t_g \times \sigma^2/\epsilon \\ \text{m-SGD:} &\quad \text{Total wall-clock time} = t_g \times \sigma^2/m\epsilon \end{aligned}$$

- If $t_r \approx t_g$, writes are not concurrent

$$\begin{aligned} \text{SGD:} &\quad \text{Total wall-clock time} = (t_g + 2t_r) \times \sigma^2/\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \\ \text{m-SGD:} &\quad \text{Total wall-clock time} = (t_g + (m+1)t_r) \times \sigma^2/m\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \end{aligned}$$

- Gains due to parallelization offset by the limited memory throughput

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

$$\begin{aligned} \text{SGD:} &\quad \text{Total wall-clock time} = t_g \times \sigma^2/\epsilon \\ \text{m-SGD:} &\quad \text{Total wall-clock time} = t_g \times \sigma^2/m\epsilon \end{aligned}$$

- If $t_r \approx t_g$, writes are not concurrent

$$\begin{aligned} \text{SGD:} &\quad \text{Total wall-clock time} = (t_g + 2t_r) \times \sigma^2/\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \\ \text{m-SGD:} &\quad \text{Total wall-clock time} = (t_g + (m+1)t_r) \times \sigma^2/m\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \end{aligned}$$

- Gains due to parallelization offset by the limited memory throughput
- Synchronization requirement cause idling of cores

Curse of Parallelization: Wall Clock Time

- Let $t_g =$ time to calculate $\nabla f(\mathbf{x}, \xi_j)$ and $t_r =$ time to read/write $\mathbf{x}_t$
- If $t_r \ll t_g$, then

$$\begin{aligned} \text{SGD:} \quad & \text{Total wall-clock time} = t_g \times \sigma^2/\epsilon \\ \text{m-SGD:} \quad & \text{Total wall-clock time} = t_g \times \sigma^2/m\epsilon \end{aligned}$$

- If $t_r \approx t_g$, writes are not concurrent

$$\begin{aligned} \text{SGD:} \quad & \text{Total wall-clock time} = (t_g + 2t_r) \times \sigma^2/\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \\ \text{m-SGD:} \quad & \text{Total wall-clock time} = (t_g + (m+1)t_r) \times \sigma^2/m\epsilon \approx \mathcal{O}(\sigma^2/\epsilon) \end{aligned}$$

- Gains due to parallelization offset by the limited memory throughput
- Synchronization requirement cause idling of cores
- Memory is locked while being written

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

where $\xi_i \subseteq \{1, \dots, n\}$ is an hyperedge

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

  where $\xi_i \subseteq \{1, \ldots, n\}$ is an hyperedge
- E.g., $\xi_i = \{1, 3, 4\}$ and $f(\mathbf{x}, \xi_i)$ depends on x_1, x_3, x_4

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

where $\xi_i \subseteq \{1, \ldots, n\}$ is an hyperedge

- E.g., $\xi_i = \{1, 3, 4\}$ and $f(\mathbf{x}, \xi_i)$ depends on x_1, x_3, x_4
- Sparsity: $|\xi_i| \ll d$

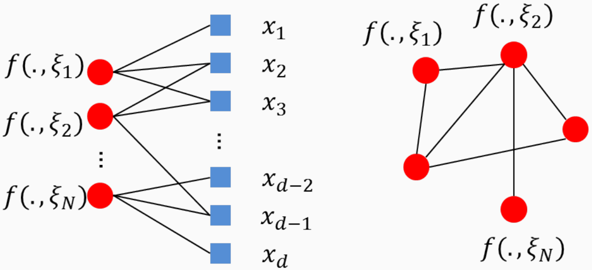


Figure 3: (a) Bipartite graph (b) conflict graph representation

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

  where $\xi_i \subseteq \{1, \ldots, n\}$ is an hyperedge
- E.g., $\xi_i = \{1, 3, 4\}$ and $f(\mathbf{x}, \xi_i)$ depends on x_1, x_3, x_4
- Sparsity: $|\xi_i| \ll d$
- Function $f : \mathbb{R}^n \times \mathcal{E} \to \mathbb{R}$ depends only on the subset of variables in ξ_i

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

  where $\xi_i \subseteq \{1, \ldots, n\}$ is an hyperedge
- E.g., $\xi_i = \{1, 3, 4\}$ and $f(\mathbf{x}, \xi_i)$ depends on x_1, x_3, x_4
- Sparsity: $|\xi_i| \ll d$
- Function $f : \mathbb{R}^n \times \mathcal{E} \to \mathbb{R}$ depends only on the subset of variables in ξ_i
- So only a few entries of $\nabla f(\mathbf{x}, \xi_i)$ are non-zero

Sparse Problem Structure

- Consider the problem [Recht et al., 2011]

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \xi_i)$$

  where $\xi_i \subseteq \{1, \ldots, n\}$ is an hyperedge
- E.g., $\xi_i = \{1, 3, 4\}$ and $f(\mathbf{x}, \xi_i)$ depends on x_1, x_3, x_4
- Sparsity: $|\xi_i| \ll d$
- Function $f : \mathbb{R}^n \times \mathcal{E} \to \mathbb{R}$ depends only on the subset of variables in ξ_i
- So only a few entries of $\nabla f(\mathbf{x}, \xi_i)$ are non-zero
- Indeed, $[\nabla f(\mathbf{x}, \xi_i)]_j = 0$ for all $j \notin \xi_i$

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:

without synchronizing with other cores

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
  - reads $\mathbf{x}$ from the memory;

without synchronizing with other cores

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
    - reads $\mathbf{x}$ from the memory;
    - evaluates $\nabla f(\mathbf{x}, \xi)$;

without synchronizing with other cores

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
    - reads $\mathbf{x}$ from the memory;
    - evaluates $\nabla f(\mathbf{x}, \xi)$;
    - updates $\mathbf{x}$; and

  without synchronizing with other cores

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
  - reads $\mathbf{x}$ from the memory;
  - evaluates $\nabla f(\mathbf{x}, \xi)$;
  - updates $\mathbf{x}$; and
  - writes $\mathbf{x}$ to memory one entry at a time

  without synchronizing with other cores

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
    - reads $\mathbf{x}$ from the memory;
    - evaluates $\nabla f(\mathbf{x}, \xi)$;
    - updates $\mathbf{x}$; and
    - writes $\mathbf{x}$ to memory one entry at a time

  without synchronizing with other cores
- This will lead to inconsistent reads and overwrites: recipe for disaster?

Hogwild!

- Go hog wild: read and write $\mathbf{x}$ without locking
- Each core does the following:
    - reads $\mathbf{x}$ from the memory;
    - evaluates $\nabla f(\mathbf{x}, \xi)$;
    - updates $\mathbf{x}$; and
    - writes $\mathbf{x}$ to memory one entry at a time

  without synchronizing with other cores
- This will lead to inconsistent reads and overwrites: recipe for disaster?
- Key idea: collisions rare if $\xi_i \cap \xi_j = \emptyset$ with high probability

Hogwild Algorithm

- Define $[\mathbf{x}]_\xi \in \mathbb{R}^{d\times 1}$ to contain only those entries that are in ξ, i.e.,

$$([\mathbf{x}]_\xi)_j = \begin{cases} 0 & j \notin \xi \\ x_j & j \in \xi \end{cases}$$

Hogwild Algorithm

- Define $[\mathbf{x}]_\xi \in \mathbb{R}^{d\times 1}$ to contain only those entries that are in ξ, i.e.,

$$([\mathbf{x}]_\xi)_j = \begin{cases} 0 & j \notin \xi \\ x_j & j \in \xi \end{cases}$$

- The full algorithm takes the form:

Hogwild Algorithm

- Define $[\mathbf{x}]_\xi \in \mathbb{R}^{d\times 1}$ to contain only those entries that are in ξ, i.e.,

$$([\mathbf{x}]_\xi)_j = \begin{cases} 0 & j \notin \xi \\ x_j & j \in \xi \end{cases}$$

- The full algorithm takes the form:

Algorithm 3 Hogwild! (at each core, in parallel)

```
1: repeat
2:     Sample an hyperedge ξ
3:     Let [x̂]_ξ = an inconsistent read of [x]_ξ
4:     Evaluate [u]_ξ = −η∇f([x̂]_ξ, ξ)
5:     for v ∈ ξ do:
6:         x_v ← x_v + u_v
7:     end for
8: until number of edges ≤ T
```

Perturbed SGD

- Cannot write Hogwild in classical SGD form

Lemma (Perturbed SGD: Strongly Convex + Smooth [Mania et al., 2017])
For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1-\eta\mu)\delta_t + \eta^2\mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t)\rangle]$$

Perturbed SGD

- Cannot write Hogwild in classical SGD form
- Instead consider perturbed SGD with some random variable ξ_t

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\hat{\mathbf{x}}_t, \xi_t)$$

where $\hat{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}_t$ with noise $\mathbf{n}_t$ independent of ξ_t

Lemma (Perturbed SGD: Strongly Convex + Smooth [Mania et al., 2017])

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu \mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta \mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t) \rangle]$$

Perturbed SGD

- Cannot write Hogwild in classical SGD form
- Instead consider perturbed SGD with some random variable ξ_t

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\hat{\mathbf{x}}_t, \xi_t)$$

  where $\hat{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}_t$ with noise $\mathbf{n}_t$ independent of ξ_t

- Defining $\delta_t := \mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^\star\|]$, then

Lemma (Perturbed SGD: Strongly Convex + Smooth [Mania et al., 2017])

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu \mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta \mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t) \rangle]$$

Perturbed SGD Proof

Lemma (Perturbed SGD: Strongly Convex + Smooth)

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t)\rangle]$$

Proof: Expand the optimality gap

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_t - \mathbf{x}^\star - \eta\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|$$

$$= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle + \eta^2 \|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2 + 2\eta\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle$$

Lemma (Perturbed SGD: Strongly Convex + Smooth)

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1-\eta\mu)\delta_t + \eta^2\mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t,\xi_t)\|^2] + 2\eta\mu\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta\mathbb{E}[\langle\hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t,\xi_t)\rangle]$$

Proof: Expand the optimality gap and add-subtract $\langle\hat{\mathbf{x}}_t, \nabla f(\hat{\mathbf{x}}_t,\xi_t)\rangle$

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_t - \mathbf{x}^\star - \eta\nabla f(\hat{\mathbf{x}}_t,\xi_t)\|$$

$$= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle\hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla f(\hat{\mathbf{x}}_t,\xi_t)\rangle + \eta^2\|\nabla f(\hat{\mathbf{x}}_t,\xi_t)\|^2 + 2\eta\langle\hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t,\xi_t)\rangle$$

Perturbed SGD Proof

Lemma (Perturbed SGD: Strongly Convex + Smooth)

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu \mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta \mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t)\rangle]$$

Proof: Expand the optimality gap and add-subtract $\langle \hat{\mathbf{x}}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle$

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_t - \mathbf{x}^\star - \eta \nabla f(\hat{\mathbf{x}}_t, \xi_t)\|$$

$$= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle + \eta^2 \|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2 + 2\eta\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle$$

$$\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] = \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla F(\hat{\mathbf{x}}_t)\rangle + \eta^2 \|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2$$
$$+ 2\eta \mathbb{E}\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle$$

Lemma (Perturbed SGD: Strongly Convex + Smooth)

For L-smooth, μ-convex functions f, perturbed SGD satisfies

$$\delta_{t+1} \leq (1-\eta\mu)\delta_t + \eta^2 \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2] + 2\eta\mu \mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] + 2\eta \mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\mathbf{x}_t, \xi_t) \rangle]$$

Proof: Expand the optimality gap and add-subtract $\langle \hat{\mathbf{x}}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t) \rangle$

$$\begin{aligned}
&\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_t - \mathbf{x}^\star - \eta \nabla f(\hat{\mathbf{x}}_t, \xi_t)\| \\
&= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta \langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla f(\hat{\mathbf{x}}_t, \xi_t) \rangle + \eta^2 \|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2 + 2\eta \langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t) \rangle \\
&\mathbb{E}_t[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2] = \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta \langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla F(\hat{\mathbf{x}}_t) \rangle + \eta^2 \|\nabla f(\hat{\mathbf{x}}_t, \xi_t)\|^2 \\
&\qquad\qquad + 2\eta \mathbb{E} \langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t) \rangle
\end{aligned}$$

Lemma follows from using μ-strong convexity and triangle inequality:
$\langle \hat{\mathbf{x}}_t - \mathbf{x}^\star, \nabla F(\hat{\mathbf{x}}_t) \rangle \geq \mu \|\hat{\mathbf{x}}_t - \mathbf{x}^\star\|^2 \geq \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \mu \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2$

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge
- Let $\bar{\mathbf{x}}_t$ be the contents before t-th read

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge
- Let $\bar{\mathbf{x}}_t$ be the contents before t-th read
- Also, recall that $[\mathbf{x}]_{\xi_t}$ is an inconsistent read, and define full vector $\hat{\mathbf{x}}_t$:

$$[\hat{\mathbf{x}}_t]_v = \begin{cases} [\hat{\mathbf{x}}_t]_v & v \in \xi_t \text{ – these are changed} \\ [\bar{\mathbf{x}}_t]_v & v \notin \xi_t \text{ – these remain same as before the read} \end{cases}$$

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge
- Let $\bar{\mathbf{x}}_t$ be the contents before t-th read
- Also, recall that $[\mathbf{x}]_{\xi_t}$ is an inconsistent read, and define full vector $\hat{\mathbf{x}}_t$:

$$[\hat{\mathbf{x}}_t]_v = \begin{cases} [\hat{\mathbf{x}}_t]_v & v \in \xi_t \text{ – these are changed} \\ [\bar{\mathbf{x}}_t]_v & v \notin \xi_t \text{ – these remain same as before the read} \end{cases}$$

- $\hat{\mathbf{x}}_t$ independent of ξ_t (can be relaxed)

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge
- Let $\bar{\mathbf{x}}_t$ be the contents before t-th read
- Also, recall that $[\mathbf{x}]_{\xi_t}$ is an inconsistent read, and define full vector $\hat{\mathbf{x}}_t$:

$$[\hat{\mathbf{x}}_t]_v = \begin{cases} [\hat{\mathbf{x}}_t]_v & v \in \xi_t \text{ – these are changed} \\ [\bar{\mathbf{x}}_t]_v & v \notin \xi_t \text{ – these remain same as before the read} \end{cases}$$

- $\hat{\mathbf{x}}_t$ independent of ξ_t (can be relaxed)
- Bounded gradients: $\|f(\hat{\mathbf{x}}, \xi)\| \leq M$ (can be relaxed)

Hogwild as Perturbed SGD

- Let ξ_t be the t-th sampled hyperedge
- Let $\bar{\mathbf{x}}_t$ be the contents before t-th read
- Also, recall that $[\mathbf{x}]_{\xi_t}$ is an inconsistent read, and define full vector $\hat{\mathbf{x}}_t$:

$$[\hat{\mathbf{x}}_t]_v = \begin{cases} [\hat{\mathbf{x}}_t]_v & v \in \xi_t \text{ – these are changed} \\ [\bar{\mathbf{x}}_t]_v & v \notin \xi_t \text{ – these remain same as before the read} \end{cases}$$

- $\hat{\mathbf{x}}_t$ independent of ξ_t (can be relaxed)
- Bounded gradients: $\|f(\hat{\mathbf{x}}, \xi)\| \leq M$ (can be relaxed)
- Key idea: after T updates are written to the memory:

$$\mathbf{x}_T = \mathbf{x}_1 - \eta \nabla f(\hat{\mathbf{x}}_1, \xi_1) - \eta \nabla f(\hat{\mathbf{x}}_2, \xi_2) - \ldots - \eta \nabla f(\hat{\mathbf{x}}_{T-1}, \xi_{T-1})$$

or

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\hat{\mathbf{x}}_t, \xi_t)$$

Hogwild Abstractions: τ and Δ

- Δ = average degree of conflict graph

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)
- τ can be proxy for number of cores: τ read-writes in parallel

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)
- τ can be proxy for number of cores: τ read-writes in parallel
- Consider, for instance, times i and j:

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)
- τ can be proxy for number of cores: τ read-writes in parallel
- Consider, for instance, times i and j:
  - if $i < j$ and $\xi_i \cap \xi_j = \emptyset$, $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ written before $\hat{\mathbf{x}}_j$ read: contribution of $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ included into $\hat{\mathbf{x}}_j$ and $\mathbf{x}_j$

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)
- τ can be proxy for number of cores: τ read-writes in parallel
- Consider, for instance, times i and j:
  - if $i < j$ and $\xi_i \cap \xi_j = \emptyset$, $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ written before $\hat{\mathbf{x}}_j$ read: contribution of $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ included into $\hat{\mathbf{x}}_j$ and $\mathbf{x}_j$
  - If $i > j$ and $\xi_i \cap \xi_j = \emptyset$, then neither $\hat{\mathbf{x}}_j$ nor $\mathbf{x}_j$ contain any contribution of $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$

Hogwild Abstractions: τ and Δ

- $\Delta =$ average degree of conflict graph
- Max. number of hyperedges that overlap with a given hyperedge $= \tau$
- $\tau = 0$ implies no overlap (classical SGD)
- τ can be proxy for number of cores: τ read-writes in parallel
- Consider, for instance, times i and j:
  - if $i < j$ and $\xi_i \cap \xi_j = \emptyset$, $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ written before $\hat{\mathbf{x}}_j$ read: contribution of $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$ included into $\hat{\mathbf{x}}_j$ and $\mathbf{x}_j$
  - If $i > j$ and $\xi_i \cap \xi_j = \emptyset$, then neither $\hat{\mathbf{x}}_j$ nor $\mathbf{x}_j$ contain any contribution of $\nabla f(\hat{\mathbf{x}}_i, \xi_i)$
- Edges $\xi_i \cap \xi_j = \emptyset$ if $|i - j| > \tau$

Hogwild: modeling inconsistent reads

- Let $\mathbf{S}_\iota^t$ be diagonal matrix with entries in $\{-1, 0, 1\}$
- Define conflicting edges: $\mathcal{I}_t := \{t-\tau, t-\tau+1, \ldots t-1, t+1, \ldots, t+\tau\}$
- Then, all possible update orders can be written as

$$\hat{\mathbf{x}}_t - \mathbf{x}_t = \eta \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)$$

- Models all patterns of possibly partial updates while ξ_t is being processed

Lemma

The following bounds hold:

$$\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] \le \eta^2 M \left(2\tau + 8\tau^2 \frac{\Delta}{d}\right)$$

$$\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, e_t)\rangle] \le 4\eta M^2 \tau \frac{\Delta}{d}$$

We use $\|\nabla f(\hat{\mathbf{x}}_t, \xi_\iota)\| \le M$

$$\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle] = \eta \sum_{\iota \in \mathcal{I}_t} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle]$$

Lemma

The following bounds hold:

$$\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] \leq \eta^2 M \left(2\tau + 8\tau^2 \frac{\Delta}{d}\right)$$

$$\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, e_t)\rangle] \leq 4\eta M^2 \tau \frac{\Delta}{d}$$

We use $\|\nabla f(\hat{\mathbf{x}}_t, \xi_\iota)\| \leq M$

$$\begin{aligned}\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle] &= \eta \sum_{\iota \in \mathcal{I}_t} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle] \\ &\leq \eta M^2 \sum_{\iota} \Pr[\xi_\iota \cap \xi_t \neq \emptyset]\end{aligned}$$

Lemma

The following bounds hold:

$$\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] \leq \eta^2 M \left(2\tau + 8\tau^2 \frac{\Delta}{d}\right)$$

$$\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, e_t)\rangle] \leq 4\eta M^2 \tau \frac{\Delta}{d}$$

We use $\|\nabla f(\hat{\mathbf{x}}_t, \xi_\iota)\| \leq M$ and $\Pr(\xi_\iota \cap \xi_t \neq \emptyset) = \frac{2\Delta}{d}$

$$\begin{aligned}\mathbb{E}[\langle \hat{\mathbf{x}}_t - \mathbf{x}_t, \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle] &= \eta \sum_{\iota \in \mathcal{I}_t} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \nabla f(\hat{\mathbf{x}}_t, \xi_t)\rangle] \\ &\leq \eta M^2 \sum_\iota \Pr\left[\xi_\iota \cap \xi_t \neq \emptyset\right] \\ &\leq 2\eta M^2 \tau \frac{2\Delta}{d}\end{aligned}$$

Hogwild Analysis

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] = \eta^2 \mathbb{E}[\| \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota) \|^2]$$

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$\begin{aligned}\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] &= \eta^2 \mathbb{E}[\| \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2] \\ &= \eta^2 \sum_{\iota \in \mathcal{I}_t} \mathbb{E} \left\| \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota) \right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \mathbf{S}_\kappa^t \nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa) \rangle]\end{aligned}$$

Hogwild Analysis

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] &= \eta^2 \mathbb{E}[\| \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2] \\
&= \eta^2 \sum_{\iota \in \mathcal{I}_t} \mathbb{E} \left\| \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota) \right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \mathbf{S}_\kappa^t \nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa) \rangle] \\
&\leq \eta^2 \sum_{\iota} \mathbb{E} \left\| \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota) \right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\| \, \|\nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\| \, \mathbb{1}_{\xi_\iota \cap \xi_\kappa \neq \emptyset}]
\end{aligned}$$

Hogwild Analysis

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$
\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] &= \eta^2 \mathbb{E}[\| \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2] \\
&= \eta^2 \sum_{\iota \in \mathcal{I}_t} \mathbb{E} \left\|\mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \mathbf{S}_\kappa^t \nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\rangle] \\
&\leq \eta^2 \sum_{\iota} \mathbb{E} \left\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\| \, \|\nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\| \, \mathbb{1}_{\xi_\iota \cap \xi_\kappa \neq \emptyset}] \\
&\leq \eta^2 M^2 (2\tau + 4\tau^2 \mathrm{Pr}\left[\xi_\iota \cap \xi_\kappa \neq \emptyset\right]) = 2\eta^2 M^2 \tau (1 + 2\tau(2\Delta/d))
\end{aligned}
$$

Hogwild Analysis

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] &= \eta^2 \mathbb{E}[\|\sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2] \\
&= \eta^2 \sum_{\iota \in \mathcal{I}_t} \mathbb{E}\left\|\mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \mathbf{S}_\kappa^t \nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\rangle] \\
&\leq \eta^2 \sum_{\iota} \mathbb{E}\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\| \, \|\nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\| \, \mathbb{1}_{\xi_\iota \cap \xi_\kappa \neq \emptyset}] \\
&\leq \eta^2 M^2 (2\tau + 4\tau^2 \mathrm{Pr}\left[\xi_\iota \cap \xi_\kappa \neq \emptyset\right]) = 2\eta^2 M^2 \tau(1 + 2\tau(2\Delta/d))
\end{aligned}$$

Substituting all bounds,

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 M^2 C_1$$

where $C_1 = 1 + 8\tau\Delta/d + 4\eta\mu\tau + 16\eta\mu\tau^2\Delta/d$.

Hogwild Analysis

Since $\|\mathbf{S}\mathbf{u}\|_2 \leq \|\mathbf{u}\|$, it holds that

$$\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2] &= \eta^2 \mathbb{E}[\| \sum_{\iota \in \mathcal{I}_t} \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2] \\
&= \eta^2 \sum_{\iota \in \mathcal{I}_t} \mathbb{E}\left\|\mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\right\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\langle \mathbf{S}_\iota^t \nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota), \mathbf{S}_\kappa^t \nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\rangle] \\
&\leq \eta^2 \sum_{\iota} \mathbb{E}\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\|^2 + \eta^2 \sum_{\iota \neq \kappa} \mathbb{E}[\|\nabla f(\hat{\mathbf{x}}_\iota, \xi_\iota)\| \, \|\nabla f(\hat{\mathbf{x}}_\kappa, \xi_\kappa)\| \, \mathbb{1}_{\xi_\iota \cap \xi_\kappa \neq \emptyset}] \\
&\leq \eta^2 M^2 (2\tau + 4\tau^2 \Pr[\xi_\iota \cap \xi_\kappa \neq \emptyset]) = 2\eta^2 M^2 \tau (1 + 2\tau(2\Delta/d))
\end{aligned}$$

Substituting all bounds,

$$\delta_{t+1} \leq (1 - \eta\mu)\delta_t + \eta^2 M^2 C_1$$

where $C_1 = 1 + 8\tau\Delta/d + 4\eta\mu\tau + 16\eta\mu\tau^2\Delta/d$.
Yields $\mathcal{O}(\frac{L}{\mu\epsilon})$ oracle complexity (same as SGD) provided τ is not too large

State-of-the-art for high-dimensional

- Asynchronous SVRG [Mania et al., 2017] is the variance-reduced version of Hogwild!

State-of-the-art for high-dimensional

- Asynchronous SVRG [Mania et al., 2017] is the variance-reduced version of Hogwild!
- Extensions to non-convex settings with more realistic assumptions [Cannelli et al., 2019]

State-of-the-art for high-dimensional

- Asynchronous SVRG [Mania et al., 2017] is the variance-reduced version of Hogwild!
- Extensions to non-convex settings with more realistic assumptions [Cannelli et al., 2019]
- Very large delays [Zhou et al., 2018]

State-of-the-art for high-dimensional

- Asynchronous SVRG [Mania et al., 2017] is the variance-reduced version of Hogwild!
- Extensions to non-convex settings with more realistic assumptions [Cannelli et al., 2019]
- Very large delays [Zhou et al., 2018]
- Proximal variants [Zhu et al., 2018]

State-of-the-art for high-dimensional

- Asynchronous SVRG [Mania et al., 2017] is the variance-reduced version of Hogwild!
- Extensions to non-convex settings with more realistic assumptions [Cannelli et al., 2019]
- Very large delays [Zhou et al., 2018]
- Proximal variants [Zhu et al., 2018]
- Decentralized variants? Skewed sparsity profile?

Conclusion

Summary

- Oracle complexity results for different SGD variants
- Intuition regarding variance reduction and coordinate descent
- When to apply which version?
- Unified and simplified proofs (extend to non-strongly convex settings also)
- State-of-the-art and open problems

Allen-Zhu, Z. (2017).
Katyusha: The first direct acceleration of stochastic gradient methods.
The Journal of Machine Learning Research, 18(1):8194–8244.

Beck, A. (2017).
First-order methods in optimization, volume 25.
SIAM.

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

Bubeck, S. (2019).
Sebastien Bubeck's blog: I'm a bandit.
https://blogs.princeton.edu/imabandit/2018/11/21/a-short-proof-for-nesterovs-momentum/.
Accessed: 14 July 2019.

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

Cannelli, L., Facchinei, F., Kungurtsev, V., and Scutari, G. (2019).
Asynchronous parallel algorithms for nonconvex optimization.
Mathematical Programming, pages 1–34.

Chen, Y. (2019).
Notes on large scale optimization for data science.
http://www.princeton.edu/~yc5/ele522_optimization/lectures.html.
Accessed: 23 June 2019.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. (2018).
Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator.
In *Advances in Neural Information Processing Systems*, pages 689–699.

Gorbunov, E., Hanzely, F., and Richtárik, P. (2019).
A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent.
arXiv preprint arXiv:1905.11261.

Hanzely, F., Mishchenko, K., and Richtárik, P. (2018).
SEGA: Variance reduction via gradient sketching.
In *Advances in Neural Information Processing Systems*, pages 2082–2093.

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

Konevcnỳ, J., Liu, J., Richtárik, P., and Takávc, M. (2015).
Mini-batch semi-stochastic gradient descent in the proximal setting.
IEEE Journal of Selected Topics in Signal Processing, 10(2):242–255.

Kovalev, D., Horváth, S., and Richtárik, P. (2019).
Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop.
arXiv preprint arXiv:1901.08689.

Krizhevsky, A. (2009).
Learning multiple layers of features from tiny images.
Master's thesis, University of Toronto.

Lin, H., Mairal, J., and Harchaoui, Z. (2015).
A universal catalyst for first-order optimization.
In *Advances in neural information processing systems*, pages 3384–3392.

Mania, H., Pan, X., Papailiopoulos, D., Recht, B., Ramchandran, K., and Jordan, M. I. (2017).
Perturbed iterate analysis for asynchronous stochastic optimization.
SIAM Journal on Optimization, 27(4):2202–2229.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takávc, M. (2017).
Sarah: A novel method for machine learning problems using stochastic recursive gradient.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621.

Recht, B., Re, C., Wright, S., and Niu, F. (2011).
Hogwild: A lock-free approach to parallelizing stochastic gradient descent.
In *Advances in neural information processing systems*, pages 693–701.

Saunders, M. (2019).
Notes on first-order methods for minimizing smooth functions.
https://web.stanford.edu/class/msande318/notes/notes-first-order-smooth.pdf.
Accessed: 23 June 2019.

Sun, H., Lu, S., and Hong, M. (2019).
Improving the sample and communication complexity for decentralized non-convex optimization: A joint gradient estimation and tracking approach.
arXiv preprint arXiv:1910.05857.

Vandenberghe, L. (2019).
Optimization methods for large-scale systems.
http://www.seas.ucla.edu/~vandenbe/ee236c.html.
Accessed: 14 Aug. 2019.

Wang, F., Dai, J., Li, M., Chan, W.-c., Kwok, C. C.-h., Leung, S.-l., Wu, C., Li, W., Yu, W.-c., Tsang, K.-h., et al. (2016).
Risk assessment model for invasive breast cancer in Hong Kong women.
Medicine, 95(32).

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. (2018).
Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization.
arXiv preprint arXiv:1810.10690.

Zhou, Z., Mertikopoulos, P., Bambos, N., Glynn, P., Ye, Y., Li, L.-J., and Fei-Fei, L. (2018).
Distributed asynchronous optimization with unbounded delays: How slow can you go?
In *International Conference on Machine Learning*, pages 5970–5979.

Zhu, R., Niu, D., and Li, Z. (2018).
Asynchronous stochastic proximal methods for nonconvex nonsmooth optimization.
arXiv preprint arXiv:1802.08880.