Decentralized Stochastic Non-convex Optimization

Usman A. Khan
Electrical and Computer Engineering, Tufts University

August 28, 2020

Graduate Students

Reza D.
(2011-15)

C. Xi
(2012-17)

F. Saadatniaki
(2014-19)

R. Xin
(2016-19)

M. I. Qureshi
(2018-)

A. Swar
(2020-)

Research Overview

Learning from Data

- Data is everywhere and holds a significant potential
  - Credit card fraud, Medical diagnosis, Political campaigns
  - Image classification, Deep Learning
  - . . .
- Key challenges: Centralized solutions are no longer practical
  - Large, private, and proprietary datasets
  - Computation and communication have practical constraints
- **Can decentralized algorithms outperform their centralized counterparts?** *How to quantify such a comparison?*
- Let us consider a classical example ...

Self-Driving Cars: Recognizing Traffic Signs

- Identify STOP vs. YIELD sign

Figure 1: Binary classification: (Left) Training phase (Right) Testing phase

- Input data: Image $\boldsymbol{\theta}$ and its label $\mathbf{y}$
- Model: $g(\mathbf{x}; \boldsymbol{\theta})$ takes the image point and predicts the label
- Loss: $\ell(g(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y})$, prediction error as a function of the parameter $\mathbf{x}$

- Problem: Find the parameter $\mathbf{x}$ that minimizes the loss

$$\min_{\mathbf{x}} f(\mathbf{x}); \qquad f(\mathbf{x}) := \ell(g(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y})$$

- Our focus: First-order methods for different function classes

Some Preliminaries

Basic Definitions

- $f : \mathbb{R}^p \to \mathbb{R}$ is L-smooth, non-convex, and $f(\mathbf{x}) \geq f^* \geq -\infty, \forall \mathbf{x}$
  - Bounded above by a quadratic
  - $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$
- f satisfies Polyak-Łojasiewics (PL) condition [Polyak '87, Karimi et al. '16]
  - Every stationary point is a global minimum (not necessarily convex)
  - Strong convexity is a special case
  - $2\mu\,(f(\mathbf{x}) - f^*) \leq \|\nabla f(\mathbf{x})\|^2.$
- f is μ-strongly convex
  - Convex and bounded below by a quadratic
  - $f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \leq f(\mathbf{y})$
- $\kappa := \frac{L}{\mu}$ is called the condition number, $L \geq \mu > 0$

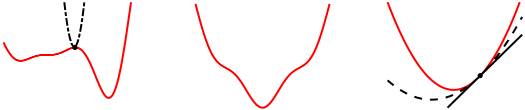

Figure 2: Non-convex: $\sin(ax)(x+bx^2)$. PL condition: $x^2+3\sin^2(x)$. Quadratic

First-order methods (Gradient Descent)

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Search for a point $\mathbf{x}^*$ where the gradient is zero, i.e., $\nabla f(\mathbf{x}^*) = \mathbf{0}$
- Intuition: Take a step in the direction opposite to the gradient
  - At $\star$, $\nabla f(\mathbf{x}^*) = 0$

Figure 3: Minimizing strongly convex functions: $\mathbb{R} \to \mathbb{R}$ and $\mathbb{R}^2 \to \mathbb{R}$

- A well-known *first-order* algorithm: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \cdot \nabla f(\mathbf{x}_k)$
- With stochastic gradients: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \cdot \mathbf{g}(\mathbf{x}_k)$

Performance Metrics and Other Criteria

- Stochastic Gradient Descent (SGD): $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \cdot \mathbf{g}(\mathbf{x}_k)$
  - $\mathbf{g}(\mathbf{x}_k)$ is an unbiased estimate of $\nabla f(\mathbf{x}_k)$ with bounded variance

- Optimality gap (PL and sc): $\mathbb{E}\left[\, f(\mathbf{x}_k) - f^* \,\right]$
- Mean-squared residual (sc): $\mathbb{E}\left[\, \|\mathbf{x}_k - \mathbf{x}^*\|^2 \,\right]$
- Mean-squared stationary gap (non-convex): $\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[\, \|\nabla f(\mathbf{x}_k)\|^2 \,\right]$

- Almost sure ($\delta > 0$): $\mathbb{P}\left[\lim_{k\to\infty} k^{1-\delta}(f(\mathbf{x}_k) - f^*) = 0\right] = 1$

- Decentralized problems: node i's iterate is $\mathbf{x}_k^i$
  - Replace $\mathbf{x}_k$ above by $\mathbf{x}_k^i$ or $\bar{\mathbf{x}}_k := \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_k^i$
  - Agreement error: $\mathbb{E}\left[\, \|\mathbf{x}_k^i - \mathbf{x}_k^j\|^2 \,\right]$ or $\mathbb{E}\left[\, \|\mathbf{x}_k^i - \bar{\mathbf{x}}_k\|^2 \,\right]$

  - Network-independent behavior
  - Speedup compared to centralized counterparts

Decentralized First-Order Methods

Decentralized Optimization

- Problem setup:

$$\text{P1} : \min_{\mathbf{x}} F(\mathbf{x}), \qquad F(\mathbf{x}) := \sum_{i=1}^{n} f_i(\mathbf{x}), \quad f_i : \mathbb{R}^p \to \mathbb{R}$$

- Search for a stationary point $\mathbf{x}^*$ such that $\nabla F(\mathbf{x}^*) = 0$

First-order methods under the following setup

- Measurement model:
  - *Online*: Each node i makes a noisy measurement $\to$ an imperfect local gradient $\nabla f_i(\mathbf{x})$ at any $\mathbf{x}$
    (Reduces to full gradient model when the variance is zero)
  - *Batch*: Each node i has access to a local dataset with m_i data points and their corresponding labels, i.e., $\nabla f_i(\mathbf{x}) = \sum_{j=1}^{m_i} \nabla f_{i,j}(\mathbf{x})$
- Each local cost f_i is L-smooth and $F^* := \inf_{\mathbf{x}} F(\mathbf{x}) \geq -\infty$
- The nodes communicate over a strongly connected graph

Local Gradient Descent

- Implement $\mathbf{x}_{k+1}^{i} = \mathbf{x}_{k}^{i} - \alpha \cdot \nabla f_i(\mathbf{x}_{k}^{i})$ at each node i
- Each node converges to a local solution

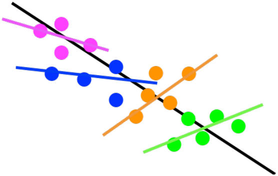

Figure 4: Linear regression: Locally optimal solutions

- Requirements for a decentralized algorithm
  - Agreement: Each node agrees to the same solution
  - Optimality: The agreed upon solution is the optimal
  - Local GD does not meet either

Decentralized Gradient Descent

- Mix and Descend: At each node i

$$\mathbf{x}^i_{k+1} = \sum_{r=1}^{n} w_{ir} \cdot \mathbf{x}^r_k - \alpha_k \cdot \nabla f_i(\mathbf{x}^i_k)$$

- The weight matrix $W = \{w_{ij}\}$ is primitive and doubly stochastic
  - $\lambda \in [0, 1)$ is the second largest singular value of W
  - $(1 - \lambda) \in (0, 1]$ is the spectral gap of the network

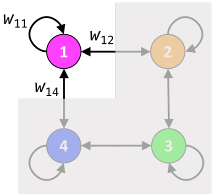

Figure 5: DGD over undirected graphs

Decentralized Gradient Descent

- DGD: At each node i

$$\mathbf{x}_{k+1}^{i} = \sum_{r=1}^{n} w_{ir} \cdot \mathbf{x}_{k}^{r} - \alpha_k \cdot \nabla f_i(\mathbf{x}_k^i)$$

- For strongly convex problems
  - Decaying step-size: convergence is sublinear $O(\frac{1}{k})$ [Nedić et al. '09]
  - Constant step-size: linear but inexact [Yuan et al. '13]

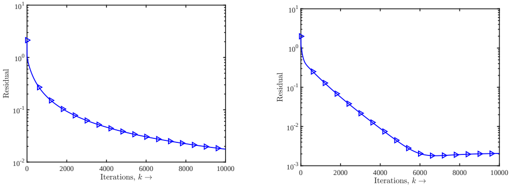

Figure 6: DGD with a decaying step-size (left) and constant step-size (right)

- Let us consider DGD with stochastic gradients

Decentralized Stochastic Gradient Descent

- Online setup–each node i makes an imperfect measurement leading to a stochastic gradient $\mathbf{g}_i$:
  - $\mathbf{g}_i(\mathbf{x}_k^i)$ is an unbiased estimate of the true gradient $\nabla f_i(\mathbf{x}_k^i)$, and
  - $\mathbf{g}_i(\mathbf{x}_k^i)$ has a bounded variance ν^2
- DSGD at node i: [Ram et al. '10], [Chen et al. '12]

$$\mathbf{x}_{k+1}^i = \sum_{r=1}^{n} w_{ir} \cdot \mathbf{x}_k^r - \alpha_k \cdot \mathbf{g}_i(\mathbf{x}_k^i)$$

- What do we know about the performance of DSGD?

Performance of DSGD (constant step-size)

- Smooth strongly convex problems:
- Mean-squared residual decays **linearly** to an error ball [Yuan et al. '19]

$$\limsup_{k\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\|\mathbf{x}_k^i - \mathbf{x}^*\|_2^2] = \mathcal{O}\Big(\frac{\alpha}{n\mu}\nu^2 + \frac{\alpha^2\kappa^2}{1-\lambda}\nu^2 + \frac{\alpha^2\kappa^2}{(1-\lambda)^2}\eta\Big),$$

where $\eta := \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\mathbf{x}^*) - \sum_i \nabla f_i(\mathbf{x}^*)\right\|_2^2$

- Smooth non-convex problems:
- Mean-squared stationary gap follows [Lian et al. '17]

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\|\nabla F(\overline{\mathbf{x}}_k)\|^2\right] \leq \mathcal{O}\left(\frac{F(\overline{\mathbf{x}}_0) - F^*}{\alpha K} + \frac{\alpha L}{n}\nu^2 + \frac{\alpha^2 L^2}{1-\lambda}\nu^2 + \frac{\alpha^2 L^2}{(1-\lambda)^2}\zeta\right),$$

where $\zeta := \sup_{\mathbf{x}} \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\mathbf{x}) - \sum_i \nabla f_i(\mathbf{x})\right\|^2$

- DSGD is impacted by three components:
  - Dissimilarity (η or ζ) between the local f_i's and the global $F = \sum_i f_i$
  - Variance ν^2 of the stochastic gradient
  - Spectral gap of the network $(1-\lambda)$

This talk

- Eliminate the dependence on local and global dissimilarity
- Eliminate the variance of the stochastic gradient
- Develop network-independent convergence rates

- Precise statements on mean-squared and almost sure convergence
- Speedup when compared with centralized counterparts

- Optimal rates

Decentralized Stochastic Gradient Descent with Gradient Tracking

Addressing the local and global dissimilarity

GT-DSGD: Intuition

- Problem: $\min_{\mathbf{x}} \sum_i f_i(\mathbf{x})$
- DSGD with full gradient and constant step-size:

$$\mathbf{x}_{k+1}^i = \sum_{r=1}^{n} w_{ir} \cdot \mathbf{x}_k^r - \alpha \cdot \nabla f_i(\mathbf{x}_k^i)$$

- Impacted by $\|\nabla f_i(\mathbf{x}^*) - \nabla F(\mathbf{x}^*)\|$ (sc) or $\|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|$ (ncvx)

- $\mathbf{x}^*$ is not a fixed point: $\mathbf{x}^* \neq \sum_{r=1}^n w_{ir} \cdot \mathbf{x}^* - \alpha \cdot \nabla f_i(\mathbf{x}^*)$
- At $\mathbf{x}^*$: $\sum_i \nabla f_i(\mathbf{x}^*) = 0$, which does not imply $\nabla f_i(\mathbf{x}^*) = 0$

- Fix: Replace ∇f_i with an estimate of the global gradient ∇F
- Full gradient: [Xu et al. '15], [Lorenzo et al. '15], [Qu et al. '16], [Xi-Xin-Khan '16], [Shi et al. '16]
- Stochastic gradient: [Pu et al. '18], [Xin-Sahu-Khan-Kar '19]

GT-DSGD: Algorithm

- Problem: $\min_{\mathbf{x}} \sum_i f_i(\mathbf{x})$
- DSGD with a constant step-size: $\mathbf{x}_{k+1}^i = \sum_{r=1}^n w_{ir} \cdot \mathbf{x}_k^r - \alpha \cdot \mathbf{g}_i(\mathbf{x}_k^i)$

Algorithm 1: GT-DSGD at each node i

Data: $\mathbf{x}_0^i$; $\{\alpha_k\}$; $\{w_{ir}\}_{r=1}^n$; $\mathbf{y}_0^i = \mathbf{0}_p$; $\mathbf{g}_r(\mathbf{x}_{-1}^i, \boldsymbol{\xi}_{-1}^i) := \mathbf{0}_p$.
for $k = 0, 1, \ldots,$ **do**

$$\mathbf{y}_{k+1}^i = \sum_{r=1}^n w_{ir}\big(\mathbf{y}_k^r + \mathbf{g}_r(\mathbf{x}_k^r, \boldsymbol{\xi}_k^r) - \mathbf{g}_r(\mathbf{x}_{k-1}^r, \boldsymbol{\xi}_{k-1}^r)\big)$$

$$\mathbf{x}_{k+1}^i = \sum_{r=1}^n w_{ir}\big(\mathbf{x}_k^r - \alpha_k \cdot \mathbf{y}_{k+1}^r\big)$$

end

- The variable $\mathbf{y}_k^i$ tracks the global gradient $\nabla F(\mathbf{x}_k^i)$ at each node i
- Dynamic average consensus: [Zhu et al. '08]

GT-DSGD: Experiment

- Decentralized linear regression (strongly convex)
- Full gradient, $n = 500$ nodes, random connected graph

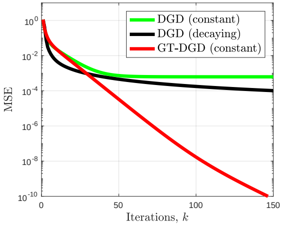

Figure 7: Performance comparison

- When perfect gradients are used:
  - Without GT, convergence is linear but inexact due to the local-vs-global dissimilarity bias
  - With GT, convergence is linear and exact
- What happens when the gradients are stochastic?

GT-DSGD (constant step-size): Addressing the local and global dissimilarity

Smooth non-convex problems satisfying PL condition

GT-DSGD (constant step-size):
Smooth non-convex problems satisfying PL condition

Theorem (abridged, Xin-Khan-Kar '20†)

Let F satisfy the PL condition. For a certain constant step-size α, the mean optimality gap decays **linearly** *at $\mathcal{O}((1-\mu\alpha)^k)$ to an error ball:*

$$\limsup_{k\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[F(\mathbf{x}_k^i) - F^*\right] \leq \underbrace{\mathcal{O}\left(\frac{\alpha\kappa}{n}\nu^2\right)}_{\text{Centralized minibatch SGD}} + \underbrace{\mathcal{O}\left(\alpha^2\kappa L \frac{\lambda^2}{(1-\lambda)^3}\nu^2\right)}_{\text{Decentralized network effect}}$$

The bias due to the local and global cost dissimilarity is eliminated

- For $\alpha \leq \mathcal{O}\left(\frac{(1-\lambda)^3}{\lambda^2 nL}\right)$, the R.H.S matches centralized minibatch SGD
  - n times better than the centralized SGD

    (with data parallelization and communication over n machines)
- The results are immediately applicable to strongly convex problems
- Perfect gradient ($\nu = 0$): ϵ-complexity is $\mathcal{O}(\kappa^{5/4}\log\frac{1}{\epsilon})$
  - Improves the best known rate under PL [Tang et al. '19]
  - Under strong convexity [Li et al. '19]: $\mathcal{O}(\kappa \log\frac{1}{\epsilon})$]
- † *An improved convergence analysis for decentralized online stochastic non-convex optimization*: `https://arxiv.org/abs/2008.04195`

GT-DSGD (constant step-size): Addressing the local and global dissimilarity

General smooth non-convex problems

GT-DSGD (constant step-size): General (smooth) non-convex problems

Theorem (abridged, Xin-Khan-Kar '20†)

For any step-size $\alpha \in \left(0, \min\left\{1, \frac{1-\lambda^2}{3\lambda}, \frac{(1-\lambda^2)^2}{4\sqrt{3}\lambda^2}\right\}\frac{1}{2L}\right]$, *we have* $\forall K > 0$,

$$\underbrace{\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}_k^i)\right\|^2\right]}_{\text{Mean-squared stationary gap}} \leq \underbrace{\frac{4(F(\bar{\mathbf{x}}_0) - F^*)}{\alpha K} + \frac{2\alpha L}{n}\nu^2}_{\text{Centralized minibatch SGD}} + \underbrace{\frac{320\alpha^2 L^2\lambda^2}{(1-\lambda^2)^3}\nu^2 + \frac{64\alpha^2 L^2\lambda^4}{(1-\lambda^2)^3 K}\frac{\|\nabla \mathbf{f}_0\|^2}{n}}_{\text{Decentralized network effect}}$$

- Asymptotic characterization, $K \to \infty$
  - For any $\alpha \leq \mathcal{O}\left(\frac{(1-\lambda)^3}{\lambda^2 nL}\right)$, the R.H.S matches the centralized minibatch SGD *(up to constant factors)*
    n times improvement over centralized SGD
- †*An improved convergence analysis for decentralized online stochastic non-convex optimization*: `https://arxiv.org/abs/2008.04195`

GT-DSGD (constant step-size): General (smooth) non-convex problems

Theorem (abridged, Xin-Khan-Kar '20[†])

Let $\|\nabla \mathbf{f}_0\|^2 = \mathcal{O}(n)$, $\alpha = \left(\frac{n}{K}\right)^{1/2}$, *and* $K \geq 4nL^2 \max\left\{1, \frac{9\lambda^2}{(1-\lambda^2)^2}, \frac{48\lambda^4}{(1-\lambda^2)^4}\right\}$, *then*

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_k^i)\|^2\right] \leq \underbrace{\frac{4(F(\overline{\mathbf{x}}_0) - F^*)}{\sqrt{nK}} + \frac{2\nu_a^2 L}{\sqrt{nK}}}_{\text{Centralized minibatch SGD}} + \underbrace{\frac{320n\lambda^2\nu_a^2L^2}{(1-\lambda^2)^3K} + \frac{64nL^2\lambda^4}{(1-\lambda^2)^3K^2}}_{\text{Decentralized network effect}}$$

Thus, with $K \geq K_{nc} := \mathcal{O}\left(\frac{n^3\lambda^4L^2}{(1-\lambda)^6}\right)$,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_k^i)\|^2\right] \leq \mathcal{O}\left(\frac{\nu_a^2 L}{\sqrt{nK}}\right).$$

- Non-asymptotic characterization
  - Linear $\mathcal{O}(n)$ speedup over centralized SGD
  - Network-independent convergence rate *(in a finite time)*
- [†] *An improved convergence analysis for decentralized online stochastic non-convex optimization*: `https://arxiv.org/abs/2008.04195`

GT-DGD (constant step-size): Demo

- Full gradient, decentralized linear regression, $n = 100$ nodes
- Each node possesses one data point
- Collaborate to learn the slope and intercept

GT-DSGD (constant step-size): Experiment

- Full vs. stochastic gradient
- Decentralized linear regression, $n = 100$ nodes

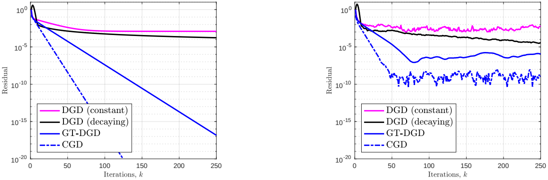

Figure 8: GT-DGD vs. GT-DSGD

- Gradient tracking eliminates the local and global dissimilarity bias
- The variance of the stochastic gradient still remains

- Addressing the variance
  - Online problems: decaying step-sizes
  - Batch problems: variance reduction

Online GT-DSGD (decaying step-sizes)

Addressing the local and global dissimilarity

Addressing the variance of the stochastic gradient

GT-DSGD (decaying step-sizes): Smooth non-convex problems satisfying PL condition

Theorem (abridged, Xin-Khan-Kar '20[†])

Consider the step-size sequence $\alpha_k = \frac{6}{\mu(k+\gamma)}$, *with* $\gamma = \max\left\{\frac{6}{\mu\overline{\alpha}}, \frac{8}{1-\lambda^2}\right\}$. *Suppose that* $\|\nabla \mathbf{f}(\mathbf{x}_0)\|^2 = \mathcal{O}(n)$, *then we have*

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[F(\mathbf{x}_k^i) - F^*\right] \leq \mathcal{O}\left(\frac{\kappa^2\left(F(\overline{\mathbf{x}}_0) - F^*\right)}{k^2} + \frac{\kappa}{n\mu k}\nu^2\right),$$

when $k \geq K_{PL} := \mathcal{O}\left(\max\left\{\frac{\lambda^2 n\kappa}{(1-\lambda)^3}, \frac{\lambda\kappa^{5/4}}{1-\lambda}, \kappa, \frac{\lambda^{3/2}\kappa^{11/8}}{(1-\lambda)^{3/2}}, \frac{\kappa^{-1/2}}{(1-\lambda)^{3/2}}\right\}\right)$.

- Non-asymptotic, asymptotic, and network-independent behaviors
- The rate matches the centralized minibatch SGD when $k \geq K_{PL}$
- Only requires the global cost $\sum_i f_i$ to satisfy the PL condition
- In contrast, existing work requires each f_i to be strongly convex and $k \geq \mathcal{O}(\frac{n^2\kappa^6}{(1-\lambda)^2})$ iterations for network-independence
- [†] *An improved convergence analysis for decentralized online stochastic non-convex optimization*: `https://arxiv.org/abs/2008.04195`

GT-DSGD (decaying step-sizes): Smooth non-convex problems satisfying PL condition

Theorem (abridged, Xin-Khan-Kar '20†)

Consider the step-size sequence: $\alpha_k = \frac{1}{(k+1)}$. *For an arbitrarily small* $\varepsilon > 0$, *we have* $\forall i, j$,

$$\mathbb{P}\Big(\lim_{k\to\infty} k^{1-\varepsilon} \big\| \mathbf{x}_k^i - \mathbf{x}_k^j \big\|^2 = 0 \Big) = 1,$$

$$\mathbb{P}\Big(\lim_{k\to\infty} k^{1-\varepsilon} \big(F(\mathbf{x}_k^i) - F^* \big) = 0 \Big) = 1.$$

- Asymptotic almost sure characterization
  - The proof uses the Robbins-Siegmund almost supermartingale convergence theorem
  - This is the first pathwise rate for decentralized stochastic optimization *(to the best of our knowledge)*
  - Leads to almost sure statements for strongly convex problems
- The analysis techniques are of value in other related problems
- † *An improved convergence analysis for decentralized online stochastic non-convex optimization*: `https://arxiv.org/abs/2008.04195`

The GT-VR framework: Batch problems

Addressing the local and global dissimilarity

Addressing the variance of the stochastic gradient

GT-VR framework: Batch problems

- Each node i possesses a local batch of m_i data samples
  - The local cost f_i is the sum over all data samples $\sum_{j=1}^{m_i} f_{i,j}$
  - Distribution is arbitrary in terms of both quantity and quality

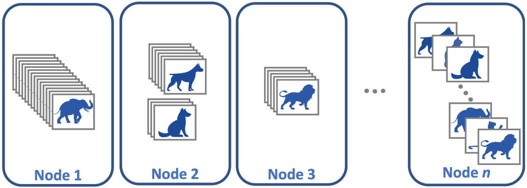

Figure 9: Data distributed within each node and over multiple nodes

- Gradient computation $\sum_{j=1}^{m_i} \nabla f_{i,j}$ is $\mathcal{O}(m_i)$ per node per iteration
  - Full gradient GD can be prohibitively expensive:
    $\mathbf{x}_{k+1}^i = \sum_r w_{ir} \cdot \mathbf{x}_k^r - \alpha \cdot \sum_{j=1}^{m_i} \nabla f_{i,j}(\mathbf{x}_k^i)$

GT-VR framework: Batch problems

- An efficient method is to sample one data point $f_{i,\tau}$ per iteration
  - $\mathbf{x}_{k+1}^i = \sum_r w_{ir} \cdot \mathbf{x}_k^r - \alpha \cdot \nabla f_{i,\tau}(\mathbf{x}_k^i)$
  - Performance is impacted due to sampling and local vs. global bias

- The GT-VR framework: From $\nabla f_{i,\tau}$ to $\nabla F = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \nabla f_{i,j}$
  - Local variance reduction at each node
  - Global gradient tracking over the node network

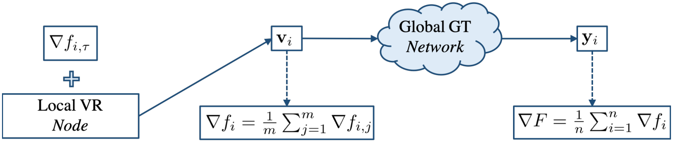

Figure 10: GT-VR: Sample, estimate using VR, and track using GT

- Popular VR methods: SAG, SAGA, SVRG, SPIDER, SARAH

GT-SAGA

- At node i [Xin-Khan-Kar '19[†]]
- Maintain a gradient table $[\widehat{\nabla f}_{i,1}, \ldots, \widehat{\nabla f}_{i,m_i}]$
- At each $k = 0, 1, \ldots$
  - Update $\mathbf{x}_{k+1}^i = \sum_r w_{ir} \cdot \mathbf{x}_k^r - \alpha \cdot \mathbf{y}_k^i$
  - Sample a random index s_k^i from $1, \ldots, m_i$
  - SAGA [Defazio et al. '14]: $\mathbf{v}_{k+1}^i = \nabla f_{i,s_k^i}(\mathbf{x}_{k+1}^i) - \widehat{\nabla f}_{i,s_k^i} + \frac{1}{m_i} \sum_j \widehat{\nabla f}_{i,j}$
  - Update the gradient table: $\widehat{\nabla f}_{i,s_k^i} \leftarrow \nabla f_{i,s_k^i}(\mathbf{x}_{k+1}^i)$
  - Use the estimated $\mathbf{v}_{k+1}^i$ to update the GT variable $\mathbf{y}_{k+1}^i$
- [†] *Variance-reduced decentralized stochastic optimization with accelerated convergence*: https://arxiv.org/abs/1912.04230

GT-SVRG

- At node i [Xin-Khan-Kar '19†]
- Outer loop iterate $\mathbf{x}_k^i$, and inner loop iterate $\underline{\mathbf{x}}_t^i$
- At each k, compute the local full gradient: $\nabla f_i(\mathbf{x}_k^i) = \frac{1}{m_i} \sum_j \nabla f_{i,j}(\mathbf{x}_k^i)$
  - At each $t = [1, \ldots, T]$
    - Update $\underline{\mathbf{x}}_{t+1}^i$ with the GT variable
    - Sample a random index τ from $1, \ldots, m_i$
    - SVRG [Johnson et al. '13]: $\mathbf{v}_{t+1}^i = \nabla f_{i,\tau}(\underline{\mathbf{x}}_{t+1}^i) - \nabla f_{i,\tau}(\mathbf{x}_k^i) + \nabla f_i(\mathbf{x}_k^i)$
    - Use the estimated $\mathbf{v}_{t+1}^i$ in GT
  - Set $\mathbf{x}_{k+1} = \underline{\mathbf{x}}_T^i$ or $\frac{1}{T} \sum_t \underline{\mathbf{x}}_t^i$

- † *Variance-reduced decentralized stochastic optimization with accelerated convergence*: https://arxiv.org/abs/1912.04230

GT-SARAH

- At node i [Xin-Khan-Kar. '20†]
- Outer loop iterate $\mathbf{x}_k^i$, and inner loop iterate $\underline{\mathbf{x}}_t^i$
- At each k, compute the local full gradient: $\nabla f_i(\mathbf{x}_k^i) = \frac{1}{m_i} \sum_j \nabla f_{i,j}(\mathbf{x}_k^i)$
  - At each $t = [1, \ldots, T]$
    - Update $\underline{\mathbf{x}}_{t+1}^i$ with the GT variable
    - Sample a random index τ from $1, \ldots, m_i$
    - SARAH [Nguyen et al. '17], [Fang et al. '18]:
      $\mathbf{v}_{t+1}^i = \nabla f_{i,\tau}(\underline{\mathbf{x}}_{t+1}^i) - \nabla f_{i,\tau}(\underline{\mathbf{x}}_t^i) + \mathbf{v}_t^i$
    - Use the estimated $\mathbf{v}_{t+1}^i$ in GT
  - Set $\mathbf{x}_{k+1} = \underline{\mathbf{x}}_T^i$ or $\frac{1}{T} \sum_t \underline{\mathbf{x}}_t^i$
- † *A near-optimal stochastic gradient method for decentralized non-convex finite-sum optimization*: `https://arxiv.org/abs/2008.07428`

GT-SAGA: Smooth and strongly convex

Theorem (Mean-squared and almost sure convergence Xin-Khan-Kar '19†)

Let $m := \min m_i$ *and* $M := \max_i m_i$. *Under a certain constant step-size* α, *GT-SAGA achieves an* ϵ*-optimal solution of* $\mathbf{x}^*$ *in*

$$\mathcal{O}\left(\max\left\{M, \frac{M}{m}\frac{\kappa^2}{(1-\lambda)^2}\right\}\log\frac{1}{\epsilon}\right)$$

component gradient computations (iterations) at each node.

In addition, we have, $\forall i \in \{1, \cdots, n\}$,

$$\mathbb{P}\left(\lim_{k\to\infty} \gamma_g^{-k}\left\|\mathbf{x}_k^i - \mathbf{x}^*\right\|^2 = 0\right) = 1,$$

where $\gamma_g = 1 - \min\left\{\mathcal{O}\left(\frac{1}{M}\right), \mathcal{O}\left(\frac{m(1-\lambda)^2}{M\kappa^2}\right)\right\}$.

- † *Variance-reduced decentralized stochastic optimization with accelerated convergence*: https://arxiv.org/abs/1912.04230

GT-SVRG: Smooth and strongly convex

Theorem (Mean-squared and almost sure convergence Xin-Khan-Kar '19[†])

Let $m := \min m_i$ *and* $M := \max_i m_i$. *Under a certain constant step-size* α, *GT-SVRG achieves an* ϵ*-optimal solution of* $\mathbf{x}^*$ *in*

$$\mathcal{O}\left(\left(M + \frac{\kappa^2 \log \kappa}{(1-\lambda^2)^2}\right) \log \frac{1}{\epsilon}\right)$$

component gradient computations at each node.

In addition, we have, $\forall i \in \{1, \cdots, n\}$,

$$\mathbb{P}\left(\lim_{k \to \infty} 0.8^{-k} \left\| \mathbf{x}_i^k - \mathbf{x}^* \right\|^2 = 0\right) = 1.$$

- [†] *Variance-reduced decentralized stochastic optimization with accelerated convergence*: https://arxiv.org/abs/1912.04230

GT-SAGA vs. GT-SVRG: Smooth and strongly convex

- $\mathcal{O}\left(\max\left\{M, \frac{M}{m}\frac{\kappa^2}{(1-\lambda)^2}\right\}\log\frac{1}{\epsilon}\right)$ vs. $\mathcal{O}\left(\left(M + \frac{\kappa^2\log\kappa}{(1-\lambda^2)^2}\right)\log\frac{1}{\epsilon}\right)$

- Big-data regime $M = m \approx \mathcal{O}(\kappa^2(1-\lambda)^{-2})$
  - $\mathcal{O}(M\log\frac{1}{\epsilon})$ vs. centralized $\mathcal{O}(nM\log\frac{1}{\epsilon})$
  - Linear speedup vs. the centralized

- Uneven data distribution: $M \gg m = 1$
  - $\mathcal{O}\left(M\frac{\kappa^2}{(1-\lambda)^2}\log\frac{1}{\epsilon}\right)$ vs. $\mathcal{O}\left(\left(M + \frac{\kappa^2\log\kappa}{(1-\lambda^2)^2}\right)\log\frac{1}{\epsilon}\right)$
  - GT-SVRG performs better at the expense of added synchronization
  - GT-SAGA on the other hand needs $\mathcal{O}(m_i)$

GT-SAGA vs. SVRG: Experiments

- Non-asymptotic network-independent convergence
- Linear speedup vs. centralized counterparts

Figure 11: GT-SAGA and GT-SVRG: Behavior in the big-data regime

GT-SAGA vs. SVRG: Experiments

- Comparison with related work

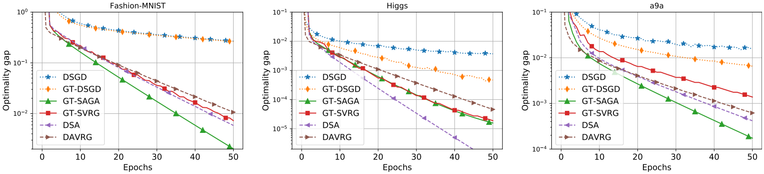

Figure 12: Performance comparison over different datasets

GT-SARAH: Smooth and non-convex

Theorem (Almost sure and mean-squared convergence Xin-Khan-Kar '20†)

For arbitrary inner loop length, as long as the constant step-size α is less than a certain upper bound, GT-SARAH's outer loop iterate $\mathbf{x}_k^i$ follows

$$\mathbb{P}\left(\lim_{k\to\infty} \|\nabla F(\mathbf{x}_k^i)\| = 0\right) = 1 \qquad \textit{and} \qquad \lim_{k\to\infty} \mathbb{E}\left[\left\|\nabla F(\mathbf{x}_k^i)\right\|^2\right] = 0.$$

- † *A near-optimal stochastic gradient method for decentralized non-convex finite-sum optimization*: `https://arxiv.org/abs/2008.07428`

GT-SARAH: Smooth and non-convex

- Total of $N = nm$ data points divided equally among n nodes

Theorem (Gradient computation complexity Xin-Khan-Kar '20[†])

Under a certain constant step-size α, GT-SARAH, with $\mathcal{O}(m)$ inner loop iterations, reaches an ϵ-optimal stationary point of the global cost F in

$$\mathcal{H} := \mathcal{O}\left(\max\left\{N^{1/2}, \frac{n}{(1-\lambda)^2}, \frac{(n+m)^{1/3}n^{2/3}}{1-\lambda}\right\}\left(Lc + \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(\overline{\mathbf{x}}_0)\right\|^2\right)\frac{1}{\epsilon}\right)$$

gradient computations across all nodes, where $c := F\left(\overline{\mathbf{x}}_0\right) - F^*$.

- In the regime $n \leq \mathcal{O}(N^{1/2}(1-\lambda)^3)$: $\mathcal{H} = \mathcal{O}(N^{1/2}\epsilon^{-1})$
  - **Matches the near-optimal algorithmic lower bound** [SPIDER: Fang et al. '18]

- [†] *A near-optimal stochastic gradient method for decentralized non-convex finite-sum optimization*: `https://arxiv.org/abs/2008.07428`

GT-SARAH: Smooth and non-convex

- Minimize a sum of $N := nm$ smooth non-convex functions
- Near-optimal Rate: $O(N^{1/2}\epsilon^{-1})$ in the regime $n \leq \mathcal{O}(N^{1/2}(1-\lambda)^3)$
  - **Matches the near-optimal algorithmic lower bound** [SPIDER: Fang et al. '18]
- Independent of the variance of local gradient estimators
- Independent of the local vs. global dissimilarity bias
- Network-independent performance
- Linear speedup

Conclusions

- Gradient tracking plus DSGD (constant step-sizes)
  - GT eliminates the local vs. global dissimilarity bias
  - Improved rates for non-convex functions (and PL condition)
- Gradient tracking plus DSGD (decaying step-sizes)
  - Decaying step-sizes eliminate the variance due to the stochastic grad
  - Improved rates and analysis for non-convex functions satisfying the PL condition
- GT-VR for batch problems
  - Linear convergence for smooth strongly convex problems
  - Near-optimal performance for non-convex finite sum problems
- Linear speedup
- Network-independent convergence behavior
- Regimes where decentralized methods "outperform" their centralized counterparts

GT-SARAH: Analysis

GT-SARAH: Analysis

- Use the L-smoothness of F to establish the following lemma
$$F(\mathbf{y}) \leq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \tfrac{L}{2} \|\mathbf{y} - \mathbf{x}\|^2 \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$$

Lemma (Descent inequality)

If the step-size follows that $0 < \alpha \leq \frac{1}{2L}$, *then we have*

$$\mathbb{E}\left[F\left(\overline{\mathbf{x}}^{T+1,K}\right)\right] \leq F(\overline{\mathbf{x}}^{0,1}) - \frac{\alpha}{2} \sum_{k,t}^{K,T} \mathbb{E}\left[\left\|\nabla F(\overline{\mathbf{x}}^{t,k})\right\|^2\right]$$
$$- \alpha \left(\frac{1}{4} \sum_{k,t}^{K,T} \mathbb{E}\left[\left\|\overline{\mathbf{v}}^{t,k}\right\|^2\right] - \sum_{k,t}^{K,T} \mathbb{E}\left[\left\|\overline{\mathbf{v}}^{t,k} - \overline{\nabla \mathbf{f}}(\mathbf{x}^{t,k})\right\|^2\right] - L^2 \sum_{k,t}^{K,T} \mathbb{E}\left[\frac{\left\|\mathbf{x}^{t,k} - \mathbf{1} \otimes \overline{\mathbf{x}}^{t,k}\right\|^2}{n}\right] \right)$$

- The object in red has two errors that we need to bound
  - Gradient estimation error: $\mathbb{E}[\|\overline{\mathbf{v}}^{t,k} - \overline{\nabla \mathbf{f}}(\mathbf{x}^{t,k})\|^2]$
  - Agreement error: $\mathbb{E}[\|\mathbf{x}^{t,k} - \mathbf{1} \otimes \overline{\mathbf{x}}^{t,k}\|^2]$

GT-SARAH: Analysis

Lemma (Gradient estimation error)

We have $\forall k \geq 1$,

$$\sum_{t=0}^{T} \mathbb{E}\left[\|\overline{\mathbf{v}}^{t,k} - \overline{\nabla \mathbf{f}}(\mathbf{x}^{t,k})\|^2\right] \leq \frac{3\alpha^2 TL^2}{n} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\overline{\mathbf{v}}^{t,k}\|^2\right] + \frac{6TL^2}{n} \sum_{t=0}^{T} \mathbb{E}\left[\frac{\|\mathbf{x}^{t,k} - \mathbf{1} \otimes \bar{\mathbf{x}}^{t,k}\|^2}{n}\right].$$

Lemma (Agreement error)

If the step-size follows $0 < \alpha \leq \frac{(1-\lambda^2)^2}{8\sqrt{42}L}$, *then*

$$\sum_{k=1}^{K} \sum_{t=0}^{T} \mathbb{E}\left[\frac{\|\mathbf{x}^{t,k} - \mathbf{1} \otimes \bar{\mathbf{x}}^{t,k}\|^2}{n}\right] \leq \frac{64\alpha^2}{(1-\lambda^2)^3} \frac{\|\nabla \mathbf{f}(\mathbf{x}^{0,1})\|^2}{n} + \frac{1536\alpha^4 L^2}{(1-\lambda^2)^4} \sum_{k=1}^{K} \sum_{t=0}^{T} \mathbb{E}\left[\|\overline{\mathbf{v}}^{t,k}\|^2\right].$$

- *Agreement error is coupled with the gradient estimation error*
- *Derive an LTI system that describes their evolution*
- *Analyze the LTI dynamics to obtain the agreement error lemma*

- Use the two lemmas back in the descent inequality

GT-SARAH: Analysis

Lemma (Refined descent inequality)

For $0 < \alpha \leq \overline{\alpha} := \min\left\{\frac{(1-\lambda^2)^2}{4\sqrt{42}}, \frac{\sqrt{n}}{\sqrt{6T}}, \left(\frac{2n}{3n+12T}\right)^{\frac{1}{4}}\frac{1-\lambda^2}{6}\right\}\frac{1}{2L}$, *we have*

$$\frac{1}{n}\sum_{i,k,t}^{n,K,T}\mathbb{E}\left[\|\nabla F(\mathbf{x}_i^{t,k})\|^2\right] \leq \frac{4(F(\overline{\mathbf{x}}^{0,1}) - F^*)}{\alpha} + \left(\frac{3}{2} + \frac{6T}{n}\right)\frac{256\alpha^2 L^2}{(1-\lambda^2)^3}\frac{\|\nabla \mathbf{f}(\mathbf{x}^{0,1})\|^2}{n}.$$

- Taking $K \to \infty$ on both sides leads to $\sum_{k,t}^{\infty,T}\mathbb{E}[\|\nabla F(\mathbf{x}_i^{t,k})\|] < \infty$
  - Mean-squared and a.s. results follow

- Divide both sides by $K \cdot T$ and solve for K when the R.H.S $\leq \epsilon$
  - Gradient computation complexity follows by nothing that GT-SARAH computes $n(m+2T)$ gradients per iteration across all nodes
  - Choose α as the maximum and $T = \mathcal{O}(m)$ to obtain the optimal rate