GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity

Arto Maranjyan

Algorithms & Computationally Intensive Inference seminars

Department of Statistics

University of Warwick

6 October 2023

GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity

Artavazd Maranjyan * KAUST† Saudi Arabia

Mher Safaryan KAUST Saudi Arabia

Peter Richtárik KAUST Saudi Arabia

Abstract

We study a class of distributed optimization algorithms that aim to alleviate high communication costs by allowing the clients to perform multiple local gradient-type training steps prior to communication. While methods of this type have been studied for about a decade, the empirically observed acceleration properties of local training eluded all attempts at theoretical understanding. In a recent breakthrough, Mishchenko et al. (ICML 2022) proved that local training, when properly executed, leads to provable communication acceleration, and this holds in the strongly convex regime without relying on any data similarity assumptions. However, their method ProxSkip requires all clients to take the same number of local training steps in each communication round. Inspired by a common sense intuition, we start our investigation by conjecturing that clients with "less important" data should be able to get away with fewer local training steps without this impacting the overall communication complexity of the method. It turns out that this intuition is correct: we managed to redesign the original ProxSkip method to achieve this. In particular, we prove that our modified method, for which coin the name GradSkip, converges linearly under the same assumptions, has the same accelerated communication complexity, while the number of local gradient steps can be reduced relative to a local condition number. We further generalize our method by extending the randomness of probabilistic alternations to arbitrary unbiased compression operators and considering a generic proximable regularizer. This generalization, which we call GradSkip+, recovers several related methods in the literature as special cases. Finally, we present an empirical study on carefully designed toy problems that confirm our theoretical claims.

1 Introduction

Federated Learning (FL) is an emerging distributed machine learning paradigm where diverse data holders or clients (*e.g.*, smart watches, mobile devices, laptops, hospitals) collectively aim to train a single machine learning model without revealing local data to each other or the orchestrating central server [McMahan et al., 2017, Kairouz et al, 2019, Wang, 2021]. Training such models amounts to solving federated optimization problems of the form

$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}. \quad (1)$$

*The work of Artavazd Maranjyan was performed during a Summer research internship in the Optimization and Machine Learning Lab at KAUST led by Peter Richtárik. Artavazd Maranjyan is a Machine Learning researcher at YerevaNN and a Master's student at Yerevan State University, Armenia.

†KAUST = King Abdullah University of Science and Technology.

Preprint. Under review.

arXiv:2210.16402v2 [cs.LG] 30 May 2023

Artavazd Maranjyan, Mher Safaryan, Peter Richtárik

GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity

arXiv:2210.16402, 2023

About me

Ph.D. in Computer Science
King Abdullah University of Science and Technology (KAUST)
Aug 2023 – Present
Thuwal, Saudi Arabia
Advisor: Peter Richtárik

M.Sc. in Applied Statistics and Data Science
Yerevan State University
2021 – 2023
Yerevan, Armenia

B.Sc. in Informatics and Applied Mathematics
Yerevan State University
2017 – 2021
Yerevan, Armenia

Co-authors

Mher Safaryan

KAUST
King Abdullah University of Science and Technology

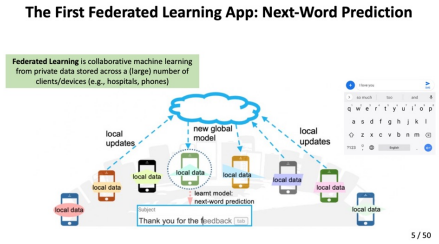
ISTA
Institute of Science and Technology Austria

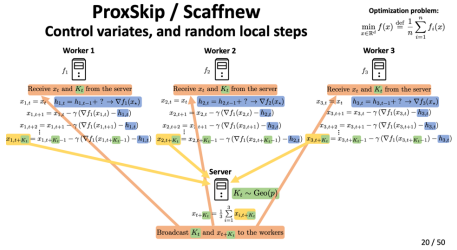
Peter Richtárik

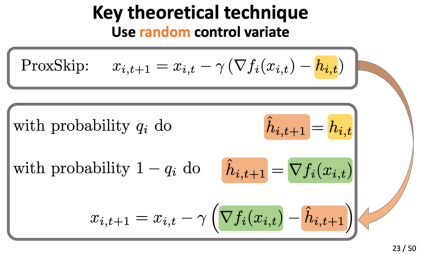
KAUST
King Abdullah University of Science and Technology

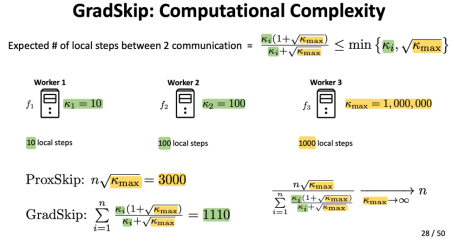
Outline of the Talk

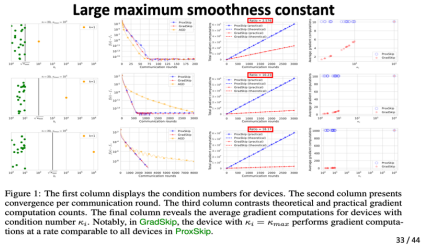
1. What is Federated Learning?
2. What is Local Training?
3. The ProxSkip Algorithm
4. **GradSkip: Algorithm**
5. **GradSkip: Theory**
6. **GradSkip: Experiments**







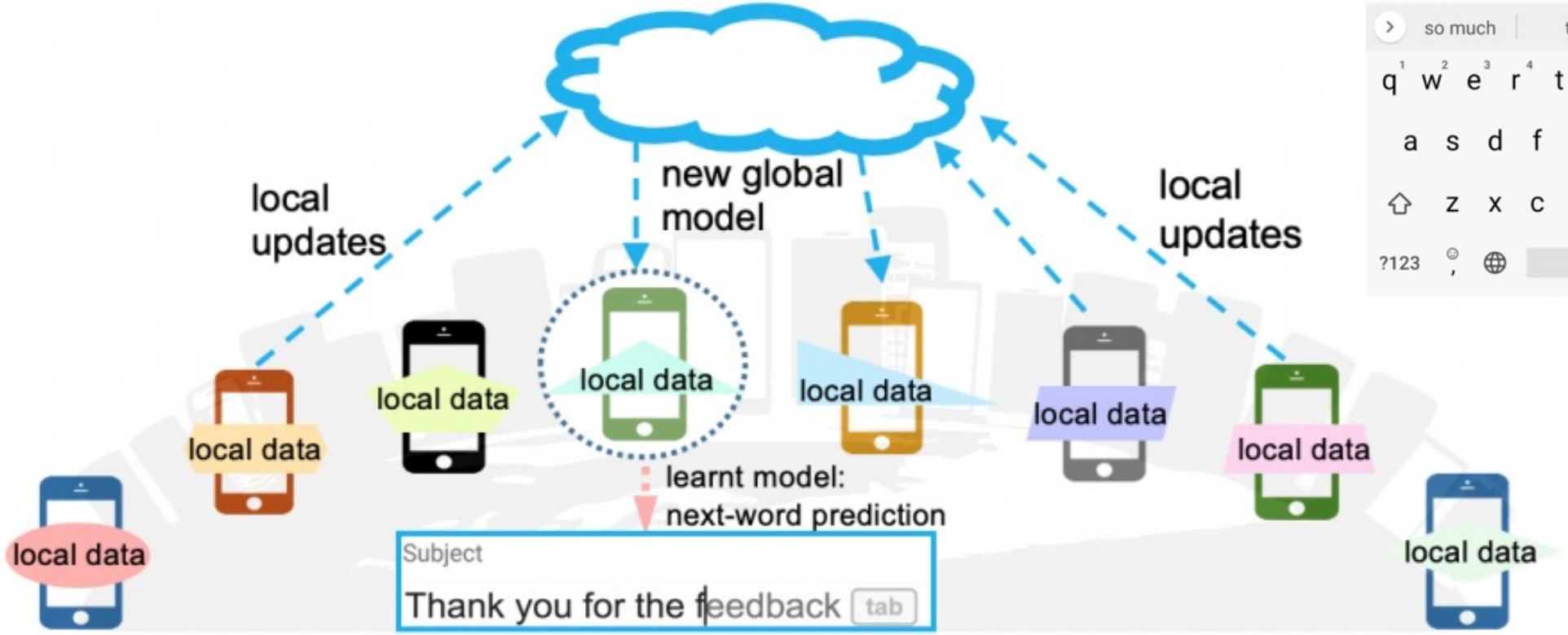


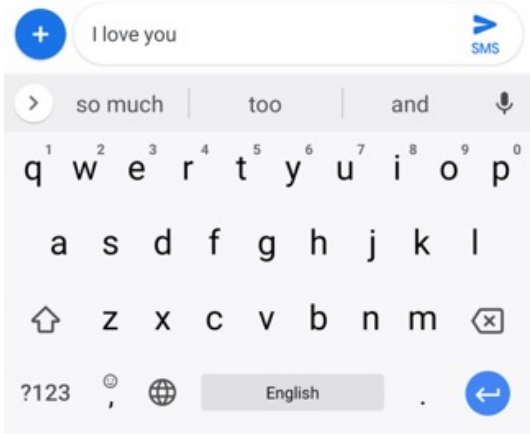


Part 1
What is Federated Learning?

The First Federated Learning App: Next-Word Prediction

Federated Learning is collaborative machine learning from private data stored across a (large) number of clients/devices (e.g., hospitals, phones)





Optimization Formulation of Federated Learning

$$\min_{x \in \mathbb{R}^{d}} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

devices / machines

model parameters / features

Loss on local data $\mathcal{D}_i$ stored on device i

$$f_i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i} f_{i,\xi}(x)$$

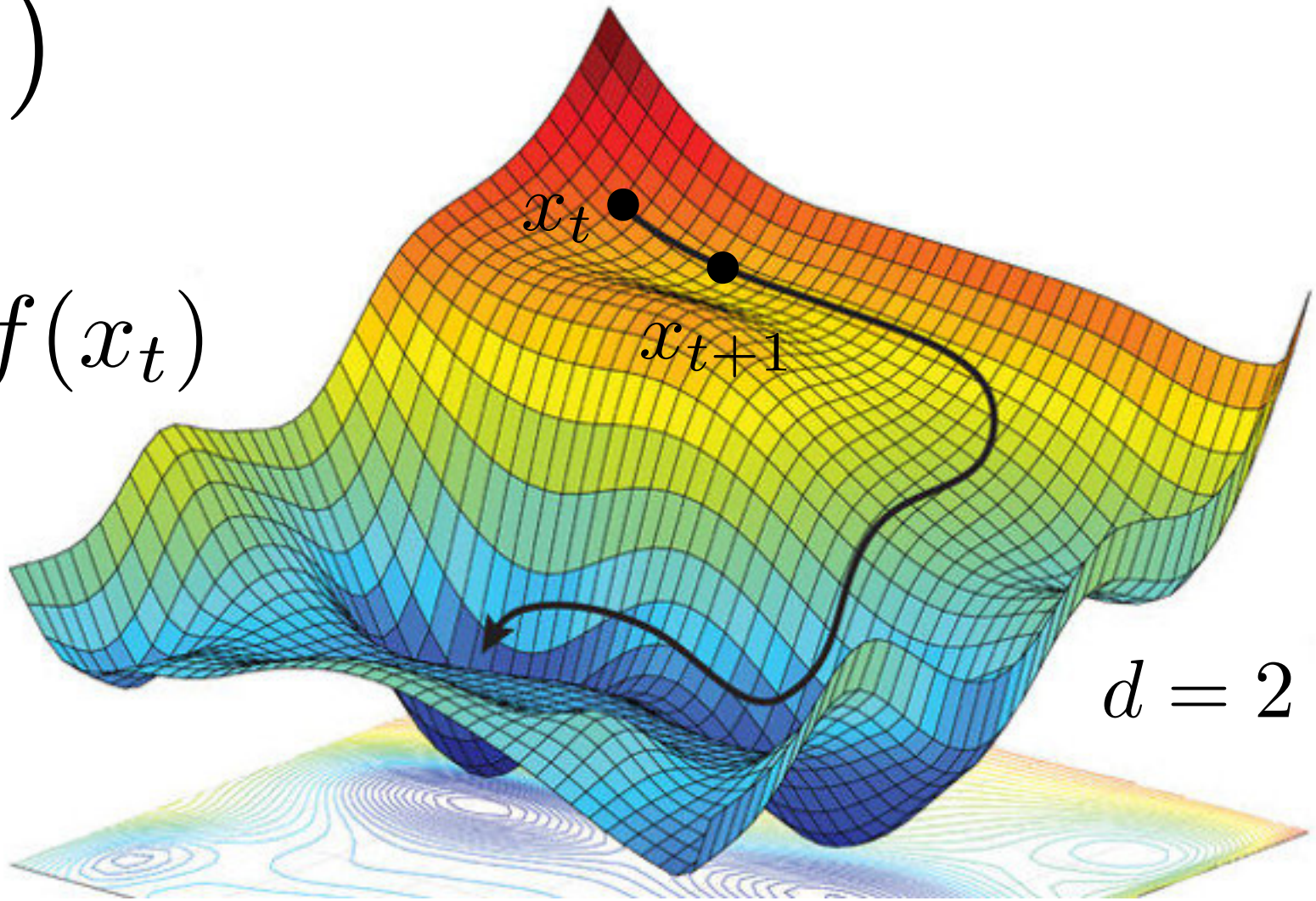
The datasets $\mathcal{D}_1, \ldots, \mathcal{D}_n$ can be arbitrarily heterogeneous

Gradient Descent

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

$$x_{t+1} = x_t - \gamma \nabla f(x_t)$$

Stepsize / Learning rate



Distributed Gradient Descent

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

$$x_{t+1} = x_t - \gamma \nabla f(x_t)$$

d-dimensional gradient computed by machine *i*

d-dimensional vector computed by machine *i*

Distributed Gradient Descent

(Each worker performs 1 GD step using its local function, and the results are averaged)

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Worker 1

f_1

Receive x_t from the server

$x_{1,t} = x_t$

$x_{1,t+1} = x_{1,t} - \gamma\nabla f_1(x_{1,t})$

Worker 2

f_2

Receive x_t from the server

$x_{2,t} = x_t$

$x_{2,t+1} = x_{2,t} - \gamma\nabla f_2(x_{2,t})$

Worker 3

f_3

Receive x_t from the server

$x_{3,t} = x_t$

$x_{3,t+1} = x_{3,t} - \gamma\nabla f_3(x_{3,t})$

Server

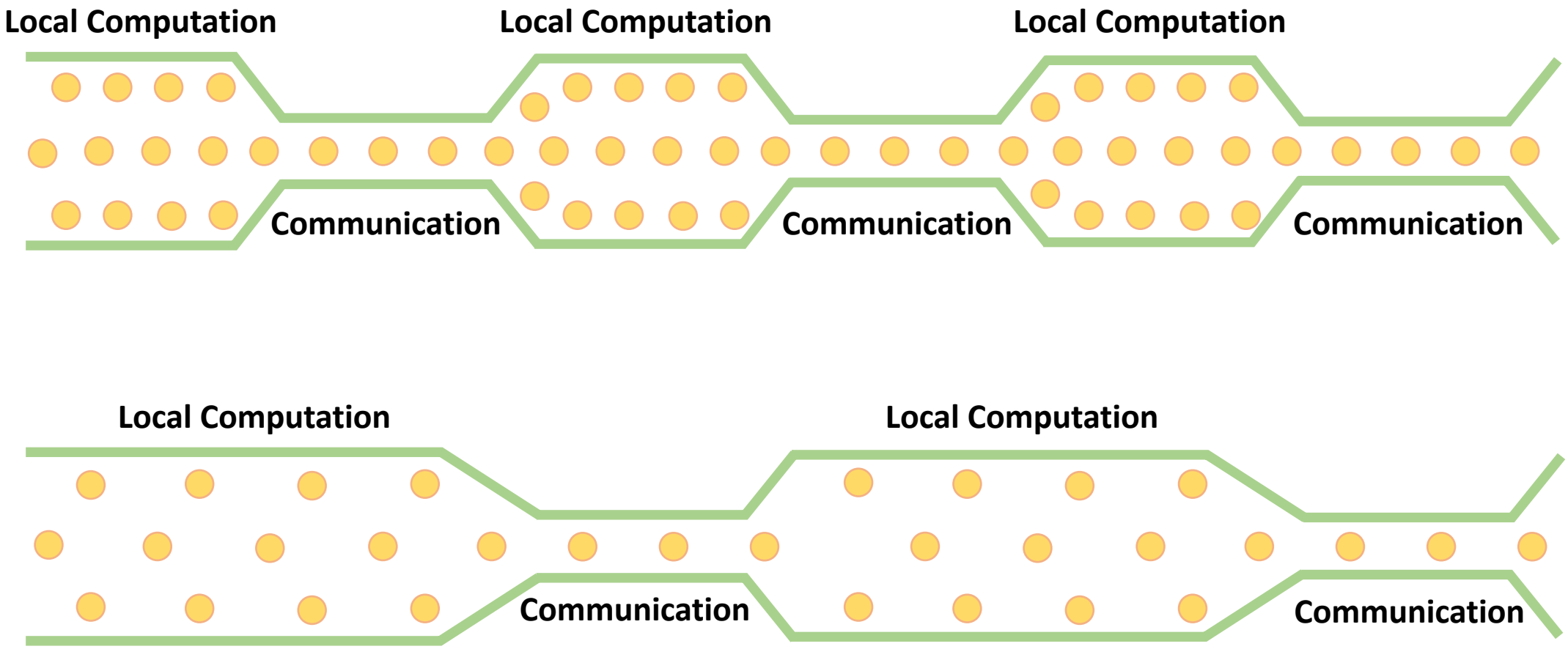
d-dimensional vector computed by machine *1*

$$x_{t+1} = \frac{1}{3}\sum_{i=1}^{3} x_{i,t+1}$$

Broadcast x_{t+1} to the workers

Part 2
What is Local Training

The idea behind Local Training



Distributed Local Gradient Descent

(Each worker performs *K* GD steps using its local function, and the results are averaged)

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Worker 1

f_1

Receive x_t from the server

$x_{1,t} = x_t$

$$x_{1,t+1} = x_{1,t} - \gamma\nabla f_1(x_{1,t})$$
$$x_{1,t+2} = x_{1,t+1} - \gamma\nabla f_1(x_{1,t+1})$$
$$\vdots$$
$$x_{1,t+K} = x_{1,t+K-1} - \gamma\nabla f_1(x_{1,t+K-1})$$

Worker 2

f_2

Receive x_t from the server

$x_{2,t} = x_t$

$$x_{2,t+1} = x_{2,t} - \gamma\nabla f_2(x_{2,t})$$
$$x_{2,t+2} = x_{2,t+1} - \gamma\nabla f_2(x_{2,t+1})$$
$$\vdots$$
$$x_{2,t+K} = x_{2,t+K-1} - \gamma\nabla f_2(x_{2,t+K-1})$$

Worker 3

f_3

Receive x_t from the server

$x_{3,t} = x_t$

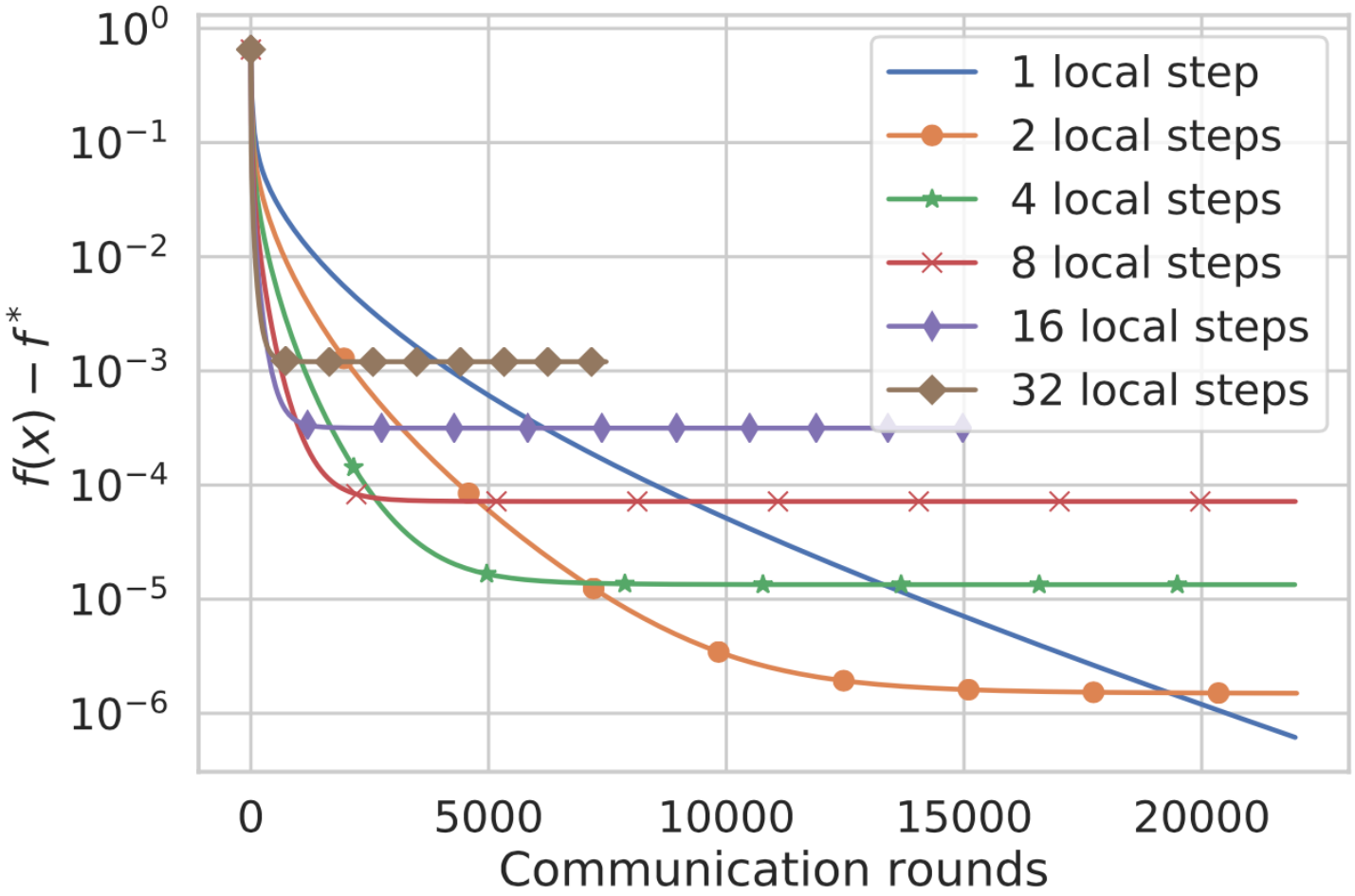
$$x_{3,t+1} = x_{3,t} - \gamma\nabla f_3(x_{3,t})$$
$$x_{3,t+2} = x_{3,t+1} - \gamma\nabla f_3(x_{3,t+1})$$
$$\vdots$$
$$x_{3,t+K} = x_{3,t+K-1} - \gamma\nabla f_3(x_{3,t+K-1})$$

Server

$$x_{t+K} = \frac{1}{3}\sum_{1=1}^{3} x_{i,t+K}$$

Broadcast x_{t+K} to the workers

What does Local Training do?



Plot taken from:

Ahmed Khaled, Konstantin Mishchenko, Peter Richtárik
First Analysis of Local GD on Heterogeneous Data
NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality, 2019

L2-regularized logistic regression
LibSVM `mushrooms` dataset

Brief History of Local Training Methods

Table 1: Five generations of local training (LT) methods summarizing the progress made by the ML/FL community over the span of 7+ years in the understanding of the *communication acceleration properties of LT*.

Generation[a]	Theory	Assumptions	Comm. Complexity[b]	Selected Key References
1. Heuristic	✗	—	empirical results only	LocalSGD [Povey et al., 2015]
	✗	—	empirical results only	SparkNet [Moritz et al., 2016]
	✗	—	empirical results only	FedAvg [McMahan et al., 2017]
2. Homogeneous	✓	bounded gradients	sublinear	FedAvg [Li et al., 2020b]
	✓	bounded grad. diversity[c]	linear but worse than GD	LFGD [Haddadpour and Mahdavi, 2019]
3. Sublinear	✓	standard[d]	sublinear	LGD [Khaled et al., 2019]
	✓	standard	sublinear	LSGD [Khaled et al., 2020]
4. Linear	✓	standard	linear but worse than GD	Scaffold [Karimireddy et al., 2020]
	✓	standard	linear but worse than GD	S-Local-GD [Gorbunov et al., 2020a]
	✓	standard	linear but worse than GD	FedLin [Mitra et al., 2021]
5. Accelerated	✓	standard	linear & better than GD	ProxSkip/Scaffnew [Mishchenko et al., 2022]
	✓	standard	linear & better than GD	ProxSkip-VR

[a] Since client sampling (CS) and data sampling (DS) can only *worsen* theoretical communication complexity, our historical breakdown of the literature into 5 generations of LT methods focuses on the full client participation (i.e., no CS) and exact local gradient (i.e., no DS) setting. While some of the referenced methods incorporate CS and DS techniques, these are irrelevant for our purposes. Indeed, from the viewpoint of communication complexity, all these algorithms enjoy best theoretical performance in the no-CS and no-DS regime.

[b] For the purposes of this table, we consider problem (1) in the *smooth* and *strongly convex* regime only. This is because the literature on LT methods struggles to understand even in this simplest (from the point of view of optimization) regime.

[c] *Bounded gradient diversity* is a uniform bound on a specific notion of gradient variance depending on client sampling probabilities. However, this assumption (as all homogeneity assumptions) is very restrictive. For example, it is not satisfied the standard class of smooth and strongly convex functions.

[d] The notorious FL challenge of handling non-i.i.d. data by LT methods was solved by Khaled et al. [2019] (from the viewpoint of *optimization*). From generation 3 onwards, there was no need to invoke any data/gradient homogeneity assumptions. Handling non-i.i.d. data remains a challenge from the point of view of *generalization*, typically by considering *personalized* FL models.

Grigory Malinovsky, Kai Yi, Peter Richtárik
Variance Reduced ProxSkip: Algorithm, Theory and Application to Federated Learning
NeurIPS 2022

What is the accelerated rate?

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\|\nabla f(x) - \nabla f(y)\| \le L\,\|x-y\|$$

$$t \ge T(L,\varepsilon) \;\Rightarrow\; \|x_t - x_\star\|^2 \le \varepsilon\,\|x_0 - x_\star\|^2$$

iterations

Error tolerance

$$x_\star := \arg\min_{x\in\mathbb{R}^d} f(x)$$

Generation	Rate	Number of communications
3 (DLGD, LGD, LSGD)	Sublinear	$T(L,\varepsilon) = 1{,}000{,}000$
4 (FedLin, Scaffold)	Linear = GD	$T(L,\varepsilon) = 10{,}000$
5 (ProxSkip/Scaffnew, ProxSkip-VR)	Accelerated	$T(L,\varepsilon) = 100$

Why treat all devices equally?

(Each worker performs *K* GD steps using its local function, and the results are averaged)

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Worker 1

f_1 L_1

Receive x_t from the server

$x_{1,t} = x_t$

$$x_{1,t+1} = x_{1,t} - \gamma\nabla f_1(x_{1,t})$$
$$x_{1,t+2} = x_{1,t+1} - \gamma\nabla f_1(x_{1,t+1})$$
$$\vdots$$
$$x_{1,t+K} = x_{1,t+K-1} - \gamma\nabla f_1(x_{1,t+K-1})$$

Worker 2

f_2 L_2

Receive x_t from the server

$x_{2,t} = x_t$

$$x_{2,t+1} = x_{2,t} - \gamma\nabla f_2(x_{2,t})$$
$$x_{2,t+2} = x_{2,t+1} - \gamma\nabla f_2(x_{2,t+1})$$
$$\vdots$$
$$x_{2,t+K} = x_{2,t+K-1} - \gamma\nabla f_2(x_{2,t+K-1})$$

Worker 3

f_3 L_3

Receive x_t from the server

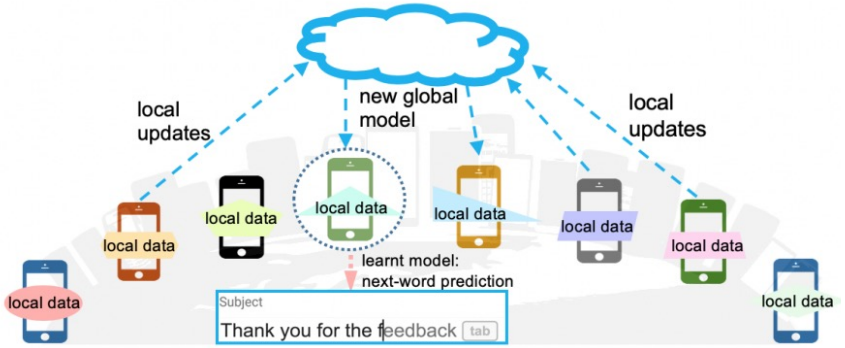
$x_{3,t} = x_t$

$$x_{3,t+1} = x_{3,t} - \gamma\nabla f_3(x_{3,t})$$
$$x_{3,t+2} = x_{3,t+1} - \gamma\nabla f_3(x_{3,t+1})$$
$$\vdots$$
$$x_{3,t+K} = x_{3,t+K-1} - \gamma\nabla f_3(x_{3,t+K-1})$$

Server

$$x_{t+K} = \frac{1}{3}\sum_{1=1}^{3} x_{i,t+K}$$

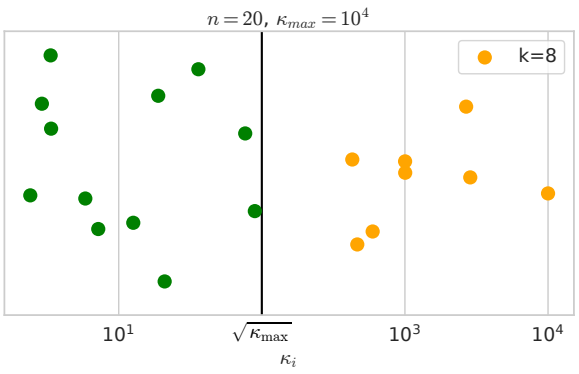
Broadcast x_{t+K} to the workers

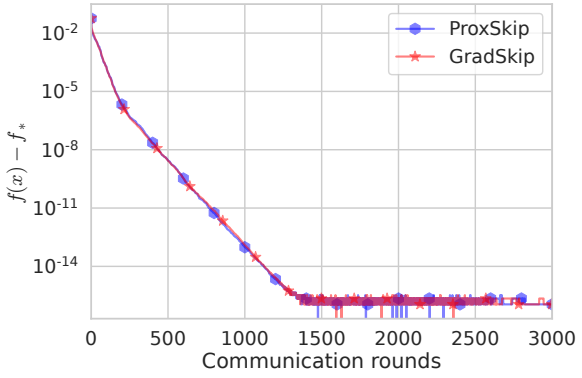


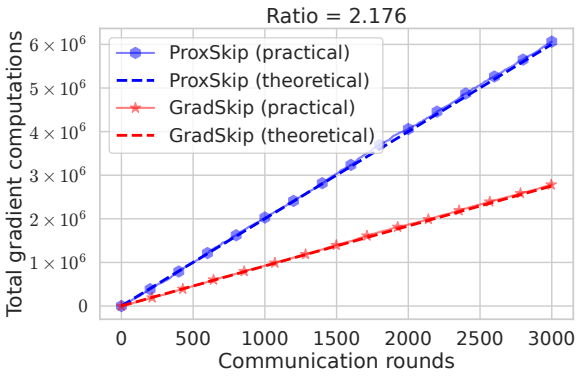
Key insight

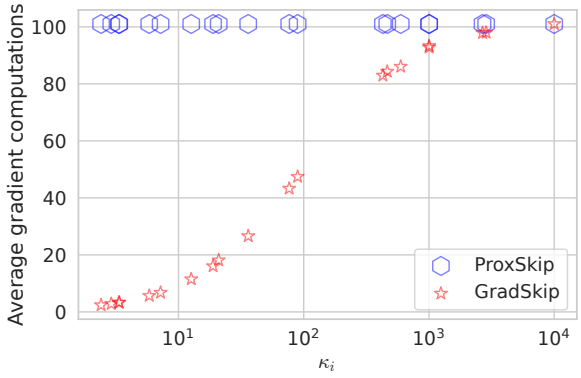
GradSkip = ProxSkip + Heterogeneity Awareness

Algorithm	Communication Complexity	Computational Complexity
ProxSkip	Accelerated (100 communications)	1000 GD steps per client
GradSkip	Accelerated (100 communications)	$\leq$ 1000 GD steps per client









Part 3
The ProxSkip Algorithm

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, Peter Richtárik **ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!** *International Conference on Machine Learning (ICML), 2022*

ProxSkip / Scaffnew

Control variates, and random local steps

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Worker 1

f_1

Receive x_t and K_t from the server

$x_{1,t} = x_t$ $\quad h_{1,t} = h_{1,t-1} + \ ? \ \to \nabla f_1(x_\star)$

$x_{1,t+1} = x_{1,t} - \gamma\left(\nabla f_1(x_{1,t}) - h_{1,t}\right)$

$x_{1,t+2} = x_{1,t+1} - \gamma\left(\nabla f_1(x_{1,t+1}) - h_{1,t}\right)$

$\vdots$

$x_{1,t+K_t} = x_{1,t+K_t-1} - \gamma\left(\nabla f_1(x_{1,t+K_t-1}) - h_{1,t}\right)$

Worker 2

f_2

Receive x_t and K_t from the server

$x_{2,t} = x_t$ $\quad h_{2,t} = h_{2,t-1} + \ ? \ \to \nabla f_2(x_\star)$

$x_{2,t+1} = x_{2,t} - \gamma\left(\nabla f_1(x_{2,t}) - h_{2,t}\right)$

$x_{2,t+2} = x_{2,t+1} - \gamma\left(\nabla f_1(x_{2,t+1}) - h_{2,t}\right)$

$\vdots$

$x_{2,t+K_t} = x_{2,t+K_t-1} - \gamma\left(\nabla f_1(x_{2,t+K_t-1}) - h_{2,t}\right)$

Worker 3

f_3

Receive x_t and K_t from the server

$x_{3,t} = x_t$ $\quad h_{3,t} = h_{3,t-1} + \ ? \ \to \nabla f_3(x_\star)$

$x_{3,t+1} = x_{3,t} - \gamma\left(\nabla f_1(x_{3,t}) - h_{3,t}\right)$

$x_{3,t+2} = x_{3,t+1} - \gamma\left(\nabla f_1(x_{3,t+1}) - h_{3,t}\right)$

$\vdots$

$x_{3,t+K_t} = x_{3,t+K_t-1} - \gamma\left(\nabla f_1(x_{3,t+K_t-1}) - h_{3,t}\right)$

Server

$K_t \sim \mathrm{Geo}(p)$

$$x_{t+K_t} = \frac{1}{3}\sum_{i=1}^{3} x_{i,t+K_t}$$

Broadcast K_t and x_{t+K_t} to the workers

? = Client Drift

$$\frac{p}{\gamma}\left(x_{i,t+K_t} - \hat{x}_{i,t}\right)$$

Part 4
The GradSkip Algorithm

GradSkip

Let workers decide how much to work

Optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Worker 1

f_1 L_1

Receive x_t and K_t from the server

$x_{1,t} = x_t \quad h_{1,t} = h_{1,t-1} + \ ? \ \to \nabla f_1(x_\star)$

$M_{1,t} \sim \mathrm{Geo}(q_1) \quad K_{1,t} = \min\{M_{1,t}, K_t\}$

$x_{1,t+1} = x_{1,t} - \gamma\left(\nabla f_1(x_{1,t}) - h_{1,t}\right)$

$x_{1,t+2} = x_{1,t+1} - \gamma\left(\nabla f_1(x_{1,t+1}) - h_{1,t}\right)$

$\vdots$

$\vdots$

$x_{1,t+K_{1,t}} = x_{1,t+K_{1,t}-1} - \gamma\left(\nabla f_1(x_{1,t+K_{1,t}-1}) - h_{1,t}\right)$

Worker 2

f_2 L_2

Receive x_t and K_t from the server

$x_{2,t} = x_t \quad h_{2,t} = h_{2,t-1} + \ ? \ \to \nabla f_2(x_\star)$

$M_{2,t} \sim \mathrm{Geo}(q_2) \quad K_{2,t} = \min\{M_{2,t}, K_t\}$

$x_{2,t+1} = x_{2,t} - \gamma\left(\nabla f_1(x_{2,t}) - h_{2,t}\right)$

$x_{2,t+2} = x_{2,t+1} - \gamma\left(\nabla f_1(x_{2,t+1}) - h_{2,t}\right)$

$\vdots$

$x_{2,t+K_{2,t}} = x_{2,t+K_{2,t}-1} - \gamma\left(\nabla f_2(x_{2,t+K_{2,t}-1}) - h_{2,t}\right)$

Worker 3

f_3 L_3

Receive x_t and K_t from the server

$x_{3,t} = x_t \quad h_{3,t} = h_{3,t-1} + \ ? \ \to \nabla f_3(x_\star)$

$M_{3,t} \sim \mathrm{Geo}(q_3) \quad K_{3,t} = \min\{M_{3,t}, K_t\}$

$x_{3,t+1} = x_{3,t} - \gamma\left(\nabla f_1(x_{3,t}) - h_{3,t}\right)$

$x_{3,t+2} = x_{3,t+1} - \gamma\left(\nabla f_1(x_{3,t+1}) - h_{3,t}\right)$

$\vdots$

$\vdots$

$\vdots$

$x_{3,t+K_{3,t}} = x_{3,t+K_{3,t}-1} - \gamma\left(\nabla f_3(x_{3,t+K_{3,t}-1}) - h_{3,t}\right)$

Server

$K_t \sim \mathrm{Geo}(p)$

$$x_{t+K_t} = \frac{1}{3}\sum_{i=1}^{3} x_{i,t+K_t}$$

Broadcast K_t and x_{t+K_t} to the workers

? = Client Drift

$$\frac{p}{\gamma}\left(x_{i,t+K_t} - \hat{x}_{i,t}\right)$$

Key theoretical technique

Use random control variate

ProxSkip: $x_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - h_{i,t}\right)$

with probability $1 - q_i$ do $\hat{h}_{i,t+1} = h_{i,t}$

with probability q_i do $\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$

$$x_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1}\right)$$

Part 5
GradSkip Theory

GradSkip: Assumptions same as in ProxSkip

Assumptions:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|$$

$$\langle \nabla f_i(x) - \nabla f_i(y), x - y \rangle \geq \mu \|x - y\|^2$$

GradSkip: Bounding the # of Iterations

Theorem:

$$\gamma \leq \min_i \left\{ \frac{1}{L_i} \frac{p^2}{p^2 + q_i(1-p^2)} \right\}$$

$$t \geq \max \left\{ \frac{1}{\gamma\mu}, \frac{1}{p^2 - q_{\min}(1-p^2)} \right\} \log \frac{1}{\varepsilon} \Rightarrow \mathbb{E}\left[\Psi_t\right] \leq \varepsilon \Psi_0$$

iterations

Lyapunov function:

$$\Psi_t := \sum_{i=1}^{n} \|x_{i,t} - x_\star\|^2 + \frac{\gamma^2}{p^2} \sum_{i=1}^{n} \|h_{i,t} - \nabla f_i(x_\star)\|^2$$

GradSkip: Optimal Probabilities

$$\kappa_i = \frac{L_i}{\mu} \qquad \kappa_{\max} = \frac{L_{\max}}{\mu}$$

$$q_i = \frac{\frac{1}{\kappa_i} - \frac{1}{\kappa_{\max}}}{1 - \frac{1}{\kappa_{\max}}} \qquad p^2 = \frac{1}{\kappa_{\max}}$$

$$\gamma \leq \min_i \left\{ \frac{1}{L_i} \frac{p^2}{p^2 + q_i(1-p^2)} \right\} = \frac{1}{L_{\max}}$$

$$t \geq \max\left\{ \frac{1}{\gamma\mu}, \frac{1}{p^2 - q_{\min}(1-p^2)} \right\} \log \frac{1}{\varepsilon} = \kappa_{\max} \log \frac{1}{\varepsilon} \Rightarrow \mathbb{E}\left[\Psi_t\right] \leq \varepsilon \Psi_0$$

$$q_{\min} = 0$$

GradSkip: Computational Complexity

Expected # of local steps between 2 communication = $\frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} \leq \min\left\{\kappa_i, \sqrt{\kappa_{\max}}\right\}$

Worker 1

f_1 $\kappa_1 = 10$

10 local steps

Worker 2

f_2 $\kappa_2 = 100$

100 local steps

Worker 3

f_3 $\kappa_{\max} = 1,000,000$

1000 local steps

$$\text{ProxSkip: } n\sqrt{\kappa_{\max}} = 3000$$

$$\text{GradSkip: } \sum_{i=1}^{n} \frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} = 1110$$

$$\frac{n\sqrt{\kappa_{\max}}}{\sum_{i=1}^{n} \frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}}} \xrightarrow[\kappa_{\max}\to\infty]{} n$$

GradSkip vs ProxSkip

	GradSkip	ProxSkip
Number of iterations	$\kappa_{\max} \log \frac{1}{\varepsilon}$	$\kappa_{\max} \log \frac{1}{\varepsilon}$
Expected number of communications	$\sqrt{\kappa_{\max}} \log \frac{1}{\varepsilon}$	$\sqrt{\kappa_{\max}} \log \frac{1}{\varepsilon}$
Expected number of local steps between two communications	$\frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} \leq \min\left\{\kappa_i, \sqrt{\kappa_{\max}}\right\}$	$\sqrt{\kappa_{\max}}$
Expected number of local steps	$\sum_{i=1}^{n} \frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} \approx \sqrt{\kappa_{\max}}$	$n\sqrt{\kappa_{\max}}$

Part 6
Experiments

Experimental setup

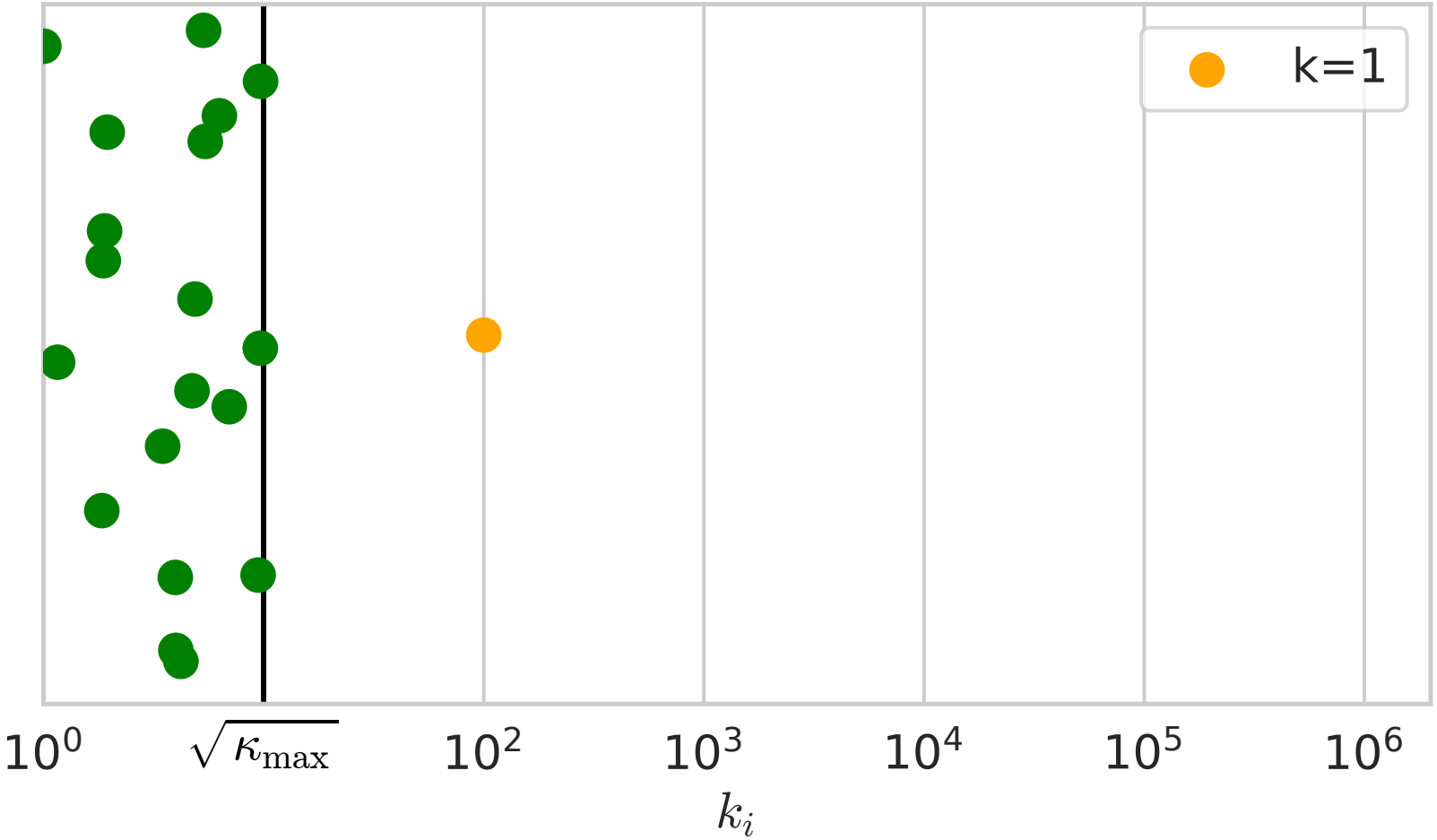
L2-regularized logistic regression:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-b_i a_i^\top x\right)\right) + \frac{\lambda}{2}\|x\|^2$$

$$b_i \in \{-1, +1\}, \lambda = 0.1,$$

$$\mathbf{A}_i = \mathbf{U}_i \mathbf{L}_i \mathbf{V}_i \in \mathbb{R}^{200,300}, \sigma_{max}(\mathbf{A}_i) = L_i$$

$n = 20,\ \kappa_{max} = 10^2$



Large maximum smoothness constant

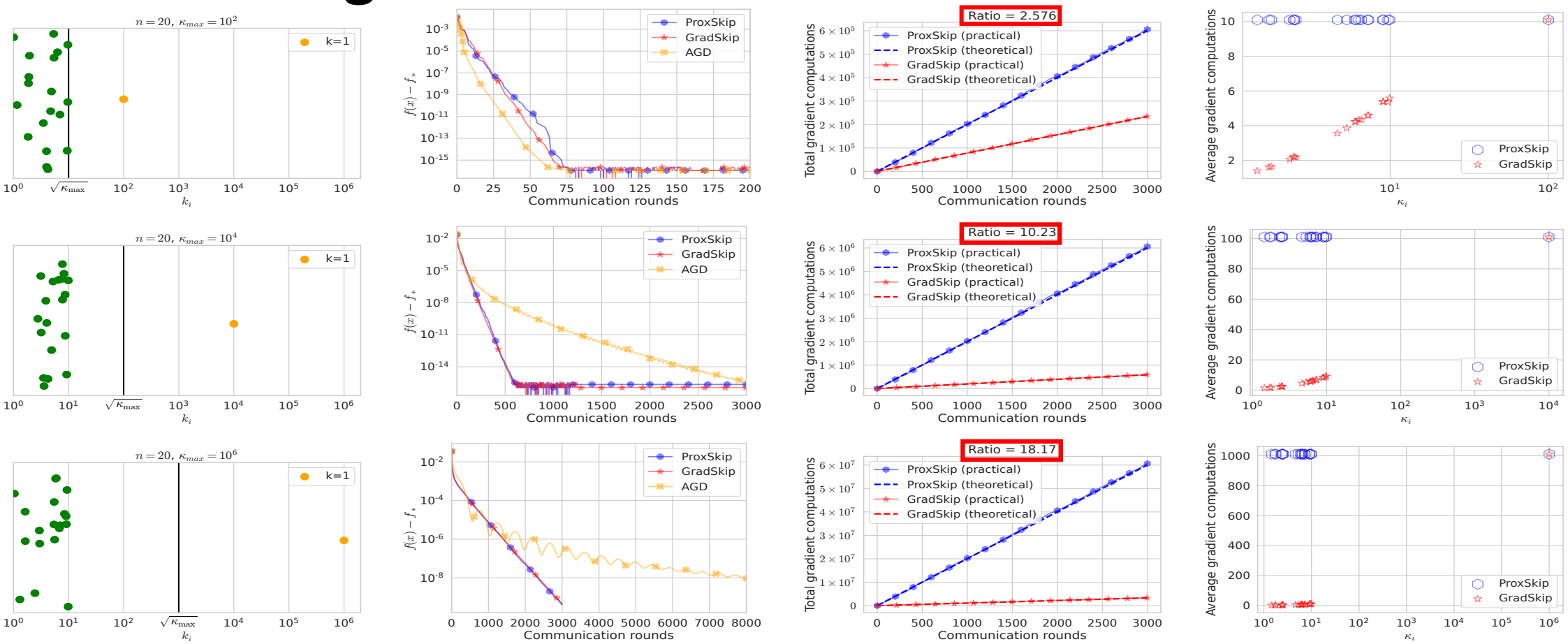


Figure 1: The first column displays the condition numbers for devices. The second column presents convergence per communication round. The third column contrasts theoretical and practical gradient computation counts. The final column reveals the average gradient computations for devices with condition number κ_i. Notably, in GradSkip, the device with $\kappa_i = \kappa_{max}$ performs gradient computations at a rate comparable to all devices in ProxSkip.

Large number of clients

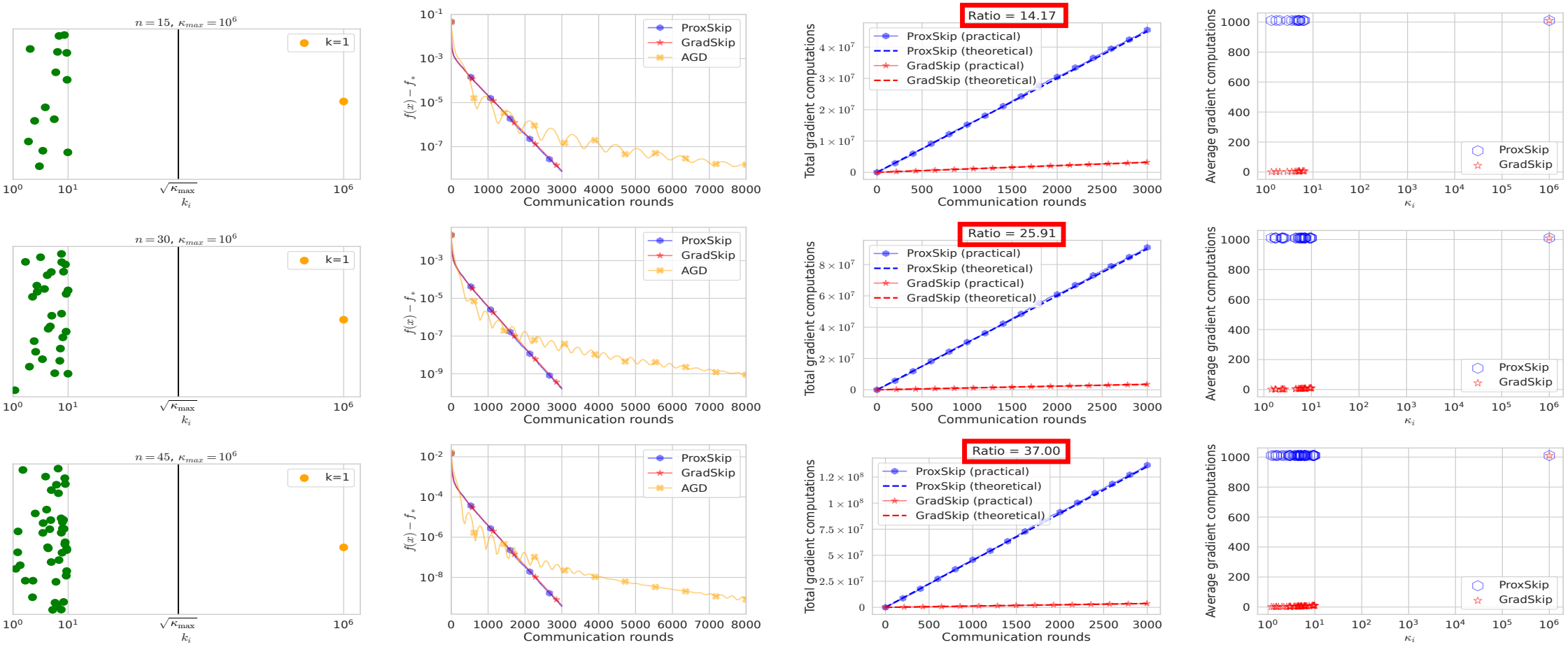


Figure 2: The columns in this figure represent the same as those in Figure 1.

Real dataset

We also do the same experiment using the *"australian"* dataset from LibSVM library (Chang and Lin, 2011). We set the regularization parameter $\lambda = 10^{-4} L_{\max}$. We split the dataset equally into $n = 20$ devices. In this case, we get $k = 8$ devices with ill-conditioned local problems, so the gradient computation ratio of ProxSkip over GradSkip should be close to $n/k = 2.5$. It can be seen in Figure 3.

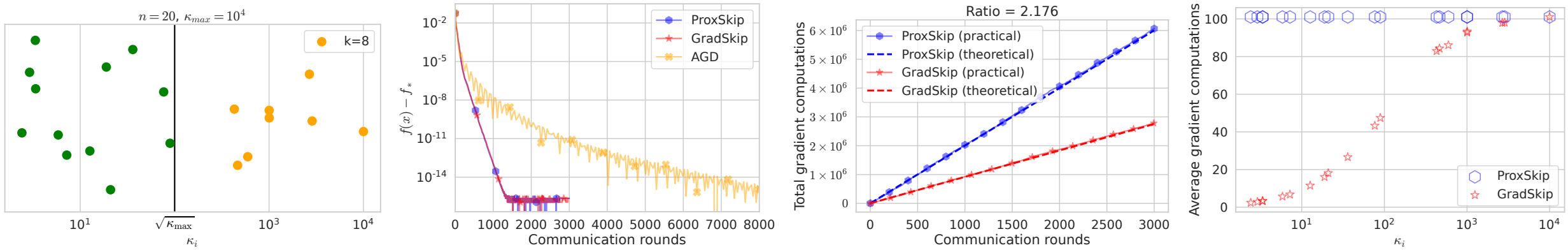


Figure 3: The plots have the same meaning as in Figure 1.

Takeaways

- Local Training is a communication acceleration technique in Federated Learning.
- ProxSkip is the first algorithm that proved the accelerated communication property.

GradSkip = ProxSkip + Heterogeneity Awareness

- GradSkip is a generalization of ProxSkip.
- GradSkip has the same communication rate as ProxSkip.
- GradSkip is a Local Training algorithm that allows devices to take a different number of local steps.
- GradSkip is the first algorithm that benefits from allowing clients to take a different number of local steps.
- GradSkip is the first Local Training algorithm with SOTA accelerated communication complexity and has reduced computational complexity.
- GradSkip can have huge benefits compared to ProxSkip in terms of computational complexity.

Future Research Directions

- GradSkip + communication compression.
- GradSkip + partial participation.
- GradSkip + differential privacy.
- GradSkip + personalization.
- GradSkip + decentralized case.
- GradSkip + asynchronous updates.
- Convex and non-convex setups.

Thank you

Appendix

The DLGD Algorithm

Algorithm DLGD (Distibuted Local Gradient Descent)

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, number of local steps $K \geq 1$, number of iterations $T \geq 1$
for $t = 0, 1, \ldots, T-1$ **do**
  in parallel on all workers $i \in [n]$ **do**
    $\hat{x}_{i,0} = x_{i,t}$ ← Receieve $x_{i,t}$ from the server
    for s = 1,2,. . . ,K **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma \nabla f_i(\hat{x}_{i,s-1})$ ← Do K local GD steps
    end for
    $x_{i,t+1} = \frac{1}{n} \sum_{j=1}^{n} \hat{x}_{j,K}$ ← Synchronize
  end local updates
end for

Theoretical problems connected with DLGD

Algorithm DLGD (Distibuted Local Gradient Descent)

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, number of local steps $K \geq 1$, number of iterations $T \geq 1$
for $t = 0, 1, \ldots, T-1$ **do**
  in parallel on all workers $i \in [n]$ **do**
    $\hat{x}_{i,0} = x_{i,t}$
    for s = 1,2,. . . ,K **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma \nabla f_i(\hat{x}_{i,s-1})$
    end for
    $x_{i,t+1} = \frac{1}{n} \sum_{j=1}^{n} \hat{x}_{j,K}$
  end local updates
end for

Inner loop

Client Drift
$\nabla f_i(x_\star) \neq 0$

Rediscovering Scaffnew (ProxSkip)

Fixing Client Drift by adding control variates

Algorithm DLGD with control variets

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, number of local steps $K \geq 1$, number of iterations $T \geq 1$
initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$,
for $t = 0, 1, \ldots, T-1$ **do**
  in parallel on all workers $i \in [n]$ **do**
    $\hat{x}_{i,0} = x_{i,t}$
    for s = 1,2,. . . , K **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma(\nabla f_i(\hat{x}_{i,s-1}) - h_{i,t})$
    end for
    $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}(\hat{x}_{j,K} - \gamma K h_{j,t})$
    $h_{i,t+1} = h_{i,t} + \frac{1}{\gamma K}(x_{i,t+1} - \hat{x}_{i,K})$
  end local updates
end for

Desire

$h_{i,t} \to \nabla f_i(x_\star)$

remove shift

update control variate

accumulated client drift

Rediscovering Scaffnew (ProxSkip)

Making loopless

Algorithm Scaffnew: Application of ProxSkip to Federated Learning

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$,
number of local steps $K \geq 1$, number of iterations $T \geq 1$
initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$,
synchonization probability p
for $t = 0, 1, \ldots, T-1$ **do**
  server: flip a coin, $\theta_t \in \{0, 1\}$, where $\mathrm{Prob}(\theta_t = 1) = p$
  in parallel on all workers $i \in [n]$ **do**
    $\hat{x}_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - h_{i,t}\right)$ ← *Local GD step*
    if $\theta_t = 1$ **then**
      $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}\left(\hat{x}_{j,t+1} - h_{j,t}\right)$ ← *synchronize*
    else
      $x_{i,t+1} = \hat{x}_{t+1}$ ← *continue doing local steps*
    end if
    $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}\left(x_{i,t+1} - \hat{x}_{i,t+1}\right)$
  end local updates
end for

Reformulated Scaffnew (ProxSkip)

Algorithm Reformulated Scaffnew

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, number of local steps $K \geq 1$, number of iterations $T \geq 1$ initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$, synchonization probability p

for $t = 0, 1, \ldots, T-1$ **do**

  Sample $K_t \sim \text{Geometric}(p)$

  in parallel on all workers $i \in [n]$ **do**

    $\hat{x}_{i,0} = x_{i,t}$

    for s $= 1,2,\ldots,K_t$ **do**

      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma\left(\nabla f_i(\hat{x}_{i,s-1}) - h_{i,t}\right)$

    end for

    $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}\left(\hat{x}_{j,K_t} - h_{j,t}\right)$

    $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}\left(x_{i,t+1} - \hat{x}_{i,t+1}\right)$

  end local updates

end for

The GradSkip Algorithm

Idea: Let workers decide how much to work

Algorithm GradSkip?

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$,
number of local steps $K \geq 1$, number of iterations $T \geq 1$
initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$,
synchonization probability p,
probabilities $q_i > 0$ controlling local steps
for $t = 0, 1, \ldots, T-1$ **do**
  Sample $K_t \sim \text{Geometric}(p)$
  in parallel on all workers $i \in [n]$ **do**
    Sample $M_{i,t} \sim \text{Geometric}(1 - q_i)$
    $M_{i,t} = \min\{M_{i,t}, K_t\}$
    $\hat{x}_{i,0} = x_{i,t}$
    for s = 1,2,...,$M_{i,t}$ **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma\left(\nabla f_i(\hat{x}_{i,s-1}) - h_{i,t}\right)$
    end for
    $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}\left(\hat{x}_{j,M_{j,t}} - h_{j,t}\right)$
    $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$
  end local updates
end for

Discovering GradSkip

Making loopless

Algorithm 1 GradSkip?

for $t = 0, 1, \ldots, T-1$ **do**
  server: flip a coin, $\theta_t \in \{0, 1\}$, where $\mathrm{Prob}(\theta_t = 1) = p$
  in parallel on all workers $i \in [n]$ **do**
    Sample $M_{i,t} \sim \mathrm{Geometric}(1 - q_i)$
    $M_{i,t} = \min\{M_{i,t}, K_t\}$
    $\hat{x}_{i,0} = x_{i,t}$
    for s $= 1,2,\ldots,M_{i,t}$ **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma\left(\nabla f_i(\hat{x}_{i,s-1}) - h_{i,t}\right)$
    end for
    if $\theta_t = 1$ **then**
      $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}\left(\hat{x}_{j,\boxed{M_{j,t}}} - h_{j,t}\right)$
    else
      $x_{i,t+1} = \hat{x}_{t+1}$
    end if
    $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$
  end local updates
end for

Use random control variate

ProxSkip: $\hat{x}_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - h_{i,t}\right)$

with probability q_i do $\hat{h}_{i,t+1} = h_{i,t}$

with probability $1 - q_i$ do $\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$

$$\hat{x}_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1}\right)$$

GradSkip: local GD steps

Algorithm 1 GradSkip

for $t = 0, 1, \ldots, T-1$ **do**
  server: flip a coin, $\theta_t \in \{0, 1\}$, where $\mathrm{Prob}(\theta_t = 1) = p$
  in parallel on all workers $i \in [n]$ **do**
    flip a coin, $\eta_{i,t} \in \{0, 1\}$, where $\mathrm{Prob}(\eta_{i,t} = 1) = 1 - q_i$
    if $\eta_{i,t} = 1$ **then**
      $\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$
    else
      $\hat{h}_{i,t+1} = h_{i,t}$
    end if
    $\hat{x}_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1})$
    if $\theta_t = 1$ **then**
      $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n} \hat{x}_{j,t+1} - \frac{\gamma}{p}\hat{h}_{j,t+1}$
    else
      $x_{i,t+1} = \hat{x}_{i,t+1}$
    end if
    $h_{i,t+1} = \hat{h}_{i,t+1} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$
  end local updates
end for

if $\eta_{i,t} = 1$ **then**
  $h_{i,t+1} = \nabla f_i(x_{i,t})$
else
  $h_{i,t+1} = h_{i,t}$
end if
$x_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - h_{i,t+1})$

$$x_{i,t+1} = x_{i,t}$$

$$h_{i,t+1} = \nabla f(x_{i,t}) = \nabla f(x_{i,t+1})$$

GradSkip

Algorithm 1 GradSkip

Input: step-size $\gamma > 0$, synchronization probabilities p, probabilities $q_i > 0$ controlling local steps, initial local iterates $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, initial shifts $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$, total number of iterations $T \geq 1$
for $t = 0, 1, \ldots, T-1$ **do**
  server: Flip a coin $\theta_t \in \{0, 1\}$ with $\mathrm{Prob}(\theta_t = 1) = p$ ◇ Decide when to skip communication
  for all devices $i \in [n]$ **in parallel do**
    Flip a coin $\eta_{i,t} \in \{0, 1\}$ with $\mathrm{Prob}(\eta_{i,t} = 1) = q_i$ ◇ Decide when to skip gradient steps (see Lemma 3.1)
    $\hat{h}_{i,t+1} = \eta_{i,t} h_{i,t} + (1 - \eta_{i,t}) \nabla f_i(x_{i,t})$ ◇ Update the local auxiliary shifts $\hat{h}_{i,t}$
    $\hat{x}_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1})$ ◇ Update the local auxiliary iterate $\hat{x}_{i,t}$ via shifted gradient step
    if $\theta_t = 1$ **then**
      $x_{i,t+1} = \frac{1}{n} \sum_{j=1}^{n} \left(\hat{x}_{j,t+1} - \frac{\gamma}{p} \hat{h}_{j,t+1} \right)$ ◇ Average shifted iterates, but only very rarely!
    else
      $x_{i,t+1} = \hat{x}_{i,t+1}$ ◇ Skip communication!
    end if
    $h_{i,t+1} = \hat{h}_{i,t+1} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$ ◇ Update the local shifts $h_{i,t}$
  end for
end for

Discovering GradSkip

Making loopless

Algorithm GradSkip?

Input: stepsize $\gamma > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$,
number of local steps $K \geq 1$, number of iterations $T \geq 1$
initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$,
synchonization probability p,
probabilities $q_i > 0$ controlling local steps
for $t = 0, 1, \ldots, T-1$ **do**
  server: flip a coin, $\theta_t \in \{0, 1\}$, where $\mathrm{Prob}(\theta_t = 1) = p$
  in parallel on all workers $i \in [n]$ **do**
    Sample $M_{i,t} \sim \mathrm{Geometric}(1 - q_i)$
    $M_{i,t} = \min\{M_{i,t}, K_t\}$
    $\hat{x}_{i,0} = x_{i,t}$
    for s $= 1, 2, \ldots, M_{i,t}$ **do**
      $\hat{x}_{i,s} = \hat{x}_{i,s-1} - \gamma\left(\nabla f_i(\hat{x}_{i,s-1}) - h_{i,t}\right)$
    end for
    if $\theta_t = 1$ **then**
      $x_{i,t+1} = \frac{1}{n}\sum_{j=1}^{n}\left(\hat{x}_{j,M_{j,t}} - h_{j,t}\right)$
    else
      $x_{i,t+1} = \hat{x}_{t+1}$
    end if
    $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$
  end local updates
end for

ProxSkip: $\hat{x}_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - h_{i,t}\right)$

with probability q_i do $\hat{h}_{i,t+1} = h_{i,t}$

with probability $1 - q_i$ do $\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$

$$\hat{x}_{i,t+1} = x_{i,t} - \gamma\left(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1}\right)$$

Stop working

$x_{i,t+1} = \hat{x}_{i,t+1} = x_{i,t}$

$$h_{i,t+1} = \hat{h}_{i,t+1} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1}) = \hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$$