FEDSKETCH: COMMUNICATION-EFFICIENT AND DIFFERENTIALLY-PRIVATE FEDERATED LEARNING VIA SKETCHING

FARZIN HADDADPOUR*, BELHAL KARIMI†, PING LI‡, AND XIAOYUN LIN§

Abstract. Communication complexity and privacy are the two key challenges in Federated Learning where the goal is to perform a distributed learning through a large volume of devices. In this work, we introduce FedSKETCH and FedSKETCHGATE algorithms to address both challenges in Federated learning jointly, where these algorithms are intended to be used for homogenous and heterogenous data distribution settings respectively. The key idea is to compress the accumulation of local gradients using count sketch, therefore, the server does not have access to the gradients themselves which provides privacy. Furthermore, due to the lower dimension of sketching used, our method exhibits communication-efficiency property as well. We provide, for the aforementioned schemes, sharp convergence guarantees. Finally, we back up our theory with various set of experiments.

Key words. Federated Learning, Compression, Sketching, Communication-efficient

1. Introduction.

Increasing applications in machine learning include the learning of a complex model across a large amount of devices in a distributed manner. In the particular case of federated learning, the training data is stored across these multiple devices and can not be centralized. Two natural problems arise from this setting. First, communications bottlenecks appear when a central server and the multiple devices must exchange gradient-informed quantities. Then, privacy-related issues due to the protection of the sensitive individual data must be taken into account.
The former has extensively been tackled via quantization [], sparsification [] and compression [] methods yielding to a drastic reduction of the number of bits required to communicate those gradient-related informations. Solving the privacy issue has been widely executed injecting an additional layer of random noise in order to respect differential-privacy property of the method.
In [6], the authors derive a single framework in order to tackle these issues jointly and introduce `DiffSketch` based on the Count Sketch operator. Compression and privacy is performed using random hash functions such that no third parties are able to access the original data.
The main contributions of this paper are summarized as follows:

- Based on the current compression methods, we provide a new algorithm – `HEAPRIX` – that displays an unbiased estimator the full gradient we ought to communicate to the central parameter server. We theoretically show that `HEAPRIX` jointly reduces the cost of communication between devices and server, preserves privacy and is unbiased.
- We develop a general algorithm for communication-efficient and privacy preserving federated learning based on this novel compression algorithm. Those methods, namely `FedSKETCH` and `FedSKETCHGATE`, are derived under *homogeneous* and *heterogeneous* data distribution settings.
- Non asymptotic analysis of our method is established for convex, strongly-convex and nonconvex functions in Theorem 5.1 and Theorem 5.5 for respec-

*Baidu Research, Seattle, USA (farzin@gmail.com).
†Baidu Research, Beijing, CN (karimibelhal@baidu.com).
‡Baidu Research, Seattle, USA (liping@baidu.com).
§Baidu Research, Seattle, USA (xiaoyun.li@rutgers.edu).

tively the i.i.d. and non i.i.d. case, and highlights an improvement in the number of iteration required to achieve a stationarity point.

Related Work for Distributed Setting: [5] develop a solution for leveraging sketches of full gradients in a distributed setting while training a global model using SGD [8, 2]. They introduce `Sketched-SGD` and establish a communication complexity of order $\mathcal{O}(\log(d))$ where d is the dimension of the parameters, i.e. the dimension of the gradient. Other recent solutions to reduce the communication cost include quantized gradient as developed in [1, 7, 9]. Yet, their dependence on the number of devices p makes them harder to be used in some settings.

Related Work for Privacy-preserving Setting: Differentially private methods for federated learning have been extensively developed and studied in the recent years.

The remaining of the paper is organized as follows. Section 2 gives a formal presentation of the general problem. Section 3 describes the various compression algorithms used for communication efficiency and privacy preservation, and introduces our new compression method. The training algorithms are provided in Section 4 and their respective analysis in the strongly-convex or nonconvex cases are provided Section 5.

Notation: For the rest of the paper we indicate the number of communication rounds and number of bits per round per device with $R(\epsilon)$ and $B(d)$ respectively. For the rest of the paper we indicate the count sketch of any vector $\boldsymbol{x}$ with $\mathbf{S}(\boldsymbol{x})$

2. Problem Setting. The federated learning optimization problem across p distributed devices is defined as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \triangleq \left[\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{p}\sum_{j=1}^{p} F_j(\boldsymbol{x})\right] \tag{2.1}$$

where $F_j(\boldsymbol{x}) = \mathbb{E}_{\xi\in\mathcal{D}_j}\left[f_j\left(\boldsymbol{x},\xi\right)\right]$ is the local cost function at device j and is (possibly) convex. ξ is a random variable with probability distribution $\mathcal{D}_j$.

3. Compression Operation. A common sketching solution employed to tackle (2.1) is called `Count Sketch` and is described Algorithm 3.1.

Algorithm 3.1 `CS`: Count Sketch to compress $\boldsymbol{x}\in\mathbb{R}^d$.

Inputs: $\boldsymbol{x}\in\mathbb{R}^d, t, k, \mathbf{S}_{t\times k}, h_j(1\le i\le t), sign_j(1\le i\le t)$
Compress vector $\boldsymbol{x}\in\mathbb{R}^d$ **into** $\mathbf{S}\left(\boldsymbol{x}\right)$**:**
for $\boldsymbol{x}_i\in\boldsymbol{x}$ **do**
  for $j=1,\cdots,t$ **do**
    $\mathbf{S}[j][h_j(i)] = \mathbf{S}[j-1][h_{j-1}(i)] + \text{sign}_j(i).\boldsymbol{x}_i$
  end for
end for
return $\mathbf{S}_{t\times k}(\boldsymbol{x})$

3.1. Unbiased Compressor.

DEFINITION 3.1 (Unbiased compressor). *A randomized function, $C:\mathbb{R}^d\to\mathbb{R}^d$ is called an unbiased compression operator with $\Delta\ge 1$, if we have*

$$\mathbb{E}\left[C(\boldsymbol{x})\right] = \boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\left[\left\|C(\boldsymbol{x})\right\|_2^2\right] \le \Delta\left\|\boldsymbol{x}\right\|_2^2$$

We indicate this class of compressor with $C\in\mathbb{U}(\Delta)$

We note that this definition leads to the property

$$\mathbb{E}\left[\|\mathrm{C}(\boldsymbol{x})-\boldsymbol{x}\|_2^2\right] \leq (\Delta-1)\|\boldsymbol{x}\|_2^2$$

Remark 3.2. Note that in case of $\Delta = 1$ our algorithm reduces for the case of no compression. This property allows us the noise of the compression.

Algorithm 3.2 PRIVIX[6]: Unbiased compressor based on sketching.

Inputs: $\boldsymbol{x} \in \mathbb{R}^d, t, k, \mathbf{S}_{t\times k}, h_j(1 \leq i \leq t), sign_j(1 \leq i \leq t)$
Query $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$ **from** $\mathbf{S}(\mathbf{x})$**:**
for $i = 1, \ldots, d$ **do**
    $\tilde{\boldsymbol{x}}[i] = \text{Median}\{\text{sign}_j(i).\mathbf{S}[j][h_j(i)] : 1 \leq j \leq t\}$
end for
Output: $\tilde{\boldsymbol{x}}$

Estimation errors:.

PROPERTY 1 ([6]). *For our proof purpose we will need the following crucial properties of the count sketch described in Algorithm 3.1, for any real valued vector* $\mathbf{x} \in \mathbb{R}^d$*:*

1) Unbiased estimation*: As it is also mentioned in [6], we have:*

$$\mathbb{E}_{\mathbf{S}}\left[\textit{PRIVIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]\right] = \mathbf{x}$$

2) Bounded variance*: With* $k = \mathcal{O}\left(\frac{e}{\mu^2}\right)$ *and* $t = \mathcal{O}\left(\ln\left(\frac{1}{\delta}\right)\right)$*, we have the following bound with probability* $1-\delta$*:*

$$\mathbb{E}_{\mathbf{S}}\left[\|\textit{PRIVIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]-\mathbf{x}\|_2^2\right] \leq \mu^2 d\|\mathbf{x}\|_2^2$$

Therefore, PRIVIX $\in \mathbb{U}(1+\mu^2 d)$ with probability $1-\delta$.

Remark 3.3. We note that $\Delta = 1+\mu^2 d$ implies that if $k \to d$, $\Delta \to 1+1 = 2$, which means that the case of no compression is not covered. Thus, the algorithms based on this may converges poorly.

Differentially Private Property:.

DEFINITION 3.4. *A randomized mechanism* $\mathcal{O}$ *satisfies* $\epsilon-$*differential privacy, if for input data* S_1 *and* S_2 *differing by up to one element, and for any output* D *of* $\mathcal{O}$*,*

$$\Pr\left[\mathcal{O}(S_1) \in D\right] \leq \exp\left(\epsilon\right)\Pr\left[\mathcal{O}(S_2) \in D\right]$$

ASSUMPTION 1 (Input vector distribution). *For the purpose of privacy analysis, similar to [?, ?], we suppose that for any input vector* S *with length* $|S| = l$*, each element* $s_i \in S$ *is drawn i.i.d. from a Gaussian distribution:* $s_i \sim \mathcal{N}(0, \sigma^2)$*, and bounded by a large probability:* $|s_i| \leq C, 1 \leq i \leq p$ *for some positive constant* $C > 0$*.*

THEOREM 3.5 ($\epsilon-$ differential privacy of count sketch, [6]). *For a sketching algorithm* $\mathcal{O}$ *using Count Sketch* $\mathbf{S}_{t\times k}$ *with* t *arrays of* k *bins, for any input vector* S *with length* l *satisfying Assumption 1,* $\mathcal{O}$ *achieves* $t.\ln\left(1+\frac{\alpha C^2 k(k-1)}{\sigma^2(l-2)}(1+\ln(l-k))\right)-$*differential privacy with high probability, where* α *is a positive constant satisfying* $\frac{\alpha C^2 k(k-1)}{\sigma^2(l-2)}(1+\ln(l-k)) \leq \frac{1}{2}-\frac{1}{\alpha}$*.*

The proof of this theorem can be found in [6].

3.2. Biased compressor.

DEFINITION 3.6 (Biased compressor). *A (randomized) function, $C: \mathbb{R}^d \to \mathbb{R}^d$ is called a compression operator with $\alpha > 0$ and $\Delta \geq 1$, if we have*

$$\mathbb{E}\left[\left\|\alpha \boldsymbol{x} - \bar{C}(\boldsymbol{x})\right\|_2^2\right] \leq \left(1 - \frac{1}{\Delta}\right) \|\boldsymbol{x}\|_2^2$$

Any biased compression operator C is indicated by $C \in \mathbb{C}(\Delta, \alpha)$.

The following Lemma links these two definitions:

LEMMA 3.7 ([4]). *We have $\mathbb{U}(\Delta) \subset \mathbb{C}(\Delta)$.*

An instance of biased compressor based on sketching is given in Algorithm 3.3.

Algorithm 3.3 `HEAVYMIX` [5]

Inputs: $\mathbf{S_g}$; parameter-k
Compress vector $\tilde{\mathbf{g}} \in \mathbb{R}^d$ into $\mathbf{S}(\tilde{\mathbf{g}})$:
Query $\hat{\ell}_2^2 = (1 \pm 0.5) \|\mathbf{g}\|^2$ from sketch $\mathbf{S_g}$
$\forall j$ query $\hat{\mathbf{g}}_j^2 = \hat{\mathbf{g}}_j^2 \pm \frac{1}{2k} \|\mathbf{g}\|^2$ from sketch $\mathbf{S_g}$
$H = \{j | \hat{\mathbf{g}}_j \geq \frac{\hat{\ell}_2^2}{k}\}$ and $NH = \{j | \hat{\mathbf{g}}_j < \frac{\hat{\ell}_2^2}{k}\}$
$\text{Top}_k = H \cup rand_\ell(NH)$, where $\ell = k - |H|$
Get exact values of Top_k
Output: $\mathbf{g}_S : \forall j \in \text{Top}_k : \mathbf{g}_{Si} = \mathbf{g}_i$ and $\forall \notin \text{Top}_k : \mathbf{g}_{Si} = 0$

LEMMA 3.8 ([5]). *`HEAVYMIX`, with sketch size $\Theta\left(k \log\left(\frac{d}{\delta}\right)\right)$ is a biased compressor with $\alpha = 1$ and $\Delta = d/k$ with probability $\geq 1 - \delta$. In other words, with probability $1 - \delta$, `HEAVYMIX` $\in C(\frac{d}{k}, 1)$.*

3.3. Sketching Based on Induced Compressor.

The following Lemma from [4] shows that how we can transfer biased compressor into an unbiased compressor:

LEMMA 3.9 (Induced Compressor [4]). *For $C_1 \in \mathbb{C}(\Delta_1)$ with $\alpha = 1$, choose $C_2 \in \mathbb{U}(\Delta_2)$ and define the induced compressor with*

$$C(\mathbf{x}) = C_1(\mathbf{x}) + C_2\left(x - C_1\left(\mathbf{x}\right)\right)$$

The induced compressor C satisfies $C \in \mathbb{U}(\mathbf{x})$ with $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1}$.

Remark 3.10. We note that if $\Delta_2 \geq 1$ and $\Delta_1 \leq 1$, we have $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1} \leq \Delta_2$

Using this concept of the induced compressor we introduce the following:

Algorithm 3.4 `HEAPRIX`

Inputs: $\boldsymbol{x} \in \mathbb{R}^d, t, k, \mathbf{S}_{t\times k}, h_j(1 \leq i \leq t), sign_j(1 \leq i \leq t)$, parameter-$k$
Approximate $\mathbf{S}(x)$ using `HEAVYMIX`
Approximate $\mathbf{S}\left(x - \texttt{HEAVYMIX}[\mathbf{S}(x)]\right)$ using `PRIVIX`
Output: $\texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right] + \texttt{PRIVIX}\left[\mathbf{S}\left(\mathbf{x} - \texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]\right)\right]$

COROLLARY 3.11. *Based on Lemma 3.9 and using Algorithm 3.4, we have $C(x) \in \mathbb{U}(\mu^2 d)$.*

Remark 3.12. We highlight that in this case if $k \to d$, then $C(x) \to x$ which means that your convergence algorithm can be improved by decreasing the noise of compression (with choice of bigger k).

In the following we define two general framework for different sketching algorithms for homogeneous and heterogeneous data distributions.

4. Algorithms for homogeneous and heterogeneous settings. In the following, first we present two algorithm for homogeneous setting. Then, we present two algorithms for heterogeneous algorithms to deal with data heterogeneity.

4.1. Homogeneous setting. In this section, we propose two algorithms for the setting where data at distributed devices is correlated. The proposed Federated Learning with averaging uses sketching to compress communication. The main difference between first algorithm and the algorithm in [6] is that we use distinct local and global learning rates. Additionally, unlike [6] we do not add add local Gaussian noise for the privacy purpose.

In `FedSKETCH`, we indicate the number of communication rounds between devices and server with R, and the number of local updates at device j is illustrated with τ, which happens between two consecutive communication rounds. Unlike [3], server node does not store any global model, instead device j has two models, $\boldsymbol{x}^{(r)}$ and $\boldsymbol{x}_j^{(\ell,r)}$. In communication round r device j, the local model $\boldsymbol{x}_j^{(\ell,r)}$ is updated using the rule

$$\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta \tilde{\mathbf{g}}_j^{(\ell,r)}, \qquad \text{for } \ell = 0, \dots, \tau - 1$$

where $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \Xi_j^{(\ell,r)}) \triangleq \frac{1}{b} \sum_{\xi \in \Xi_j^{(\ell,r)}} \nabla L_j(\boldsymbol{x}_j^{(\ell,r)}, \xi)$ is a stochastic gradient of f_j evaluated using the mini-batch $\Xi_j^{(\ell,r)} = \{\xi_{j,1}^{(\ell,r)}, \dots, \xi_{j,b_j}^{(\ell,r)}\}$ of size b_j. η is the local learning rate. After τ local updates locally, model at device j and communication round r is indicated by $\boldsymbol{x}_j^{(\tau,r)}$. The next step of our algorithm is that device j sends the count sketch $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j \left(\boldsymbol{x}_j^{(\tau,r)} - \boldsymbol{x}_j^{(0,r)} \right)$ back to the server. We highlight that

$$\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j \left(\boldsymbol{x}_j^{(\tau,r)} - \boldsymbol{x}_j^{(0,r)} \right) = \mathbf{S}_j \left(\eta \sum_{\ell=0}^{\tau-1} \tilde{\mathbf{g}}_j^{(\ell,r)} \right) = \eta \mathbf{S}_j \left(\sum_{\ell=0}^{\tau-1} \tilde{\mathbf{g}}_j^{(\ell,r)} \right),$$

which is the aggregation of the consecutive stochastic gradients multiplied with local updates η.

Upon receiving all $\mathbf{S}_j^{(r)}$ from devices, the server computes $\mathbf{S}^{(r)} = \frac{1}{p} \sum_{j=1}^{p} \mathbf{S}_j^{(r)}$ and broadcasts it to all devices. Devices after receiving $\mathbf{S}^{(r)}$ from server updates global model $\boldsymbol{x}^{(r)}$ using rule

$$\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \texttt{PRIVIX} \left[\mathbf{S}^{(r-1)} \right]$$

All these steps are summarized in `FedSKETCH` (Algorithm 4.1). A variant of this algorithm which using a different compression scheme, called `HEAPRIX` is also described in Algorithm 4.1. We note that for this variant we need to have an additional communication round between server and worker j to aggregate $\delta_j^{(r)} \triangleq \mathbf{S}_j \left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)}) \right]$. Then, server averages all $\delta_j^{(r)}$ and broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p} \sum_{j=1}^{p} \delta_j^{(r)}$ to all devices.

Remark 4.1. An important feature of our algorithm is that due to lower dimension of the count sketch, the resulting average of the sketching taken by the server ($\mathbf{S}^{(r)}$

and $\tilde{\mathbf{S}}^{(r)}$) are also of lower dimension. Therefore, these algorithms are considered as methods where exploit bidirectional compression in communication from server to device back and forth.

Algorithm 4.1 `FedSKETCH`$(R,\, \tau, \eta, \gamma)$: Private Federated Learning with Sketching.

Inputs: $\boldsymbol{x}^{(0)}$ as an initial model shared by all local devices, the number of communication rounds R, the the number of local updates τ, and global and local learning rates γ and η, respectively
for $r = 0, \ldots, R-1$ **do**
  parallel for device $j = 1, \ldots, n$ **do**:
    If PRIVIX variant:
      Computes $\boldsymbol{\Phi}^{(r-1)} \triangleq \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)}\right]$
    If HEAPRIX variant:
      Computes $\boldsymbol{\Phi}^{(r-1)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \boldsymbol{\Phi}^{(r-1)}$
    Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    for $c = 0, \ldots, \tau - 1$ **do**
      Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(c,r)})$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\, \tilde{\mathbf{g}}_j^{(\ell,r)}$
    end for
    Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$ back to the server.
  Server **computes**
    $\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1} \mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
  If HEAPRIX variant:
    Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)})\right]$
    Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p} \delta_j^{(r)}$ to devices
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

4.2. Heterogeneous setting. In the previous section, we discussed two algorithms `FedSKETCH-I` and `FedSKETCH-II`, which are originally designed for homogeneous setting where data distribution available at devices are identical.

Algorithm 4.2 `FedSKETCHGATE`(R, τ, η, γ): Private Federated Learning with Sketching and gradient tracking.

Inputs: $\boldsymbol{x}^{(0)} = \boldsymbol{x}_j^{(0)}$ as an initial model shared by all local devices, the number of communication rounds R, the the number of local updates τ, and global and local learning rates γ and η, respectively
for $r = 0, \ldots, R-1$ **do**
  parallel for device $j = 1, \ldots, n$ **do**:
    If PRIVIX variant:
      Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\texttt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \texttt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$
      Computes $\boldsymbol{\Phi} \triangleq \texttt{PRIVIX}(\mathbf{S}^{(r-1)})$
    If HEAPRIX variant:
      Computes $\boldsymbol{\Phi} \triangleq \texttt{HEAVYMIX}(\mathbf{S}^{(r-1)}) + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
      Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\boldsymbol{\Phi} - \boldsymbol{\Phi}_j\right)$
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\boldsymbol{\Phi}$
    Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    for $\ell = 0, \ldots, \tau - 1$ **do**
      Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\left(\tilde{\mathbf{g}}_j^{(\ell,r)} - \mathbf{c}_j^{(r)}\right)$
    end for
    Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$ back to the server.
  Server **computes**
    $\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1} \mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
  If HEAPRIX variant:
    Device j computes

$$\boldsymbol{\Phi}_j \triangleq \texttt{HEAVYMIX}[\mathbf{S}_j^{(r)}] + \texttt{PRIVIX}\left[\mathbf{S}_j^{(r)} - \mathbf{S}_j{}^{(r)}\left(\texttt{HEAVYMIX}[\mathbf{S}_j{}^{(r)}]\right)\right]$$

    Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left(\texttt{HEAVYMIX}[\mathbf{S}^{(r)}]\right)$
    Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p} \delta_j^{(r)}$ to devices
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

5. Convergence Analysis. The following assumptions are required for our analysis:

ASSUMPTION 2 (Smoothness and Lower Boundedness). *The local objective function $f_j(\cdot)$ of jth device is differentiable for $j \in [m]$ and L-smooth, i.e., $\|\nabla f_j(\boldsymbol{u}) - \nabla f_j(\mathbf{v})\| \leq L\|\boldsymbol{u} - \mathbf{v}\|$, $\forall\ \boldsymbol{u}, \mathbf{v} \in \mathbb{R}^d$. Moreover, the optimal objective function $f(\cdot)$ is bounded below by $f^* = \min_{\boldsymbol{x}} f(\boldsymbol{x}) > -\infty$.*

ASSUMPTION 3 (Polyak-Lojasiewicz (PL)). *A function f satisfies the PL conditon with constant μ if $\frac{1}{2}\|\nabla f(\boldsymbol{x})\|_2^2 \geq \mu\big(f(\boldsymbol{x}) - f(\boldsymbol{x}^*)\big)$, $\forall \boldsymbol{x} \in \mathbb{R}^d$ with $\boldsymbol{x}^*$ is an optimal solution.*

5.1. Convergence of `FEDSKETCH` for homogeneous setting. Now we focus on the homogeneous case in which the stochastic local gradient of each worker is an

unbiased estimator of the global gradient.

ASSUMPTION 4 (Bounded Variance). *For all $j \in [m]$, we can sample an independent mini-batch ℓ_j of size $|\Xi_j^{(\ell,r)}| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w}; \Xi_j), \mathbb{E}_{\xi_j}[\tilde{\mathbf{g}}_j] = \nabla f(\boldsymbol{w}) = \mathbf{g}$ with the variance bounded is bounded by a constant σ^2, i.e., $\mathbb{E}_{\Xi_j}\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}\|^2\right] \leq \sigma^2$.*

THEOREM 5.1. *Suppose that the conditions in Assumptions 2-4 hold. Given $0 < k = \mathcal{O}\left(\frac{e}{\mu^2}\right) \leq d$, and Consider `FedSKETCH` in Algorithm 4.1 with sketch size $B = \mathcal{O}\left(k \log\left(\frac{dR}{\delta}\right)\right)$. If the local data distributions of all users are identical (homogeneous setting), then with probability $1-\delta$ we have*

- ***Nonconvex:***

`PRIVIX` *Set $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau\left(\frac{\mu^2 d}{p}+1\right)}}$ and $\gamma \geq m$, the sequence of iterates satisfies*

$$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\frac{1}{\epsilon}\right) \text{ and } \tau = \mathcal{O}\left(\frac{\frac{\mu^2 d}{p}+1}{p\epsilon}\right).$$

`HEAPRIX` *Set $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau\left(\frac{\mu^2 d-1}{p}+1\right)}}$ and $\gamma \geq m$, the sequence of iterates satisfies*

$$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\frac{1}{\epsilon}\right) \text{ and } \tau = \mathcal{O}\left(\frac{\frac{\mu^2 d-1}{p}+1}{p\epsilon}\right).$$

- ***Strongly convex or PL:***

`PRIVIX` *Set $\eta = \frac{1}{2L\left(\frac{\mu^2 d}{p}+1\right)\tau\gamma}$ and $\gamma \geq m$, we obtain that the iterates satisfy*

$$\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\left(\frac{\mu^2 d}{p}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right) \text{ and } \tau = \mathcal{O}\left(\frac{1}{m\epsilon}\right).$$

`HEAPRIX` *Set $\eta = \frac{1}{2L\left(\frac{\mu^2 d-1}{p}+1\right)\tau\gamma}$ and $\gamma \geq m$, we obtain that the iterates satisfy*

$$\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\left(\frac{\mu^2 d-1}{p}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right) \text{ and } \tau = \mathcal{O}\left(\frac{1}{m\epsilon}\right).$$

- ***Convex:***

`PRIVIX` *Set $\eta = \frac{1}{2L\left(\frac{\mu^2 d}{p}+1\right)\tau\gamma}$ and $\gamma \geq m$, we obtain that the iterates satisfy*

$$\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\frac{L\left(1+\frac{\mu^2 d}{p}\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right) \text{ and } \tau = \mathcal{O}\left(\frac{1}{m\epsilon^2}\right).$$

`HEAPRIX` *Set $\eta = \frac{1}{2L\left(\frac{\mu^2 d-1}{p}+1\right)\tau\gamma}$ and $\gamma \geq m$, we obtain that the iterates satisfy*

$$\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon \text{ if we set } R = \mathcal{O}\left(\frac{L\left(\frac{\mu^2 d-1}{p}+1\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right) \text{ and } \tau = \mathcal{O}\left(\frac{1}{m\epsilon^2}\right).$$

Several auxiliary results regarding communication cost can be derived as follows:

COROLLARY 5.2 (Total communication cost). *As a consequence of Remark **??**, the total communication cost per-worker becomes*

$$\mathcal{O}(RB) = \mathcal{O}\left(Rk\log\left(\frac{dR}{\delta}\right)\right) = \mathcal{O}\left(\frac{k}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)$$

*We note that this result in addition to improving over the communication complexity of federated learning of the state-of-the-art from $\mathcal{O}\left(\frac{d}{\epsilon}\right)$ in [**?**, **?**, **?**] to $\mathcal{O}\left(\frac{kp}{\epsilon}\log\left(\frac{dp}{\epsilon\delta}\right)\right)$,*

it also implies differential privacy. As a result, total communication cost is

$$BpR = \mathcal{O}\left(\frac{kp}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right).$$

We note that the state-of-the-art in [?] the total communication cost is

$$BpR = \mathcal{O}\left(pd\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(\frac{pd}{\epsilon}\right)$$

Thus, we improve this result, in terms of dependency to d, from pd to $p\log(d)$. In comparison to [5], we improve the total communication per worker from $RB = \mathcal{O}\left(\frac{k}{\epsilon^2}\log\left(\frac{d}{\epsilon^2\delta}\right)\right)$ to $RB = \mathcal{O}\left(\frac{k}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)$.

Remark 5.3. It is worthy to note that most of the available communication-efficient algorithm with quantization or compression only consider communication-efficiency from devices to server. However, Algorithm 4.1 also improves the communication efficiency from server to devices as well.

COROLLARY 5.4 (Total communication cost for PL or strongly convex). *To achieve the convergence error of ϵ, we need to have $R = \mathcal{O}\left(\kappa(\frac{\mu^2 d}{p}+1)\log\frac{1}{\epsilon}\right)$ and $\tau = \left(\frac{1}{\epsilon}\right)$. This leads to the total communication cost per worker of*

$$BR = \mathcal{O}\left(k\kappa(\frac{\mu^2 d}{p}+1)\log\left(\frac{\kappa(\frac{\mu^2 d^2}{p}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right)$$

As a consequence, the total communication cost becomes:

$$BpR = \mathcal{O}\left(k\kappa(\mu^2 d+p)\log\left(\frac{\kappa(\frac{\mu^2 d^2}{p}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right)$$

We note that the state-of-the-art in [?] the total communication cost is

$$BpR = \mathcal{O}\left(\kappa pd\log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(\kappa pd\log\left(\frac{1}{\epsilon}\right)\right)$$

We improve this result, in terms of dependency to d, to

$$BpR = \mathcal{O}\left(k\kappa(\mu^2 d+p)\log\left(\frac{\kappa(\frac{\mu^2 d}{p}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right)$$

Improving from pd to $p+d$.

5.2. Convergence of FedSKETCHGATE in data heterogeneous setting.

ASSUMPTION 5 (Bounded Local Variance). *For all $j \in [m]$, we can sample an independent mini-batch Ξ_j of size $|\xi_j| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w};\Xi_j), \mathbb{E}_\xi[\tilde{\mathbf{g}}_j] = \nabla f_j(\boldsymbol{w}) = \mathbf{g}_j$. Moreover, the variance of local stochastic gradients is bounded above by a constant σ^2, i.e., $\mathbb{E}_\Xi\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}_j\|^2\right] \le \sigma^2$.*

THEOREM 5.5. *Suppose that the conditions in Assumptions 2 and 5 hold. Given $0 < k = \mathcal{O}\left(\frac{e}{\mu^2}\right) \le d$, and Consider FedSKETCHGATE in Algorithm ?? with sketch size $B = \mathcal{O}\left(k\log\left(\frac{dR}{\delta}\right)\right)$. If the local data distributions of all users are identical (homogeneous setting), then with probability $1-\delta$ we have*

- ***Nonconvex:***

  `PRIVIX` *Set* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\mu^2 d)}}$ *and* $\gamma \geq m$, *the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\frac{\mu^2 d+1}{\epsilon}\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{p\epsilon}\right)$.

  `HEAPRIX` *Set* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\mu^2 d)}}$ *and* $\gamma \geq m$, *the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\frac{\mu^2 d}{\epsilon}\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{p\epsilon}\right)$.

- ***PL or Strongly convex:***

  `PRIVIX` *Set* $\eta = \frac{1}{2L(\mu^2 d+1)\tau\gamma}$ *and* $\gamma \geq m$, *we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\left(\mu^2 d+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{m\epsilon}\right)$.

  `HEAPRIX` *Set* $\eta = \frac{1}{2L(\mu^2 d)\tau\gamma}$ *and* $\gamma \geq m$, *we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\left(\mu^2 d\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{m\epsilon}\right)$.

- ***Convex:***

  `PRIVIX` *Set* $\eta = \frac{1}{2L(\mu^2 d+1)\tau\gamma}$ *and* $\gamma \geq m$, *we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\frac{L\left(1+\mu^2 d\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{m\epsilon^2}\right)$.

  `HEAPRIX` *Set* $\eta = \frac{1}{2L(\mu^2 d)\tau\gamma}$ *and* $\gamma \geq m$, *we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\frac{L\left(\mu^2 d\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{m\epsilon^2}\right)$.

6. Conclusions. Conclude

Appendix A. Proofs.

REFERENCES

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

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

[3] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, *Federated learning with compression: Unified analysis and sharp guarantees*, arXiv preprint arXiv:2007.01154, (2020).

[4] S. Horváth and P. Richtárik, *A better alternative to error feedback for communication-efficient distributed learning*, arXiv preprint arXiv:2006.11077, (2020).

[5] N. Ivkin, D. Rothchild, E. Ullah, I. Stoica, R. Arora, et al., *Communication-efficient distributed sgd with sketching*, in Advances in Neural Information Processing Systems, 2019, pp. 13144–13154.

[6] T. Li, Z. Liu, V. Sekar, and V. Smith, *Privacy for free: Communication-efficient learning with differential privacy using sketches*, arXiv preprint arXiv:1911.00972, (2019).

[7] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, *Deep gradient compression: Reducing the communication bandwidth for distributed training*, arXiv preprint arXiv:1712.01887, (2017).

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

[9] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, *Sparsified sgd with memory*, in Advances in Neural Information Processing Systems, 2018, pp. 4447–4458.