FedSketch: Communication-Efficient and Private Federated Learning via Sketching

Farzin Haddadpour, Belhal. Karimi, Ping Li, Xiaoyun Li

August 3, 2020

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 homogeneous and heterogeneous 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.

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 [1], sparsification [18] and compression [3] 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.

With the focus of communication-efficiency, [8] proposes a distributed SGD algorithm using sketching and they provide the convergence analysis in homogeneous data distribution setting.

Also with focus on privacy, in [11], 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. Yet, [11] does not provide the convergence analysis for the `DiffSketch` in Federated setting. In this work, we provide a thorough convergence analysis for the Federated Learning using sketching.

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 of 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, Polyak-Łojasiewicz (generalization of strongly-convex) and nonconvex functions in Theorem 2 and Theorem 3 for respectively the i.i.d. and non i.i.d. case, and highlight an improvement in the number of iteration required to achieve a stationarity point.

Related Work for Communication-efficient Distributed Setting: [8] develop a solution for leveraging sketches of full gradients in a distributed setting while training a global model using SGD [15, 4]. 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, 13, 16]. Yet, their dependence on the number of devices p makes them harder to be used in some settings. Additionally, there are other research efforts such as [6, 14, 2] that exploit compression in Federated Learning. Finally, the recent work in [7] exploits variance reduction technique with compression jointly in distributed optimization.

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:

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

where $F_j(\boldsymbol{x}) = \mathbb{E}_{\xi\in\mathcal{D}_j}\left[L_j\left(\boldsymbol{x},\xi\right)\right]$ is the local cost function at device j. ξ is a random variable with probability distribution $\mathcal{D}_j$, and L_j is a loss function that measures the performance of model $\boldsymbol{x}$. We note that, while for the homogeneous data distribution, we assume $\mathcal{D}_j$ for $1 \leq j \leq p$ have the same distribution and $L_1 = L_2 = \ldots = L_p$, in the heterogeneous setting these data distributions and loss functions L_j can be different from device to device.

3 Compression Operation

A common sketching solution employed to tackle (1) called `Count Sketch` (for more detail see the seminal works [?, 5, 10]) is described Algorithm 1.

Algorithm 1 `CS` [10]: Count Sketch to compress $\boldsymbol{x}\in\mathbb{R}^d$.

```
Inputs: $\boldsymbol{x}\in\mathbb{R}^d, t, k, \mathbf{S}_{t\times k}, h_j(1\leq i\leq t), sign_j(1\leq i\leq t)$
Compress vector $\boldsymbol{x}\in\mathbb{R}^d$ into $\mathbf{S}(\boldsymbol{x})$:
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 1 (Unbiased compressor)**.** *A randomized function, $C:\mathbb{R}^d\to\mathbb{R}^d$ is called an unbiased compression operator with $\Delta\geq 1$, if we have*

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

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

We note that this definition leads to the property

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

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

Algorithm 2 `PRIVIX`[11]: 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: $\tilde{\boldsymbol{x}}$
```

Property 1 ([11])**.** *For our proof purpose we will need the following crucial properties of the count sketch described in Algorithm 1, for any real valued vector* $\mathbf{x} \in \mathbb{R}^d$*:*

1) Unbiased estimation*: As it is also mentioned in [11], we have:*

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

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* $1-\delta$*:*

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

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

Remark 2. *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.*

Definition 2. *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 3, 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 1 ($\epsilon-$ differential privacy of count sketch, [11])**.** *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 [11].

3.2 Biased compressor

Definition 3 (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*

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

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

The following Lemma links these two definitions:

Lemma 1 ([7])**.** *We have* $\mathbb{U}(\Delta) \subset \mathbb{C}(\Delta)$*.*

An instance of biased compressor based on sketching is given in Algorithm 3.

Algorithm 3 `HEAVYMIX` [8]

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

Lemma 2 ([8])**.** *`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 [7] shows that how we can transfer biased compressor into an unbiased compressor:

Lemma 3 (Induced Compressor [7])**.** *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*

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

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

Remark 3. *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 `HEAPRIX`:

Algorithm 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 1. *Based on Lemma 3 and using Algorithm 4, we have $C(x) \in \mathbb{U}(\mu^2 d)$.*

Remark 4. *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 [11] is that we use distinct local and global learning rates. Additionally, unlike [11] 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 [6], 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

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

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 $\Xi_j^{(\ell,r)} = \{\xi_{j,1}^{(\ell,r)}, \ldots, \xi_{j,b_j}^{(\ell,r)}\}$ 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

$$\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1}^{p}\mathbf{S}_j^{(r)} \tag{2}$$

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 5). A variant of this algorithm which uses a different compression scheme, called `HEAPRIX` is also described in Algorithm 5. 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 to all devices the following quantity:

$$\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p}\delta_j^{(r)}. \tag{3}$$

Upon receiving $\tilde{\mathbf{S}}^{(r)}$ all devices compute

$$\mathbf{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r)} - \tilde{\mathbf{S}}^{(r)}\right] \tag{4}$$

and then updates his global model using $\boldsymbol{x}^{(r+!)} = \boldsymbol{x}^{(r)} - \gamma\mathbf{\Phi}^{(r)}$.

Remark 5 (Improvement over [6]). *An important feature of our algorithm is that due to lower dimension of the count sketch, the resulting averages ($\mathbf{S}^{(r)}$ and $\tilde{\mathbf{S}}^{(r)}$) taken by the server, are also of lower dimension. Therefore, these algorithms exploit bidirectional compression in communication from server to device back and forth. As a result, due to this bidirectional property of communicating sketching for the case of large quantiziation error shown by $q = \theta(\frac{d}{k})$ in [6], our algorithms outperform `FedCom` algorithm in [6]. Furthermore, sketching-based server-devices communication algorithm such as ours also provides privacy as a by-product.*

Algorithm 5 `FedSKETCH`(R, τ, η, γ): 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 number of local updates $\tau$, and global and local learning rates $\gamma$ and $\eta$, respectively
for $r = 0, \ldots, R-1$ do
    parallel for device $j = 1, \ldots, n$ do:
        if PRIVIX variant:
            Computes $\mathbf{\Phi}^{(r)} \triangleq \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)}\right]$
        if HEAPRIX variant:
            Computes $\mathbf{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
        Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\mathbf{\Phi}^{(r)}$
        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 $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
            $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\, \tilde{\mathbf{g}}_j^{(\ell,r)}$
        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 algorithm `FedSKETCH`, which is originally designed for homogeneous setting where data distribution available at devices are identical. However, in a heterogeneous setting where data distribution could be different, the aforementioned algorithms may fail to perform well in practice. The main reason to cause this issue is that in Federated learning devices are using local stochastic descent direction which could be different than global descent direction when the data distribution are non-identical. Therefore, to mitigate the effect of data heterogeneity, we introduce new algorithm `FedSKETCHGATE` based on sketching. This algorithm uses the idea of gradient tracking introduced in [6] (with compression) and a variation in [12] (without compression). The main idea is that using an approximation of global gradient, $\mathbf{c}_j^{(r)}$, we correct the local gradient direction. For the `FedSKETCH GATE` with `PRIVIX` variant, the correction vector $\mathbf{c}_j^{(r)}$ at device j and communication round r is computed using the update rule $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\texttt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \texttt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$ where $\mathbf{S}_j^{(r-1)} \triangleq \mathbf{S}\left(\boldsymbol{x}_j^{(0,r-1)} - \boldsymbol{x}_j^{(\tau,r-1)}\right)$ is computed and stored at device j from previous communication round $r-1$. The term $\mathbf{S}^{(r-1)}$ is computed similar to `FedSKETCH` in (2). For `FedSKETCHGATE`, the server needs to compute $\tilde{\mathbf{S}}^{(r)}$ using (3). Then, device j computes $\mathbf{\Phi}_j \triangleq \texttt{HEAPRIX}[\mathbf{S}_j^{(r)}]$ and $\mathbf{\Phi} \triangleq \texttt{HEAPRIX}(\mathbf{S}^{(r-1)})$ and updates the correction vector $\mathbf{c}_j^{(r)}$ using the recursion $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}(\mathbf{\Phi} - \mathbf{\Phi}_j)$.

Algorithm 6 FedSKETCHGATE(R, τ, η, γ): Private Federated Learning with Sketching and gradient tracking.

```
Inputs: $\boldsymbol{x}^{(0)} = \boldsymbol{x}_j^{(0)}$ shared by all local devices, communication rounds $R$, local updates $\tau$, global and local learning rates $\gamma$ and $\eta$.
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 $\mathbf{\Phi}^{(r)} \triangleq \texttt{PRIVIX}(\mathbf{S}^{(r-1)})$
        if HEAPRIX variant:
            Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\mathbf{\Phi}^{(r)} - \mathbf{\Phi}_j^{(r)}\right)$
            Computes $\mathbf{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
        Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \mathbf{\Phi}^{(r)}$ and $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
        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 $\mathbf{\Phi}_j^{(r)} \triangleq \texttt{HEAPRIX}[\mathbf{S}_j^{(r)}]$
            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 2. *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 5 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* $\gamma \geq p$*, 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* $R = \mathcal{O}\left(\frac{1}{\epsilon}\right)$ *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*

$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\frac{1}{\epsilon}\right)$ *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* $\gamma \geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = \mathcal{O}\left(\left(\frac{\mu^2 d}{p}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \mathcal{O}\left(\frac{1}{p\epsilon}\right)$.

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

- ***Convex:***

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

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

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

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

Corollary 2 (Total communication cost)**.** *The total communication cost per-worker becomes*

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

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 [9, 12, 17] 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 [9] 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 [8], we improve the total communication per worker from* $\mathcal{O}\left(\frac{k}{\epsilon^2}\log\left(\frac{d}{\epsilon^2\delta}\right)\right)$ *to* $\mathcal{O}\left(\frac{k}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)$.

Remark 6. *It is worth noting that most of the available communication-efficient algorithm with quantization or compression only consider communication-efficiency from devices to server. However, Algorithm 5 also improves the communication efficiency from server to devices as well.*

Corollary 3 (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 of ***`FedSKETCH`***, *Alg.5, 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 [9] 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, 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 3. *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 6 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 $\gamma \ge m$, the sequence of iterates satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \le \epsilon$ 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 $\gamma \ge m$, the sequence of iterates satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \le \epsilon$ 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 $\gamma \ge m$, we obtain that the iterates satisfy $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \le \epsilon$ if we set $R = \mathcal{O}\left(\left(\mu^2 d + 1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ and $\tau = \mathcal{O}\left(\frac{1}{p\epsilon}\right)$.

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

- ***Convex:***

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

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

Remark 7. *We note that to the best of our knowledge, only [11] provides the convergence analysis for heterogeneous data distribution for distributed SGD. Therefore, Theorem 3 is the first to provide the convergence analysis of Federated Learning using sketching for heterogeneous setting.*

6 Numerical Applications

7 Experiments

In this section, we provide empirical results on MNIST dataset to demonstrate the effectiveness of our proposed algorithms. The model we use is the LeNet [ ] CNN architecture, with 60,000 model parameters in total.

Four methods are compared in our experiments: Federated SGD (FedSGD), SketchSGD [8], FedSketch-PRIVIX (FS-PRIVIX for short) and FedSketch-HEAPRIX (FS-HEAPRIX). We implement the algorithms in Pytorch by simulating the distributed and federated environment. Note that in Algorithm 5, FS-PRIVIX with global learning rate $\gamma = 1$ is equivalent to the DiffSketch approach proposed in [**?**]. In all experiments, we set the number of workers to 50. For federated learning algorithms, we use different number of local updates τ. For SketchedSGD which is under synchronous distributed learning framework, τ is fixed as 1. For all methods, we tune the learning rates (both local and global, if applicable) over the log-scale and report the best results.

In each round of local update, we randomly choose half of the local devices to be active, which is more practical in real-world applications. For the data distribution on each device, we test both homogeneous and heterogeneous setting. The the first case, each device receives uniformly chosen data samples (each class has equal probability). In the later case, every device only receives samples from one or two classes among ten digits in the MNIST dataset. Since data is not distributed i.i.d. among local devices, training is expected to be harder in the heterogeneous case.

Homogeneous case. In Figure **??** we provide the training loss and test accuracy of four algorithms with $\tau = 1$ (since SketchSGD requires single local update per round). We also test different sizes of sketching matrix, $(t, k) = (20, 40)$ and $(50, 100)$. Note that these two choices of sketch size correspond to a 75× and 12× compression ratio, respectively. In general, as one would expect, higher compression ratio leads to worse learning performance. In both cases, FS-HEAPRIX performs the best in terms of both training objective and test accuracy. FS-PRIVIX is better when sketch size is large (i.e. when the estimation from sketches are more accurate), while SketchSGD performs better with small sketch size.

The results for multiple local updates are given in Figure **??**, where we set $\tau = 2, 5$. We see that FS-HEAPRIX is significantly better than FS-PRIVIX, either with small or large sketching matrix. In both cases, FS-HEAPRIX yields acceptable extra test error compared to FedSGD, especially taking the high compression ratio (e.g. 75×) into consideration. However, FS-PRIVIX performs poorly with small sketch size $(20, 40)$, and even diverges with $\tau = 5$. Another observation is that, the performance of FS-HEAPRIX improves with increasing number of local updates. That is, the proposed method is able to further reduce the communication cost by reducing the number of rounds required for communication. This is also consistent with our theoretical claims established in this paper.

8 Conclusion

In this paper, we introduced `FedSKETCH` and `FedSKETCHGATE` algorithms for homogeneous and heterogeneous data distribution setting respectively for Federated Learning wherein communication between server and devices is only performed using count sketch. Our algorithms, thus, provide communication-efficiency and privacy. We analyze the convergence error for *non-convex*, *Polyak-Łojasiewicz* and *general convex* objective functions in the scope of Federated Optimization.

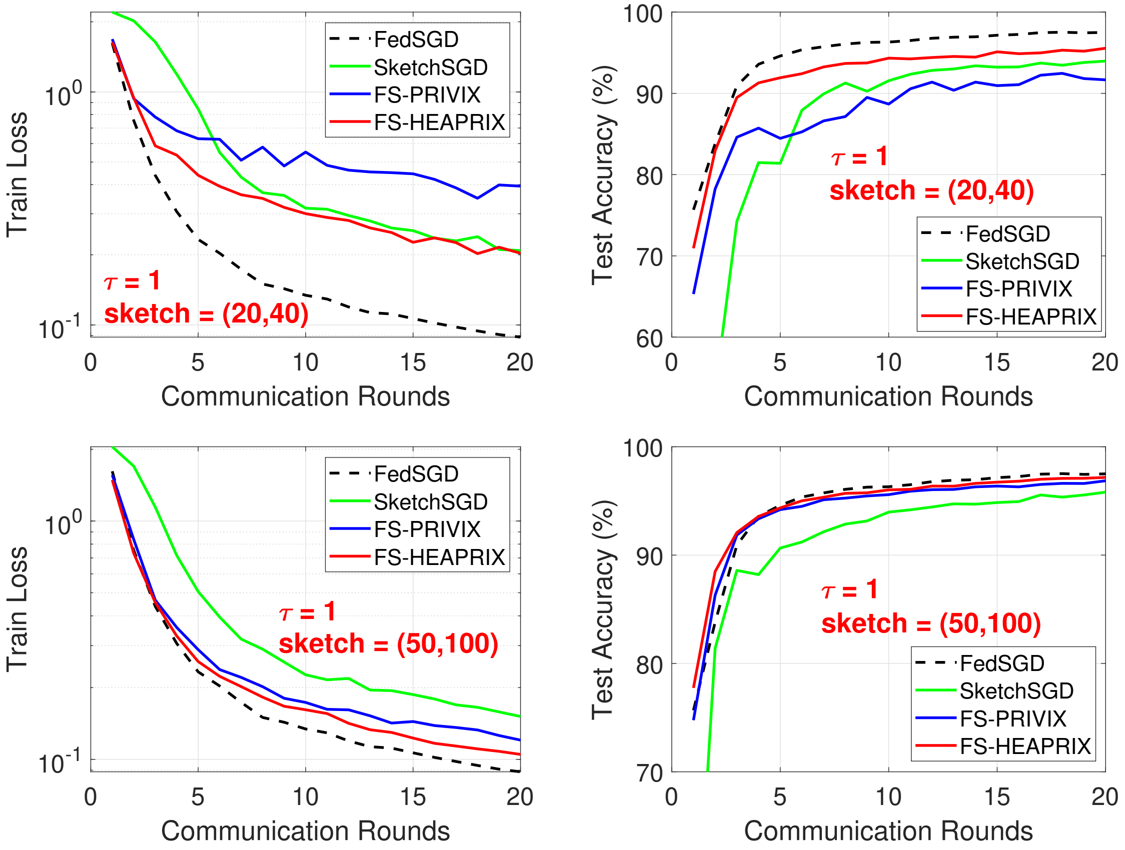

Figure 1: The comparison of four algorithms on LeNet CNN architecture.

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] D. Basu, D. Data, C. Karakus, and S. Diggavi, *Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations*, in Advances in Neural Information Processing Systems, 2019, pp. 14695–14706.

[3] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, *signsgd: Compressed optimisation for non-convex problems*, arXiv preprint arXiv:1802.04434, (2018).

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

[5] G. Cormode and S. Muthukrishnan, *An improved data stream summary: the count-min sketch and its applications*, Journal of Algorithms, 55 (2005), pp. 58–75.

[6] 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).

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

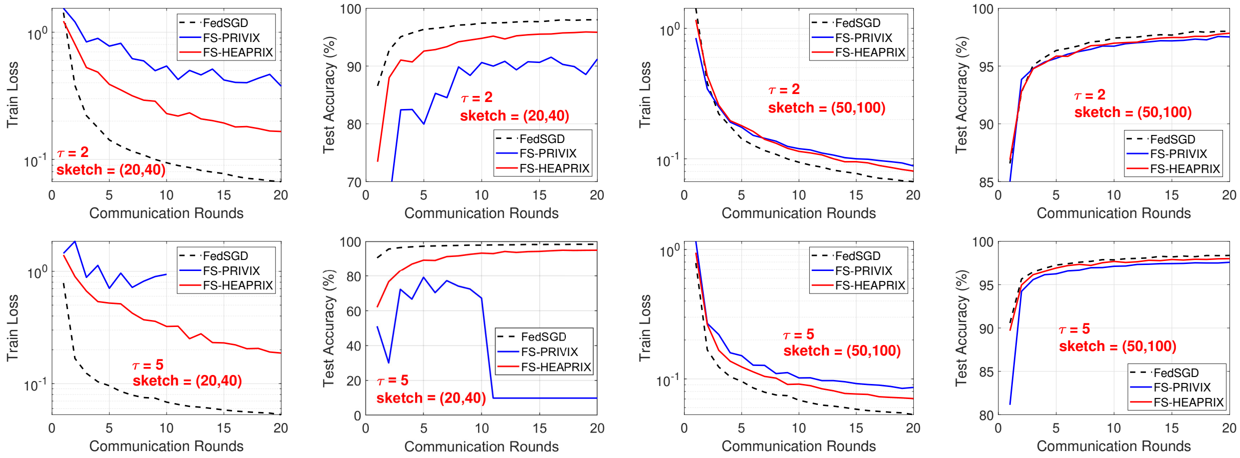

Figure 2: Comparison of FedSGD, FS-PRIVIX and FS-HEAPRIX on LeNet CNN architecture, with number of local updates being 2 and 5.

[8] 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.

[9] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, *Scaffold: Stochastic controlled averaging for on-device federated learning*, arXiv preprint arXiv:1910.06378, (2019).

[10] J. Kleinberg, *Bursty and hierarchical structure in streams*, Data Mining and Knowledge Discovery, 7 (2003), pp. 373–397.

[11] 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).

[12] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng, *Variance reduced local sgd with lower communication complexity*, arXiv preprint arXiv:1912.12844, (2019).

[13] 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).

[14] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, *Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization*, arXiv preprint arXiv:1909.13014, (2019).

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

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

[17] J. Wang and G. Joshi, *Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms*, arXiv preprint arXiv:1808.07576, (2018).

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

A Proof of main Theorems

The proof of Theorem 2 follows directly from the results in [6]. For the sake of the completeness we review an assumptions from this reference for the quantiziation with their notation.

Assumption 6 ([6])**.** *The output of the compression operator $Q(\mathbf{x})$ is an unbiased estimator of its input $\mathbf{x}$, and its variance grows with the squared of the squared of ℓ_2-norm of its argument, i.e., $\mathbb{E}\left[Q(\mathbf{x})\right] = \mathbf{x}$ and $\mathbb{E}\left[\left\|Q(\mathbf{x}) - \mathbf{x}\right\|^2\right] \leq q \left\|\mathbf{x}\right\|^2$.*

A.1 Proof of Theorem 2

Based on Assumption 6 we have:

Theorem 4 ([6])**.** *Consider `FedCOM` in [6]. Suppose that the conditions in Assumptions 2, 4 and 6 hold. If the local data distributions of all users are identical (homogeneous setting), then we have*

- ***Nonconvex:*** *By choosing stepsizes as $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau\left(\frac{q}{p}+1\right)}}$ and $\gamma \geq p$, 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 $R = O\left(\frac{1}{\epsilon}\right)$ and $\tau = O\left(\frac{\frac{q}{p}+1}{p\epsilon}\right)$.*
- ***Strongly convex or PL:*** *By choosing stepsizes as $\eta = \frac{1}{2L\left(\frac{q}{p}+1\right)\tau\gamma}$ 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 $R = O\left(\left(\frac{q}{p}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ and $\tau = O\left(\frac{1}{p\epsilon}\right)$.*
- ***Convex:*** *By choosing stepsizes as $\eta = \frac{1}{2L\left(\frac{q}{p}+1\right)\tau\gamma}$ and $\gamma \geq p$, we obtain that the iterates satisfy $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ if we set $R = O\left(\frac{L\left(1+\frac{q}{p}\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ and $\tau = O\left(\frac{1}{p\epsilon^2}\right)$.*

Proof. Since the sketching `PRIVIX` and `HEAPRIX`, satisfy the Assumption 6 with $q = \mu^2 d$ and $q = \mu^2 d - 1$ respectively with probablity $1-\delta$. Therefore, all the results in Theorem 2, conclude from Theorem 4 with probability $1-\delta$ and plugging $q = \mu^2 d$ and $q = \mu^2 d - 1$ respectively into the corresponding convergence bounds. □

A.2 Proof of Theorem 3

For the heterogeneous setting, the results in [6] requires the following extra assumption that naturally holds for the sketching:

Assumption 7 ([6])**.** *The compression scheme Q for the heterogeneous data distribution setting satisfies the following condition*

$$\mathbb{E}_Q[\|\frac{1}{m}\sum_{j=1}^{m} Q(\boldsymbol{x}_j)\|^2 - \|Q(\frac{1}{m}\sum_{j=1}^{m}\boldsymbol{x}_j)\|^2] \leq G_q\,.$$

We note that since sketching is a linear compressor, in the case of our algorithms for heterogeneous setting we have $G_q = 0$.
Next, we restate the Theorem in [6] here as follows:

Theorem 5. *Consider `FedCOMGATE` in [6]. If Assumptions 2, 5, 6 and 7 hold, then even for the case the local data distribution of users are different (heterogeneous setting) we have*

- ***Non-convex:*** *By choosing stepsizes as $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(q+1)}}$ and $\gamma \geq p$, we obtain that the iterates satsify $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ if we set $R = O\left(\frac{q+1}{\epsilon}\right)$ and $\tau = O\left(\frac{1}{p\epsilon}\right)$.*

- ***Strongly convex or PL:*** *By choosing stepsizes as* $\eta = \frac{1}{2L\left(\frac{q}{p}+1\right)\tau\gamma}$ *and* $\gamma \geq \sqrt{p\tau}$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = O\left((q+1)\,\kappa \log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.
- ***Convex:*** *By choosing stepsizes as* $\eta = \frac{1}{2L(q+1)\tau\gamma}$ *and* $\gamma \geq \sqrt{p\tau}$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set*
  $R = O\left(\frac{L(1+q)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon^2}\right)$.

Proof. Since the sketching `PRIVIX` and `HEAPRIX`, satisfy the Assumption 6 with $q = \mu^2 d$ and $q = \mu^2 d - 1$ respectively with probablity $1-\delta$. Therefore, all the results in Theorem 3, conclude from Theorem 5 with probability $1-\delta$ and plugging $q = \mu^2 d$ and $q = \mu^2 d - 1$ respectively into the convergence bounds. □