Communication-efficient and Differentially-private Distributed SGD

Ananda Theertha Suresh

with

Naman Agarwal, Felix X. Yu
Sanjiv Kumar, H. Brendan McMahan

Google Research

November 16, 2018

Outline

Motivation

Distributed SGD, federated learning

Problem formulation

Distributed SGD $\rightarrow$ distributed mean estimation

Communication efficiency

Three schemes, optimality

Differential privacy

Privacy via Binomial mechanism

Outline

Motivation

Distributed SGD, federated learning

Problem formulation

Distributed SGD $\rightarrow$ distributed mean estimation

Communication efficiency

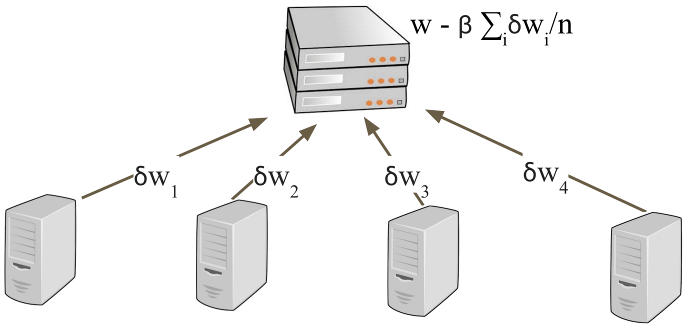
Three schemes, optimality

Differential privacy

Privacy via Binomial mechanism

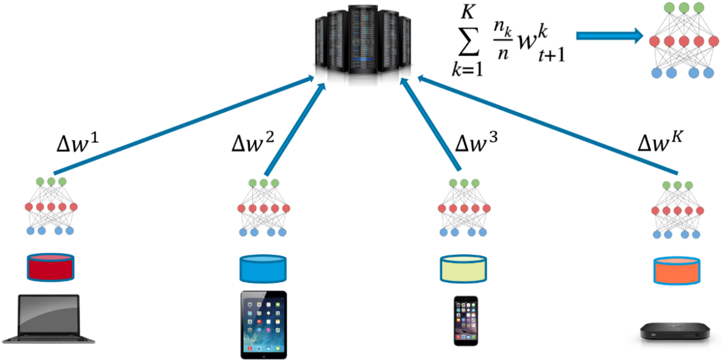
New in distributed SGD?

Server based distributed SGD



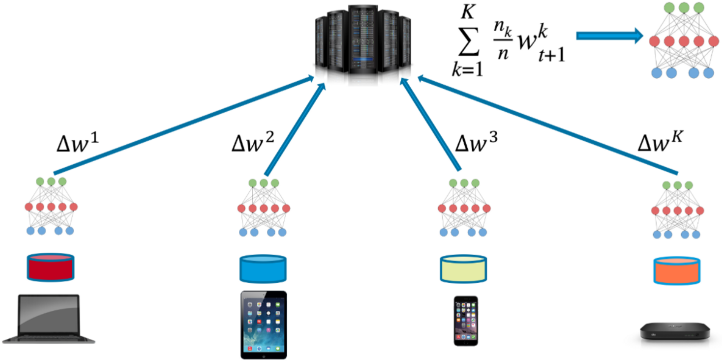
- A tool for computation efficiency
- Workers are efficient
- Well designed architecture e.g., data is i.i.d.

Client based distributed SGD



- Data is inherently distributed e.g., cellphones
- Clients (cellphones) are not computationally efficient
- Not well designed architecture e.g., data is not i.i.d.
- Constrained resources e.g., memory, bandwidth, power
- Privacy e.g., sensitive data

Client based distributed SGD



- Data is inherently distributed e.g., cellphones
- Clients (cellphones) are not computationally efficient
- Not well designed architecture e.g., data is not i.i.d.
- Constrained resources e.g., memory, bandwidth, power
- Privacy e.g., sensitive data
- Federated learning: Knoecny et al '17

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Suppose g_i^t is a vector of 32-bit floats: $32 \cdot d$ bits

Medium LSTM model for PTB: 13.5 million parameters $\sim$ 50 MB

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Suppose g_i^t is a vector of 32-bit floats: $32 \cdot d$ bits

Medium LSTM model for PTB: 13.5 million parameters $\sim$ 50 MB

This work

How to reduce the uplink communication cost?

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Suppose g_i^t is a vector of 32-bit floats: $32 \cdot d$ bits

Medium LSTM model for PTB: 13.5 million parameters $\sim$ 50 MB

This work

How to reduce the uplink communication cost?

- Uplink is more expensive than downlink

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Suppose g_i^t is a vector of 32-bit floats: $32 \cdot d$ bits

Medium LSTM model for PTB: 13.5 million parameters $\sim$ 50 MB

This work

How to reduce the uplink communication cost?

- Uplink is more expensive than downlink

What type of privacy guarantees for the users?

Distributed SGD over clients: challenges

Standard approach

Client i sends gradient g_i^t

Uplink communication cost

To send g_i^t to constant accuracy: $\approx d \log d$ bits

Expensive for large models with millions of parameters

Suppose g_i^t is a vector of 32-bit floats: $32 \cdot d$ bits

Medium LSTM model for PTB: 13.5 million parameters $\sim$ 50 MB

This work

How to reduce the uplink communication cost?

- Uplink is more expensive than downlink

What type of privacy guarantees for the users?

Can we do both?

Distributed SGD formulation

Goal

Given m functions $F_1(w), F_2(w), \ldots, F_m(w)$ on m clients, minimize

$$\frac{1}{m}\sum_{i=1}^{m} F_i(w)$$

Distributed SGD formulation

Goal

Given m functions $F_1(w), F_2(w), \ldots, F_m(w)$ on m clients, minimize

$$\frac{1}{m}\sum_{i=1}^{m} F_i(w)$$

Algorithm

Initialize w to w^0

For t from 0 to T:

- Randomly select n clients
- Selected clients compute gradients $g_i^t = \nabla F_i(w^t)$
- Selected client send gradients to server
- Server computes average $g^t = \frac{1}{n}\sum g_i^t$ and updates model by

$$w^{t+1} = w^t - \eta_t \cdot g^t$$

Distributed SGD guarantees

Non-convex problems Ghadimi et al '13

After T steps

$$\mathbb{E}_{t\sim T}\left\|\nabla F(w^t)\right\|_2^2 \lessapprox \frac{\sigma}{\sqrt{T}}$$

where

$$\sigma^2 = \max_{1\leq t\leq T} \mathbb{E}\left\|g^t(w^t) - \nabla F(w^t)\right\|_2^2$$

Similar results for convex and strongly convex problems

Distributed SGD guarantees

Non-convex problems Ghadimi et al '13

After T steps

$$\mathbb{E}_{t\sim T}\left\|\nabla F(w^t)\right\|_2^2 \lessapprox \frac{\sigma}{\sqrt{T}}$$

where

$$\sigma^2 = \max_{1\leq t\leq T} \mathbb{E}\left\|g^t(w^t) - \nabla F(w^t)\right\|_2^2$$

Similar results for convex and strongly convex problems

SGD guarantees depend on the MSE in gradients

Distributed SGD guarantees

Non-convex problems Ghadimi et al '13

After T steps

$$\mathbb{E}_{t\sim T}\left\|\nabla F(w^t)\right\|_2^2 \lessapprox \frac{\sigma}{\sqrt{T}}$$

where

$$\sigma^2 = \max_{1\le t\le T} \mathbb{E}\left\|g^t(w^t) - \nabla F(w^t)\right\|_2^2$$

Similar results for convex and strongly convex problems

SGD guarantees depend on the MSE in gradients

Guarantees with post-processing

If $g_i^t \to \tilde{g}_i^t$, such that $\mathbb{E}[\tilde{g}_i^t] = g_i^t$ and $\tilde{g}^t = \frac{1}{n}\sum_i \tilde{g}_i^t$:

$$\tilde{\sigma}^2 = \max_{1\le t\le T} \mathbb{E}\left\|\tilde{g}^t(w^t) - \nabla F(w^t)\right\|_2^2$$

Distributed mean estimation

Setting

n vectors $X^n \stackrel{\text{def}}{=} X_1, X_2 \ldots, X_n \in \mathbb{R}^d$ that reside on n clients

Distributed mean estimation

Setting

n vectors $X^n \stackrel{\text{def}}{=} X_1, X_2 \ldots, X_n \in \mathbb{R}^d$ that reside on n clients

Goal

Estimate the mean of the vectors:

$$\bar{X} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} X_i$$

Distributed mean estimation

Setting

n vectors $X^n \stackrel{\text{def}}{=} X_1, X_2 \ldots, X_n \in \mathbb{R}^d$ that reside on n clients

Goal

Estimate the mean of the vectors:

$$\bar{X} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} X_i$$

Applications

- Distributed SGD: X_i is the gradient
- Distributed power iteration
- Distributed Lloyd's algorithm
- ...

Communication efficiency: approach

Problem statement

n vectors $X^n \stackrel{\text{def}}{=} X_1, X_2 \ldots, X_n \in \mathbb{R}^d$ that reside on n clients
Estimate the mean:

$$\bar{X} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} X_i$$

Communication efficiency: approach

Problem statement

n vectors $X^n \stackrel{\text{def}}{=} X_1, X_2 \ldots, X_n \in \mathbb{R}^d$ that reside on n clients
Estimate the mean:

$$\bar{X} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} X_i$$

Communication protocol

- Client i transmits a compressed or private vector $q(X_i)$
- Server estimates the mean by some function of $q(X_1), q(X_2), \ldots, q(X_n)$

Definitions

π : communication protocol

$\hat{\bar{X}}$: estimated mean

Definitions

π : communication protocol

$\hat{\bar{X}}$: estimated mean

Mean squared error (MSE)

$$\mathcal{E}(\pi, X^n) = \mathbb{E}\left[\left\|\hat{\bar{X}} - \bar{X}\right\|_2^2\right]$$

Expectation over the randomization in the protocol

Definitions

π : communication protocol

$\hat{\bar{X}}$: estimated mean

Mean squared error (MSE)

$$\mathcal{E}(\pi, X^n) = \mathbb{E}\left[\left\|\hat{\bar{X}} - \bar{X}\right\|_2^2\right]$$

Expectation over the randomization in the protocol

Communication cost

- $\mathcal{C}_i(\pi, X_i)$: expected number of bits sent by i-th client
- Total number of bits transmitted by all clients

$$\mathcal{C}(\pi, X^n) = \sum_{i=1}^{n} \mathcal{C}_i(\pi, X_i)$$

Definitions

π : communication protocol

$\hat{\bar{X}}$: estimated mean

Mean squared error (MSE)

$$\mathcal{E}(\pi, X^n) = \mathbb{E}\left[\left\|\hat{\bar{X}} - \bar{X}\right\|_2^2\right]$$

Expectation over the randomization in the protocol

Communication cost

- $\mathcal{C}_i(\pi, X_i)$: expected number of bits sent by i-th client
- Total number of bits transmitted by all clients

$$\mathcal{C}(\pi, X^n) = \sum_{i=1}^{n} \mathcal{C}_i(\pi, X_i)$$

MSE ($\mathcal{E}$) vs communication ($\mathcal{C}$) trade-off?

Minimax formulation

No assumptions on $X_1, X_2, \ldots, X_n$

Minimax formulation

No assumptions on $X_1, X_2, \ldots, X_n$

To characterize optimality:

- B^d: unit Euclidean ball in d dimensions

Minimax formulation

No assumptions on $X_1, X_2, \ldots, X_n$

To characterize optimality:

- B^d: unit Euclidean ball in d dimensions

Minimax MSE

$$\mathcal{E}(c) \stackrel{\text{def}}{=} \min_{\pi:\mathcal{C}(\pi)<c} \max_{X^n \in B^d} \mathcal{E}(\pi, X^n)$$

Related works

Garg et al '14, Braverman et al '16

- Assume $X_1, X_2, \ldots, X_n$ are generated i.i.d. from a distribution, typically Gaussian
- Estimate the distribution mean instead of the empirical mean

Related works

Garg et al '14, Braverman et al '16

- Assume $X_1, X_2, \ldots, X_n$ are generated i.i.d. from a distribution, typically Gaussian
- Estimate the distribution mean instead of the empirical mean
- Algorithms are tailored to the assumptions e.g.,
  - Estimate the mean of a Gaussian with unit variance
  - Compute $\hat{p} = \Pr(X_i \geq 0)$
  - Output $\hat{\mu}$ such that $\Pr(N(\hat{\mu}, 1) > 0) = \hat{p}$

Related works

Garg et al '14, Braverman et al '16

- Assume $X_1, X_2, \ldots, X_n$ are generated i.i.d. from a distribution, typically Gaussian
- Estimate the distribution mean instead of the empirical mean
- Algorithms are tailored to the assumptions e.g.,
  - Estimate the mean of a Gaussian with unit variance
  - Compute $\hat{p} = \Pr(X_i \geq 0)$
  - Output $\hat{\mu}$ such that $\Pr(N(\hat{\mu}, 1) > 0) = \hat{p}$

We assume no distribution over data and want empirical mean

Related works

Garg et al '14, Braverman et al '16

- Assume $X_1, X_2, \ldots, X_n$ are generated i.i.d. from a distribution, typically Gaussian
- Estimate the distribution mean instead of the empirical mean
- Algorithms are tailored to the assumptions e.g.,
  - Estimate the mean of a Gaussian with unit variance
  - Compute $\hat{p} = \Pr(X_i \geq 0)$
  - Output $\hat{\mu}$ such that $\Pr(N(\hat{\mu}, 1) > 0) = \hat{p}$

We assume no distribution over data and want empirical mean

Alistarh et al '16

Use quantization and Elias coding

Three protocols

- Stochastic uniform quantization
  - Binary
  - k-level
- Stochastic rotated quantization
  - Quantization error significantly reduced by random rotation
- Variable length coding
  - Optimal communication-MSE trade-off

Stochastic binary quantization

For client i with vector $X_i \in \mathbb{R}^d$:

$X_i^{\max} = \max_{1 \le j \le d} X_i(j)$

$X_i^{\min} = \min_{1 \le j \le d} X_i(j)$

The quantized value for each coordinate j:

$$Y_i(j) = \begin{cases} X_i^{\max} & \text{w.p. } \frac{X_i(j) - X_i^{\min}}{X_i^{\max} - X_i^{\min}} \\ X_i^{\min} & \text{otherwise} \end{cases}$$

Stochastic binary quantization

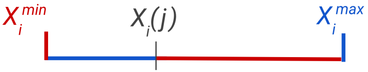
For client i with vector $X_i \in \mathbb{R}^d$:

$X_i^{\max} = \max_{1 \leq j \leq d} X_i(j)$

$X_i^{\min} = \min_{1 \leq j \leq d} X_i(j)$

The quantized value for each coordinate j:

$$Y_i(j) = \begin{cases} X_i^{\max} & \text{w.p. } \frac{X_i(j) - X_i^{\min}}{X_i^{\max} - X_i^{\min}} \\ X_i^{\min} & \text{otherwise} \end{cases}$$



Stochastic binary quantization

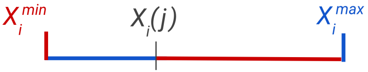
For client i with vector $X_i \in \mathbb{R}^d$:

$X_i^{\max} = \max_{1 \le j \le d} X_i(j)$

$X_i^{\min} = \min_{1 \le j \le d} X_i(j)$

The quantized value for each coordinate j:

$$Y_i(j) = \begin{cases} X_i^{\max} & \text{w.p. } \frac{X_i(j) - X_i^{\min}}{X_i^{\max} - X_i^{\min}} \\ X_i^{\min} & \text{otherwise} \end{cases}$$



Observe $\mathbb{E}\, Y_i(j) = X_i(j)$

Stochastic binary quantization

Estimated mean $$\hat{\hat{X}} = \frac{1}{n} \cdot \sum_{i=1}^{n} Y_i$$

Stochastic binary quantization

Estimated mean

$$\hat{\bar{X}} = \frac{1}{n} \cdot \sum_{i=1}^{n} Y_i$$

Communication cost

$d + \tilde{\mathcal{O}}(1)$ bits per client and hence

$$\mathcal{C} = n \cdot \left(d + \tilde{\mathcal{O}}(1)\right)$$

Stochastic binary quantization

Estimated mean

$$\hat{\bar{X}} = \frac{1}{n} \cdot \sum_{i=1}^{n} Y_i$$

Communication cost

$d + \tilde{\mathcal{O}}(1)$ bits per client and hence

$$\mathcal{C} = n \cdot \left(d + \tilde{\mathcal{O}}(1)\right)$$

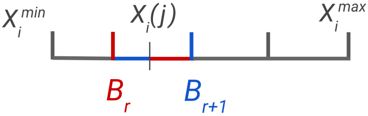
Mean squared error

$$\mathcal{E} = \mathcal{O}\left(\frac{d}{n} \cdot \frac{1}{n} \sum_{i=1}^{n} \|X_i\|_2^2\right)$$

d bits per client $\implies$ MSE is $\mathcal{O}(d/n)$

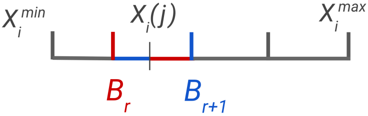
Stochastic k-level quantization

Divide the interval into k-levels



Stochastic k-level quantization

Divide the interval into k-levels



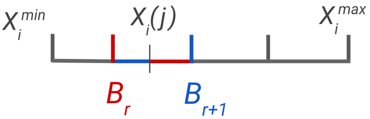
Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client and hence

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1) \right)$$

Stochastic k-level quantization

Divide the interval into k-levels



Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client and hence

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1) \right)$$

Mean squared error

$$\mathcal{E} = \mathcal{O}\left(\frac{d}{n(k-1)^2} \cdot \frac{1}{n} \sum_{i=1}^{n} \|X_i\|_2^2 \right)$$

$d \log_2 k$ bits per client $\implies$ MSE is $\mathcal{O}(d/nk^2)$

Two improvements

1. Stochastic rotated k-level quantization

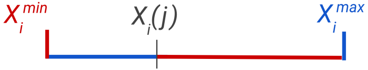
Rotates the vectors before quantization

2. Variable length coding

Use Huffman/Arithmetic coding + universal compression

Stochastic rotated k-level quantization

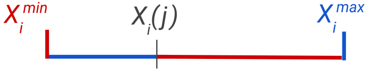
Observe



Smaller $X_i^{\max} - X_i^{\min}$, smaller the error

Stochastic rotated k-level quantization

Observe

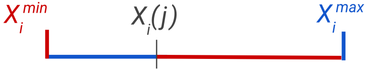


Smaller $X_i^{\max} - X_i^{\min}$, smaller the error

Random rotation reduces $X_i^{\max} - X_i^{\min}$ to $\mathcal{O}\left(\sqrt{\frac{\log d}{d}}\|X_i\|_2\right)$

Stochastic rotated k-level quantization

Observe



Smaller $X_i^{\max} - X_i^{\min}$, smaller the error

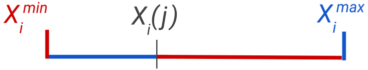
Random rotation reduces $X_i^{\max} - X_i^{\min}$ to $\mathcal{O}\left(\sqrt{\frac{\log d}{d}}\|X_i\|_2\right)$

For each client

Rotate the vector using a random rotation matrix: $Z_i = RX_i$

Stochastic rotated k-level quantization

Observe



Smaller $X_i^{\max} - X_i^{\min}$, smaller the error

Random rotation reduces $X_i^{\max} - X_i^{\min}$ to $\mathcal{O}\left(\sqrt{\frac{\log d}{d}}\|X_i\|_2\right)$

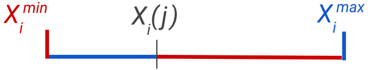
For each client

Rotate the vector using a random rotation matrix: $Z_i = RX_i$

Quantize each coordinate of Z_i in k levels to obtain Y_i

Stochastic rotated k-level quantization

Observe



Smaller $X_i^{\max} - X_i^{\min}$, smaller the error

Random rotation reduces $X_i^{\max} - X_i^{\min}$ to $\mathcal{O}\left(\sqrt{\frac{\log d}{d}}\|X_i\|_2\right)$

For each client

Rotate the vector using a random rotation matrix: $Z_i = RX_i$

Quantize each coordinate of Z_i in k levels to obtain Y_i

Server

Estimate the mean by

$$\hat{\bar{X}} = R^{-1} \cdot \frac{1}{n} \sum_{i=1}^{n} Y_i$$

Stochastic rotated k-level quantization

Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1) \right)$$

Stochastic rotated k-level quantization

Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)\right)$$

Mean squared error

$$\mathcal{E} = \mathcal{O}\left(\frac{\log d}{n(k-1)^2} \cdot \frac{1}{n}\sum_{i=1}^{n} \|X_i\|_2^2\right)$$

Stochastic rotated k-level quantization

Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)\right)$$

Mean squared error

$$\mathcal{E} = \mathcal{O}\left(\frac{\log d}{n(k-1)^2} \cdot \frac{1}{n}\sum_{i=1}^{n} \|X_i\|_2^2\right)$$

$\mathcal{O}(d \log k)$ bits/client, MSE $\mathcal{O}(d/nk^2) \to \mathcal{O}(\log d/nk^2)$

Stochastic rotated k-level quantization

Communication cost

$d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)$ bits per client

$$\mathcal{C} = n \cdot \left(d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1)\right)$$

Mean squared error

$$\mathcal{E} = \mathcal{O}\left(\frac{\log d}{n(k-1)^2} \cdot \frac{1}{n}\sum_{i=1}^{n} \|X_i\|_2^2\right)$$

$\mathcal{O}(d \log k)$ bits/client, MSE $\mathcal{O}(d/nk^2) \to \mathcal{O}(\log d/nk^2)$

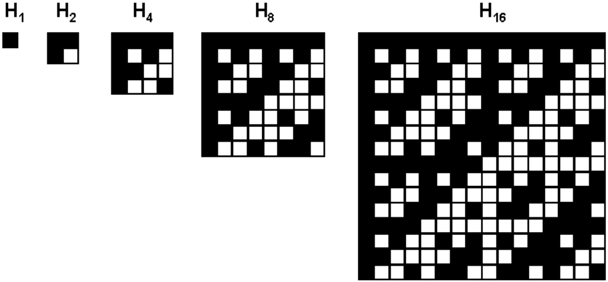
Rotation of high-dimensional vector is slow: $\mathcal{O}(d^2)$!

Fast random rotation

Use structured rotation $R = HD$

H : Walsh-Hadamard matrix

D : Diagonal matrix with independent Radamacher entries



Matrix-vector multiplication time: $\mathcal{O}(d \log d)$

Used in other contexts: Ailon et al '09, Yu et al '16

Two improvements

1. Stochastic rotated k-Level Quantization

Rotates the vectors before quantization

2. Variable length coding

Uses Huffman/Arithmetic coding + universal compression

Information theoretically optimal

Variable length coding

Previous schemes used fixed $\log k$ bits for each dimension

Variable length: use different number of bits for each dimension

Key Idea: Use fewer bits for more frequent bins

Described quantized distribution, encode using arithmetic coding for quantized distribution

Variable length coding

Previous schemes used fixed $\log k$ bits for each dimension

Variable length: use different number of bits for each dimension

Key Idea: Use fewer bits for more frequent bins

Described quantized distribution, encode using arithmetic coding for quantized distribution

Communication Cost

Arithmetic/Huffman coding + universal compression yields

$$\mathcal{C} \leq n \cdot \mathcal{O}\Big(d(1 + \log(k^2/d + 1)) + \tilde{\mathcal{O}}(1)\Big)$$

For $k \leq \sqrt{d}$, $\mathcal{O}(d \log k) \to \mathcal{O}(d)$ bits per client, MSE $\mathcal{O}(d/nk^2)$

$k = \sqrt{d}$, $\mathcal{O}(d)$ bits per client, MSE $\mathcal{O}(1/n)$

Information theoretic optimality

What is the best protocol for any worst case dataset?

Variable length coding combined with client sampling is optimal

Information theoretic optimality

What is the best protocol for any worst case dataset?

Variable length coding combined with client sampling is optimal

Min-max result

Let $t < 1$ be a constant. For communication cost $c \leq tnd$,

$$\mathcal{E}(c) = \min_{\pi:\mathcal{C}(\pi)<c} \max_{X^n \in B^d} \mathcal{E}(\pi, X^n) = \Theta\left(\min\left(1, \frac{d}{c}\right)\right)$$

Product of communication cost and MSE scales linearly with d

Information theoretic optimality

What is the best protocol for any worst case dataset?

Variable length coding combined with client sampling is optimal

Min-max result

Let $t < 1$ be a constant. For communication cost $c \leq tnd$,

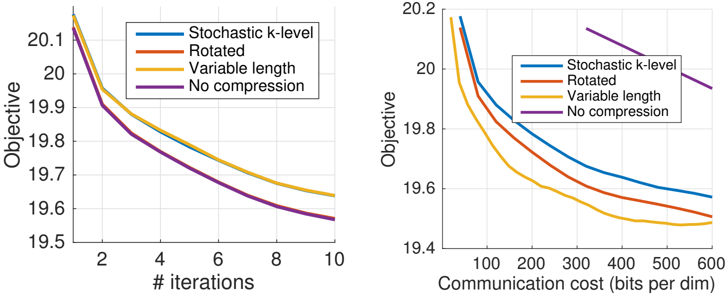
$$\mathcal{E}(c) = \min_{\pi : \mathcal{C}(\pi) < c} \max_{X^n \in B^d} \mathcal{E}(\pi, X^n) = \Theta\left(\min\left(1, \frac{d}{c}\right)\right)$$

Product of communication cost and MSE scales linearly with d

Lower bound from results in Zhang et al NIPS '13

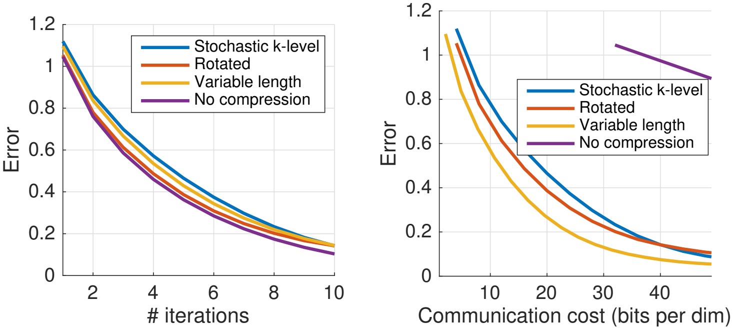
Experiments: Lloyd's algorithm (kmeans)

MNIST, $m = 60K$, $d = 784$, $n = 10$ clients, 10 centers, $k = 16$



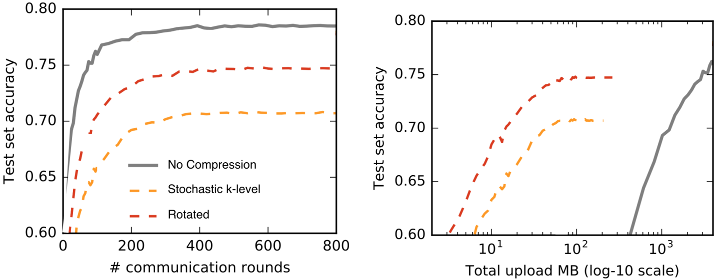
Experiments: Power iteration (PCA)

MNIST, $m = 60K$, $d = 784$, $n = 10$ clients, $k = 16$ levels



Experiments: distributed SGD

CIFAR, $m = 50K$, $d > 10^6$, $n = 100$ clients (500 examples each), $k = 2$



Communication efficiency: conclusion

Three approaches for compressed distributed mean estimation without any assumption on data distribution

For $k = 2$,

	Bits per client	MSE
Stochastic k-level	d	$\mathcal{O}(d/n)$
Rotated	d	$\mathcal{O}(\log(d)/n)$
Variable-length*	$\mathcal{O}(d)$	$\mathcal{O}(1/n)$

*Communication optimal in minimax sense

Communication efficiency: conclusion

Three approaches for compressed distributed mean estimation without any assumption on data distribution

For $k = 2$,

	Bits per client	MSE
Stochastic k-level	d	$\mathcal{O}(d/n)$
Rotated	d	$\mathcal{O}(\log(d)/n)$
Variable-length*	$\mathcal{O}(d)$	$\mathcal{O}(1/n)$

*Communication optimal in minimax sense

Privacy?

Differential privacy

Communication protocol

Client i transmits a compressed / private vector $q(X_i)$

Server estimates the mean by some function of $q(X_1), \ldots, q(X_n)$

Let the estimate be $\hat{\bar{X}}$

Differential privacy

Communication protocol

Client i transmits a compressed / private vector $q(X_i)$

Server estimates the mean by some function of $q(X_1), \ldots, q(X_n)$

Let the estimate be $\hat{\bar{X}}$

Differential privacy

Two sets: $X^n = X_1, X_2, \ldots, X_n$ and $X'^n = X_1, X_2, \ldots, X'_n$

A mean estimator is (ϵ, δ) differential private, if for any set S

$$\Pr(\hat{\bar{X}}(X^n) \in S) \leq e^{\epsilon} \Pr(\hat{\bar{X}}(X'^n) \in S) + \delta$$

Differential privacy

Communication protocol

Client i transmits a compressed / private vector $q(X_i)$

Server estimates the mean by some function of $q(X_1), \ldots, q(X_n)$

Let the estimate be $\hat{\hat{X}}$

Differential privacy

Two sets: $X^n = X_1, X_2, \ldots, X_n$ and $X'^n = X_1, X_2, \ldots, X'_n$

A mean estimator is (ϵ, δ) differential private, if for any set S

$$\Pr(\hat{\hat{X}}(X^n) \in S) \leq e^{\epsilon} \Pr(\hat{\hat{X}}(X'^n) \in S) + \delta$$

No one can identify X_i of a single user

Gaussian mechanism

Algorithm

Server computes the average $\hat{\hat{X}}$ using $q(X_1), q(X_2), \ldots, q(X_n)$ and releases

$$\hat{\hat{X}} + N(0, \sigma^2 \mathbb{I}_d),$$

where $\sigma \approx \frac{1}{n\epsilon} \log \frac{1}{\delta}$

Gaussian mechanism

Algorithm

Server computes the average $\hat{\bar{X}}$ using $q(X_1), q(X_2), \ldots, q(X_n)$ and releases

$$\hat{\bar{X}} + N(0, \sigma^2 \mathbb{I}_d),$$

where $\sigma \approx \frac{1}{n\epsilon} \log \frac{1}{\delta}$

Guarantees

Released model is (ϵ, δ) differential private

Gaussian mechanism

Algorithm

Server computes the average $\hat{\hat{X}}$ using $q(X_1), q(X_2), \ldots, q(X_n)$ and releases

$$\hat{\hat{X}} + N(0, \sigma^2 \mathbb{I}_d),$$

where $\sigma \approx \frac{1}{n\epsilon} \log \frac{1}{\delta}$

Guarantees

Released model is (ϵ, δ) differential private

Issues

Server may not add noise

Server knows true $\hat{\hat{X}}$

Distributed Gaussian mechanism

Algorithm

Clients send $g(X_i) = q(X_i) + N(0, \sigma\sqrt{n})$
Server computes average by

$$\frac{1}{n} \cdot \sum_{i=1}^{n} g(X_i)$$

Distributed Gaussian mechanism

Algorithm

Clients send $g(X_i) = q(X_i) + N(0, \sigma\sqrt{n})$
Server computes average by

$$\frac{1}{n} \cdot \sum_{i=1}^{n} g(X_i)$$

Analysis

Average of Gaussians is a Gaussian
Total noise variance:

$$\frac{1}{n} \cdot \sigma^2 n = \sigma^2$$

$\implies$ Learned model is (ϵ, δ) differentially private

Differential privacy guarantees

Server is negligent but not malicious

Estimate of the average is differentially private

Learned model is private and safe to release

Differential privacy guarantees

Server is negligent but not malicious

Estimate of the average is differentially private

Learned model is private and safe to release

Clients do not trust the server

Secure aggregation: cryptographic scheme to ensure that the server learns only the average $\hat{\hat{X}}$

Server does not learn anything about individual users

Differential privacy guarantees

Server is negligent but not malicious

Estimate of the average is differentially private

Learned model is private and safe to release

Clients do not trust the server

Secure aggregation: cryptographic scheme to ensure that the server learns only the average $\hat{\hat{X}}$

Server does not learn anything about individual users

Issue

Algorithm is not communication efficient

Cryptographic protocol operates over discrete values

Distributed Gaussian mechanism: attempt 2

Quantized Gaussians

- Clients send $g(X_i) = X_i + N(0, \sigma\sqrt{n})$
- Quantizes $g(X_i)$ to obtain $q(g(X_i))$
- Servers compute average by

$$\frac{1}{n} \cdot \sum_{i=1}^{n} q(g(X_i))$$

Distributed Gaussian mechanism: attempt 2

Quantized Gaussians

- Clients send $g(X_i) = X_i + N(0, \sigma\sqrt{n})$
- Quantizes $g(X_i)$ to obtain $q(g(X_i))$
- Servers compute average by

$$\frac{1}{n} \cdot \sum_{i=1}^{n} q(g(X_i))$$

Differential privacy

Sum of quantized Gaussians is not Gaussian
Needs proof that quantization does not affect privacy

Binomial mechanism

Algorithm

- Client i computes the compressed vector $q(X_i)$
- Adds Binomial noise $Z_i = \mathrm{Bin}(m, p)$ and transmits

$$q(X_i) + Z_i$$

Binomial mechanism

Algorithm

- Client i computes the compressed vector $q(X_i)$
- Adds Binomial noise $Z_i = \text{Bin}(m, p)$ and transmits

$$q(X_i) + Z_i$$

Advantages

- Finite number of bits to send Binomial random variable
- Communication cost

$$n \cdot d \cdot \lceil \log_2(m + k) \rceil$$

- Sum of Binomial noise $\sum_i Z_i$ is also binomial
- By CLT: Binomial $\rightarrow$ Gaussian

Binomial mechanism results

Gaussian mechanism Dwork et al '06

$N(0, \sigma^2)$ is (ϵ, δ) differentially private for

$$\epsilon \geq \frac{\Delta_2 \sqrt{2 \log \frac{1.25}{\delta}}}{\sigma}$$

Binomial mechanism

$\mathrm{Bin}(N, p)$ is (ϵ, δ) differentially private for

$$\epsilon \geq \frac{\Delta_2 \sqrt{2 \log \frac{1.25}{\delta}}}{\sigma} + \tilde{\mathcal{O}}\left(\frac{\Delta_1 + \Delta_\infty}{\sigma^2}\right)$$

$\sigma = Np(1-p)$

High privacy regime: $\epsilon \to 0, \sigma \to \infty$, mechanisms are same

cpSGD

Rotated quantization + Binomial mechanism

Achieves same MSE as Gaussian mechanism and has a communication cost

$$n \cdot d \cdot \left(\log_2 \left(1 + \frac{d}{n\epsilon^2} \right) + \mathcal{O} \left(\log \log \frac{nd}{\epsilon\delta} \right) \right)$$

If $n \approx d$, then $\log \log (nd/\epsilon\delta)$ bits are sufficient

Conclusions and future directions

Conclusions

- Distributed SGD $\rightarrow$ distributed mean estimation
- Minimax optimal scheme for distributed mean estimation
- Communication-efficient DP by Binomial mechanism

Future directions

- Distributed SGD: correlation between gradients over time
- Distributed SGD: better privacy algorithms
- Distributed mean estimation: competitive / instance optimal

Thank you!