Distributed optimization: compression and local updates

Optimization in ML

Aleksandr Beznosikov

Skoltech

5 December 2023

Contemporary learning challenges

- Exponential growth in model sizes and data volumes.

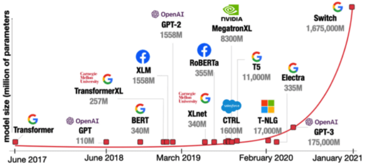
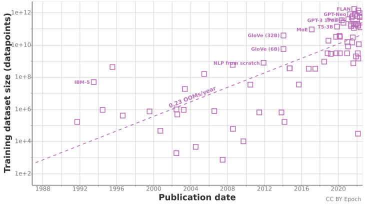


Figure: Dynamics of growth of modern language models



Varieties of distributed learning

- Cluster learning (large players): we train within one large and powerful computing cluster
- Collaborative learning (all players): pooling computing resources over the Internet

Varieties of distributed learning

- Cluster learning (large players): we train within one large and powerful computing cluster
- Collaborative learning (all players): pooling computing resources over the Internet
- Federated learning (another paradigm): learn on users' local data using their computational power

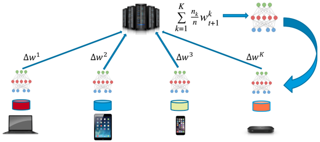


Figure: Federated learning

The most popular distributed setup

- Staging (horizontal, offline):

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{M} \sum_{m=1}^{M} f_m(w) := \frac{1}{M} \sum_{m=1}^{M} \frac{1}{n_m} \sum_{i=1}^{n_m} l(g(w, x_i), y_i).$$

- w – model weights, g – model, l – loss function.
- The data is shared among M computing devices, each device m has its own local subsample $\{x_i, y_i\}_{i=1}^{n_m}$ of size n_m.

Communicating through the server

- Let us look at an example of how an ordinary undefined GD becomes centralized.

Algorithm 1 Centralized GD

Input: Stepsize $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, number of iterations K

```
for k = 0, 1, ..., K − 1 do
    Send w_k to all workers                              ▷ by server
    for i = 1, ..., n in parallel do
        Recieve w_k from server                          ▷ by workers
        Compute ∇f_m(w_k) in w_k                         ▷ by workers
        Send ∇f_m(w_k) to server                         ▷ by workers
    end for
    Recieve ∇f_m(w_k) from all workers                   ▷ by server
    Compute ∇f(w_k) = (1/M) Σ_{m=1}^{M} ∇f_m(w_k)        ▷ by server
    w_{k+1} = w_k − γ∇f(w_k)                             ▷ by server
end for
```

Output: w^K

What are we fighting for?

- **Question:** distribution is necessary for parallelization, but why can't we achieve full parallelization?

What are we fighting for?

- **Question:** distribution is necessary for parallelization, but why can't we achieve full parallelization?
- Communication costs are a waste of time.
- The communication bottleneck problem is relevant for all distributed learning productions.
- There are many ways to fight for effective communications.

Compression: unbiased and biased operators

Unbiased compression (quantization)

Unbiased compression (quantization)

Let us call the stochastic operator $\mathcal{Q}(x)$ an unbiased compression (quantization) operator if for any $x \in \mathbb{R}^d$ it is fulfilled:

$$\mathbb{E}[\mathcal{Q}(x)] = x, \quad \mathbb{E}[\|\mathcal{Q}(x)\|_2^2] \leq \omega \|x\|_2^2,$$

where $\omega \geq 1$.

Unbiased compression: examples

Random sparsification (selection of random components)

Consider a stochastic operator

$$\text{Randk}(x) = \frac{d}{k} \sum_{i \in S} [x]_i e_i,$$

where k — some fixed number from the set $\{1, \ldots, d\}$ (the number of components of vector x that we pass; for example, we can choose $k = 1$), S —a random subset of the set $\{1, \ldots, d\}$ of size k (the subset S is chosen randomly and equally likely among all possible subsets of size d), $[\cdot]_i$ – i-th component of the vector, $(e_1, \ldots, e_d)$ —the standard basis in $\mathbb{R}^d$.

Richtárik P. and Takáč M. Parallel coordinate descent methods for big data optimization

Unbiased compression: examples

Random sparsification (selection of random components)

Consider a stochastic operator

$$\mathrm{Randk}(x) = \frac{d}{k} \sum_{i \in S} [x]_i e_i,$$

where k — some fixed number from the set $\{1, \ldots, d\}$ (the number of components of vector x that we pass; for example, we can choose $k = 1$), S —a random subset of the set $\{1, \ldots, d\}$ of size k (the subset S is chosen randomly and equally likely among all possible subsets of size d), $[\cdot]_i$ – i-th component of the vector, $(e_1, \ldots, e_d)$ —the standard basis in $\mathbb{R}^d$.

Richtárik P. and Takáč M. Parallel coordinate descent methods for big data optimization

- **Question:** why do we need the multiplier $\frac{d}{k}$?

Unbiased compression: examples

Random sparsification (selection of random components)

Consider a stochastic operator

$$\mathrm{Randk}(x) = \frac{d}{k} \sum_{i \in S} [x]_i e_i,$$

where k — some fixed number from the set $\{1, \ldots, d\}$ (the number of components of vector x that we pass; for example, we can choose $k = 1$), S —a random subset of the set $\{1, \ldots, d\}$ of size k (the subset S is chosen randomly and equally likely among all possible subsets of size d), $[\cdot]_i$ – i-th component of the vector, $(e_1, \ldots, e_d)$ —the standard basis in $\mathbb{R}^d$.

Richtárik P. and Takáč M. Parallel coordinate descent methods for big data optimization

- **Question:** why do we need the multiplier $\frac{d}{k}$? For unbiasedness.

Unbiased compression: examples

- **Question:** What is ω for random sparsification?

Unbiased compression: examples

- **Question:** What is ω for random sparsification? $\frac{d}{k}k$.
  Each coordinate takes part in $\mathcal{Q}(x)$ with probability $\frac{k}{d}$, so

$$
\begin{aligned}
\mathbb{E}\left[\|\mathcal{Q}(x)\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^{d}[\mathcal{Q}(x)]_i^2\right] \\
&= \frac{d^2}{k^2}\left[\sum_{i=1}^{d}\frac{k}{d}[x]_i^2\right] \\
&= \frac{d}{k}\|x\|^2.
\end{aligned}
$$

  Here $[\cdot]_i$ – ith coordinate of the vector.

Unbiased compression: examples

Three-level ℓ_2-quantization

Consider the following operator: $[\mathcal{Q}(x)]_i = \|x\|_2 \text{sign}(x_i)\xi_i, \;\; i = 1, \ldots, d$, where $[\cdot]_i$ is the i-th component of the vector, and ξ_i —a random variable having a Bernoulli distribution with parameter $\frac{|x_i|}{\|x\|_2}$, i.e.

$$\xi_i = \begin{cases} 1 & \text{with probability } \frac{|x_i|}{\|x\|_2}, \\ 0 & \text{with probability } 1 - \frac{|x_i|}{\|x\|_2}. \end{cases}$$

Thus, if we want to pass a vector $\mathcal{Q}(x)$, we need to pass a vector consisting of zeros and ± 1, and a real number $\|x\|_2$, with the probability of zeroing a component the greater the component is smaller modulo it. It can be shown that this operator is an unbiased compression with constant $\omega = \sqrt{d}$.

Alistarh D. et al. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding

Unbiased compression: examples

- **Question:** Is rounding be an unbiased operator?
- **Question:** Which rounding seems to be the most natural for calculations on a computer?

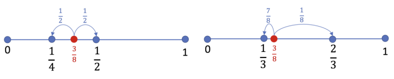
Unbiased compression: examples

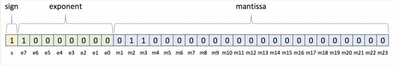
Natural compression (random rounding to the power of two)

Consider the following operator:

$$[\mathcal{Q}(x)]_i = \begin{cases} \lfloor [x]_i \rfloor_2\,, & \text{with probabilty } p = \frac{[x]_i - \lceil [x]_i \rceil_2}{\lceil [x]_i \rceil_2 - \lfloor [x]_i \rfloor_2} \\ \lceil [x]_i \rceil_2\,, & \text{with probability } 1-p \end{cases},$$

where $[\cdot]_i$ – i-vector component, $\lfloor \cdot \rfloor_2$ – is the nearest degree of two from the bottom, $\lceil \cdot \rceil_2$ is the nearest degree of two from the top. We round to the two nearest powers of two, the probability of rounding is greater the closer the real number is to the corresponding power of two. It can be shown that $\omega = \frac{9}{8}$.





Horváth S. et al. Natural compression for distributed deep learning

Unbiased compression: an idea

- The simplest idea that comes to mind is to use parallel `GD`, but apply unbiased compression to the gradients sent from the workers to the server.

Quantized GD (QGD)

Algorithm 1 QGD

Input: step size $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, number of iterations K

for $k = 0, 1, \ldots, K-1$ **do**
    Send w_k to all workers ▷ by server
    for $m = 1, \ldots, M$ in parallel **do**
        Recieve w_k from server ▷ by workers
        Compute $\nabla f_m(w_k)$ in point w_k ▷ by workers
        Independently generate $g_{k,m} = \mathcal{Q}(\nabla f_m(w_k))$ ▷ by workers
        Send $g_{k,m}$ to master ▷ by workers
    end for
    Recieve $g_{k,m}$ from all workers ▷ by server
    Compute $g_k = \frac{1}{M}\sum_{m=1}^{M} g_{k,m}$ ▷ by server
    $w_{k+1} = w_k - \gamma g_k$ ▷ by server
end for

Output: w^K

Unbiased compression: a proof

- We prove in the case where all f_m are L-smooth and μ-simply convex.
- Let us consider one iteration of the method:

$$\|w_{k+1} - w^*\|^2 = \|w_k - w^*\|^2 - 2\gamma\langle g_k, w_k - w^*\rangle + \|g_k\|^2.$$

Unbiased compression: a proof

- We prove in the case where all f_m are L-smooth and μ-simply convex.
- Let us consider one iteration of the method:

$$\|w_{k+1} - w^*\|^2 = \|w_k - w^*\|^2 - 2\gamma\langle g_k, w_k - w^*\rangle + \|g_k\|^2.$$

- We take the conditional mat expectation by randomness only at iteration k:

$$\begin{aligned}\mathbb{E}\left[\|w_{k+1} - w^*\|^2 \mid w_k\right] =& \|w_k - w^*\|^2 - 2\gamma\langle \mathbb{E}\left[g_k \mid w_k\right], w_k - w^*\rangle \\ &+ \gamma^2 \mathbb{E}\left[\|g_k\|^2 \mid w_k\right].\end{aligned}$$

Unbiased compression: a proof

- work with $\mathbb{E}\left[g_k \mid w_k\right]$:

$$
\begin{aligned}
\mathbb{E}\left[g_k \mid w_k\right] &= \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\left[g_{k,m} \mid w_k\right] \\
&= \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\left[\mathbb{E}\left[\mathcal{Q}(\nabla f_m(w_k)) \mid \nabla f_m(w_k)\right] \mid w_k\right] \\
&= \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\left[\nabla f_m(w_k) \mid w_k\right] = \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w_k) = \nabla f(w_k).
\end{aligned}
$$

Unbiased compression: a proof

- work with $\mathbb{E}\left[g_k \mid w_k\right]$:

$$\mathbb{E}\left[g_k \mid w_k\right] = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[g_{k,m} \mid w_k\right]$$
$$= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{E}\left[\mathcal{Q}(\nabla f_m(w_k)) \mid \nabla f_m(w_k)\right] \mid w_k\right]$$
$$= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[\nabla f_m(w_k) \mid w_k\right] = \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(w_k) = \nabla f(w_k).$$

- work with $\mathbb{E}\left[\|g_k\|^2 \mid w_k\right]$:

$$\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] = \mathbb{E}\left[\left\|\frac{1}{M}\sum_{m=1}^{M} g_{k,m}\right\|^2 \mid x_k\right] = \frac{1}{M^2}\mathbb{E}\left[\left\|\sum_{m=1}^{M} g_{k,m}\right\|^2 \mid w^k\right]$$

Unbiased compression: a proof

- We continue and apply the first property (unbiasedness) in the definition of compression:

$$
\begin{aligned}
\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] =& \frac{1}{M^2}\mathbb{E}\left[\left\|\sum_{m=1}^{M} g_{k,m}\right\|^2 \mid w_k\right] \\
=& \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|g_{k,m}\|^2 \mid w_k\right] + \frac{2}{M^2}\sum_{m\neq l}\mathbb{E}\left[\langle g_{k,m}, g_{k,l}\rangle \mid w_k\right] \\
=& \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|g_{k,m}\|^2 \mid w_k\right] \\
&+ \frac{1}{M^2}\sum_{m\neq l}\mathbb{E}\left[\langle \mathbb{E}\left[g_{k,m} \mid \nabla f_m(w_k)\right], \mathbb{E}\left[g_{k,l} \mid \nabla f_l(x_k)\right]\rangle \mid w_k\right].
\end{aligned}
$$

Unbiased compression: a proof

- We continue and apply the first property (unbiasedness) in the definition of compression:

$$
\begin{aligned}
\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] =& \frac{1}{M^2}\mathbb{E}\left[\left\|\sum_{m=1}^{M} g_{k,m}\right\|^2 \mid w_k\right] \\
=& \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|g_{k,m}\|^2 \mid w_k\right] + \frac{2}{M^2}\sum_{m\neq l}\mathbb{E}\left[\langle g_{k,m}, g_{k,l}\rangle \mid w_k\right] \\
=& \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|g_{k,m}\|^2 \mid w_k\right] \\
&+ \frac{1}{M^2}\sum_{m\neq l}\mathbb{E}\left[\langle \mathbb{E}\left[g_{k,m} \mid \nabla f_m(w_k)\right], \mathbb{E}\left[g_{k,l} \mid \nabla f_l(x_k)\right]\rangle \mid w_k\right].
\end{aligned}
$$

Unbiased compression: a proof

- We continue and apply the second property in the definition of compression:

$$
\begin{aligned}
\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] =& \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|\mathcal{Q}(\nabla f_m(w_k))\|^2 \mid w_k\right] \\
&+ \frac{1}{M^2}\sum_{m\neq l}\langle \nabla f_m(w_k), \nabla f_l(w_k)\rangle \\
\leq& \frac{\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w_k)\|^2 + \|\nabla f(w_k)\|^2 \\
\leq& \frac{2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w_k) - \nabla f_m(w^*)\|^2 \\
&+ \frac{2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 + \|\nabla f(w_k) - \nabla f(w^*)\|^2.
\end{aligned}
$$

Unbiased compression: a proof

- We continue and apply the second property in the definition of compression:

$$
\begin{aligned}
\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] \leq & \frac{4\omega L}{M^2}\sum_{m=1}^{M}(f_m(w_k) - f_m(w^*) - \langle \nabla f_m(w^*), w_k - w^* \rangle) \\
& + \frac{2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 + 2L(f(w_k) - f(x^*)) \\
= & \frac{4\omega L}{M}(f(w_k) - f(w^*)) \\
& + \frac{2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 + 2L(f(w_k) - f(w^*)).
\end{aligned}
$$

Unbiased compression: a proof

- Everything we got:

$$\mathbb{E}\left[\|w_{k+1} - w^*\|^2 \mid w_k\right] = \|w_k - w^*\|^2 - 2\gamma\langle \mathbb{E}\left[g_k \mid w_k\right], w_k - w^*\rangle + \gamma^2 \mathbb{E}\left[\|g_k\|^2 \mid w_k\right].$$

$$\mathbb{E}\left[g_k \mid w_k\right] = \nabla f(w_k).$$

$$\mathbb{E}\left[\|g_k\|^2 \mid w_k\right] \leq \frac{4\omega L}{M}(f(w_k) - f(w^*)) + \frac{2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 + 2L(f(w_k) - f(w^*)).$$

Unbiased compression: a proof

- Connect:

$$\mathbb{E}\left[\|w_{k+1} - w^*\|^2 \mid w_k\right] \leq \|w_k - w^*\|^2 - 2\gamma\langle \nabla f(w_k), w_k - w^*\rangle + 2\gamma^2 L\left(\frac{2\omega}{M} + 1\right)(f(w_k) - f(w^*)) + \frac{2\gamma^2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2.$$

- We use a strong convexity:

$$\mathbb{E}\left[\|w_{k+1} - w^*\|^2 \mid w_k\right] \leq \|w_k - w^*\|^2 - 2\gamma\left(\frac{\mu}{2}\|w_k - w^*\|^2 + f(w_k) - f(w^*)\right) + 2\gamma^2 L\left(\frac{2\omega}{M} + 1\right)(f(w_k) - f(w^*)) + \frac{2\gamma^2\omega}{M^2}\sum^{M}\|\nabla f_m(w^*)\|^2$$

Unbiased compression: a proof

- If we take the total mathematical expectation

$$
\begin{aligned}
\mathbb{E}\left[\|w_{k+1} - w^*\|^2\right] \leq & (1-\gamma\mu)\mathbb{E}\left[\|w_k - w^*\|^2\right] \\
& - 2\gamma\left[1 - \gamma L\left(\frac{2\omega}{M} + 1\right)\right]\mathbb{E}\left[(f(x_k) - f(x^*))\right] \\
& + \frac{2\gamma^2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 .
\end{aligned}
$$

- If $\gamma \leq L^{-1}\left(\frac{2\omega}{M} + 1\right)^{-1}$, then

$$
\begin{aligned}
\mathbb{E}\left[\|w_{k+1} - w^*\|^2\right] \leq & (1-\gamma\mu)\mathbb{E}\left[\|w_k - w^*\|^2\right] \\
& + \frac{2\gamma^2\omega}{M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2 .
\end{aligned}
$$

QGD: convergence

Theorem (QGD)

Let all local functions f_m be μ-simply convex and have L-Lipschitz gradient, then if $\eta \leq L^{-1}\left(\frac{2\omega}{M}+1\right)^{-1}$ then

$$\mathbb{E}\left[\|w_K - w^*\|^2\right] = \mathcal{O}\left((1-\gamma\mu)^K\|w_0 - w^*\|^2 + \frac{1}{K}\cdot\frac{2\omega}{\mu M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2\right)$$

The selection of γ from the paper was also used to obtain this result:

PDF Stich S. Unified Optimal Analysis of the (Stochastic) Gradient Method

QGD: convergence

Theorem (QGD)

Let all local functions f_m be μ-simply convex and have L-Lipschitz gradient, then if $\eta \leq L^{-1}\left(\frac{2\omega}{M}+1\right)^{-1}$ then

$$\mathbb{E}\left[\|w_K - w^*\|^2\right] = \mathcal{O}\left((1-\gamma\mu)^K\|w_0 - w^*\|^2 + \frac{1}{K}\cdot\frac{2\omega}{\mu M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2\right)$$

The selection of γ from the paper was also used to obtain this result:

PDF Stich S. Unified Optimal Analysis of the (Stochastic) Gradient Method

- **Question:** what are the problems with this estimation? (recall the convergence estimate GD)

QGD: convergence

Theorem (QGD)

Let all local functions f_m be μ-simply convex and have L-Lipschitz gradient, then if $\eta \leq L^{-1}\left(\frac{2\omega}{M} + 1\right)^{-1}$ then

$$\mathbb{E}\left[\|w_K - w^*\|^2\right] = \mathcal{O}\left((1-\gamma\mu)^K\|w_0 - w^*\|^2 + \frac{1}{K}\cdot\frac{2\omega}{\mu M^2}\sum_{m=1}^{M}\|\nabla f_m(w^*)\|^2\right)$$

The selection of γ from the paper was also used to obtain this result:

PDF Stich S. Unified Optimal Analysis of the (Stochastic) Gradient Method

- **Question:** what are the problems with this estimation? (recall the convergence estimate GD) Sublinear convergence (depends on the heterogeneity of the data).

QGD: convergence

- Behaviour in practice:

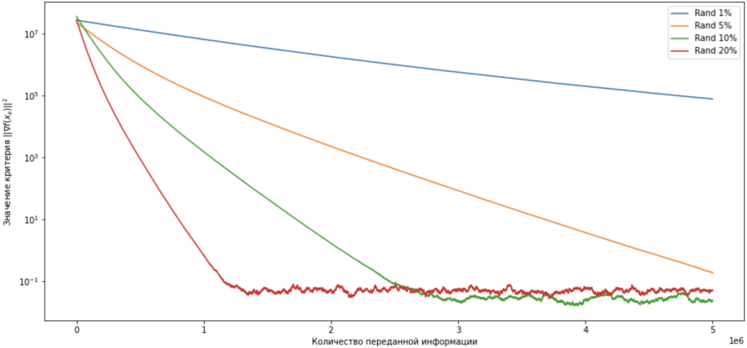


Figure: Behaviour of methods with unbiased compression operator and constant step size

- In theory the pitch was chosen cleverly, with constant step the theory predicts exactly the same effect – early plateauing.

Unbiased compression: solving the plateau problem

- Method DIANA – QGD with memory:

Algorithm 1 DIANA (sketch)

1: Each device m possesses a vector of "memory" $h_0^m = 0$

2: The server stores $h_0 = \frac{1}{M}\sum\limits_{m=1}^{M} h_0^m = 0$

3: Send a compressed version of the difference to the server $\mathcal{Q}(\nabla f_m(w^k) - h_k^m)$

4: Refreshing the memory $h_{k+1}^m = h_k^m + \alpha\mathcal{Q}(\nabla f_m(w^k) - h_k^m)$

5: The server calculates $g_k = h_k + \frac{1}{M}\sum\limits_{m=1}^{M} \mathcal{Q}(\nabla f_m(w^k) - h_k^m)$

6: For the update $w_{k+1} = w_k - \gamma g_k$

7: The server updates $h_{k+1} = h_k + \alpha\frac{1}{M}\sum\limits_{m=1}^{M} \mathcal{Q}(\nabla f_m(w^k) - h_k^m)$

Mishchenko K. et al. Distributed Learning with Compressed Gradient Differences

QGD and DIANA: convergence

- **Question:** What are some other questions about convergence/estimates of convergence?

QGD and DIANA: convergence

- **Question:** What are some other questions about convergence/estimates of convergence? Does it converge better at all?
- Best estimate per number of communications for the unaccelerated method with unbiased compression (DIANA):

$$\mathcal{O}\left(\left[1 + \frac{\omega}{M}\right] \frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- Estimate on the number of communications for GD:

$$\mathcal{O}\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- In terms of **number** of communications, compressed methods are inferior to basic methods – this is to be expected (compression fees). BUT!

QGD and DIANA: convergence

- Compressors compress information β times and it is typical that $\beta \geq \omega$.

QGD and DIANA: convergence

- Compressors compress information β times and it is typical that $\beta \geq \omega$.
- Best estimate on the number of information for the unaccelerated method with unbiased compression (DIANA):

$$\mathcal{O}\left(\left[\frac{1}{\beta} + \frac{1}{M}\right]\frac{L}{\mu}\log\frac{1}{\varepsilon}\right).$$

- Estimate on the number of information for GD:

$$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right).$$

QGD and DIANA: convergence

- Compressors compress information β times and it is typical that $\beta \geq \omega$.
- Best estimate on the number of information for the unaccelerated method with unbiased compression (DIANA):

$$\mathcal{O}\left(\left[\frac{1}{\beta} + \frac{1}{M}\right] \frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- Estimate on the number of information for GD:

$$\mathcal{O}\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- The unbiased compressor provably improves the number of transmitted information, an improvement factor: $\left[\frac{1}{\beta} + \frac{1}{M}\right]$.

There may not be a server

- Often in practice "centralised communications via a server" are implemented without a "server".
- Architecture with AllGather/AllReduce procedure: some graph of links/communications is given, messages are exchanged according to this graph, including averaging can be organised.

Chan, E. et al. Collective communication: theory, practice, and experience

Centralised communications without a server

Operation	Before: Node 0	Before: Node 1	Before: Node 2	Before: Node 3	After: Node 0	After: Node 1	After: Node 2	After: Node 3
Broadcast	x				x	x	x	x
Reduce(-to-one)	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$\sum_j x^{(j)}$			
Scatter	x_0 x_1 x_2 x_3				x_0	x_1	x_2	x_3
Gather	x_0	x_1	x_2	x_3	x_0 x_1 x_2 x_3			
Allgather	x_0	x_1	x_2	x_3	x_0 x_1 x_2 x_3	x_0 x_1 x_2 x_3	x_0 x_1 x_2 x_3	x_0 x_1 x_2 x_3
Reduce-scatter	$x_0^{(0)}$ $x_1^{(0)}$ $x_2^{(0)}$ $x_3^{(0)}$	$x_0^{(1)}$ $x_1^{(1)}$ $x_2^{(1)}$ $x_3^{(1)}$	$x_0^{(2)}$ $x_1^{(2)}$ $x_2^{(2)}$ $x_3^{(2)}$	$x_0^{(3)}$ $x_1^{(3)}$ $x_2^{(3)}$ $x_3^{(3)}$	$\sum_j x_0^{(j)}$	$\sum_j x_1^{(j)}$	$\sum_j x_2^{(j)}$	$\sum_j x_3^{(j)}$
Allreduce	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$\sum_j x^{(j)}$	$\sum_j x^{(j)}$	$\sum_j x^{(j)}$	$\sum_j x^{(j)}$

Ring AllReduce

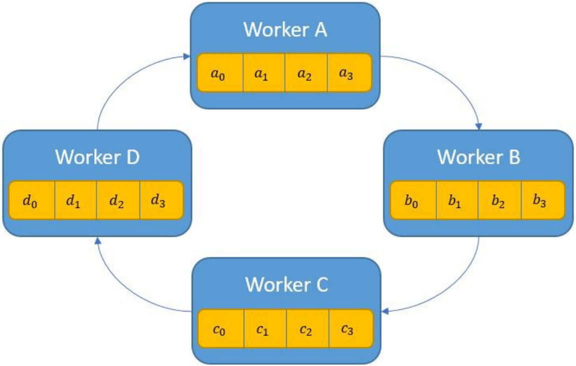


Figure: Picture from here

Ring AllReduce: first step of averaging

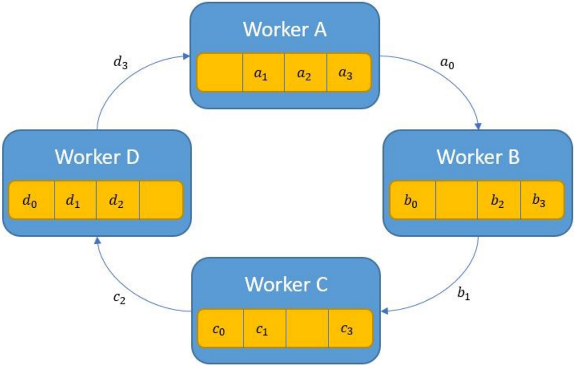


Figure: Picture from here

Ring AllReduce: second step of averaging

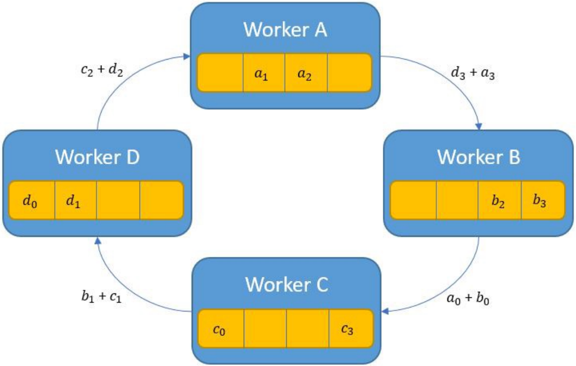


Figure: Picture from here

Ring AllReduce: first step of backprogation

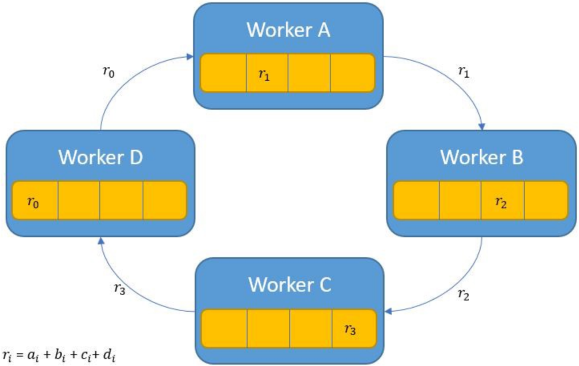


Figure: Картинка отсюда

Ring AllReduce: final

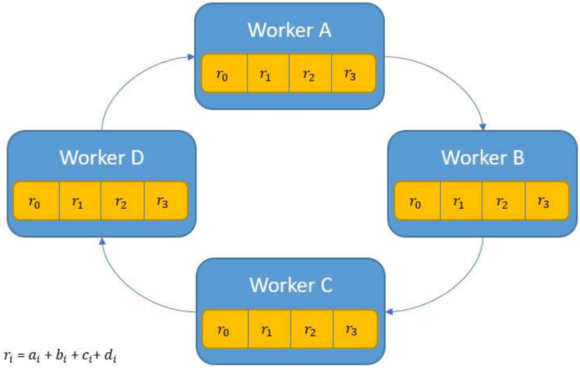


Figure: Картинка отсюда

Quantized GD (QGD) with AllReduce

Algorithm 1 QGD

Input: step size $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, number of iterations K

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: **for** $m = 1, \ldots, M$ in parallel **do**
3: Compute $\nabla f_m(w_k)$ in w_k
4: Indipendetly generate $g_{k,m} = \mathcal{Q}(\nabla f_m(w_k))$
5: Run AllReduce $\{g_{k,m}\}$ and get $g_k = \frac{1}{M}\sum_{m=1}^{M} g_{k,m}$
6: $w_{k+1} = w_k - \gamma g_k$
7: **end for**
8: **end for**

Output: w^K

Quantized GD (QGD) with AllReduce

Algorithm 1 QGD

Input: step size $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, number of iterations K

for $k = 0, 1, \ldots, K-1$ **do**
  for $m = 1, \ldots, M$ in parallel **do**
    Compute $\nabla f_m(w_k)$ in w_k
    Indipendetly generate $g_{k,m} = \mathcal{Q}(\nabla f_m(w_k))$
    Run AllReduce $\{g_{k,m}\}$ and get $g_k = \frac{1}{M}\sum_{m=1}^{M} g_{k,m}$
    $w_{k+1} = w_k - \gamma g_k$
  end for
end for

Output: w^K

- **Question:** What kind of problems might appear with (for example) Randk?

Quantized GD (QGD) with AllReduce

Algorithm 1 QGD

Input: step size $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, number of iterations K

```
for k = 0, 1, ..., K − 1 do
    for m = 1, ..., M in parallel do
        Compute ∇f_m(w_k) in w_k
        Indipendetly generate g_{k,m} = Q(∇f_m(w_k))
        Run AllReduce {g_{k,m}} and get g_k = (1/M) Σ_{m=1}^{M} g_{k,m}
        w_{k+1} = w_k − γ g_k
    end for
end for
```

Output: w^K

- **Question:** What kind of problems might appear with (for example) Randk? The same non-zero coordinates on different devices can cause collisions.

PermK: do a dependent randomisation

Permutation compressor (dependent RandK)

Suppose that $d \geq n$ and $d = qn$, where $q \geq 1$ is an integer. Let $\pi = (\pi_1, \ldots, \pi_d)$ be a random permutation of $\{1, \ldots, d\}$. Then for each $i \in \{1, 2, \ldots, n\}$ we have the following compression operator

$$\mathcal{Q}_i(u) = n \cdot \sum_{j=q(i-1)+1}^{qi} u_{\pi_j} e_{\pi_j}.$$

Szlendak, R. et al. Permutation Compressors for Provably Faster Distributed Nonconvex Optimization

PermK: do a dependent randomisation

Permutation compressor (dependent RandK)

Suppose that $d \geq n$ and $d = qn$, where $q \geq 1$ is an integer. Let $\pi = (\pi_1, \ldots, \pi_d)$ be a random permutation of $\{1, \ldots, d\}$. Then for each $i \in \{1, 2, \ldots, n\}$ we have the following compression operator

$$\mathcal{Q}_i(u) = n \cdot \sum_{j=q(i-1)+1}^{qi} u_{\pi_j} e_{\pi_j}.$$

Szlendak, R. et al. Permutation Compressors for Provably Faster Distributed Nonconvex Optimization

- Friendly to centralised communications without a server.
- In the homogeneous case have the physics of cheap full gradient forwarding.

Biased compression

- Randomisation is good, but there's gap for improvement.

Biased compression

We call the (stochastic) operator $C(x)$ a shifted compression operator if for any $x \in \mathbb{R}^d$ it is fulfilled:

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

where $\delta \geq 0$.

Biased compression: examples

"Greedy" sparsification (selection of the largest modular components)

Consider an operator

$$\mathrm{Top}_k(x) = \sum_{i=d-k+1}^{d} x_{(i)} e_{(i)},$$

where k — some fixed number from the set $\{1, \ldots, d\}$ (the number of components of vector x that we pass; for example, we can choose $k = 1$), with the coordinates sorted modulo: $|x_{(1)}| \leq |x_{(2)}| \leq \ldots \leq |x_{(d)}|$, $(e_1, \ldots, e_d)$ — the standard basis in $\mathbb{R}^d$. It can be shown that this operator is a shifted compression with constant $\delta = \frac{d}{k} k$.

Alistarh D. et al. The convergence of sparsified gradient methods

Offset compression: examples

- Various examples of compressors (sparsifiers, rounding, etc):
  - Beznosikov A. et al. On Biased Compression for Distributed Learning
- A practical biased compressor based on iterative SVD decomposition:
  - Vogels T. et al. PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization

Biased compression: idea and proof in the case of 1 node

- Use the same approach as in the unbiased case (QGD).

Biased compression: idea and proof in the case of 1 node

- Use the same approach as in the unbiased case (`QGD`).
- Let's prove it in the case of a single node:

$$w_{k+1} = w_k - \gamma C(\nabla f(w_k)).$$

Let f have L-Lipschitz gradient and be μ-simply convex.

Biased compression: idea and proof in the case of 1 node

- Use the same approach as in the unbiased case (`QGD`).
- Let's prove it in the case of a single node:

$$w_{k+1} = w_k - \gamma C(\nabla f(w_k)).$$

  Let f have L-Lipschitz gradient and be μ-simply convex.
- Let's start by taking advantage of the Lispiness of the gradient:

$$\begin{aligned} f(w_{k+1}) &= f(w_k - \gamma C(\nabla f(w_k))) \\ &\leq f(w_k) + \langle \nabla f(w_k), -\gamma C(\nabla f(w_k)) \rangle + \frac{L}{2} \| -\gamma C(\nabla f(w_k)) \|^2 \\ &= f(w_k) - \gamma \langle C(\nabla f(w_k)), \nabla f(w_k) \rangle + \frac{\gamma^2 L}{2} \| C(\nabla f(w_k)) \|^2. \end{aligned}$$

Biased compression: idea and proof in the case of 1 node

- Compressor definition:

$$
\begin{aligned}
&\|\nabla f(w_k)\|^2 - 2\mathbb{E}_C\left[\langle C(\nabla f(w_k)), \nabla f(w_k)\rangle\right] + \mathbb{E}_C\left[\|C(\nabla f(w_k))\|^2\right] \\
&\qquad = \mathbb{E}_C\left[\|C(\nabla f(w_k)) - \nabla f(w_k)\|^2\right] \leq \left(1 - \frac{1}{\delta}\right)\|\nabla f(w_k)\|^2.
\end{aligned}
$$

Biased compression: idea and proof in the case of 1 node

- Compressor definition:

$$\|\nabla f(w_k)\|^2 - 2\mathbb{E}_C\left[\langle C(\nabla f(w_k)), \nabla f(w_k)\rangle\right] + \mathbb{E}_C\left[\|C(\nabla f(w_k))\|^2\right]$$
$$= \mathbb{E}_C\left[\|C(\nabla f(w_k)) - \nabla f(w_k)\|^2\right] \leq \left(1 - \frac{1}{\delta}\right)\|\nabla f(w_k)\|^2.$$

- From where:

$$-\gamma\mathbb{E}_C\left[\langle C(\nabla f(w_k)), \nabla f(w_k)\rangle\right] + \frac{\gamma}{2}\mathbb{E}_C\left[\|C(\nabla f(w_k))\|^2\right] \leq -\frac{\gamma}{2\delta}\|\nabla f(w_k)\|^2.$$

Biased compression: idea and proof in the case of 1 node

- From the previous two slides:

$$f(w_{k+1}) - \leq f(w_k) - \gamma \langle C(\nabla f(w_k)), \nabla f(w_k) \rangle + \frac{\gamma^2 L}{2} \| C(\nabla f(w_k)) \|^2.$$

$$-\gamma \mathbb{E}_C \left[\langle C(\nabla f(w_k)), \nabla f(w_k) \rangle \right] + \frac{\gamma}{2} \mathbb{E}_C \left[\| C(\nabla f(w_k)) \|^2 \right] \leq -\frac{\gamma}{2\delta} \| \nabla f(w_k) \|^2.$$

Biased compression: idea and proof in the case of 1 node

- From the previous two slides:

$$f(w_{k+1}) - \leq f(w_k) - \gamma \langle C(\nabla f(w_k)), \nabla f(w_k) \rangle + \frac{\gamma^2 L}{2} \| C(\nabla f(w_k)) \|^2.$$

$$-\gamma \mathbb{E}_C \left[\langle C(\nabla f(w_k)), \nabla f(w_k) \rangle \right] + \frac{\gamma}{2} \mathbb{E}_C \left[\| C(\nabla f(w_k)) \|^2 \right] \leq -\frac{\gamma}{2\delta} \| \nabla f(w_k) \|^2.$$

- Add, subtract $f(w^*)$ from both parts, and take the full expectation matrix:

$$\begin{aligned} \mathbb{E}\left[f(w_{k+1}) - f(w^*) \right] \leq & \mathbb{E}\left[f(w_k) - f(w^*) \right] - \frac{\gamma}{2} (1 - \gamma L) \mathbb{E}\left[\| C(\nabla f(w_k)) \|^2 \right] \\ & - \frac{\gamma}{2\delta} \mathbb{E}\left[\| \nabla f(w_k) \|^2 \right]. \end{aligned}$$

Biased compression: idea and proof in the case of 1 node

- From the previous two slides:

$$f(w_{k+1}) - \leq f(w_k) - \gamma \langle C(\nabla f(w_k)), \nabla f(w_k) \rangle + \frac{\gamma^2 L}{2} \| C(\nabla f(w_k)) \|^2.$$

$$-\gamma \mathbb{E}_C \left[\langle C(\nabla f(w_k)), \nabla f(w_k) \rangle \right] + \frac{\gamma}{2} \mathbb{E}_C \left[\| C(\nabla f(w_k)) \|^2 \right] \leq -\frac{\gamma}{2\delta} \| \nabla f(w_k) \|^2.$$

- Add, subtract $f(w^*)$ from both parts, and take the full expectation matrix:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*) \right] \leq \mathbb{E}\left[f(w_k) - f(w^*) \right] - \frac{\gamma}{2} (1 - \gamma L) \mathbb{E}\left[\| C(\nabla f(w_k)) \|^2 \right] - \frac{\gamma}{2\delta} \mathbb{E}\left[\| \nabla f(w_k) \|^2 \right].$$

- Let's take $\gamma \leq \frac{1}{L}$:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*) \right] \leq \mathbb{E}\left[f(w_k) - f(w^*) \right] - \frac{\gamma}{2\delta} \mathbb{E}\left[\| \nabla f(w_k) \|^2 \right].$$

Biased compression: idea and proof in the case of 1 node

- From the previous slide:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*)\right] \leq \mathbb{E}\left[f(w_k) - f(w^*)\right] - \frac{\gamma}{2\delta}\mathbb{E}\left[\|\nabla f(w_k)\|^2\right].$$

Biased compression: idea and proof in the case of 1 node

- From the previous slide:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*)\right] \leq \mathbb{E}\left[f(w_k) - f(w^*)\right] - \frac{\gamma}{2\delta}\mathbb{E}\left[\|\nabla f(w_k)\|^2\right].$$

- Strong convexity (or even the weaker PL condition):

$$2\mu(f(w_k) - f(w^*)) \leq \|\nabla f(w_k)\|^2.$$

Biased compression: idea and proof in the case of 1 node

- From the previous slide:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*)\right] \leq \mathbb{E}\left[f(w_k) - f(w^*)\right] - \frac{\gamma}{2\delta}\mathbb{E}\left[\|\nabla f(w_k)\|^2\right].$$

- Strong convexity (or even the weaker PL condition):

$$2\mu(f(w_k) - f(w^*)) \leq \|\nabla f(w_k)\|^2.$$

- Let's combine the previous two:

$$\mathbb{E}\left[f(w_{k+1}) - f(w^*)\right] \leq \left(1 - \frac{\gamma\mu}{\delta}\right)\mathbb{E}\left[f(w_k) - f(w^*)\right].$$

Biased compression: a theorem in the case of 1 node

Theorem (convergence of QGD with shifted compression in case of 1 node)

Let f μ be strongly convex (or PL) and have L-Lipschitz gradient, then QGD for one node with step $\gamma \leq 1/L$ and with biased compressor with parameter δ converges and is satisfied:

$$f(w_K) - f(w^*) \leq \left(1 - \frac{\gamma\mu}{\delta}\right)^K (f(w_0) - f(w^*)).$$

Beznosikov A. et al. On Biased Compression for Distributed Learning

Biased compression: it's not that simple

- Consider the following distributed problem with $M = 3$, $d = 3$ and local functions:

$$f_1(w) = \langle a, w\rangle^2 + \frac{1}{4}\|w\|^2,\ f_2(w) = \langle b, w\rangle^2 + \frac{1}{4}\|w\|^2,\ f_3(w) = \langle c, w\rangle^2 + \frac{1}{4}\|w\|$$

where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ и $c = (2, 2, -3)$.

Biased compression: it's not that simple

- Consider the following distributed problem with $M = 3$, $d = 3$ and local functions:

$$f_1(w) = \langle a, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_2(w) = \langle b, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_3(w) = \langle c, w \rangle^2 + \frac{1}{4}\|w\|$$

where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ и $c = (2, 2, -3)$.
- **Question:** where is her optimum?

Biased compression: it's not that simple

- Consider the following distributed problem with $M = 3$, $d = 3$ and local functions:

$$f_1(w) = \langle a, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_2(w) = \langle b, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_3(w) = \langle c, w \rangle^2 + \frac{1}{4}\|w\|$$

  where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ и $c = (2, 2, -3)$.
- **Question:** where is her optimum? $(0, 0, 0)$.

Biased compression: it's not that simple

- Consider the following distributed problem with $M = 3$, $d = 3$ and local functions:

$$f_1(w) = \langle a, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_2(w) = \langle b, w \rangle^2 + \frac{1}{4}\|w\|^2,\ f_3(w) = \langle c, w \rangle^2 + \frac{1}{4}\|w\|$$

where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ и $c = (2, 2, -3)$.

- **Question:** where is her optimum? $(0, 0, 0)$.
- Let the starting point $w_0 = (t, t, t)$ for some $t > 0$. Then the local gradients are:

$$\nabla f_1(w_0) = \frac{t}{2}(-11, 9, 9), \quad \nabla f_2(w_0) = \frac{t}{2}(9, -11, 9), \quad \nabla f_3(w_0) = \frac{t}{2}(9, 9, -11).$$

- **Question:** what will the `QGD` (gradient descent with compressions) step look like if we use $Top1$ compression?

Biased compression: it's not that simple

- Consider the following distributed problem with $M = 3$, $d = 3$ and local functions:

$$f_1(w) = \langle a, w \rangle^2 + \frac{1}{4}\|w\|^2, \; f_2(w) = \langle b, w \rangle^2 + \frac{1}{4}\|w\|^2, \; f_3(w) = \langle c, w \rangle^2 + \frac{1}{4}\|w\|$$

  where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ и $c = (2, 2, -3)$.
- **Question:** where is her optimum? $(0, 0, 0)$.
- Let the starting point $w_0 = (t, t, t)$ for some $t > 0$. Then the local gradients are:

$$\nabla f_1(w_0) = \frac{t}{2}(-11, 9, 9), \quad \nabla f_2(w_0) = \frac{t}{2}(9, -11, 9), \quad \nabla f_3(w_0) = \frac{t}{2}(9, 9, -11).$$

- **Question:** what will the `QGD` (gradient descent with compressions) step look like if we use *Top*1 compression?

$$w_1 = (t, t, t) + \eta \cdot \frac{11}{6}(t, t, t) = \left(1 + \frac{11\eta}{6}\right) w_0.$$

- We move away from the solution geometrically for any $\eta > 0$.

Offset compression: error compensation

- Let's try to remember what we didn't pass on in the communication process:

$$e_{1,m} = 0 + \gamma \nabla f_m(w_0) - C(0 + \gamma \nabla f_m(w_0)).$$

Offset compression: error compensation

- Let's try to remember what we didn't pass on in the communication process:
$$e_{1,m} = 0 + \gamma \nabla f_m(w_0) - C(0 + \gamma \nabla f_m(w_0)).$$

- And add this to future parcels:
$$C(e_{1,m} + \gamma \nabla f_m(w_1))$$

Offset compression: error compensation

- Let's try to remember what we didn't pass on in the communication process:

$$e_{1,m} = 0 + \gamma \nabla f_m(w_0) - C(0 + \gamma \nabla f_m(w_0)).$$

- And add this to future parcels:

$$C(e_{1,m} + \gamma \nabla f_m(w_1))$$

- In an arbitrary iteration, it is written as follows:

$$\begin{aligned} &\text{Parcel:} C(e_{k,m} + \gamma \nabla f_m(w_k)), \\ &e_{k+1,m} = e_{k,m} + \gamma \nabla f_m(w_k) - C(e_{k,m} + \gamma \nabla f_m(w_k)) \end{aligned}$$

Offset compression: error compensation

- Let's try to remember what we didn't pass on in the communication process:

$$e_{1,m} = 0 + \gamma \nabla f_m(w_0) - C(0 + \gamma \nabla f_m(w_0)).$$

- And add this to future parcels:

$$C(e_{1,m} + \gamma \nabla f_m(w_1))$$

- In an arbitrary iteration, it is written as follows:

$$\begin{aligned} &\text{Parcel:} C(e_{k,m} + \gamma \nabla f_m(w_k)), \\ &e_{k+1,m} = e_{k,m} + \gamma \nabla f_m(w_k) - C(e_{k,m} + \gamma \nabla f_m(w_k)) \end{aligned}$$

- This technique is called error compensation (error feedback).

PDF Stich S. et al. Sparsified SGD with memory

QGD with error feedback

Algorithm 1 QGD с error feedback

Input: step size $\gamma > 0$, starting point $w_0 \in \mathbb{R}^d$, starting errors $e_{0,m} = 0$ for all m from 1 to M, number of iterations K

for $k = 0, 1, \ldots, K-1$ **do**
  Send x_k to all workers ▷ by server
  for $m = 1, \ldots, M$ in parallel **do**
    Recieve w_k from the master ▷ by workers
    Compute $\nabla f(w_k)$ in w_k ▷ by workers
    Generate $g_{k,m} = C(e_{k,m} + \gamma \nabla f(w_k))$ ▷ by workers
    Вычислить $e_{k+1,m} = e_{k,m} + \gamma \nabla f_m(w_k) - g_{k,m}$ ▷ by workers
    Send $g_{k,m}$ to master ▷ by workers
  end for
  Recieve $g_{k,m}$ from all workers ▷ by server
  Compute $g_k = \frac{1}{M} \sum_{m=1}^{M} g_{k,m}$ ▷ by server
  $w_{k+1} = w_k - g_k$ ▷ by server
end for
Output: w_K

QGD with error feedback: convergence

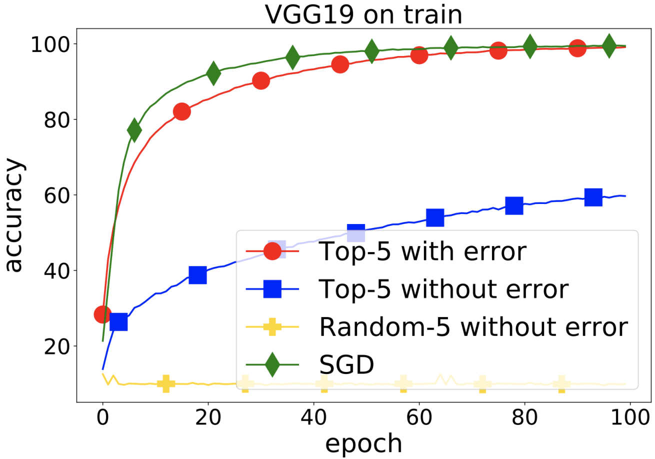


Figure: Accuracy during training of VGG19 on CIFAR10 using different compressors

QGD c error feedback: convergence

Theorem GD with error feedback

Let all local functions f_m be μ-simply convex and have L-Lipschitz gradient, then if $\eta \leq \frac{1}{28\delta L}$ then

$$\mathbb{E}\left[f(\tilde{w}_K) - f(x^*)\right] \leq \mathcal{O}\left(\delta L \|w_0 - w^*\|^2 \exp\left(-\frac{\gamma\mu K}{2}\right) + \frac{\delta}{\mu K} \cdot \frac{1}{M}\sum_{m=1}^{M} \|\nabla f_m(w^*)\|^2\right).$$

Stich S. and Karimireddy S. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication

Beznosikov A. et al. On Biased Compression for Distributed Learning

- Same problem as QGD – the second term in the evaluation should be eliminated

Biased compression: solving the plateau issue

- Idea like DIANA: memory + compression of difference

Algorithm 1 EF21 (sketch)

1: Each device m possesses a vector of "memory" $g_0^m = 0$

2: The server stores $h_0 = \frac{1}{M}\sum\limits_{m=1}^{M} h_0^m = 0$

3: Send a compressed version of the difference to the server $C(\nabla f_m(w^k) - h_k^m)$

4: Refreshing the memory $h_{k+1}^m = h_k^m + C(\nabla f_m(w^k) - h_k^m)$

5: Server compute $h_{k+1} = h_k + \frac{1}{M}\sum\limits_{m=1}^{M} C(\nabla f_m(w^k) - h_k^m)$

6: Update: $w_{k+1} = w_k - \gamma h_{k+1}$

Richtarik P. et al. EF21: A New, Simpler, Theoretically Better, and Practically Faster Error Feedback

Unbiased vs biased

- Best estimate on the number of communications for the unaccelerated method with unbiased compression (DIANA):

$$\mathcal{O}\left(\left[1 + \frac{\omega}{M}\right] \frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- Best estimate on the number of communications for the unaccelerated method with biased compression (EF-21):

$$\mathcal{O}\left([1 + \delta] \frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- It has already been discussed that these estimates are worse than for the baseline GD.

Unbiased vs biased

- Compressors compress information β times and it is typical that $\beta \geq \omega$ and $\beta \geq \delta$.

Unbiased vs biased

- Compressors compress information β times and it is typical that $\beta \geq \omega$ and $\beta \geq \delta$.
- Best estimate on the number of information for the unaccelerated method with unbiased compression (`DIANA`):

$$\mathcal{O}\left(\left[\frac{1}{\beta} + \frac{1}{M}\right] \frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- As discussed, the unbiased compressor provably improves the number of transmitted information.

Unbiased vs biased

- Compressors compress information β times and it is typical that $\beta \geq \omega$ and $\beta \geq \delta$.
- Best estimate on the number of information for the unaccelerated method with unbiased compression (DIANA):
$$\mathcal{O}\left(\left[\frac{1}{\beta} + \frac{1}{M}\right]\frac{L}{\mu}\log\frac{1}{\varepsilon}\right).$$
- As discussed, the unbiased compressor provably improves the number of transmitted information.
- The biased compressor has an estimate:
$$\mathcal{O}\left(\left[\frac{1}{\beta} + \frac{\delta}{\beta}\right]\frac{L}{\mu}\log\frac{1}{\varepsilon}\right).$$
- The biased compressor does not improve the number of transmitted information in the general case. **And this is an open question: how to see the theoretical superiority of biased operators, which often manifests itself in practice.**

The idea – more localised computing

- In the basic approach, communications occur every iteration.
- If counting (stochastic) gradients is much cheaper, why not count multiple times between communications.

Local gradient descent

Idea:

- Do local steps:
$$x_m^{k+1} = x_m^k - \gamma \nabla f_m(x_m^k, \xi_m^k).$$
- Every tth iteration, forward the current x_m^k to the server. The server averages $x^k = \frac{1}{M}\sum\limits_{m=1}^{M} x_m^k$, and forwards x^k to the devices. The devices update: $x_m^k = x^k$.
- A centralised SGD is a Local SGD with $K = 1$.

Convergence

- Typical convergence of this type of methods:

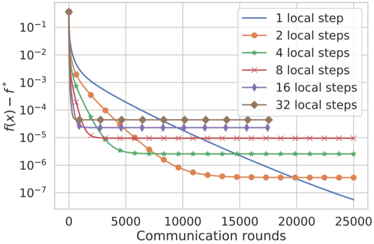


Figure: Convergence of the Local Method in practice for logistic regression.

- Faster in terms of communications, worse quality of marginal accuracy.

PDF Khaled A. et al. Tighter Theory for Local SGD on Identical and Heterogeneous Data

Convergence

- **Question:** Why does this effect occur? It occurs because of the heterogeneity of local data on different devices.
- In the upper estimates of the convergence of the method, this also shows up:

$$\mathcal{O}\left(\frac{\|x^0 - x^*\|^2}{\gamma T} + \frac{\gamma \sigma_{opt}^2}{M}\right),$$

where $\gamma \leq \mathcal{O}\left(\frac{1}{Lt}\right)$ – step, K – number of local iterations on each device, . The estimation is given for the case of convex and L-smooth f_m.

Khaled A. et al. Tighter Theory for Local SGD on Identical and Heterogeneous Data

- Moreover, the σ_{opt}^2 factor is not eliminated.

Glasgow M.R. et al. Sharp bounds for federated averaging (local sgd) and continuous perspective

Solving the problem

- **Question:** the problem of the local method is convergence to a neighbourhood. How can it be solved?

Solving the problem

- **Question:** the problem of the local method is convergence to a neighbourhood. How can it be solved?
- Regularisation of the local problem:

$$\tilde{f}_m(x) := f_m(x) + \frac{\lambda}{2}\|x - v\|^2,$$

where v is some reference point.

Karimireddy S. P. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning

What do we want to achieve, anyway?

- Lower bounds:

$$K = \Omega\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon}\right).$$

  L и μ – smoothness and strong convexity constants f.
- **Question:** which method will give these estimates?

What do we want to achieve, anyway?

- Lower bounds:

$$K = \Omega\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right).$$

L и μ – smoothness and strong convexity constants f.

- **Question:** which method will give these estimates? Distributed version of Nesterov's method with 1 local step between communications.
- We note that local steel methods for stochastic productions.

What do we want to achieve, anyway?

- Lower bounds:

$$K = \Omega\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right).$$

L и μ – smoothness and strong convexity constants f.

- **Question:** which method will give these estimates? Distributed version of Nesterov's method with 1 local step between communications.
- We note that local steel methods for stochastic productions.
- But even here, there is generally no improvement.

  Woodworth B. The Min-Max Complexity of Distributed Stochastic Convex Optimization with Intermittent Communication
- But there are productions where localised methods shoot out.

Data similarity

Data similarity

- A distributed learning task:

$$f(w) = \frac{1}{M}\sum_{m=1}^{M} f_m(w) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{i=1}^{N}\ell(w, z_i)\right],$$

where z_i is the sampling element (x_i, y_i), ℓ is the loss function (it has l and g sewn into it).

Data similarity

- A distributed learning task:

$$f(w) = \frac{1}{M}\sum_{m=1}^{M} f_m(w) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{i=1}^{N}\ell(w, z_i)\right],$$

where z_i is the sampling element $(x_i,\, y_i)$, ℓ is the loss function (it has l and g sewn into it).

- Suppose we can split the training sample evenly across devices (e.g., if cluster or collaborative computing on open data is used).

Data similarity

- A distributed learning task:

$$f(w) = \frac{1}{M}\sum_{m=1}^{M} f_m(w) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{i=1}^{N}\ell(w, z_i)\right],$$

where z_i is the sampling element $(x_i,\ y_i)$, ℓ is the loss function (it has l and g sewn into it).

- Suppose we can split the training sample evenly across devices (e.g., if cluster or collaborative computing on open data is used).
- This gives the similarity of the local loss functions.

Data similarity

- A distributed learning task:

$$f(w) = \frac{1}{M}\sum_{m=1}^{M} f_m(w) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{i=1}^{N}\ell(w, z_i)\right],$$

where z_i is the sampling element (x_i, y_i), ℓ is the loss function (it has l and g sewn into it).

- Suppose we can split the training sample evenly across devices (e.g., if cluster or collaborative computing on open data is used).
- This gives the similarity of the local loss functions.
- It is argued that for any w

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta.$$

Höfding's matrix inequality

Theorem (Höfding's matrix inequality)

Consider a finite sequence of random square matrices $\{X_i\}_{i=1}^N$. Let the matrices in this sequence be independent, Hermite and of dimension d. Suppose also that $\mathbb{E}[X_i] = 0$, и $X_i^2 \preceq A^2$ almost probably, where A is a non-random Hermite matrix. Then with probability $1 - p$ it is fulfilled that

$$\left\| \sum_{i=1}^N X_i \right\| \leq \sqrt{8N\|A^2\| \cdot \ln\left(d/p\right)}.$$

Tropp J. An introduction to matrix concentration inequalities

Tropp J. User-friendly tail bounds for sums of random matrices

Similarity parameter

- Local loss function:

$$f_m(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(w, z_i).$$

- ℓ – L-smooth (L-Lipschitz gradient), convex, twice differentiable function (e.g., quadratic or loggressive). Then we have $\nabla^2 \ell(w, z_i) \preceq LI$ for any w and z_i (here I is a unit matrix.).
- Let's distribute all data evenly over all nodes. $X_i = \frac{1}{N}\left[\nabla \ell(w, z_i) - \nabla f(w)\right]$. It is easy to check that all conditions of Höfding's matrix inequality are satisfied for it, in particular, $A^2 = \frac{4L^2}{N^2} I$.

Similarity parameter: bottom line

- As a result, we have

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta \sim \frac{L}{\sqrt{N}}.$$

Similarity parameter: bottom line

- As a result, we have

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta \sim \frac{L}{\sqrt{N}}.$$

- For quadratic problems, we can obtain an estimate of the form:

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta \sim \frac{L}{N}.$$

Hendrikx H. et al. Statistically Preconditioned Accelerated Gradient Method for Distributed Optimization

Similarity parameter: bottom line

- As a result, we have

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta \sim \frac{L}{\sqrt{N}}.$$

- For quadratic problems, we can obtain an estimate of the form:

$$\|\nabla^2 f_m(w) - \nabla^2 f(w)\| \leq \delta \sim \frac{L}{N}.$$

Hendrikx H. et al. Statistically Preconditioned Accelerated Gradient Method for Distributed Optimization

- In any case, the conclusion follows: the larger the local sample size, the smaller the similarity parameter (hessians are similar to each other).

Method in general terms

- Consider the mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by a strictly-convex function $\varphi(x)$:

$$V(x, y) = \varphi(x) - \varphi(y) - \langle \nabla \varphi(y); x - y \rangle.$$

Method in general terms

- Consider the mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by a strictly-convex function $\varphi(x)$:

$$V(x, y) = \varphi(x) - \varphi(y) - \langle \nabla \varphi(y); x - y \rangle.$$

- **Question:** Which method will result if $\varphi(x) = \frac{1}{2}\|x\|^2$?

Method in general terms

- Consider the mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by a strictly-convex function $\varphi(x)$:

$$V(x, y) = \varphi(x) - \varphi(y) - \langle \nabla \varphi(y); x - y \rangle.$$

- **Question:** Which method will result if $\varphi(x) = \frac{1}{2}\|x\|^2$? Gradient Descent.

Convergence in general

Definition (relative smoothness and strong convexity)

Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ is convex and doubly differentiable. Let us say that the function f is L_φ-smooth and μ_φ-strongly convex with respect to φ if for any $x \in \mathbb{R}^d$ it is fulfilled that

$$\mu_\varphi \nabla^2 \varphi(x) \preceq \nabla^2 f(x) \preceq L_\varphi \nabla^2 \varphi(x),$$

or equivalently for any $x, y \in \mathbb{R}^d$

$$\mu_\varphi V(x,y) \leq f(x) - f(y) - \langle \nabla f(y); x - y \rangle \leq L_\varphi V(x,y).$$

Lu H. et al. Relatively-Smooth Convex Optimization by First-Order Methods, and Applications

Convergence in general: a proof

- The first optimality condition for the mirror descent step:

$$\gamma \nabla f(w_k) + \nabla \varphi(w_{k+1}) - \nabla \varphi(w_k) = 0.$$

Convergence in general: a proof

- The first optimality condition for the mirror descent step:

$$\gamma \nabla f(w_k) + \nabla \varphi(w_{k+1}) - \nabla \varphi(w_k) = 0.$$

- From it (here w^* – optimum):

$$\langle \gamma \nabla f(w_k) + \nabla \varphi(w_{k+1}) - \nabla \varphi(w_k), w_{k+1} - w^* \rangle = 0.$$

$$\begin{aligned} \langle \gamma \nabla f(w_k), w^{k+1} - w^* \rangle &= \langle \nabla \varphi(w_k) - \nabla \varphi(w_{k+1}), w^{k+1} - w^* \rangle \\ &= V(w^*, w_k) - V(w^*, w_{k+1}) - V(w_{k+1}, w_k). \end{aligned}$$

(the last statement is called Pythagoras' theorem for Bregman divergences and is verified by definition)

Convergence in general: a proof

- The first optimality condition for the mirror descent step:

$$\gamma \nabla f(w_k) + \nabla \varphi(w_{k+1}) - \nabla \varphi(w_k) = 0.$$

- From it (here w^* – optimum):

$$\langle \gamma \nabla f(w_k) + \nabla \varphi(w_{k+1}) - \nabla \varphi(w_k), w_{k+1} - w^* \rangle = 0.$$

$$\begin{aligned} \langle \gamma \nabla f(w_k), w^{k+1} - w^* \rangle &= \langle \nabla \varphi(w_k) - \nabla \varphi(w_{k+1}), w^{k+1} - w^* \rangle \\ &= V(w^*, w_k) - V(w^*, w_{k+1}) - V(w_{k+1}, w_k). \end{aligned}$$

(the last statement is called Pythagoras' theorem for Bregman divergences and is verified by definition)

- Small permutations will give:

$$\begin{aligned} &\langle \gamma \nabla f(w_k), w_{k+1} - w_k \rangle + V(w_{k+1}, w_k) \\ &\quad = V(w^*, w_k) - V(w^*, w_{k+1}) - \langle \gamma \nabla f(w_k), w_k - w^* \rangle. \end{aligned}$$

Convergence in general: a proof

- Подставим $\gamma = \frac{1}{L_\varphi}$:

$$
\begin{aligned}
\langle \nabla f(w_k), w^{k+1} - w^k \rangle + L_\varphi V(w_{k+1}, w_k) \\
= & L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) \\
& - \langle \nabla f(w_k), w_k - w^* \rangle.
\end{aligned}
$$

Convergence in general: a proof

- Подставим $\gamma = \frac{1}{L_\varphi}$:

$$\begin{aligned}\langle \nabla f(w_k), w^{k+1} - w^k \rangle + L_\varphi V(w_{k+1}, w_k) \\ = & L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) \\ & - \langle \nabla f(w_k), w_k - w^* \rangle.\end{aligned}$$

- Let us use the definition of smoothness with respect to φ c $x = w_{k+1}$, $y = w_k$:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k); w_{k+1} - w_k \rangle + L_\varphi V(w_{k+1}, w_k).$$

Convergence in general: a proof

- Подставим $\gamma = \frac{1}{L_\varphi}$:

$$\langle \nabla f(w_k), w^{k+1} - w^k \rangle + L_\varphi V(w_{k+1}, w_k) \\ = L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) \\ - \langle \nabla f(w_k), w_k - w^* \rangle.$$

- Let us use the definition of smoothness with respect to φ c $x = w_{k+1}$, $y = w_k$:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k); w_{k+1} - w_k \rangle + L_\varphi V(w_{k+1}, w_k).$$

- Let's combine the previous two:

$$f(w_{k+1}) - f(w_k) \leq L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) - \langle \nabla f(w_k), w_k - w$$

Convergence in general: a proof

- From the previous slide:

$$
\begin{aligned}
f(w_{k+1}) - f(w_k) \leq & L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) \\
& - \langle \nabla f(w_k), w_k - w^* \rangle.
\end{aligned}
$$

Convergence in general: a proof

- From the previous slide:

$$f(w_{k+1}) - f(w_k) \leq L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) \\ - \langle \nabla f(w_k), w_k - w^* \rangle.$$

- Relative strong convexity:

$$\mu_\varphi V(w^*, w_k) \leq f(w^*) - f(w_k) - \langle \nabla f(w_k); w^* - w_k \rangle$$

Convergence in general: a proof

- From the previous slide:

$$f(w_{k+1}) - f(w_k) \leq L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) - \langle \nabla f(w_k), w_k - w^* \rangle.$$

- Relative strong convexity:

$$\mu_\varphi V(w^*, w_k) \leq f(w^*) - f(w_k) - \langle \nabla f(w_k); w^* - w_k \rangle$$

- Put the previous two together and move them around a bit:

$$f(w_{k+1}) - f(w^*) \leq (L_\varphi - \mu_\varphi) V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}).$$

Convergence in general: a proof

- From the previous slide:

$$f(w_{k+1}) - f(w_k) \leq L_\varphi V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}) - \langle \nabla f(w_k), w_k - w^* \rangle.$$

- Relative strong convexity:

$$\mu_\varphi V(w^*, w_k) \leq f(w^*) - f(w_k) - \langle \nabla f(w_k); w^* - w_k \rangle$$

- Put the previous two together and move them around a bit:

$$f(w_{k+1}) - f(w^*) \leq (L_\varphi - \mu_\varphi) V(w^*, w_k) - L_\varphi V(w^*, w_{k+1}).$$

- By virtue of the fact that w^* – optimum:

$$V(w^*, w_{k+1}) \leq \left(1 - \frac{\mu_\varphi}{L_\varphi}\right) V(w^*, w_k).$$

Convergence in general: theorem

Theorem (convergence of mirror descent)

Let φ and f satisfy the definition above, then the mirror descent with step $\gamma = \frac{1}{L_\varphi}$ converges and is satisfied:

$$V(w^*, w_K) \leq \left(1 - \frac{\mu_\varphi}{L_\varphi}\right)^K V(w^*, w_0).$$

Lu H. et al. Relatively-Smooth Convex Optimization by First-Order Methods, and Applications

Method for data similarity

- Mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by the function $\varphi(x)$ (here we need to require that f_1 is convex):

$$\varphi(x) = f_1(x) + \frac{\delta}{2}\|x\|^2.$$

The function f_1 is stored on the server.

Method for data similarity

- Mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by the function $\varphi(x)$ (here we need to require that f_1 is convex):

$$\varphi(x) = f_1(x) + \frac{\delta}{2}\|x\|^2.$$

The function f_1 is stored on the server.

- **Question:** What is the number of communications that occur in K iterations of such a mirror descent?

Method for data similarity

- Mirror descent:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by the function $\varphi(x)$ (here we need to require that f_1 is convex):

$$\varphi(x) = f_1(x) + \frac{\delta}{2}\|x\|^2.$$

The function f_1 is stored on the server.

- **Question:** What is the number of communications that occur in K iterations of such a mirror descent? K of communications (the number of gradient counts ∇f), computing $\arg\min$ requires only computations on the server.

Method for data similarity

Algorithm 1 `Mirror descent for data similarity`

Input: Step size $\gamma > 0$, strating point $w^0 \in \mathbb{R}^d$, number of iterations K

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: Send x_k to workers ▷ by server
3: **for** $m = 1, \ldots, M$ in parallel **do**
4: Receive w_k from server ▷ by workers
5: Compute $\nabla f_m(w_k)$ in w_k ▷ by workers
6: Send $\nabla f_m(w_k)$ to server ▷ by workers
7: **end for**
8: Receive $\nabla f_m(w_k)$ from workers ▷ by server
9: Compute $\nabla f(w_k) = \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w_k)$ ▷ by server
10: $w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma \langle \nabla f(w_k), x \rangle + V(w, w_k)\right)$ ▷ by server
11: **end for**

Output: w_K

Convergence for data similarity: a proof

- Recall that convergence is defined in terms of constants from the relation:

$$\mu_\varphi \nabla^2 \varphi(w) \preceq \nabla^2 f(w) \preceq L_\varphi \nabla^2 \varphi(w),$$

Convergence for data similarity: a proof

- Recall that convergence is defined in terms of constants from the relation:

$$\mu_\varphi \nabla^2 \varphi(w) \preceq \nabla^2 f(w) \preceq L_\varphi \nabla^2 \varphi(w),$$

- In our case

$$\mu_\varphi \left(\delta I + \nabla^2 f_1(w)\right) \preceq \nabla^2 f(w) \preceq L_\varphi \left(\delta I + \nabla^2 f_1(w)\right)$$

Convergence for data similarity: a proof

- Recall that convergence is defined in terms of constants from the relation:

$$\mu_\varphi \nabla^2 \varphi(w) \preceq \nabla^2 f(w) \preceq L_\varphi \nabla^2 \varphi(w),$$

- In our case

$$\mu_\varphi \left(\delta I + \nabla^2 f_1(w)\right) \preceq \nabla^2 f(w) \preceq L_\varphi \left(\delta I + \nabla^2 f_1(w)\right)$$

- Let us find L_φ:

$$\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \le \delta \Rightarrow \nabla^2 f(w) - \nabla^2 f_1(w) \preceq \delta I$$
$$\Rightarrow \nabla^2 f(w) \preceq \delta I + \nabla^2 f_1(w) \Rightarrow L_\varphi = 1.$$

Convergence for data similarity: a proof

- Recall that convergence is defined in terms of constants from the relation:

$$\mu_\varphi \nabla^2 \varphi(w) \preceq \nabla^2 f(w) \preceq L_\varphi \nabla^2 \varphi(w),$$

- In our case

$$\mu_\varphi \left(\delta I + \nabla^2 f_1(w)\right) \preceq \nabla^2 f(w) \preceq L_\varphi \left(\delta I + \nabla^2 f_1(w)\right)$$

- Let us find L_φ:

$$\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta \Rightarrow \nabla^2 f(w) - \nabla^2 f_1(w) \preceq \delta I$$
$$\Rightarrow \nabla^2 f(w) \preceq \delta I + \nabla^2 f_1(w) \Rightarrow L_\varphi = 1.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

- From $\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta$ we have:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \delta I.$$

- Connecting two previous estimates:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

- And then

$$\nabla^2 f_1(w) + \delta I \preceq \frac{2\delta + \mu}{\mu}\nabla^2 f(w) \Rightarrow \mu_\varphi = \frac{\mu}{2\delta + \mu}.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu} \nabla^2 f(w) - \delta I.$$

- From $\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta$ we have:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \delta I.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

- From $\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta$ we have:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \delta I.$$

- Connecting two previous estimates:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

- From $\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta$ we have:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \delta I.$$

- Connecting two previous estimates:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \frac{2\delta}{\mu}\nabla^2 f(w) - \delta I.$$

- And then

$$\nabla^2 f_1(w) + \delta I \preceq \frac{2\delta + \mu}{\mu}\nabla^2 f(w) \Rightarrow \mu_\varphi = \frac{\mu}{2\delta + \mu}.$$

Convergence for data similarity: a proof

- Let us find μ_φ. From the strongly convexity of the function f:

$$\mu I \preceq \nabla^2 f(w) \Rightarrow \delta I \preceq \frac{2\delta}{\mu} \nabla^2 f(w) - \delta I.$$

- From $\|\nabla^2 f_1(w) - \nabla^2 f(w)\| \leq \delta$ we have:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \delta I.$$

- Connecting two previous estimates:

$$\nabla^2 f_1(w) - \nabla^2 f(w) \preceq \frac{2\delta}{\mu} \nabla^2 f(w) - \delta I.$$

- And then

$$\nabla^2 f_1(w) + \delta I \preceq \frac{2\delta + \mu}{\mu} \nabla^2 f(w) \Rightarrow \mu_\varphi = \frac{\mu}{2\delta + \mu}.$$

Convergence for data similarity: theorem

Theorem (convergence for data similarity)

Let f be strongly convex, f_i be convex, and ℓ be smooth, and $\varphi(w) = f_1(w) + \delta\|w\|\|^2$, then mirror descent with step $\gamma = 1$ converges and is satisfied:

$$V(w^*, w_K) \leq \left(1 - \frac{\mu}{\mu + 2\delta}\right)^K V(w^*, w_0).$$

Convergence for data similarity: theorem

Theorem (convergence for data similarity)

Let f be strongly convex, f_i be convex, and ℓ be smooth, and $\varphi(w) = f_1(w) + \delta \|w\||^2$, then mirror descent with step $\gamma = 1$ converges and is satisfied:

$$V(w^*, w_K) \leq \left(1 - \frac{\mu}{\mu + 2\delta}\right)^K V(w^*, w_0).$$

- It means that if we need to achieve an accuracy ε $(V(w^*, w_K) \sim \varepsilon)$, then we need to

$$K = \left(\left[1 + \frac{\delta}{\mu}\right] \log \frac{V(w^*, w_0)}{\varepsilon}\right) \text{ communications.}$$

Better?

- The estimate on the number of communications under data similarity:

$$K = \mathcal{O}\left(\left[1 + \frac{\delta}{\mu}\right] \log \frac{1}{\varepsilon}\right).$$

- The estimate on the number of communications for the basic distributed gradient descent:

$$K = \mathcal{O}\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

Better?

- The estimate on the number of communications under data similarity:

$$K = \mathcal{O}\left(\left[1 + \frac{\delta}{\mu}\right] \log \frac{1}{\varepsilon}\right).$$

- The estimate on the number of communications for the basic distributed gradient descent:

$$K = \mathcal{O}\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right).$$

- Recall that $\delta \sim \frac{L}{\sqrt{N}}$, i.e., there can be a significant improvement.

Another look at the mirror escapement

- Mirror descent with $\gamma = 1$:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\langle \nabla f(w_k), w \rangle + V(w, w_k) \right),$$

where $V(x, y)$ is the Bregman divergence generated by the function $\varphi(x)$:

$$\varphi(x) = f_1(x) + \frac{\delta}{2}\|x\|^2.$$

- Substitute $\varphi(x)$:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(f_1(w) + \langle \nabla f(w_k) - \nabla f_1(w_k), w \rangle + \frac{\delta}{2}\|w - w_k\|^2 \right).$$

- Or:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\frac{1}{\delta} f_1(w) + \frac{1}{2} \left\| w - \left(w_k - \frac{1}{\delta}(\nabla f(w_k) - \nabla f_1(w_k)) \right) \right\|^2 \right)$$

Bottom line about mirror descent

- The idea of regularizing the local subproblem came up.
- The idea of sliding $\approx$ proximal method with inexactness.
- Proximal method for a composite target function $g_1(w) + g_2(w)$:

$$w_{k+1} = \arg\min_{w \in \mathbb{R}^d} \left(\gamma g_2(w) + \frac{1}{2} \| w - (w_k - \gamma g_1(w_k)) \|^2 \right).$$

- In our case, $g_1 = f - f_1$, $g_2 = f_1$.

Better?

- We get:

$$K = \mathcal{O}\left(\left[1 + \frac{\delta}{\mu}\right] \log \frac{1}{\varepsilon}\right).$$

- But there's also an accelerated gradient method that gives estimates:

$$K = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon}\right).$$

- It is not clear which is better. Moreover, is it possible to speed up the method for a problem with data similarity?
- Lower estimates are also available for the data similarity problem:

$$K = \Omega\left(\sqrt{1 + \frac{\delta}{\mu}} \log \frac{1}{\varepsilon}\right),$$

i.e. a possible acceleration is assumed.

Arjevani Y. and Shamir O. Communication complexity of distributed convex learning and optimization

Optimal algorithm

- This problem has quite a history:

Reference	Communication complexity	Local gradient complexity	Order	Limitations
DANE [42]	$\mathcal{O}\left(\frac{\delta^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$	— (2)	1st	quadratic
DiSCO [51]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}(\log\frac{1}{\varepsilon} + C^2\Delta F_0)\log\frac{L}{\mu}\right)$	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}(\log\frac{1}{\varepsilon} + C^2\Delta F_0)\log\frac{L}{\mu}\right)$	2nd	C - self-concordant (3)
AIDE [40]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\log\frac{L}{\delta}\right)$	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\log\frac{L}{\delta}\right)$ (4)	1st	quadratic
DANE-LS [50]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\frac{\delta^{3/2}}{\mu^{3/2}}\log\frac{1}{\varepsilon}\right)$ (5)	1st/2nd	quadratic (6)
DANE-HB [50]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$ (5)	1st/2nd	quadratic (6)
SONATA [45]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	— (2)	1st	decentralized
SPAG [21]	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$ (1)	— (2)	1st	M - Lipshitz hessian
DiRegINA [12]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon} + \sqrt{\frac{M\delta R_0}{\mu}}\right)$	— (2)	2nd	M -Lipshitz hessian
ACN [1]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon} + \sqrt[3]{\frac{M\delta R_0}{\mu}}\right)$	— (2)	2nd	M -Lipshitz hessian
Acc SONATA [46]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\log\frac{\delta}{\mu}\right)$	— (2)	1st	decentralized
This paper	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	1st	

In particular, the mirror descent approach with unusual divergence is called DANE.

Optimal algorithm

- This problem has quite a history:

Reference	Communication complexity	Local gradient complexity	Order	Limitations
DANE [42]	$\mathcal{O}\left(\frac{\delta^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$	— [2]	1st	quadratic
DiSCO [51]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}(\log\frac{1}{\varepsilon} + C^2\Delta F_0)\log\frac{L}{\mu}\right)$	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}(\log\frac{1}{\varepsilon} + C^2\Delta F_0)\log\frac{L}{\mu}\right)$	2nd	C - self-concordant [3]
AIDE [40]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\log\frac{L}{\delta}\right)$	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\log\frac{L}{\delta}\right)$ [4]	1st	quadratic
DANE-LS [50]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\frac{\delta^{3/2}}{\mu^{3/2}}\log\frac{1}{\varepsilon}\right)$ [5]	1st/2nd	quadratic [6]
DANE-HB [50]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$ [5]	1st/2nd	quadratic [6]
SONATA [45]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right)$	— [2]	1st	decentralized
SPAG [21]	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$ [1]	— [2]	1st	M - Lipshitz hessian
DiRegINA [12]	$\mathcal{O}\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon} + \sqrt{\frac{M\delta R_0}{\mu}}\right)$	— [2]	2nd	M -Lipshitz hessian
ACN [1]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon} + \sqrt[3]{\frac{M\delta R_0}{\mu}}\right)$	— [2]	2nd	M -Lipshitz hessian
Acc SONATA [46]	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\log\frac{\delta}{\mu}\right)$	— [2]	1st	decentralized
This paper	$\mathcal{O}\left(\sqrt{\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	1st	

In particular, the mirror descent approach with unusual divergence is called DANE.

- The optimal algorithm was proposed in 2022:

PDF Kovalev D. et al. Optimal Gradient Sliding and its Application to Distributed Optimization Under Similarity

Optimal algorithm

For the problem

$$f(w) = g_1(w) + g_2(w),$$

with $g_1 = f - f_1$ и $g_2 = f_1$.

Algorithm 2 Accelerated Extragradient

1: **Input:** $w^0 = w_f^0 \in \mathbb{R}^d$
2: **Parameters:** $\tau \in (0,1]$, $\eta, \theta, \alpha > 0$, $K \in \{1, 2, \dots\}$
3: **for** $k = 0, 1, 2, \dots, K-1$ **do**
4: $w_g^k = \tau w^k + (1-\tau) w_f^k$
5: $w_f^{k+1} \approx \operatorname*{argmin}_{x \in \mathbb{R}^d} \left[\left\langle \nabla g_1(w_g^k), w - w_g^k \right\rangle + \frac{1}{2\theta} \| w - w_g^k \|^2 + g_2(w) \right]$
6: $w^{k+1} = w^k + \eta\alpha(w_f^{k+1} - w^k) - \eta \nabla g(w_f^{k+1})$
7: **end for**
8: **Output:** w^K

Three ideas

- 1st idea – acceleration
- 2nd idea – sliding
- 3d idea – extragradient
- The first two ideas are clear, the key is the third idea.

Optimal. But may be better?

- **Question:** let's forget about the distribution for 1 slide, and remember is Nesterov's method always optimal?

Optimal. But may be better?

- **Question:** let's forget about the distribution for 1 slide, and remember is Nesterov's method always optimal? No, if we consider the specificity that the target function can be of the sum species $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$.
- **Question:** what then is the optimal method? what are its upper estimates of convergence?

Optimal. But may be better?

- **Question:** let's forget about the distribution for 1 slide, and remember is Nesterov's method always optimal? No, if we consider the specificity that the target function can be of the sum species $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$.
- **Question:** what then is the optimal method? what are its upper estimates of convergence?
- The method is called Katyusha, and it has the following upper bound on convergence (oracle complexity by calling f_i):

$$\mathcal{O}\left(\left[n + \sqrt{n\frac{L}{\mu}}\right] \log \frac{1}{\varepsilon}\right).$$

PDF Allen-Zhu Z. Katyusha: the first direct acceleration of stochastic gradient methods

Question: What is the upper bound on oracular complexity for Nesterov's method?

Optimal. But may be better?

- **Question:** let's forget about the distribution for 1 slide, and remember is Nesterov's method always optimal? No, if we consider the specificity that the target function can be of the sum species $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$.
- **Question:** what then is the optimal method? what are its upper estimates of convergence?
- The method is called Katyusha, and it has the following upper bound on convergence (oracle complexity by calling f_i):

$$\mathcal{O}\left(\left[n + \sqrt{n\frac{L}{\mu}}\right]\log\frac{1}{\varepsilon}\right).$$

PDF Allen-Zhu Z. Katyusha: the first direct acceleration of stochastic gradient methods

Question: What is the upper bound on oracular complexity for Nesterov's method? $\mathcal{O}\left(n\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$

Variance reduction for similarity

- The idea behind the variance reduction method:

$$\nabla f(x)$$
$$\downarrow$$
$$\nabla f_i(x) - \nabla f_i(w) + \nabla f(w),$$

where i is generated randomly at each iteration from $[n]$, w – a reference point that is updated infrequently (randomly or deterministically).

- The idea of variance reduction method for data similarity:

$$\nabla f(x) - \nabla f_1(x)$$
$$\downarrow$$
$$\nabla f_i(x) - \nabla f_i(w) + \nabla f(w) - f_1(x),$$

where i is generated randomly at each iteration from $[M]$.

Variance reduction for similarity

Beznosikov A. & Gasnikov A. Compression and data similarity: Combination of two techniques for communication-efficient solving of distributed variational inequalities

Beznosikov A. & Gasnikov A. Similarity, Compression and Local Steps: Three Pillars of Efficient Communications for Distributed Variational Inequalities

Khaled A. & Jin C. Faster federated optimization under second-order similarity

- Previous estimate:

$$\mathcal{O}\left(M\sqrt{1+\frac{\delta}{\mu}}\log\frac{1}{\varepsilon}\right).$$

- We can achieve:

$$\mathcal{O}\left(\left[M+\frac{\delta^2}{\mu^2}\right]\log\frac{1}{\varepsilon}\right) \quad \text{or} \quad \mathcal{O}\left(\left[M+\sqrt{M}\frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right) \quad \text{or}$$

$$\mathcal{O}\left(\left[M+M^{3/4}\sqrt{\frac{\delta}{\mu}}\right]\log\frac{1}{\varepsilon}\right).$$