QLSD: Quantized Langevin stochastic dynamics for Bayesian federated learning

Lagrange Mathematical and Computing Research Center, Paris

Vincent Plassier, Joint work with Maxime Vono, Alain Durmus, Aymeric Dieuleveut, Eric Moulines

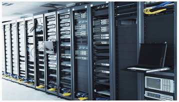
July 15, 2021

Federated Learning context

What is Federated Learning?

- Actors collaborate to learn a model
- Data privacy
- Communication constraints
- Each client has a different data distribution

Server



$$\theta_\star = \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{b} U_i(\theta)$$

$$U_i(\theta) = \sum_{(x,y) \in \mathcal{D}_i} \ell(f_\theta(x), y)$$

User 1
Local data

• • •

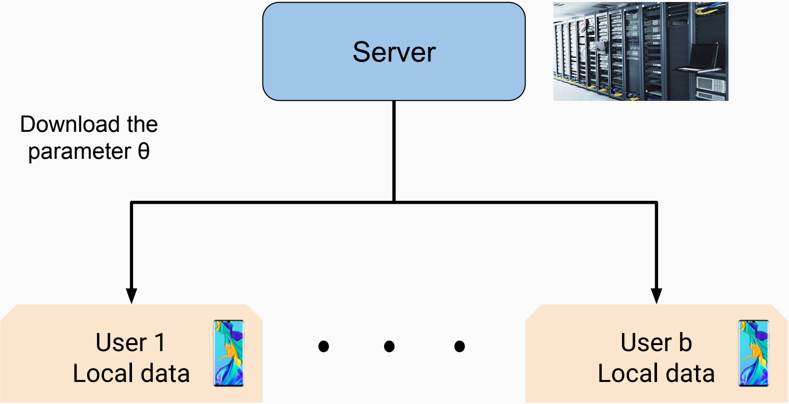
User b
Local data

$\mathcal{D}_1$ $\arg\min U_1(\theta)$?

$\mathcal{D}_b$ $\arg\min U_b(\theta)$?

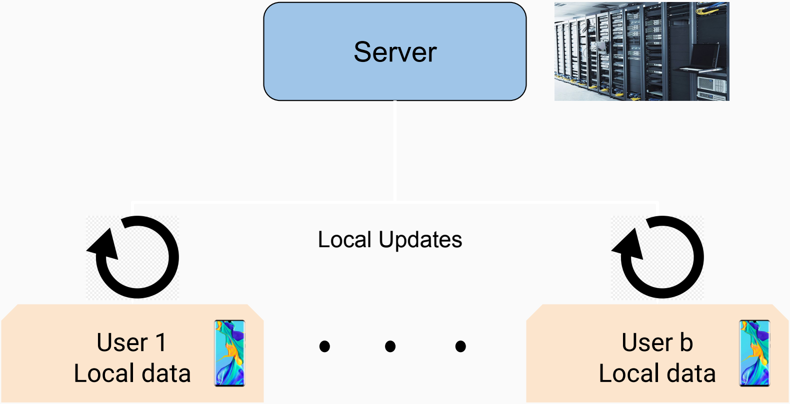
What is Federated Learning?

- Actors collaborate to learn a model
- Data privacy
- Communication constraints
- Each client has a different data distribution



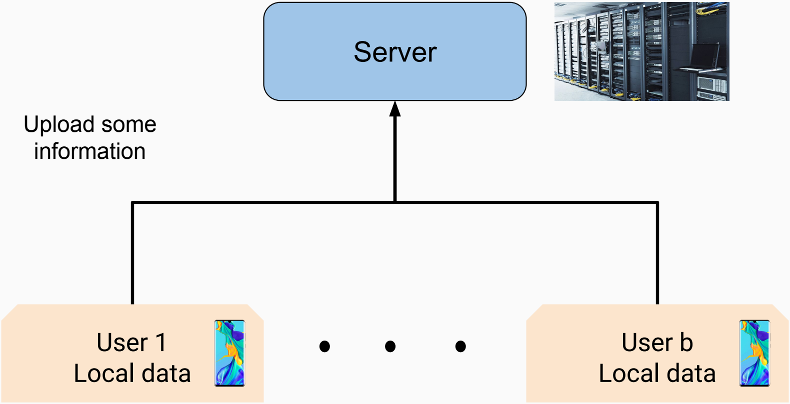
What is Federated Learning?

- Actors collaborate to learn a model
- Data privacy
- Communication constraints
- Each client has a different data distribution



What is Federated Learning?

- Actors collaborate to learn a model
- Data privacy
- Communication constraints
- Each client has a different data distribution



Why do we use Federated Learning?

- Communication constraints

Why do we use Federated Learning?

- Communication constraints

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Depth
Xception	88 MB	0.790	0.945	126
InceptionV3	92 MB	0.779	0.937	159
ResNet50	98 MB	0.749	0.921	-
ResNet152	232 MB	0.766	0.931	-
MobileNet	16 MB	0.704	0.895	88
VGG16	528 MB	0.713	0.901	23
VGG19	549 MB	0.713	0.900	26

Table 1: Keras Webpage

Why do we use Federated Learning?

- Communication constraints
- Data ownership
- Learn from each client dataset

Why do we use Federated Learning?

- Communication constraints
- Data ownership
- Learn from each client dataset

What is the difference with distributed learning?

- Non-IID data
- Unbalanced data: unequal amount of data on each node
- Massively distributed data
- limited communication

Method	**FedAvg**	**QSGD**
Article	McMahan et al. (2017)	Alistarh et al. (2017)
Num local iter	$\mathbb{N}^*$	1
Compression	No	Yes

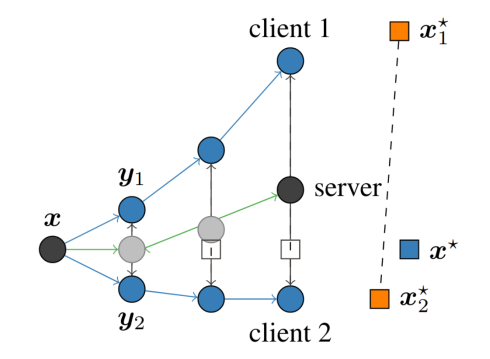


Figure 1: Karimireddy et al. (2020)

Method	**FedAvg**	**QSGD**
Article	McMahan et al. (2017)	Alistarh et al. (2017)
Num local iter	$\mathbb{N}^*$	1
Compression	No	Yes

Algorithm 1: `QSGD`

Initialize $\theta_0 \in \mathbb{R}^d$
for $k = 0$ to $K - 1$ **do**
  // In parallel on the b clients
  for $i \in \{1, \dots, b\}$ **do**
    Send $\mathscr{C}\left(\widehat{\nabla U_i(\theta_k)}\right)$
  // On the central server
  Set $\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\widehat{\nabla U_i(\theta_k)}\right)$
Output: θ_K

Compression operator.

$$\underbrace{\widehat{\nabla U_i} = \begin{pmatrix} [\widehat{\nabla U_{i1}}] \\ \vdots \\ [\widehat{\nabla U_{id}}] \end{pmatrix}}_{32d \text{ bits}} \longrightarrow \boxed{\mathscr{C}} \longrightarrow \underbrace{\mathscr{C}(\widehat{\nabla U_i})}_{d \text{ bits}}$$

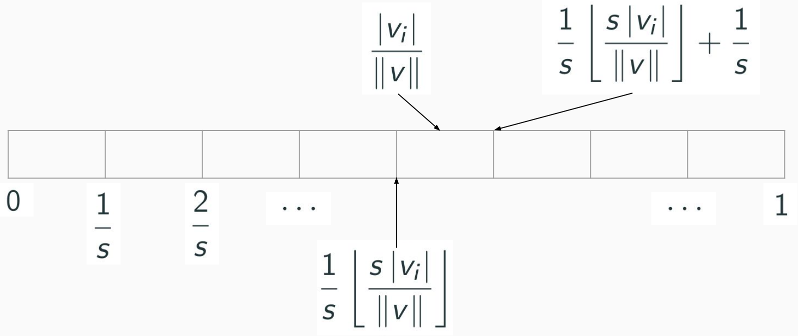
Assumptions.

$$\exists\, \omega > 0, \forall x \in \mathbb{R}^d, \qquad \begin{cases} \mathbb{E}\left[\mathscr{C}(x)\right] = x \\ \mathbb{E}\left[\|\mathscr{C}(x) - x\|^2\right] \leq \omega \|x\|^2 \end{cases}$$

An example of unbiased compressor.

$$\mathscr{C}^{(s)}(v) = \|v\| \cdot \mathrm{sign}(v) \cdot \xi_s(v)$$

$$\xi_s(v) = \left(\frac{1}{s} \left\lfloor \frac{s\,|v_i|}{\|v\|} \right\rfloor + \frac{1}{s} 1_{\text{with proba } \frac{s|v_i|}{\|v\|} - \left\lfloor \frac{s|v_i|}{\|v\|} \right\rfloor} \right)_{i=1}^{d}$$



QSGD result.

- $U_i = \sum_{j=1}^{N} U_{i,j}$ strongly-convex $\rightsquigarrow \theta_\star = \arg\min \sum_{i=1}^{b} U_i$
- $\|\nabla U_{i,j}(\theta') - \nabla U_{i,j}(\theta)\| \leq L\|\theta' - \theta\|$
- ${\sigma_\star}^2 = \sum_{i=1}^{b} \mathbb{E}\left[\|\widehat{\nabla U_i}(\theta_\star) - \nabla U_i(\theta_\star)\|^2\right]$
- $\beta^2 = \sum_{i=1}^{b} \|\nabla U_i(\theta_\star)\|^2$

Convergence result:

$$\mathbb{E}\left[\|\theta_k - \theta_\star\|^2\right] \leq (1-\gamma\mu)^k \left(\|\theta_0 - \theta_\star\|^2 + 2C\gamma^2\beta^2\right) + \frac{2\gamma}{\mu}\left((\omega+1){\sigma_\star}^2 + \omega\beta^2\right)$$

What is a memory term?

- Mishchenko et al. (2019) introduces the "memory term"
- Decreases the bias when the datasets are heterogeneous
- Mechanism to learn $\nabla U_i(\theta_\star) \neq 0$

In practice. **Update the memory term at each iteration**

- On the clients

$$\text{Send } \mathscr{C}\left(\widehat{\nabla U_i(\theta_k)} - \eta_k^{(i)}\right)$$

$$\eta_{k+1}^{(i)} = \eta_k^{(i)} + \alpha \mathscr{C}\left(\widehat{\nabla U_i(\theta_k)} - \eta_k^{(i)}\right)$$

- On the central server

$$\text{Update } \theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\widehat{\nabla U_i(\theta_k)} - \eta_k^{(i)}\right) + \gamma \eta_k$$

$$\eta_{k+1} = \eta_k + \alpha \sum_{i=1}^{b} \mathscr{C}\left(\widehat{\nabla U_i(\theta_k)} - \eta_k^{(i)}\right)$$

What is a memory term?

- Mishchenko et al. (2019) introduces the "memory term"
- Decreases the bias when the datasets are heterogeneous
- Mechanism to learn $\nabla U_i(\theta_\star) \neq 0$

Convergence results:

Client	$\eta_k^{(i)} \to \nabla U_i(\theta_\star)$
Server	$\eta_k \to \nabla U(\theta_\star) = \sum_{i=1}^{b} \nabla U_i(\theta_\star)$

What is a memory term?

- Mishchenko et al. (2019) introduces the "memory term"
- Decreases the bias when the datasets are heterogeneous
- Mechanism to learn $\nabla U_i(\theta_\star) \neq 0$

How it works?

- Each device has its own memory term $\eta_k^{(i)} \in \mathbb{R}^d$
- Since $\mathbb{E}\left[\|\mathscr{C}(x) - x\|^2\right] \leq \omega \|x\|^2 \Rightarrow$ we want to transfert $\|x\| \ll 1$ to accelerate convergence
- Transfert $\mathscr{C}\left(\underbrace{\widehat{\nabla U_i(\theta_k)} - \eta_k^{(i)}}_{\text{tends to zero}}\right)$ instead of $\mathscr{C}\left(\underbrace{\widehat{\nabla U_i(\theta_k)}}_{\neq 0 \text{ due to heterogeneity}}\right)$

QSGD with memory term.

- $U_i = \sum_{j=1}^{N} U_{i,j}$ strongly-convex $\rightsquigarrow \theta_\star = \arg\min \sum_{i=1}^{b} U_i$
- $\|\nabla U_{i,j}(\theta') - \nabla U_{i,j}(\theta)\| \leq L\|\theta' - \theta\|$
- ${\boldsymbol{\sigma}_\star}^2 = \sum_{i=1}^{b} \mathbb{E}\left[\|\widehat{\nabla U_i}(\theta_\star) - \nabla U_i(\theta_\star)\|^2\right]$
- $\boldsymbol{\beta}^2 = \sum_{i=1}^{b} \|\nabla U_i(\theta_\star)\|^2$

Convergence result:

$$\mathbb{E}\left[\|\theta_k - \theta_\star\|^2\right] \leq (1-\gamma\mu)^k \left(\|\theta_0 - \theta_\star\|^2 + 2C\gamma^2\boldsymbol{\beta}^2\right) + \frac{2\gamma}{\mu}(\boldsymbol{\omega}+1)\left({\sigma_\star}^2 + 4\underbrace{\alpha^2 C}_{\text{replace } \boldsymbol{\beta}^2}\right)$$

Objectives:

- Sample parameters $(\theta_k)_{k\in\mathbb{N}}$ in large dimension

$$\theta_k \sim \pi(\cdot \mid \mathcal{D}) = Z_\pi^{-1} \cdot \prod_{i=1}^{b} \mathrm{e}^{-U_i(\cdot)}$$

- Obtain estimators from the sampled points

Example:

- θ can be a neural network parameter

$$U_i(\theta) = \sum_{(x,y)\in\mathcal{D}_i} \ell(f_\theta(x), y)$$

- (MAP) in optimization

$$\theta_\star = \arg\min U = \arg\max \pi(\cdot \mid \mathcal{D})$$

But what can I do with those samples?

Instead of having one sample we have a family of samples

- Compute expectation $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}$ based on the samples $\theta_0, \dots, \theta_{K-1}$

$$\mathbb{E}_{\theta \sim \pi(\cdot|\mathcal{D})}[\mathcal{T}(\theta)] \simeq \frac{1}{K} \sum_{k=0}^{K-1} \mathcal{T}(\theta_k)$$

But what can I do with those samples?

Instead of having one sample we have a family of samples

- Compute expectation $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}$ based on the samples $\theta_0, \dots, \theta_{K-1}$

$$\mathbb{E}_{\theta \sim \pi(\cdot|\mathcal{D})}[\mathcal{T}(\theta)] \simeq \frac{1}{K} \sum_{k=0}^{K-1} \mathcal{T}(\theta_k)$$

- Compute predictive distribution on a new test example

$$\underbrace{\pi(y|x, \mathcal{D})}_{\text{predictive distribution}} = \int_\theta \underbrace{\pi(y|x, \theta)}_{\text{likelihood}} \underbrace{\pi(\theta|\mathcal{D})}_{\text{posterior}} \mathrm{d}\theta$$

$$\simeq \frac{1}{K} \sum_{k=0}^{K-1} \pi(y|x, \theta_k)$$

Quantized Langevin stochastic dynamic

- Langevin dynamic.

$$\mathrm{d}\vartheta_t = -\sum_{i=1}^{b} \nabla U_i(\vartheta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

- Langevin dynamic.

$$\mathrm{d}\vartheta_t = -\sum_{i=1}^{b} \nabla U_i(\vartheta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

- Invariant distribution.

$$\pi : \theta \mapsto \mathrm{e}^{-\sum_{i=1}^{b} U_i(\theta)} / \int_{\mathbb{R}^d} \mathrm{e}^{-\sum_{i=1}^{b} U_i(\theta')}\mathrm{d}\theta'$$

- Langevin dynamic.

$$\mathrm{d}\vartheta_t = -\sum_{i=1}^{b} \nabla U_i(\vartheta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

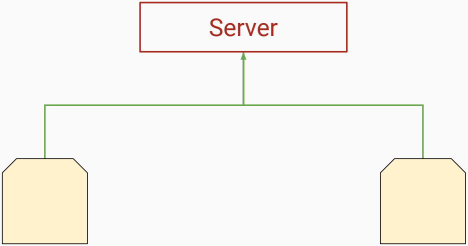
- Invariant distribution.

$$\pi : \theta \mapsto \mathrm{e}^{-\sum_{i=1}^{b} U_i(\theta)} \Big/ \int_{\mathbb{R}^d} \mathrm{e}^{-\sum_{i=1}^{b} U_i(\theta')}\mathrm{d}\theta'$$

- Discretization. Euler-Maruyama $\longrightarrow$ $\boxed{\mathtt{LSD}^{\#}}$

$$\forall k \in \{0, T-1\}, \quad \theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \nabla U_i(\theta_k) + \sqrt{2\gamma} Z_{k+1}$$

- $\boxed{\texttt{LSD}^{\#}}$ example: $|\mathcal{D}_i| = N$ and $\nabla U_{i,j}(\theta_k) = \nabla_\theta \ell(f_{\theta_k}(x_j), y_j)$



$$\nabla U_1(\theta_k) = \sum_{j=1}^{N} \nabla U_{1,j}(\theta_k) \qquad \nabla U_b(\theta_k) = \sum_{j=1}^{N} \nabla U_{b,j}(\theta_k)$$

$\hookrightarrow$ Computational cost?

- $\boxed{\mathtt{LSD}^{\#}}$ + Stochastic gradient/mini-batch

  $\hookrightarrow$ Computational cost?

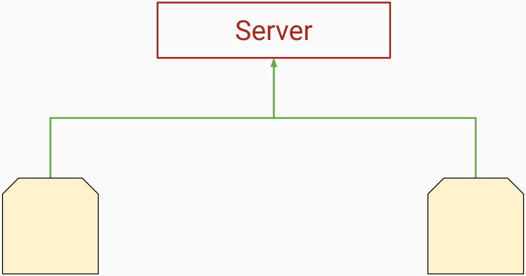
  $\hookrightarrow$ Communication constraint?

- $\boxed{\texttt{LSD}^{\#}}$ + Stochastic gradient/mini-batch
  $\hookrightarrow$ Computational cost?

- $\boxed{\texttt{LSD}^{\#}}$ + Stochastic gradient/mini-batch + Compression
  $\hookrightarrow$ Communication constraint?

- $\boxed{\texttt{LSD}^{\#}}$ + Stochastic gradient/mini-batch
  $\hookrightarrow$ Computational cost?

- $\boxed{\texttt{LSD}^{\#}}$ + Stochastic gradient/mini-batch + Compression
  $\hookrightarrow$ Communication constraint?



$$\mathscr{C}\left(\frac{N}{|S_k^1|}\sum_{j\in S_k^1}\nabla U_{1,j}(\theta_k)\right)$$

$$\mathscr{C}\left(\frac{N}{|S_k^b|}\sum_{j\in S_k^b}\nabla U_{b,j}(\theta_k)\right)$$

$\texttt{QLSD}^{\#}$ = $\boxed{\textit{Quantized Langevin Stochastic Dynamics}}^{\#}$

$$\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k) \right) + \sqrt{2\gamma} Z_{k+1}$$

- Computational cost
- Communication constraint

- $\theta_0 \sim \mu$
- Markov kernel $Q_{\#,\gamma}(\theta, \mathsf{A}) = \mathbb{P}\left(\theta_{k+1} \in \mathsf{A} \mid \theta_k = \theta\right)$

$$\boxed{\theta_k \sim \mu Q_{\#,\gamma}^k}$$

Algorithm 1: $\texttt{QLSD}^{\#}$

$\boxed{\text{Initialize } \theta_0 \in \mathbb{R}^d}$

Algorithm 1: $\texttt{QLSD}^{\#}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $\boxed{g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k)\right)}$

    Send g_k^i to the server

Algorithm 1: $\texttt{QLSD}^{\#}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \dots, b\}$ *// In parallel on the b clients* **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k)\right)$

    Send g_k^i to the server

  // On the central server

  Set $\boxed{\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} g_k^i + \sqrt{2\gamma} Z_{k+1}}$

Output: $(\theta_k)_{k=0}^{K}$

Algorithm 1: `QLSD`$^{\#}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k)\right)$

    Send g_k^i to the server

  // On the central server

  Set $\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} g_k^i + \sqrt{2\gamma} Z_{k+1}$

Output: $(\theta_k)_{k=0}^{K}$

Algorithm 2: `QSGD`

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \ldots, b\}$ **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k)\right)$

  Set $\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} g_k^i$

Output: θ_K

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x)-x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^{\#}, B_\gamma^{\#} > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q_{\#,\gamma}^k, \pi\right) \leq \underbrace{(1-\gamma\mathtt{m}/2)^k}_{\text{Contraction term}} \overbrace{W_2^2\left(\mu, \pi\right)}^{\substack{\text{Distance between}\\ \text{the initialization}\\ \text{and the target}}}$$

$$+\gamma B_\gamma^{\#} + \gamma^2 A_\gamma^{\#} (1-\mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x)-x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}}\,\langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_{\gamma}^{\#}, B_{\gamma}^{\#} > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q_{\#,\gamma}^k, \pi\right) \leq (1-\gamma \mathtt{m}/2)^k W_2^2(\mu,\pi) + \overbrace{\gamma B_{\gamma}^{\#}}^{\substack{\text{Heterogeneity} \\ +\text{ Discretization error}}}$$
$$+ \gamma^2 A_{\gamma}^{\#}(1-\mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega \|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^\#, B_\gamma^\# > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q_{\#,\gamma}^k, \pi\right) \leq (1 - \gamma \mathtt{m}/2)^k W_2^2(\mu, \pi) + \gamma B_\gamma^\# + \underbrace{\gamma^2 A_\gamma^\# (1 - \mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)}_{\text{Mini-batch + Compression}}$$

Sketch of proof.

- Based on couplings

Initialization	$\nu_0 \sim \pi$	$\theta_0 \sim \mu$
k-th iteration	$\nu_{k\gamma} \sim \pi$	$\theta_k \sim \mu Q_\gamma^k$

- Update

$$\begin{cases} \mathrm{d}\nu_t = -\nabla U(\nu_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \\ \theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\widehat{\nabla U_i}(\theta_k)\right) + \sqrt{2}(B_{\gamma(k+1)} - B_{\gamma k}) \end{cases}$$

Sketch of proof.

- Based on couplings

Initialization	$\nu_0 \sim \pi$	$\theta_0 \sim \mu$
k-th iteration	$\nu_{k\gamma} \sim \pi$	$\theta_k \sim \mu Q_\gamma^k$

- Update

$$\begin{cases} \mathrm{d}\nu_t = -\nabla U(\nu_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \\ \theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\widehat{\nabla U_i}(\theta_k)\right) + \sqrt{2}(B_{\gamma(k+1)} - B_{\gamma k}) \end{cases}$$

- Main idea $\rightsquigarrow$ to find a contraction

$$\mathbb{E}^{\mathcal{F}_k}\left[\|\nu_{\gamma(k+1)} - \theta_{k+1}\|^2\right] \leq A\|\nu_{k\gamma} - \theta_k\|^2 + \gamma^2 B\|\theta_k - \theta_\star\|^2 + \gamma^2 C - \gamma D\left\langle \nu_{k\gamma} - \theta_k, \nabla U(\nu_{k\gamma}) - \nabla U(\theta_k)\right\rangle$$

- Wasserstein distance $\rightsquigarrow$ infimum over couplings between $(\mu Q_\gamma^k, \pi)$

$$W_2^2(\mu Q_\gamma^k, \pi) \leq \mathbb{E}\left[\|\nu_{\gamma k} - \theta_k\|^2\right]$$

Drawback.

- When $\gamma \propto N^{-1}$

$$\liminf_{N\to\infty} \gamma B^{\#}_{\gamma} > 0$$

Drawback.

- When $\gamma \propto N^{-1}$

$$\liminf_{N \to \infty} \gamma B_{\gamma}^{\#} > 0$$

Solution: Variance-reduction scheme.

- Fixed-point approach based on the minimizer $\theta_\star = \arg\min U$ (Brosse et al., 2018; Baker et al., 2019).

$$\widehat{\nabla U}_i(\theta) = \frac{N}{|S_k^i|} \sum_{j \in S_k^i} \{\nabla U_{i,j}(\theta) - \nabla U_{i,j}(\theta_\star)\}$$

- Biased operator

$$\mathbb{E}\left[\widehat{\nabla U}_i(\theta)\right] = \nabla U_i(\theta) - \nabla U_i(\theta_\star) \neq \nabla U_i(\theta)$$

- QLSD*:

$$\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \{\nabla U_{i,j}(\theta) - \nabla U_{i,j}(\theta_\star)\}\right) + \sqrt{2\gamma} Z_{k+1}$$

Algorithm 3: $\texttt{QLSD}^{\star}$

$\boxed{\text{Initialize } \theta_0 \in \mathbb{R}^d}$

Algorithm 3: `QLSD`*

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K-1$ **do**

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $\boxed{g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|}\sum_{j \in S_k^i}\{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\theta_\star)\}\right)}$

    Send g_k^i to the server

Algorithm 3: QLSD*

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K-1$ **do**

  for $i \in \{1,\dots,b\}$ *// In parallel on the b clients* **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|}\sum_{j\in S_k^i}\{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\theta_\star)\}\right)$

    Send g_k^i to the server

  // On the central server

  Set $\boxed{\theta_{k+1} = \theta_k - \gamma\sum_{i=1}^b g_k^i + \sqrt{2\gamma}Z_{k+1}}$

Output: $(\theta_k)_{k=0}^K$

Algorithm 3: QLSD*

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\theta_\star)\}\right)$

    Send g_k^i to the server

  // On the central server

  Set $\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} g_k^i + \sqrt{2\gamma} Z_{k+1}$

Output: $(\theta_k)_{k=0}^{K}$

Algorithm 4: QLSD$^{\#}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $g_k^i = \mathscr{C}\left(\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \nabla U_{i,j}(\theta_k)\right)$

    Send g_k^i to the server

  // On the central server

  Set $\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^{b} g_k^i + \sqrt{2\gamma} Z_{k+1}$

Output: $(\theta_k)_{k=0}^{K}$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A^{\star}_{\gamma}, B^{\star}_{\gamma} > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q^k_{\star,\gamma}, \pi\right) \leq \underbrace{(1 - \gamma \mathtt{m}/2)^k}_{\text{Contraction term}} \overbrace{W_2^2\left(\mu, \pi\right)}^{\substack{\text{Distance between} \\ \text{the initialization} \\ \text{and the target}}} + \gamma B^{\star}_{\gamma}$$
$$+ \gamma^2 A^{\star}_{\gamma} (1 - \mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^\star, B_\gamma^\star > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q_{\star,\gamma}^k, \pi\right) \leq (1 - \gamma \mathtt{m}/2)^k \cdot W_2^2(\mu, \pi) + \overbrace{\gamma B_\gamma^\star}^{\substack{\text{Discretization error} \\ + \text{ Mini-batch} \\ + \text{ Compression}}} + \gamma^2 A_\gamma^\star (1 - \mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega \|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A^\star_\gamma, B^\star_\gamma > 0$
- $\forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2\left(\mu Q^k_{\#,\gamma}, \pi\right) \leq (1 - \gamma \mathtt{m}/2)^k W_2^2(\mu, \pi) + \gamma B^{\#}_\gamma$$
$$+ \gamma^2 \underbrace{A^\star_\gamma}_{\text{Compression}} (1 - \mathtt{m}\gamma/2)^{k-1} k \int_{\mathbb{R}^d} \|\theta - \theta_\star\|^2 \mu(\mathrm{d}\theta)$$

For non-iid data in $\mathcal{D}_1, \ldots, \mathcal{D}_b$

- $\liminf_{N\to\infty} \gamma B_\gamma^{\#} > 0$ when the stepsize $\gamma \propto N^{-1} \to 0$
- $\lim_{N\to\infty} \gamma B_\gamma^{\star} = 0$ when $\gamma \propto N^{-1} \to 0$
- $B_\gamma^{\star}$ independant of the heterogeneity !

For non-iid data in $\mathcal{D}_1, \ldots, \mathcal{D}_b$

- $\liminf_{N\to\infty} \gamma B_\gamma^{\#} > 0$ when the stepsize $\gamma \propto N^{-1} \to 0$
- $\lim_{N\to\infty} \gamma B_\gamma^{\star} = 0$ when $\gamma \propto N^{-1} \to 0$
- $B_\gamma^{\star}$ independant of the heterogeneity !

Drawback.

- Difficult estimation of $\theta_\star$, especially in a FL context

For non-iid data in $\mathcal{D}_1, \ldots, \mathcal{D}_b$

- $\liminf_{N\to\infty} \gamma B_\gamma^{\#} > 0$ when the stepsize $\gamma \propto N^{-1} \to 0$
- $\lim_{N\to\infty} \gamma B_\gamma^{\star} = 0$ when $\gamma \propto N^{-1} \to 0$
- $B_\gamma^{\star}$ independant of the heterogeneity !

Drawback.

- Difficult estimation of $\theta_\star$, especially in a FL context

Solution: Variance-reduction scheme without $\theta_\star$.

- `SVRG`: variance reduction (Johnson and Zhang, 2013)
- Memory Term: heterogeneity (Horváth et al., 2019; Dieuleveut et al., 2020)
- $\texttt{QLSD}^{++}$:

$$g_k^i = \underbrace{\mathscr{C}}_{\text{Compression}} \left(\left[\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\zeta_k)\} + h_k^i - \eta_k^i \right] \right)$$

For non-iid data in $\mathcal{D}_1, \ldots, \mathcal{D}_b$

- $\liminf_{N\to\infty} \gamma B_\gamma^{\#} > 0$ when the stepsize $\gamma \propto N^{-1} \to 0$
- $\lim_{N\to\infty} \gamma B_\gamma^{\star} = 0$ when $\gamma \propto N^{-1} \to 0$
- $B_\gamma^{\star}$ independant of the heterogeneity !

Drawback.

- Difficult estimation of $\theta_\star$, especially in a FL context

Solution: Variance-reduction scheme without $\theta_\star$.

- SVRG: variance reduction (Johnson and Zhang, 2013)
- Memory Term: heterogeneity (Horváth et al., 2019; Dieuleveut et al., 2020)
- $\texttt{QLSD}^{++}$:

$$g_k^i = \mathscr{C}\left(\left[\frac{N}{|S_k^i|}\sum_{j\in S_k^i}\left\{\nabla U_{i,j}(\theta_k)\underbrace{-\nabla U_{i,j}(\zeta_k)}_{\text{Control Variate}}\right\}\underbrace{+h_k^i}_{\text{Control Variate}} - \eta_k^i\right]\right)$$

For non-iid data in $\mathcal{D}_1, \ldots, \mathcal{D}_b$

- $\liminf_{N\to\infty} \gamma B_\gamma^{\#} > 0$ when the stepsize $\gamma \propto N^{-1} \to 0$
- $\lim_{N\to\infty} \gamma B_\gamma^{\star} = 0$ when $\gamma \propto N^{-1} \to 0$
- $B_\gamma^{\star}$ independant of the heterogeneity !

Drawback.

- Difficult estimation of $\theta_\star$, especially in a FL context

Solution: Variance-reduction scheme without $\theta_\star$.

- `SVRG`: variance reduction (Johnson and Zhang, 2013)
- Memory Term: heterogeneity (Horváth et al., 2019; Dieuleveut et al., 2020)
- `QLSD`$^{++}$:

$$g_k^i = \mathscr{C}\left(\left[\frac{N}{|S_k^i|}\sum_{j\in S_k^i}\{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\zeta_k)\} + h_k^i \underbrace{-\eta_k^i}_{\text{Memory term}}\right]\right)$$

Algorithm 5: $\mathtt{QLSD}^{++}$

$\boxed{\text{Initialize } \theta_0 \in \mathbb{R}^d}$

Algorithm 5: $\texttt{QLSD}^{++}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K-1$ **do**

  if $k \equiv 0 \, [\mathrm{mod}\, \ell]$ **then**

    Set $\boxed{\zeta_k = \theta_k}$

    for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

      Store $\boxed{h_k^i = \sum_{j=1}^{N} \nabla U_{i,j}(\zeta_k)}$

  else

    Set $\boxed{\zeta_k = \zeta_{k-1}}$

Algorithm 5: $\texttt{QLSD}^{++}$

Initialize $\theta_0 \in \mathbb{R}^d$

for $k = 0$ to $K - 1$ **do**

  if $k \equiv 0 \, [\mathrm{mod}\, \ell]$ **then**

    Set $\zeta_k = \theta_k$

    for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

      Store $h_k^i = \sum_{j=1}^{N} \nabla U_{i,j}(\zeta_k)$

  else

    Set $\zeta_k = \zeta_{k-1}$

  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**

    Set $\boxed{g_k^i = \mathscr{C}\left(\left[\frac{N}{|S_k^i|} \sum_{j \in S_k^i} \{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\zeta_k)\} + h_k^i - \eta_k^i\right]\right)}$

    Send g_k^i to the server

    Update $\boxed{\eta_{k+1}^i = \eta_k^i + \alpha g_k^i}$

Algorithm 5: QLSD^{++}

Initialize $\theta_0 \in \mathbb{R}^d$
for $k = 0$ to $K-1$ **do**
  if $k \equiv 0 \,[\mathrm{mod}\, \ell]$ **then**
    Set $\zeta_k = \theta_k$
    for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**
      Store $h_k^i = \sum_{j=1}^N \nabla U_{i,j}(\zeta_k)$
  else
    Set $\zeta_k = \zeta_{k-1}$
  for $i \in \{1, \ldots, b\}$ *// In parallel on the b clients* **do**
    Set $g_k^i = \mathscr{C}\left(\left[\frac{N}{|S_k^i|}\sum_{j \in S_k^i}\{\nabla U_{i,j}(\theta_k) - \nabla U_{i,j}(\zeta_k)\} + h_k^i - \eta_k^i\right]\right)$
    Send g_k^i to the server
    Update $\eta_{k+1}^i = \eta_k^i + \alpha g_k^i$
  // On the central server
  Set $\boxed{\theta_{k+1} = \theta_k - \gamma \sum_{i=1}^b g_k^i - \gamma\eta_k + \sqrt{2\gamma} Z_{k+1}}$
  Update $\boxed{\eta_{k+1} = \eta_k + \alpha \sum_{i=1}^b g_k^i}$
Output: $(\theta_k)_{k=0}^K$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x)-x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2)-\nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}}\left\langle \nabla U_{i,j}(\theta_2)-\nabla U_{i,j}(\theta_1), \theta_2-\theta_1\right\rangle$

Then.

- $\exists \bar{\gamma}>0$, $\forall \gamma<\bar{\gamma}$, $\exists A_\gamma^{\oplus}, B_\gamma^{\oplus}>0$
- $\forall \alpha \leq 1/(1+\omega), \forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2(\mu Q_{\oplus,\gamma}^k, \pi) \leq \underbrace{(1-\gamma\mathtt{m}/2)^k}_{\text{Contraction term}} \overbrace{W_2^2(\mu,\pi)}^{\substack{\text{Distance between}\\ \text{the initialization}\\ \text{and the target}}} + \frac{\gamma}{\mathtt{m}}(1-\gamma\mathtt{m}/2)^{k/\ell} A_\gamma^{\oplus}$$

$$+ \frac{4\omega\gamma}{\mathtt{m}}(1-\alpha)^k \sum_{i=1}^{b} \|\nabla U_i(\theta_\star)-\eta_0^{(i)}\|^2 + \frac{d\gamma}{\mathtt{m}^2} B_\gamma^{\oplus}$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x)-x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^\oplus, B_\gamma^\oplus > 0$
- $\forall \alpha \leq 1/(1+\omega), \forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2(\mu Q_{\oplus,\gamma}^k, \pi) \leq (1-\gamma\mathtt{m}/2)^k W_2^2(\mu,\pi) + \frac{\gamma}{\mathtt{m}}(1-\gamma\mathtt{m}/2)^{k/\ell} \overbrace{A_\gamma^\oplus}^{\substack{\text{Mini-batch} \\ +\ \text{Compression}}}$$
$$+ \frac{d\gamma}{\mathtt{m}^2} B_\gamma^\oplus + \frac{4\omega\gamma}{\mathtt{m}}(1-\alpha)^k \sum_{i=1}^b \|\nabla U_i(\theta_\star) - \eta_0^{(i)}\|^2$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^{\oplus}, B_\gamma^{\oplus} > 0$
- $\forall \alpha \leq 1/(1+\omega), \forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

$$W_2^2(\mu Q_{\oplus,\gamma}^k, \pi) \leq (1 - \gamma \mathtt{m}/2)^k W_2^2(\mu, \pi) + \frac{\gamma}{\mathtt{m}}(1 - \gamma\mathtt{m}/2)^{k/\ell} A_\gamma^{\oplus} + \frac{d\gamma}{\mathtt{m}^2} \underbrace{B_\gamma^{\oplus}}_{\text{Residue}} + \frac{4\omega\gamma}{\mathtt{m}}(1-\alpha)^k \sum_{i=1}^{b} \|\nabla U_i(\theta_\star) - \eta_0^{(i)}\|^2$$

Assumptions.

- The potential U is $\mathtt{m}$-strongly convex, $\mathtt{L}$-Lipschitz
- The compression $\mathscr{C}$ is unbiased and $\mathbb{E}\|\mathscr{C}(x) - x\|^2 \leq \omega\|x\|^2$
- There exists $\bar{\mathtt{M}} \geq 0$,
  $\|\nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1)\|^2 \leq \bar{\mathtt{M}} \langle \nabla U_{i,j}(\theta_2) - \nabla U_{i,j}(\theta_1), \theta_2 - \theta_1 \rangle$

Then.

- $\exists \bar{\gamma} > 0$, $\forall \gamma < \bar{\gamma}$, $\exists A_\gamma^\oplus, B_\gamma^\oplus > 0$
- $\forall \alpha \leq 1/(1+\omega), \forall \mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$

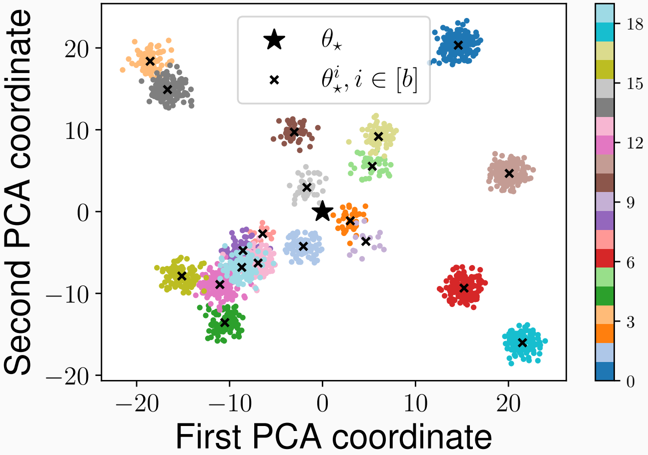
$$W_2^2(\mu Q_{\oplus,\gamma}^k, \pi) \leq (1 - \gamma \mathtt{m}/2)^k W_2^2(\mu, \pi) + \frac{\gamma}{\mathtt{m}}(1 - \gamma\mathtt{m}/2)^{k/\ell} A_\gamma^\oplus + \frac{d\gamma}{\mathtt{m}^2} B_\gamma^\oplus + \frac{4\omega\gamma}{\mathtt{m}}(1-\alpha)^k \underbrace{\sum_{i=1}^b \|\nabla U_i(\theta_\star) - \eta_0^{(i)}\|^2}_{\text{Memory initialization}}$$

To summarize.

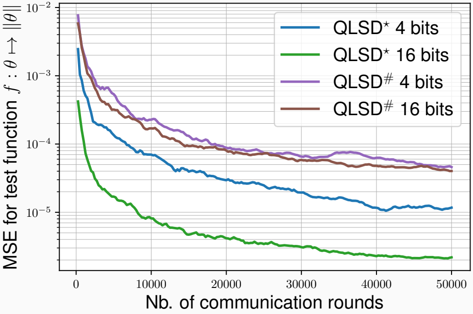
- $\boxed{1}$ We proposed $\texttt{QLSD}^{\#}$
- $\boxed{2}$ The bias $\liminf_{N\to\infty} \gamma B_{\gamma}^{\#} > 0$ when $\gamma \propto N^{-1} \to 0$
- $\boxed{3}$ $\Rightarrow$ $\texttt{QLSD}^{\star}$: control variates using $\theta_{\star} = \arg\min U$.
- $\boxed{4}$ $\hookrightarrow$ hard to compute
- $\boxed{5}$ $\Rightarrow$ $\texttt{QLSD}^{++}$: memory term $\to$ heterogeneity & control variates $\to$ fixes bias when $\gamma \propto N^{-1} \to 0$

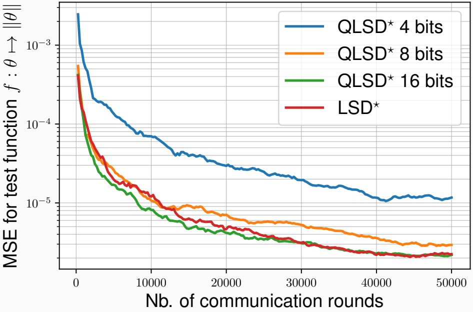
Numerical experiments

Toy Gaussian example.



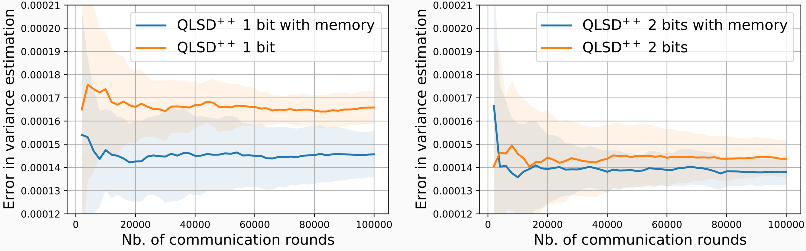
- number clients = 20, dimension = 50,
- dataset size = 200, mini-batch size = 20


MSE for test function $f: \theta \mapsto \|\theta\|$
Nb. of communication rounds
QLSD* 4 bits
QLSD* 16 bits
QLSD$^\#$ 4 bits
QLSD$^\#$ 16 bits


MSE for test function $f: \theta \mapsto \|\theta\|$
Nb. of communication rounds
QLSD* 4 bits
QLSD* 8 bits
QLSD* 16 bits
LSD*

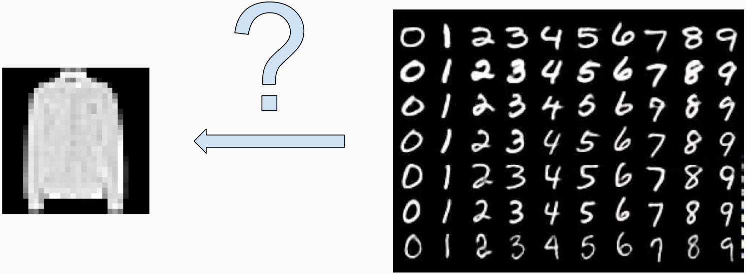
Logistic regression.

- number clients $= 50$, dimension $= 2$,
- dataset size $= 200$, mini-batch size $= 20$
- control variates update $\ell = 100$



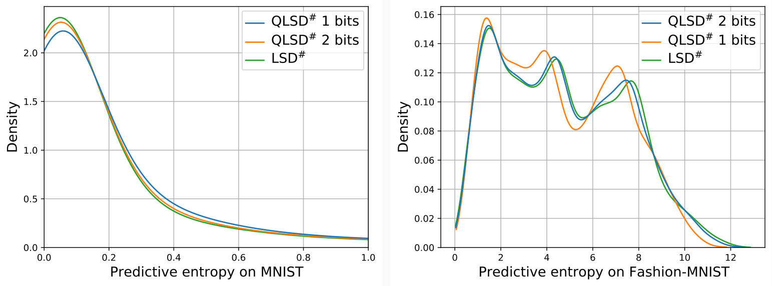
Shallow BNN.

- number clients = 100, dimension = 784,
- dataset size = 600, mini-batch size = 80
- Trained on MNIST



- Conditional predictive entropy

$$\int_\theta \log p\left(y_{\text{pred}} | x, \theta\right) \pi(\theta \mid \mathcal{D}) \, \mathrm{d}\theta \simeq \frac{1}{K} \sum_{k=0}^{K-1} \log p\left(y_{\text{pred}} | x, \theta_k\right)$$



Conclusion.

- Introduce 3 algorithms
- Analyse theoretically
- Numerically the compression does not hurt the convergence

Perspective.

- Non-convex potential U
- Biased compression
- Hamiltonian instead of Langevin diffusion
- Several local updates before communicating

References

Alistarh, D., D. Grubic, J. Li, R. Tomioka, and M. Vojnovic (2017). QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Baker, J., P. Fearnhead, E. B. Fox, and C. Nemeth (2019). Control variates for stochastic gradient MCMC. *Statistics and Computing 29*(3), 599–615.

Brosse, N., A. Durmus, and E. Moulines (2018). The promises and pitfalls of stochastic gradient langevin dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 31. Curran Associates, Inc.

Dieuleveut, A., A. Durmus, and F. Bach (2020, 06). Bridging the gap between constant step size stochastic gradient descent and Markov chains. *Annals of Statistics 48*(3), 1348–1382.

Horváth, S., D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik (2019). Stochastic Distributed Learning with Gradient Quantization and Variance Reduction . *arXiv preprint arXiv:1904.05115*.

Johnson, R. and T. Zhang (2013). Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. In *Neural Information Processing Systems*, pp. 315–323.

Karimireddy, S. P., S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR.

McMahan, B., E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR.

Mishchenko, K., E. Gorbunov, M. Takáč, and P. Richtárik (2019). Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*.

Philippenko, C. and A. Dieuleveut (2020). Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees . *arXiv preprint arXiv:2006.14591*.

DoStoVoQ: Doubly Stochastic Voronoi Vector Quantization SGD for Federated Learning

Louis Leconte, Aymeric Dieuleveut, Edouard Oyallon, Eric Moulines, Gilles Pages.

Submitted to NeurIPS 2021 Conference

June 24, 2021

Overview

Vector Quantization

Let X a random vector in $\mathbb{R}^d$

- Discretize (spatially) X i.e. replace X by a r.v. taking finitely many values close to X in some sense;
- Let $q : \mathbb{R}^d \to \Gamma \subset \mathbb{R}^d$ be a Borel function, and Γ a finite subset of $\mathbb{R}^d$(grid). $\hat{X} = q(X)$ is called a quantization of X.
- Example: if X is [0,1]-valued, one may choose a mid-point quantization: $q(x) = \frac{2k-1}{2N}$, if $\frac{k-1}{N} \leq x \leq \frac{k}{N}, x \in [0,1]$.

[0] slide inspired from G. Pages talk at CIRM, 2017.

Voronoi Quantization

Voronoi quantization [PP03, PW18], aims at selecting the closest codeword from C_M, i.e.:

$$\mathrm{VQ}(x, \mathcal{C}_M) \triangleq \mathrm{argmin}_{c \in \mathcal{C}_M} \|x - c\| .$$

Voronoi Quantization

Voronoi quantization [PP03, PW18], aims at selecting the closest codeword from $\mathcal{C}_M$, i.e.:

Figure: Voronoi quantization for $d = 2$

Voronoi Quantization

Voronoi quantization [PP03, PW18], aims at selecting the closest codeword from $\mathcal{C}_M$

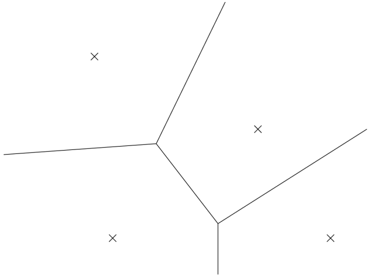


Figure: Voronoi quantization for $d = 2$

Voronoi Quantization

Voronoi quantization [PP03, PW18], aims at selecting the closest codeword from $\mathcal{C}_M$

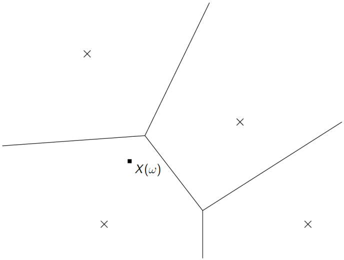


Figure: Voronoi quantization for $d = 2$

Voronoi Quantization

Voronoi quantization [PP03, PW18], aims at selecting the closest codeword from C_M, i.e.:

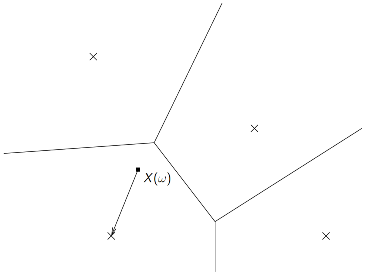


Figure: Voronoi quantization for $d = 2$

Unbiased random scalar quantization

Consider a (scalar) codebook $\mathcal{O}_Q = \{o_1, \ldots, o_Q\} \subset \mathbb{R}$ where $Q \geq 2$, and $-\infty < o_1 < \cdots < o_Q < \infty$.

- Compute the index $j(x) \in [Q]$ such that $x \in \left[o_{j(x)}, o_{j(x)+1}\right)$.
- Note that $x = \lambda^*_{j(x)}(x) o_{j(x)} + (1 - \lambda^*_{j(x)}(x)) o_{j(x)+1}$ where

$$\lambda^*_{j(x)}(x) = (x - o_{j(x)})/(o_{j(x)+1} - o_{j(x)}) \in (0, 1] \ .$$

- Unbiased scalar quantifier:

$$\mathrm{SQ}(x, \mathcal{O}_Q, u) = \mathbb{1}(\{u \leq \lambda^*_{j(x)}(x)\}) o_{j(x)} + \mathbb{1}(\{u > \lambda^*_{j(x)}(x)\}) o_{j(x)+1}$$

Unbiased random scalar quantization

- $\mathrm{SQ}(x, \mathcal{O}_Q, u) = \mathbb{1}(\{u \leq \lambda^*_{j(x)}(x)\}) o_{j(x)} + \mathbb{1}(\{u > \lambda^*_{j(x)}(x)\}) o_{j(x)+1}$ is an unbiased quantization.

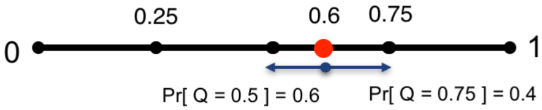


Figure: Illustration of an unbiased scalar quantization (taken from [AGL+17b])

Dual Vector Quantization

- Find weights $(\lambda_1^*(x), \ldots, \lambda_M^*(x))$, $\lambda_i^*(x) \geq 0$, $\sum_{j=1}^M \lambda_j^*(x) = 1$, such as: for all $x \in \mathrm{ConvHull}(\mathcal{C}_M)$, we get

$$\text{Dual-VQ}(x, \mathcal{C}_M, U) = \sum_{i=1}^{M} \lambda_i^*(x) c_i = x\,.$$

- The Delaunay quantizer minimizes the inertia :

$$\sum_{i=1}^{M} \lambda_i^*(x) \|x - c_i\|^2$$

Dual Vector Quantization

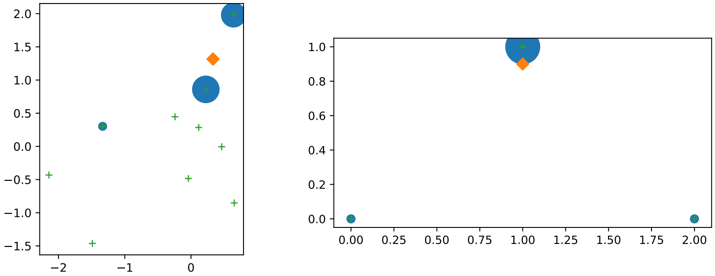


Figure: `Delaunay quantization` for a vector x (orange diamond), for a given set of codewords (green $+$), and corresponding weights (area of the blue spheres). Remark that all but three points have a 0 probability of being picked, making the quadratic error much smaller than for `HSQ-span`.

Our contributions

- Unbiased Vector Quantization
- High-compression rate
- Small computational overhead
- Theoretical guarantees on distortion and optimality.

From StoVoQ to DoStoVoQ

Why unbiasedness is important

- A compression operator Comp is unbiased if for any $x \in \mathbb{R}^d$, $\mathbb{E}[\mathrm{Comp}(x)] = x$.
- A compression operator has a ω-bounded relative variance (for some $\omega > 0$), if for all $x \in \mathbb{R}^d$, $\mathbb{E}[\|\mathrm{Comp}(x) - x\|^2] \leq \omega\|x\|^2$.

Why unbiasedness is important

- K workers compress independently the same vector x.
- Unbiasedness

$$\mathbb{E}\left[K^{-1}\sum_{k=1}^{K}\mathrm{Comp}_k(x)\right] = x$$

- Independence and bounded relative variance

$$\mathbb{E}\left[\left\|x - K^{-1}\sum_{k=1}^{K}\mathrm{Comp}_k(x)\right\|^2\right] \leq (\omega/K)\|x\|^2$$

StoVoQ Algorithm

- Voronoi Vector quantization The input vector $x \in \mathbb{R}^d$ is mapped onto its nearest neighbor in a codebook $\mathcal{C}_M = \{c_i\}_{i=1}^M$.

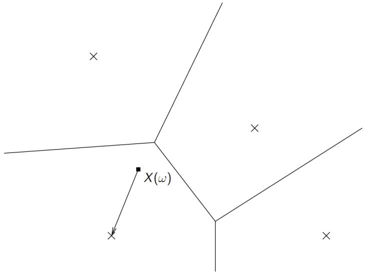


Figure: Nearest neighbor quantization

StoVoQ Algorithm

- **Voronoi Vector quantization**
- Random codebook. A new codebook is sampled every time a new quantization operation is performed.
- StoVoQdiffers from classical random VQ which typically uses a random codebook, but which is sampled once and then kept fixed.

The codebook, is not transmitted: the transmitter and the receiver use the same random seed!

StoVoQ Algorithm

- **Voronoi Vector quantization**
- **Random codebook.**
- **Unitary invariant codewords** The distribution of the codewords p is invariant under the unitary group, i.e. for any unitary matrix, U ($U^T U = \mathrm{I}_d$), and $x \in \mathbb{R}^d$,

$$p(Ux) = p(x).$$

StoVoQ Algorithm

- **Voronoi Vector quantization**
- **Random codebook.**
- **Unitary invariant codewords**
- **Bias removal.** By relying on unitarily invariant distribution for the codewords generation, the quantized value of each vector $x \in \mathbb{R}^d$ is **directionnally unbiased**. The bias only depends on the number and distributions of the random of codewords and on $\|x\|$. This key property allows to derive a simple way to remove the quantization bias.

Key Property : the quantization bias is radial

Lemma

Assume that the codebook distribution is unitarily invariant. Then, for any nonnegative measurable function f, any $U \in \mathrm{U}(d)$, and $x \in \mathbb{R}^d$,

$$\mathbb{E}_{\mathcal{C}_M \sim p}[f(\mathrm{VQ}(Ux, \mathcal{C}_M))] = \mathbb{E}_{\mathcal{C}_M \sim p}[f(U\,\mathrm{VQ}(x, \mathcal{C}_M))].$$

Taking $f(x) = x$, $\Rightarrow$ for any $x \in \mathbb{R}^d$ and $U \in \mathrm{U}(d)$, it holds that $\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(Ux, \mathcal{C}_M)] = U\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(x, \mathcal{C}_M)]$.

Key Property: the quantization bias is radial

Theorem (Quantization bias)

Assume that the codebook distribution is unitarily invariant. Then, for all $M \in \mathbb{N}$, there exists a function $r_M^p : \mathbb{R}_+ \mapsto \mathbb{R}_+$ such that for all $x \in \mathbb{R}^d$,

$$\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(x, \mathcal{C}_M)] = r_M^p(\|x\|)x.$$

In words, the expectation of the quantized vector $\mathrm{VQ}(x, \mathcal{C}_M)$ is colinear to the vector x, i.e., $\mathrm{VQ}(x, \mathcal{C}_M)$ is directionally unbiased. Moreover, the radial bias only depends on $\|x\|$, M and the distribution p.

Bias function

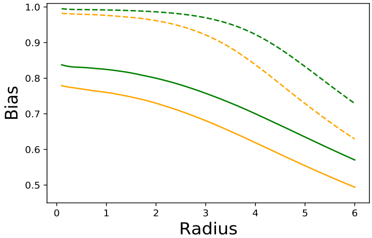


Figure: function r_M^p for $d = 4$ (dashed) and $d = 16$ (solid), $p = \mathcal{N}(0, \mathrm{I}_d)$ and $M = 2^{10}$(orange), and $M = 2^{13}$(green).

Regularity Assumptions

1. there exists $\epsilon > 0$ such that $\int r^{2+\epsilon} p_{\mathrm{rad}}(r) \mathrm{d}r < \infty$
2. for some $\delta > 0$, $m_\delta = \inf_{r \leq \delta} p_{\mathrm{rad}}(r) > 0$, and (3) p_{rad} is unimodal, i.e. the super level sets $\{r \in \mathbb{R}_+, p_{\mathrm{rad}}(r) \geq t\}$, for $t \geq 0$ are convex subsets of $\mathbb{R}_+$.

Regularity assumptions obviously satisfied if we take $p = \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for any $\sigma^2 > 0$.

Distortion of a random codebook

Theorem

Assume that the codebook distribution is (a) unitarily invariant (b) regular. Define $C_d = \pi^{-1}\Gamma(1+2/d)\Gamma(1+d/2)^{2/d}$. *Then, for every* $x \in \mathbb{R}^d$,

$$\lim_{M\to\infty} M^{2/d}\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2] = C_d\, p_{\mathrm{rad}}^{-2/d}(\|x\|)\,.$$

- Note that $C_d \approxeq_{d\to\infty} d/(2\pi\mathrm{e})$ hence C_d grows only linearly with the dimension d.
- Since $|r_M^p(\|x\|)-1| \leq \|x\|^{-1}\{\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2]\}^{1/2}$,

$$\limsup_{M\to\infty} M^{1/d}|r_M^p(\|x\|)-1| \leq C_d^{1/2} p_{\mathrm{rad}}^{-1/d}(\|x\|)/\|x\|\,.$$

Optimal Codebook, Zador's theorem

- For a given pdf q of the input the *(quadratic) distortion* is defined as:

$$\mathrm{Dist}(q, \mathcal{C}_M) = \int_{\mathbb{R}^d} \|x - \mathrm{VQ}(x, \mathcal{C}_M)\|^2 \, q(x) \, \mathrm{d}x \, .$$

We stress that in this case the expectation is taken w.r.t. the input distribution q, the codebook being deterministic in (**??**).

- A *Voronoi optimal codebook* $\mathcal{C}_M^{q,*}$ is a minimizer of the distortion over the set of codebooks: $\mathrm{Dist}(q, \mathcal{C}_M^{q,*}) = \min_{|\mathcal{C}_M|=M} \mathrm{Dist}(q, \mathcal{C}_M)$.
- Zador's theorem gives the distortion of the Voronoi optimal codebook in the limit of $M \to \infty$; as $M \to \infty$,

$$\mathrm{Dist}(q, \mathcal{C}_M) \approxeq M^{-2/d} J_d \|q\|_{d/(d+2)}$$

and J_d is a universal constant satisfying $J_d \approxeq_{d \to \infty} d/2\pi\mathrm{e}$.

Do we need an optimal codebook ?

- Objective Quantify the loss between random codebook distributed according to p and the Voronoi optimal codebook for a given input distribution q when $M \to \infty$. Define

$$\mathrm{C}(q, p, d) = \int_{\mathbb{R}^d} p(x)^{-2/d} q(x) \mathrm{d}x \,.$$

- If $\|q\|_{d/(d+2)} < \infty$, using the Hölder inequality with negative exponents, it holds that

$$\mathrm{C}(q, p, d) \geq \|q\|_{d/(d+2)}$$

.

Do we need an optimal codebook ?

Theorem

Under the "standard assumptions", $\|q\|_{d/(d+2)} < \infty$, $\int_{\mathbb{R}^d} \|x\|^{2+\delta} q(x)\mathrm{d}x < \infty$ *for some* $\delta > 0$, *and* $\mathrm{C}(q,p,d) < \infty$. *Then,*

$$\lim_{M\to\infty} \mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{Dist}(q, \mathcal{C}_M)] / \mathrm{Dist}(q, \mathcal{C}_M^{q,*}) = C_d J_d^{-1}\, \mathrm{C}(q,p,d) \|q\|_{d/(d+2)}^{-1}.$$

If the codeword distribution is given by $p_{q,d,*} = q^{d/(d+2)}(x) / \int q^{d/(d+2)}(x)\mathrm{d}x$, *then,* $\mathrm{C}(q, p_{q,d,*}, d) = \|q\|_{d/(d+2)}$.

Take-home message

- The distortion achieved by a random quantizer $\mathrm{VQ}(\cdot, \mathcal{C}_M)$, $\mathcal{C}_M \sim p$ is rate optimal (with rate $M^{-2/d}$).
- If in addition q is unitarily invariant and unimodal, then a random codebook distributed according to $p_{q,d,*}$ reaches the optimal distortion bound, up to universal constants (depending only on the dimension d).
- Moreover, as $d \to \infty$, then $C_d J_d^{-1} \approxeq_{d \to \infty} 1$ and the efficiency gap vanishes.

Take-home message

- As an illustration, assume that the input distribution is standard Gaussian $q = \mathcal{N}(0, \mathrm{I}_d)$ and set the codeword distribution to be $p_\alpha = \mathcal{N}(0, \alpha^2 \mathrm{I}_d)$ where $\alpha^2 \in \mathbb{R}_+^*$.
- If $\alpha^2 d > 2$, then $\mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha^2 \mathrm{I}_d), d) = 2\pi\alpha^2\{\alpha^2 d/(\alpha^2 d - 2)\}^{d/2}$ and $\|\mathcal{N}(0, \mathrm{I}_d)\|^{(2+d)/2} = (2\pi)(1 + 2/d)^{1+2/d}$.
- The function $\alpha \to \mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha^2 \mathrm{I}_d), d)$ has a unique minimum at $\alpha_d^2 = 1 + 2/d$ for which $\mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha_d^2 \mathrm{I}_d), d) = \|\mathcal{N}(0, \mathrm{I}_d)\|^{(2+d)/2}$ showing that a random codebook sampled from $\mathcal{N}(0, \alpha_d^2 \mathrm{I}_d)$ is optimal.

Related works

- **QSGD**: [AGL+17b] compresses each coordinate of the scaled vector $x/\|x\|$ on $s+1$ codewords. QSGD is a scalar quantizer which requires $\mathcal{O}(\sqrt{d}\log_2(d))$ bits in its highest compression setting ($s=1$, only two possible levels for each coordinate). The vector norm is transmitted with high (full) precision $\|x\|$ (16 bits). In deep learning problems, it reduces the communication cost by a factor of 4 to 7.

Related works

- **QSGD**
- **Top-H/Rand H.** map the vector to either its H largest coordinates, or a random subset of cardinality H, rescaled by d/H to ensure unbiasedness.

Related works

- **QSGD**
- **Top-H/Rand H.**
- **HyperSphere Quantization (HSQ).** HSQ was introduced by [DYZ+19]. Two versions are considered: (1) a - greedy-Voronoi VQ, which is biased; (2) an unbiased version VQ (HSQ-span), which uses a minimum-norm decomposition of $x \in \mathrm{Span}(C_M)$ the linear subspace generated by the codewords - this version suffers from a large variance and is potentially an ill-conditioning. Moreover, the performance of HSQ-span does not improve with M.

Related works

- QSGD
- Top-H/Rand H.
- **HyperSphere Quantization (HSQ).**

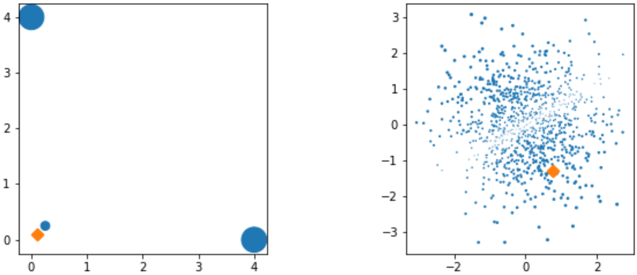


Figure: `HSQ-Span`: weights (size of the blue point) on each of the codewords of $\mathcal{C}_M$ when decomposing x (orange diamond) .

Related works

- **QSGD**
- **Top-H/Rand H.**
- **HyperSphere Quantization (HSQ).**

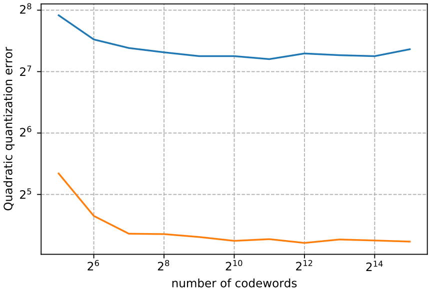


Figure: `HSQ-Span`: Distortion as a function of M (log-scale): $K = 1$ (blue) $K = 8$ (orange).

Related works

- **QSGD**
- **Top-H/Rand H.**
- **HyperSphere Quantization (HSQ).**
- **Cross-polytope.** [GKMM21] is a simple instance of Dual Quantization, with a codebook $C_M[2d]$ composed of the $2d$ canonical vectors $\{\pm\sqrt{d}e_i = \pm(0,\ldots,0,\sqrt{d},0\ldots0), i \in [d]\}$, that relies on the inclusion $B_2(0;1) \subset B_1(0;\sqrt{d}) = \mathrm{ConvHull}(C_M[2d])$. The barycentric decomposition can then easily be computed. Unfortunately, this method suffers from a large variance, as the quantization error is *lower bounded* by $\sqrt{d}-1$, which means the error has the same quadratic error than the `Rand-1` compressor.

Related works

- **QSGD**
- **Top-H/Rand H.**
- **HyperSphere Quantization (HSQ).**
- **Cross-polytope.**

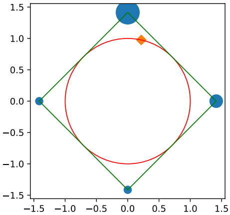


Figure: The codewords are the vertices of $B_1(0;\sqrt{d})$. A vector x (orange diamond) lying on the unit Ball $B_2(0;1)$ (red circle) is decomposed with weights (area of the blue spheres) of codewords on the Ball of radius $\sqrt{d}$

Numerical Comparisons

Table: Distortion for Gaussian inputs, for a fixed budget of 16 bits with $d = 16$.

Method	Sign [BWAA18]	Top-2	Rand-2	Polytope [GKMM21]	HSQ-span [DYZ+19]	HSQ-greed [DYZ+19]	Stovoq
# Bits (obj =16)	16	2×8	2×8	$\log_2(2 \times 16) \times 2 + 6$	$\log_2(2^{10}) + 6$	$\log_2(2^{10}) + 6$	$\log_2(2^{13}) + 3$
Unbiased			✓	✓	✓		✓
$K = 1$	6.21 (0.02)	8.40 (0.04)	102.8 (0.9)	113.9 (0.6)	146.9 (0.6)	9.03 (0.04)	6.97 (0.02)
$K = 20$	6.26 (0.02)	8.76 (0.04)	5.40 (0.04)	5.98 (0.03)	7.58 (0.04)	9.10 (0.04)	0.838 (0.005)

DoStoVoQ Algorithm

Algorithm 1: Dostovoq-SGD over T iterations

Input : T nb of steps, $(\gamma_t)_{t\geq 0}$ LR, θ_0, p, M, P ;
Output: $(\theta_t)_{t\geq 0}$
for $t = 1, \ldots, T$ **do**
  w_0 sends θ_{t-1} and different seeds $s_{k,t}$ to each w_k;
  for $k = 1, \ldots, K$ **do**
    Compute local gradient $g_{k,t}$ at θ_{t-1};
    Split $g_{k,t} \times \sqrt{D}/\|g_{k,t}\|$ on $[b^1_{k,t}, \ldots, b^L_{k,t}]$;
    for $\ell = 1, \ldots, L$ *(in parallel)* **do**
      $(\mathrm{i}_c^{t,k,\ell}, \mathrm{i}_r^{t,k,\ell}) = \mathtt{Stovoq}(b^\ell_{k,t}, \mathrm{p}, \mathrm{d}, \mathrm{P}, s_{k,t})$
    end
    Send $(\|g_{k,t}\|, (\mathrm{i}_c^{t,k,\ell}, \mathrm{i}_r^{t,k,\ell})_{\ell\in[L]})$ to w_0 ;
  end
  Reconstruct $(\hat{g}_{k,t})_{k\in K}$;
  Update: $\theta_t = \theta_{t-1} - \gamma_t \frac{1}{K}\sum_{k=1}^{K} \hat{g}_{k,t}$;
end

DoStoVoQ Algorithm

- **Splitting and renormalizing gradients.** Each worker k split its gradient into $\lfloor \frac{D}{d} \rfloor$ buckets, and apply StoVoQ for each bucket.

DoStoVoQ Algorithm

- **Splitting and renormalizing gradients.**
- **Synchronisation of random sequences of codebooks.** Independent codebooks are used to ensure that the quantizers remain conditionally independent. Generating new codebook at each time by initially sharing (different) random seeds.

Convergence Results

Consider a Smooth and Strongly Convex function $F = \sum_{k=1}^{K} f_k$, with condition number $\kappa > 1$. We measure the complexity of the algorithm by the number of iterations t required to obtain a model θ_t such that $\mathbb{E}[F(\theta_t)] - \min_{\mathbb{R}^D} F \leq \epsilon$.

- *Uncompressed* variance reduced distributed methods [DBLJ14] achieve a complexity of $\boxed{O_{\kappa\to\infty}\left(\kappa \log(\epsilon^{-1})\right)}$;

Convergence Results

Consider a Smooth and Strongly Convex function $F = \sum_{k=1}^{K} f_k$, with condition number $\kappa > 1$. We measure the complexity of the algorithm by the number of iterations t required to obtain a model θ_t such that $\mathbb{E}[F(\theta_t)] - \min_{\mathbb{R}^D} F \leq \epsilon$.

- *Uncompressed* variance reduced distributed methods
- Biased compression operators obtain $\boxed{O_{\kappa\to\infty}(\kappa(1+\delta)\log(\epsilon^{-1}))}$ for compressed GD (independently of the number of workers);

Convergence Results

Consider a Smooth and Strongly Convex function $F = \sum_{k=1}^{K} f_k$, with condition number $\kappa > 1$. We measure the complexity of the algorithm by the number of iterations t required to obtain a model θ_t such that $\mathbb{E}[F(\theta_t)] - \min_{\mathbb{R}^D} F \leq \epsilon$.

- *Uncompressed* variance reduced distributed methods
- Biased compression operators
- The result of VR-DIANA [HKM+19], which provides a complexity of $\boxed{O_{\kappa\to\infty}\left(\kappa\left(1 + \omega_M/K\right)\log(\epsilon^{-1})\right)}$, applies to `Dostovoq`-VR-DIANA.

Numerical Experiments

Least Squares Regression

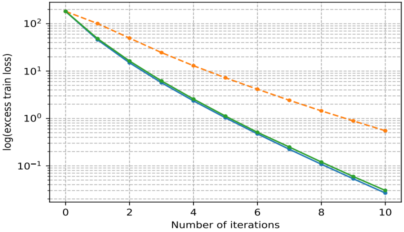


Figure: Comparison between GD (blue), `HSQ-greed` (orange) and `Dostovoq` (green), on a LSR problem in dimension $D = 2^9$.

We consider a least-squares problem with $n = 2^{14}$ samples, a bucket size $d = 16$, $D = 2^9$, and $K = 32$ workers; each worker has access to a subset $m = 2^{11}$ samples (picked with replacement) to introduce a dependency in the data used by the workers.

CIFAR10 and Imagenet

Table: Average accuracy over 5 experiments, after 100 epochs on CIFAR-10.

Algorithm	SGD	QSGD 2 bits	QSGD 4 bits	QSGD 8 bits	HSQ $d = 16$	HSQ $d = 8$	Dos. $d = 16$	Dos. $d = 8$
Raw bits per bucket	$32d$	$\sqrt{d}\log(d)$			$\log(d)$			
Effective Compression factor	1	~ 13	~ 8	~ 4	34	17	38	20
$K = 1$ worker	91.9	91.7	92.1	91.9	92.0	92.0	92.0	92.1
$K = 8$ worker	92.0	91.8	91.8	92.0	91.8	92.0	91.8	92.1

Imagenet: A ResNet here obtains 69.9%, and with a compression factor of 8, the performance drops by 2.5%. Using $d = 16$, we reach a compression factor of 38, while the Top-1 accuracy drops by only 4.8%: this is a substantially higher compression rate than the concurrent work QSGD on the ImageNet dataset.

Detailed Distortion

$Q(x) = Q_{\|}(x) + Q_{\perp}(x)$, where $Q_{\|}(x) = \|x\|^{-2} x x^{\top} Q(x)$ is the colinear distortion, and $Q_{\perp}(x)$ the orthogonal one.

Table: Distortion for Gaussian inputs

Method	Sign	Top-2	Rand-2	Polytope	
Variant					norm-quant.
$K=1$	1.0 ‖ 5.4	4.8 ‖ 3.9	12 ‖ 98	5.8 ‖ 115	5.8 ‖ 115
$K=20$	1.0 ‖ 5.4	4.7 ‖ 3.8	0.6 ‖ 4.8	0.3 ‖ 5.6	0.3 ‖ 5.6
Method	HSQ-span	HSQ-greed	StoVoQ		
Variant	norm-quant.	norm-quant.	GRVQ	Unbiased	Unbiased+quant.
$K=1$	3.8 ‖ 143	1.3 ‖ 7.8	1.8 ‖ 5.0	0.5 ‖ 10.5	0.5 ‖ 10.5
$K=20$	0.2 ‖ 7.0	1.3 ‖ 7.5	1.7 ‖ 0.25	0.03 ‖ 0.5	0.03 ‖ 0.5

Histogram of gradients

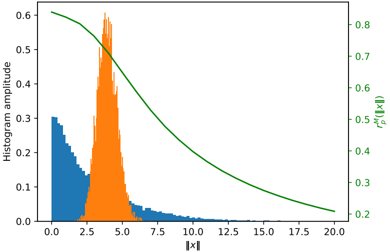


Figure: Histograms of the VGG16 gradient buckets (blue), of Gaussian vectors (orange), and the radial bias for the associated dimension $d = 16$ (green).

Influence of correlation between workers

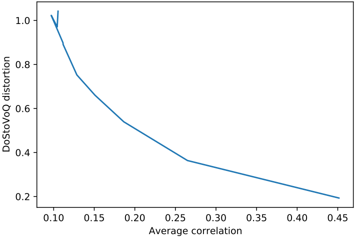


Figure: Distortion wrt correlation between gradients of $K = 8$ different workers.

Conclusion

Conclusion

- Unbiasedness is key;
- Codebook optimality is not worth it;
- High compression rate can be achieved and lead to important energy savings.

References I

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic, *Qsgd: Communication-efficient sgd via gradient quantization and encoding*, NIPS, 2017.

______, *QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding*, Advances in Neural Information Processing Systems **30** (2017), 1709–1720 (en).

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar, *signsgd: Compressed optimisation for non-convex problems*, International Conference on Machine Learning, PMLR, 2018, pp. 560–569.

References II

Aaron Defazio, Francis R Bach, and Simon Lacoste-Julien, *Saga: A fast incremental gradient method with support for non-strongly convex composite objectives*, NIPS, 2014.

Xinyan Dai, Xiao Yan, Kaiwen Zhou, Han Yang, Kelvin KW Ng, James Cheng, and Yu Fan, *Hyper-sphere quantization: Communication-efficient sgd for federated learning*, arXiv preprint arXiv:1911.04655 (2019).

Frithjof Gressmann, Zach Eaton-Rosen, and Carlo Luschi, *Improving neural network training in low dimensional random bases*, Advances in Neural Information Processing Systems **33** (2020).

References III

Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar, *vqsgd: Vector quantized stochastic gradient descent*, International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 2197–2205.

Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik, *Stochastic Distributed Learning with Gradient Quantization and Variance Reduction*, arXiv:1904.05115 [math] (2019), arXiv: 1904.05115.

Gilles Pagès and Jacques Printems, *Optimal quadratic quantization for numerics: the gaussian case*, Monte Carlo methods and applications **9** (2003), no. 2, 135–165.

References IV

Gilles Pagès and Benedikt Wilbertz, *Sharp rate for the dual quantization problem*, Séminaire de Probabilités XLIX, Springer, 2018, pp. 405–454.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi, *Powersgd: Practical low-rank gradient compression for distributed optimization*, Advances in Neural Information Processing Systems **32** (2019), 14259–14268.