



# EP3260: Machine Learning Over Networks

## Lecture 8: Communication Efficiency

Carlo Fischione

Division of Network and Systems Engineering  
School of Electrical Engineering and Computer Science  
KTH Royal Institute of Technology, Stockholm, Sweden  
<https://sites.google.com/view/mlons2020/home>

April 2020

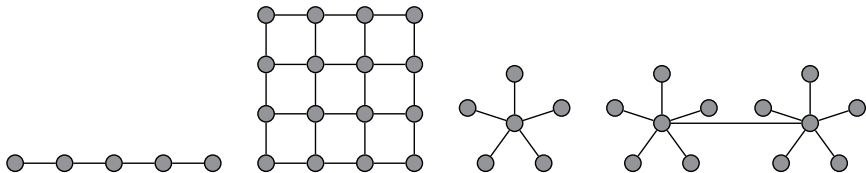
# Learning outcomes

- What is the computation-communication tradeoff in a general approach to primal-dual optimizations in ML?
- How quantization affects Gradient Descent Algorithm in ML?
- How quantization affects Stochastic Gradient Descent Algorithm in ML?

# Outline

1. Computation-communication tradeoff in a general approach
2. Quantized Distributed Gradient Descent
3. Parallel Quantized Stochastic Gradient Descent

# Recap of previous two lectures



- ML over Master-Workers networks
  - Duality methods (Lec 5)
  - Alternating Direction Methods of Multipliers (ADMM) (Lec 6)
- ML over general networks
  - Duality methods with consensus (Lec 5)
  - ADMM with consensus (Lec 6)

# Outline

1. Computation-communication tradeoff in a general approach
2. Quantized Distributed Gradient Descent
3. Parallel Quantized Stochastic Gradient Descent

# A general framework for primal-dual methods

- **Definition** (L-Lipschitz Continuity). A function  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  is L-Lipschitz Continuous if  $\forall \mathbf{u}$  and  $\mathbf{v} \in \mathbb{R}^m$ , we have  $|h(\mathbf{u}) - h(\mathbf{v})| \leq L\|\mathbf{u} - \mathbf{v}\|$
- **Definition** (L-Bounded Support). A function  $h : \mathbb{R}^m \rightarrow \mathbb{R} \cup +\infty$  has L bounded support if its effective domain is bounded by L  
 $h(\mathbf{u}) < +\infty \implies \|\mathbf{u}\| \leq L$
- **Definition** ( $\frac{1}{\mu}$ -Smoothness). A function  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  is  $\frac{1}{\mu}$  smooth if it is differentiable and its derivative is  $\frac{1}{\mu}$ -Lipschitz continuous

$$h(\mathbf{u}) \leq h(\mathbf{v}) + \nabla h(\mathbf{v})^T (\mathbf{u} - \mathbf{v}) + \frac{1}{2\mu} \|\mathbf{u} - \mathbf{v}\|^2 \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^m$$

- **Definition** ( $\mu$ -Strong Convexity). A function  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  is  $\mu$  strongly convex for  $\mu \geq 0$  if

$$h(\mathbf{u}) \geq h(\mathbf{v}) + \mathbf{s}^T (\mathbf{u} - \mathbf{v}) + \frac{\mu}{2} \|\mathbf{u} - \mathbf{v}\|^2 \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^m$$

for any  $\mathbf{s} \in \partial h(\mathbf{v})$ , where  $\partial h(\mathbf{v})$  denotes the subgradient of  $h$  at  $\mathbf{v}$

# A general framework for primal-dual methods

- We now study a general framework to ML problems having the form

$$\min_{\mathbf{u} \in \mathbb{R}^n} \ell(\mathbf{u}) + r(\mathbf{u}) \quad (I)$$

for convex functions  $\ell(\mathbf{u}) = \sum_i \ell_i(\mathbf{u})$  (the loss function) and  $r(\mathbf{u})$  (the regularizer function, e.g.  $\lambda \|\mathbf{u}\|_p$ ).

- This formulation includes ML problems such as Support Vector Machines, Linear and Logistic Regression, Lasso or Sparse Logistic Regression
- This general framework maps the ML problem (I) into one of the two following problems

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} O_A(\boldsymbol{\alpha}) = f(A\boldsymbol{\alpha}) + g(\boldsymbol{\alpha}) \quad (A)$$

$$\min_{\mathbf{w} \in \mathbb{R}^n} O_B(\mathbf{w}) = f^*(\mathbf{w}) + g^*(-A^T \mathbf{w}) \quad (B)$$

where  $\boldsymbol{\alpha} \in \mathbb{R}^n$  and  $\mathbf{w} \in \mathbb{R}^m$ ,  $A = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$  is a data matrix with column vectors  $\mathbf{x}_i \in \mathbb{R}^m \forall i$ , and  $f^*$  and  $g^*$  are the convex conjugates of  $f$  and  $g$  respectively. (A) is called primal. (B) dual.

# A general framework for primal-dual methods

- Optimization Problem (A) and (B) are equivalent according to the Fenchel-Rockafellar duality
- Given  $\alpha$  from (A), we achieve  $w$  of (B) as  $w = w(\alpha) := \nabla f(A\alpha)$
- (A) and (B) give the duality gap  $G(\alpha) := O_A(\alpha) - [-O_B(w(\alpha))]$
- Recall that the duality gap is always non negative and is zero if the pair  $(\alpha^*, w^*)$  is optimal. It gives an upper bound on the unknown primal or dual optimization error (certificate of the suboptimality) since

$$O_A(\alpha) \geq O_A(\alpha^*) \geq -O_B(w^*) \geq -O_B(w(\alpha))$$

- Assumption: Problem (A) is with  $f$   $\frac{1}{\tau}$ -smooth and the function  $g$  are separable  $g(\alpha) = \sum_i g_i(\alpha)$ , with  $g_i(\alpha)$  having  $L$ -bounded support.
- Given the equivalence between (A) and (B), this gives that in problem (B)  $f^*$  is  $\tau$ -strongly convex and the function  $g^*(-A^T w) = \sum_i g_i^*(-x_i^T w)$  is separable with each  $g_i^*$  being  $L$ -Lipschitz



# Common Losses and Regularizers

(i) Losses

Loss	Obj	$f / g^*$
Least Squares	(A)	$f = \frac{1}{2} \ A\alpha - \mathbf{b}\ _2^2$
	(B)	$g^* = \frac{1}{2} \ A^\top \mathbf{w} - \mathbf{b}\ _2^2$
Logistic Reg.	(A)	$f = \frac{1}{m} \sum_j \log(1 + \exp(b_j \mathbf{x}_j^\top \alpha))$
	(B)	$g^* = \frac{1}{n} \sum_i \log(1 + \exp(b_i \mathbf{x}_i^\top \mathbf{w}))$
SVM	(B)	$g^* = \frac{1}{n} \sum_i \max(0, 1 - y_i \mathbf{x}_i^\top \mathbf{w})$
Absolute Dev.	(B)	$g^* = \frac{1}{n} \sum_i  \mathbf{x}_i^\top \mathbf{w} - y_i $

(ii) Regularizers

Regularizer	Obj	$g / f^*$
Elastic Net	(A)	$g = \lambda(\eta \ \alpha\ _1 + \frac{1-\eta}{2} \ \alpha\ _2^2)$
	(B)	$f^* = \lambda(\eta \ \mathbf{w}\ _1 + \frac{1-\eta}{2} \ \mathbf{w}\ _2^2)$
$L_2$	(A)	$g = \frac{\lambda}{2} \ \alpha\ _2^2$
	(B)	$f^* = \frac{\lambda}{2} \ \mathbf{w}\ _2^2$
$L_1$	(A)	$g = \lambda \ \alpha\ _1$
Group Lasso	(A)	$g = \lambda \sum_p \ \alpha_{\mathcal{I}_p}\ _2, \mathcal{I}_p \subseteq [n]$

# Assumptions

- Our main interest is now to apply (A) or (B) for deriving a distributed solution to the initial ML problem (I).
- The data set  $A$  is distributed over  $K$  machines according to a partition  $\{\mathcal{P}_k\}_{k=1}^K$  of the columns of  $A \in \mathbb{R}^{m \times n}$ . The size of the partition on the machine  $k$  is  $n_k = |\mathcal{P}_k|$
- For machine  $k \in \{1, \dots, K\}$  and vector  $\alpha \in \mathbb{R}^n$ , let  $\alpha_{[k]} \in \mathbb{R}^n$  a vector with elements  $(\alpha_{[k]})_i := \alpha_i$  if  $i \in \mathcal{P}_k$  and  $(\alpha_{[k]})_i := 0$  otherwise
- Analogously, let  $A_{[k]}$  be a matrix with columns corresponding to those of  $A$  according to the partition, and zeros elsewhere
- The function  $g$  in (A) can be easily distributed, since  $g(\alpha) = \sum_i g_i(\alpha)$
- However, the function  $f(A\alpha)$  is not in general separable
- The main idea of the general framework for primal-dual methods is a separable approximation of the function  $O_A(\alpha)$ . See next

## Approximation of $O_A(\alpha)$

- Let  $\mathbf{v} := A\alpha \in \mathbb{R}^m$  and let  $\alpha_{[k]}^{(t+1)} := \alpha_{[k]}^{(t)} + \gamma \Delta\alpha_{[k]}$ , where  $\Delta\alpha_{[k]}$  denotes a certain change of variables  $\alpha_i$  for  $i \in \mathcal{P}_k$  and  $(\Delta\alpha_{[k]})_i := 0 \ \forall i \ni \mathcal{P}_k$
- Then,  $O_A(\alpha)$  can be exactly decomposed as follows

$$\sum_{i \in [n]} g_i(\alpha_i^{(t)} + \Delta\alpha_i) + f(\mathbf{v}^{(t)}) + \nabla f(\mathbf{v}^{(t)})^T A \Delta\alpha +$$

$$\frac{\sigma'}{2\tau} \Delta\alpha^T \begin{bmatrix} A_{[1]}^T A_{[1]} & & 0 \\ & \ddots & \\ 0 & & A_{[K]}^T A_{[K]} \end{bmatrix} \Delta\alpha = \sum_{k=1}^K G_k^{\sigma'}(\Delta\alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]})$$

$$G_k^{\sigma'}(\Delta\alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]}) := \frac{1}{K} f(\mathbf{v}^{(t)}) + \mathbf{w}^T A_{[k]} \Delta\alpha_{[k]} + \frac{\sigma'}{2\tau} \|A_{[k]} \Delta\alpha_{[k]}\|^2 + \sum_{i \in \mathcal{P}_k} g_i(\alpha_i^{(t)} + \Delta\alpha_{[k]i})$$

## Approximation of $O_A(\alpha)$

- The function  $G_k^{\sigma'}(\Delta\alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]}^{(t)})$  is completely local at processor  $k$  except the coupling variable  $\mathbf{v}^{(t)} = A\alpha^{(t)}$  which is global
- The decomposition of  $O_A(\alpha)$  suggests that we can iteratively solve local problems and exchange  $\alpha_k$  to reconstruct  $\mathbf{v}^{(t)}$

$$\min_{\Delta\alpha_k \in \mathbb{R}^n} G_k^{\sigma'}(\Delta\alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]}^{(t)})$$

- Each processor can do the local minimisation and just exchange to the others the variables  $\alpha_k$  at each iteration  $t$
- Note that the minimization is done independently from other processors  $k$  and thus the resulting  $G_k^{\sigma'}(\Delta\alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]}^{(t)})$  will not give the exact term to perfectly reconstruct  $O_A(\alpha)$ . However, this is enough to approximately compute the optimal solution with approximation  $\Theta$

# Algorithm 1: Generalized primal-dual algorithm

## Algorithm 1: Generalized primal-dual algorithm

**Input** Data matrix  $A$  distributed column-wise according to the partition  $\{\mathcal{P}_k\}_{k=1}^K$ , aggregation parameter  $\gamma \in (0, 1]$ , and  $\sigma'$ .

Starting point  $\alpha^{(0)} := 0 \in \mathbb{R}^n$ ,  $\mathbf{v}^{(0)} := 0 \in \mathbb{R}^m$

**for**  $t = 0, 1, \dots$  **do**

**for**  $k = 1, 2, \dots, K$  in parallel in each processor **do**

        Compute a  $\Theta$  approximate solution to

$$\min_{\Delta \alpha_k \in \mathbb{R}^n} G_k^{\sigma'}(\Delta \alpha_k; \mathbf{v}^{(t)}, \alpha_{[k]}^{(t)})$$

$$\alpha_{[k]}^{(t+1)} := \alpha_{[k]}^{(t)} + \gamma \Delta \alpha_{[k]}$$

$\Delta \mathbf{v}_k := A_{[k]} \Delta \alpha_{[k]}$ . Transmit to the other processors  $\Delta \mathbf{v}_k$

**end for**

        Compute  $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + \gamma \sum_{k=1}^K \Delta \mathbf{v}_k$

**end for**

# Application to primal and dual

## Algorithm 2: Primal mapping

**Map** Problem (I) into (A)

**Distribute** dataset  $A$  by columns (here typically features) according to the partition  $\{\mathcal{P}_k\}_{k=1}^K$

**Run** Algorithm 1 with appropriate choice of parameter  $\gamma$  and sub-problem parameter  $\sigma'$

## Algorithm 3: Dual mapping

**Map** Problem (I) into (B)

**Distribute** dataset  $A$  by columns (here typically training points) according to the partition  $\{\mathcal{P}_k\}_{k=1}^K$

**Run** Algorithm 1 with appropriate choice of parameter  $\gamma$  and sub-problem parameter  $\sigma'$

## Algorithm 1 for convex $g_i$ and L-Lipschitz $g_i^*$

- **Theorem 1:** Consider Algorithm 1 with  $\gamma := 1$ , and let  $\Theta$  be the quality of the local solver at processor  $k$ . Let  $g_i$  have  $L$  bounded support, and let  $f$  be  $\frac{1}{\tau}$ -mooth. Let  $T$  be such that

$$T \geq T_0 + \max \left( \left\lceil \frac{1}{1 - \Theta} \right\rceil, \frac{4L^2}{\tau \varepsilon_G (1 - \Theta)} \right)$$

$$T_0 \geq t_0 + \left\lceil \frac{2}{1 - \Theta} \left( \frac{8L^2}{\tau \varepsilon_G} - 1 \right) \right\rceil$$

$$t_0 \geq \max \left( 0, \left\lceil \frac{1}{1 - \Theta} \log \left( \frac{\tau n (O_A(\boldsymbol{\alpha}^{(0)})) - O_A(\boldsymbol{\alpha}^*)}{2L^2 K} \right) \right\rceil \right)$$

Then

$$\mathbb{E}[O_A(\bar{\boldsymbol{\alpha}}) - (-O_B(\mathbf{w}(\bar{\boldsymbol{\alpha}})))] \leq \varepsilon_G \quad \bar{\boldsymbol{\alpha}} = \frac{1}{T - T_0} \sum_{t=T_0+1}^{T-1} \boldsymbol{\alpha}^{(t)}$$

## Algorithm 1 for strong. convex $g_i$ and smooth $g_i^*$

- **Theorem 2:** Consider Algorithm 1 with  $\gamma := 1$ , and let  $\Theta$  be the quality of the local solver. Let  $g_i$  be  $\mu$  strongly convex  $\forall i$  and let  $f$  be  $\frac{1}{\tau}$ -mooth. Let  $T$  be such that

$$T \geq \frac{1}{1 - \Theta} \frac{\mu\tau + 1}{\mu\tau} \log \frac{1}{\varepsilon_{O_A}}$$

Then  $\mathbb{E}[O_A(\boldsymbol{\alpha}^{(T)}) - O_A(\boldsymbol{\alpha}^*)] \leq \varepsilon_{O_A}$

Moreover, if

$$T \geq \frac{1}{1 - \Theta} \frac{\mu\tau + 1}{\mu\tau} \log \left( \frac{1}{1 - \Theta} \frac{\mu\tau + 1}{\mu\tau} \frac{1}{\varepsilon_{O_A}} \right)$$

then the expected duality gap

$$\mathbb{E}[O_A(\boldsymbol{\alpha}^{(T)}) - (-O_B(\boldsymbol{w}(\boldsymbol{\alpha}^T)))] \leq \varepsilon_G$$



# Criteria for Running Algorithms 2 vs. 3

	Smooth $\ell$	Non-smooth and separable $\ell$
Strongly convex $r$	Alg. 2 or 3	Alg. 3
Non-strongly convex and separable $r$	Alg. 2	-

	Smooth $\ell$	Non-smooth and separable $\ell$
Strongly convex $r$	Theorem 3	Theorem 2
Non-strongly convex and separable $r$	Theorem 2	-

# Comparison with ADMM

- We can apply consensus ADMM to (B) (or (A)):

$$\min_{\mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{w}} \sum_{k=1}^K \sum_{i \in \mathcal{P}_k} g^*(-\mathbf{x}_i^T \mathbf{w}_k) + f^*(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{w}_k = \mathbf{w} \quad \forall k$$

- We solve the problem by the augmented Lagrangian

$$\mathbf{w}_k^{(t+1)} = \arg \min_{\mathbf{w}_k} \sum_{i \in \mathcal{P}_k} g^*(-\mathbf{x}_i^T \mathbf{w}_k) + \rho \mathbf{u}_k^{(t)T} (\mathbf{w}_k - \mathbf{w}^{(t)}) + \frac{\rho}{2} \|\mathbf{w}_k - \mathbf{w}^{(t)}\|^2$$

$$\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w}} f^*(\mathbf{w}) + \frac{\rho K}{2} \|\mathbf{w} - (\bar{\mathbf{w}}_k^{(t+1)} - \bar{\mathbf{u}}_k^{(t)})\|^2$$

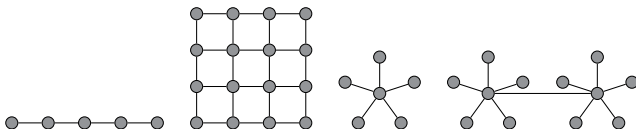
$$\mathbf{u}_k^{(t+1)} = \mathbf{u}_k^{(t)} + \mathbf{w}_k^{(t+1)} - \mathbf{w}^{(t+1)}$$

- ADMM has the drawback of the proximal updating

# Outline

1. Computation-communication tradeoff in a general approach
2. Quantized Distributed Gradient Descent
3. Parallel Quantized Stochastic Gradient Descent

# Problem formulation



- Set of  $n$  nodes  $\mathcal{V} = (1, \dots, n)$ , a set of edges  $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ . The nodes communicate over a connected and undirected graph  $\mathcal{G} = (\mathcal{G}, \mathcal{E})$
- $\mathcal{N}_i$  is the set of neighbours that node  $i$  communicates with
- Each node  $i$  has a strongly convex and smooth function  $f_i(\mathbf{w}) : \mathbb{R}^p \rightarrow \mathbb{R}$
- All the nodes wish to solve the ML optimization problem
$$\underset{\mathbf{w} \in \mathbb{R}^p}{\text{minimize}} \quad f(\mathbf{w}) = \underset{\mathbf{w} \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{|\mathcal{N}_i|} \sum_{i \in \mathcal{N}_i} f_i(\mathbf{w})$$
- Clearly,  $f(\mathbf{w})$  is strongly convex and smooth and there is a unique minimizer  $\mathbf{w}^*$

# Problem formulation

- A node has only access to its local function and it can communicate only with the neighbours  $\mathcal{N}_i$
- As we have seen in the previous lectures, we can equivalently rewrite the ML optimization problem by the consensus method as

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{|\mathcal{N}_i|} \sum_{i \in \mathcal{N}_i} f_i(\mathbf{w}) \\ & \text{s.t.} && \mathbf{w}_i = \mathbf{w}_j \quad \forall i, j \in \mathcal{N}_i \end{aligned}$$

- We could solve the problem by the methods of the previous lectures with local iterates
- However, the nodes cannot exchange the decision variables  $\mathbf{w}_{i,t}$ , but a quantized version  $\mathbf{z}_{i,t} = Q(\mathbf{w}_{i,t})$ , where  $Q(\cdot)$  is a quantizer function
- The quantization can substantially reduce the amount of information to exchange, which is very important, e.g., in IoT applications

# Quantized Distributed Gradient Descent (QDGD)

## Algorithm 4: QDGD

Node  $i$  requires Weights  $\{a_{i,j}\}_{j=1}^n$

Set  $\mathbf{w}_{i,0} = 0$  and compute  $\mathbf{z}_{i,0} = Q(\mathbf{w}_{i,0})$

**for**  $t = 0, 1, \dots, T - 1$  **do**

Transmit  $\mathbf{z}_{i,t} = Q(\mathbf{w}_{i,t})$  to  $\mathcal{N}_i$  and receive  $\mathbf{z}_{j,t}$

Compute the local decision variable as

$$\mathbf{w}_{i,t+1} = (1 - \varepsilon + \varepsilon a_{i,i})\mathbf{w}_{i,t} + \varepsilon \sum_{j \in \mathcal{N}_i} a_{i,j} \mathbf{z}_{j,t} - \alpha \varepsilon \nabla f_i(\mathbf{w}_{i,t})$$

**end for**

Return  $\mathbf{w}_{i,T}$

- $\varepsilon$  and  $\alpha$  are positive step sizes to be appropriately chosen
- There are no particular restrictions on the type of quantizer (see later)

# QDGD Convergence analysis

- **Assumption 1:**  $\forall \mathbf{w} \in \mathbb{R}^p, \mathbf{y} \in \mathbb{R}^p, f_i$  is differentiable and smooth with parameter  $L$

$$\|\nabla f_i(\mathbf{w}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{w} - \mathbf{y}\| \quad \forall i$$

- **Assumption 2:**  $\forall \mathbf{w} \in \mathbb{R}^p, \mathbf{y} \in \mathbb{R}^p, f_i$  is strongly convex with parameter  $\mu$

$$(\nabla f_i(\mathbf{w}) - \nabla f_i(\mathbf{y}))^T (\mathbf{w} - \mathbf{y}) \geq \mu\|\mathbf{w} - \mathbf{y}\|^2 \quad \forall i$$

- **Assumption 3:** The quantizer is unbiased and has a bounded variance:

$$\mathbb{E}[Q(\mathbf{w})|\mathbf{w}] = \mathbf{w} \quad \mathbb{E}[\|Q(\mathbf{w}) - \mathbf{w}\|^2|\mathbf{w}] \leq \sigma^2$$

- **Assumption 4:** The matrix  $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n,n}$  is symmetric and doubly stochastic:

$$\mathbf{A} = \mathbf{A}^T \quad \mathbf{A}\mathbf{1} = \mathbf{1} \quad \mathbf{A}^T\mathbf{1} = \mathbf{1}$$

# QDGD Convergence analysis

- **Theorem 4:** Consider the QDGD Algorithm. Suppose Assumptions 1 ~ 4 hold. Let  $\delta$  be an arbitrary scalar in  $(0, 1/2)$  and let  $\varepsilon = c_1/T^{3\delta/2}$  and  $\alpha = c_2/T^{\delta/2}$ , where  $c_1$  and  $c_2$  are arbitrary positive constants independent of  $T$ . Then, for each node  $i$

$$\mathbb{E} [\|\mathbf{w}_{i,T} - \mathbf{w}^*\|^2] \leq \mathcal{O} \left( \left( \frac{4nc_2^2 D^2 (3 + 2L/\mu)^2}{(1 - \beta)^2} + \frac{2c_1 n \sigma^2 \|\mathbf{A} - \mathbf{A}_D\|}{\mu c_2} \right) \frac{1}{T^\delta} \right)$$

where

$$D^2 = 2L \sum_{i=1}^n (f_i(0) - f_i^*), \quad f_i^* = \min_{\mathbf{w} \in \mathbb{R}^p} f_i(\mathbf{w})$$

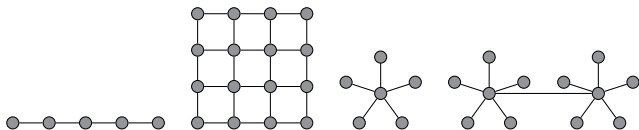
- The theorem shows that QDGD provides an approximation solution with vanishing deviation from the optimal solution, despite the quantization noise that does not vanish with the iterations
- The convergence rate is sublinear



# Outline

1. Computation-communication tradeoff in a general approach
2. Quantized Distributed Gradient Descent
3. Parallel Quantized Stochastic Gradient Descent

# Stochastic Gradient Descent (SGD)



- Set of  $n$  nodes  $\mathcal{V} = (1, \dots, n)$ , a set of edges  $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ . The nodes communicate over a connected and undirected graph  $\mathcal{G} = (\mathcal{G}, \mathcal{E})$
- Let  $\mathcal{W}$  be a known convex set. There is a global function  $f(\mathbf{w}) : \mathcal{W} \rightarrow \mathbb{R}$  which is unknown to the nodes
- Each node  $i$  has access to its measurement of the stochastic gradient of  $f(\mathbf{w})$
- All the nodes wish to solve the ML optimization problem  $\underset{\mathbf{w} \in \mathbb{R}^p}{\text{minimize}} f(\mathbf{w})$

# SGD

- **Definition 1:** Given the function  $f(\mathbf{w}) : \mathcal{W} \rightarrow \mathbb{R}$ , a stochastic gradient of  $f$  is a random function  $\tilde{g}(\mathbf{w})$  so that  $\mathbb{E}[\tilde{g}(\mathbf{w})] = \nabla f(\mathbf{w})$
- **Definition 2:** The stochastic gradient has second order moment at most  $B$  if  $\mathbb{E}[\|\tilde{g}(\mathbf{w})\|^2] \leq B$  for  $\mathbf{w} \in \mathcal{W}$
- **Definition 3:** The stochastic gradient has variance at most  $\sigma^2$  if  $\mathbb{E}[\|\tilde{g}(\mathbf{w}) - \nabla f(\mathbf{w})\|^2] \leq \sigma^2$  for  $\mathbf{w} \in \mathcal{W}$ .

# SGD

- A standard instance of the Stochastic Gradient Descent (SGD) is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \tilde{g}(\mathbf{w}_t)$$

where  $\eta_t$  is variable step size

- **Theorem 5:** Let  $\mathcal{W} \subseteq \mathbb{R}^n$  be convex and let the function  $f(\mathbf{w}) : \mathcal{W} \rightarrow \mathbb{R}$  be unknown, convex, and  $L$ -smooth. Let  $\mathbf{w}_0 \in \mathcal{W}$  be given and let  $R^2 = \sup_{\mathbf{w} \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}_0\|^2$ . Let  $T \geq 0$  be fixed. Given repeated and independent access to stochastic gradients with variance bound  $\sigma^2$ , the SGD with constant step size  $\eta_t = 1/(L + 1/\gamma)$  where  $\gamma = R/\sigma\sqrt{2/T}$  achieves

$$\mathbb{E} \left[ f \left( \frac{1}{T} \sum_{t=0}^T \mathbf{w}_t \right) \right] - \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \leq R \sqrt{\frac{2\sigma^2}{T}} + \frac{LR^2}{T}$$

# Parallel SGD

- If we have  $K$  processors each making an independent measurement of the stochastic gradient  $\tilde{g}^i(\mathbf{w})$ , and each processor  $i$  communicates to each other such measurement at every time step  $t$ , a parallel SGD is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{K} \sum_{i=1}^K \tilde{g}^i(\mathbf{w}_t)$$

- **Corollary 1:** Let  $\mathcal{W}$ ,  $f(\mathbf{w})$ ,  $\mathbf{w}_0$  and  $R$  as in the previous theorem. Fix  $\varepsilon \geq 0$ . Suppose to run parallel SGD on  $K$  processors each with access to independent stochastic gradients with second moment bound  $B$ , with step size  $\eta_t = 1/(L + \sqrt{K}/\gamma)$  with  $\gamma$  as in the previous theorem. If  $T = \mathcal{O}\left(R^2 \max\left(\frac{2B}{K\varepsilon^2}, \frac{L}{\varepsilon}\right)\right)$  then

$$\mathbb{E} \left[ f \left( \frac{1}{T} \sum_{t=0}^T \mathbf{w}_t \right) \right] - \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \leq \varepsilon$$

# Parallel Quantized SGD

## Algorithm 5: PQSGD

**for**  $t = 0, 1, \dots, T - 1$  **do**

Let  $\tilde{g}^i(\mathbf{w}_t)$  be an independent stochastic gradient

Broadcast  $\mathbf{z}_{i,t} = Q(\tilde{g}^i(\mathbf{w}_t))$  to all nodes and receive  $\mathbf{z}_{j,t}$

Compute the local estimate of the global decision variable as

$$\mathbf{w}_{i,t+1} = \mathbf{w}_{i,t} - \frac{\eta_t}{K} \sum_{i=1}^K \mathbf{z}_{i,t}$$

**end for**

Return  $\mathbf{w}_{i,T}$

- Where  $Q(\cdot)$  is a quantizer (see below)
- Does the algorithm converge? Not in general...

# Quantization

- Let  $\mathbf{v} \in \mathbb{R}^n$  with  $\mathbf{v} \neq 0$ , and let  $s \geq 1$ . The “low precision quantizer” is

$$Q_s(\mathbf{v}) = [Q_s(v_i) = \|\mathbf{v}\|_2 \operatorname{sgn}(v_i) \xi_i(\mathbf{v}, s)]$$

where  $\xi_i(\mathbf{v}, s)$  are independent random variables with outcome

$$\xi_i(\mathbf{v}, s) = \begin{cases} \ell/s & \text{with probability } 1 - p\left(\frac{|v_i|}{\|\mathbf{v}\|_2}, s\right) \\ (\ell + 1)/s & \text{otherwise} \end{cases}$$

with  $p(a, s) = as - \ell$  for any  $a \in [0, 1]$ , and the integer  $0 \leq \ell < s$  to be chosen such that  $|w_i|/\|\mathbf{w}\| \in [\ell/s, (\ell + 1)/s]$

- $\ell$  is the quantization index, and  $s$  is the upper bound of the quantization levels
- Example: if  $s = 1$ , the quantization levels are  $0, 1, -1$

# Quantization

- Motivation:  $\xi_i(\mathbf{v}, s)$  has minimal variance over distributions with support  $\{0, 1/s, \dots, 1\}$
- **Lemma:** For any vector  $\mathbf{v} \in \mathbb{R}^n$ , 1)  $\mathbb{E}[Q_s(\mathbf{v})] = \mathbf{v}$  (unbiasedness) 2)  $\mathbb{E}[\|Q_s(\mathbf{v}) - \mathbf{v}\|_2^2] \leq \min(n/s^2, \sqrt{n}/s) \|\mathbf{v}\|_2^2$  (variance bound), and 3)  $\mathbb{E}[\|Q_s(\mathbf{v})\|_0] \leq s(s + \sqrt{n})$
- **Theorem:** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be fixed, and let  $\mathbf{w} \in \mathbb{R}^n$  be arbitrary. Fix  $s \geq 2$  quantization levels. If  $\tilde{g}(\mathbf{w})$  is a stochastic gradient for  $f$  at  $\mathbf{w}$  with second order moment  $B$ , then  $Q_s(\tilde{g}(\mathbf{w}))$  is a stochastic gradient for  $f$  at  $\mathbf{w}$  with variance bound  $\min(n/s^2, \sqrt{n}/s)B$ . There is an encoding scheme so that in expectation, the number of bits to communicate  $Q_s(\tilde{g}(\mathbf{w}))$  is upper bounded by

$$\left(3 + \left(\frac{3}{2} + o(1)\right) \log \left(\frac{2(s^2 + n)}{s(s + \sqrt{n})}\right)\right) s(s + \sqrt{n}) + 32$$



# Convergence of Parallel QSGD

- **Theorem 6** (Smooth Convex Parallel QSGD). Let  $\mathcal{W}$ ,  $f(\mathbf{w})$ ,  $\mathbf{w}_0$ ,  $R$  and  $\gamma$  as in the main SGD convergence theorem. Let  $\varepsilon > 0$ . Suppose to run the Parallel QSGD algorithm on  $K$  processors accessing independent stochastic gradients with second moment bound  $B$ , with step size  $\eta_t = 1/(L + \sqrt{K}/\gamma)$  with  $\sigma = B'$  with  $B' = \min(\frac{n}{s^2}, \frac{\sqrt{n}}{s})B$ . If  $T = \mathcal{O}\left(R^2 \max\left(\frac{2B'}{K\varepsilon^2}, \frac{L}{\varepsilon}\right)\right)$  then

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=0}^T \mathbf{w}_t\right)\right] - \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \leq \varepsilon$$

Moreover, the Parallel QSGD requires a number of bits given by the previous theorem per communication round. If  $s = \sqrt{n}$ , the number of bits is reduced to  $2.8n + 32$ .

# Convergence of Parallel QSGD

- **Theorem** (Smooth non Convex Parallel QSGD). Let  $\mathcal{W}$ ,  $\mathbf{w}_0$ ,  $R$  and  $\gamma$  as in the main SGD convergence theorem. Let  $f(\mathbf{w})$  be an  $L$ -smooth possibly non-convex function, and let  $\mathbf{w}_1$  be an arbitrary initial point. Let  $T > 0$  be fixed, and  $s > 0$ .

Then there is a random stopping time  $R$  supported on  $\{1, \dots, N\}$  so that the Parallel QSGD with quantization level  $s$  constant stepsizes  $\eta = \mathcal{O}(1/L)$  and access to stochastic gradients of  $f$  with second moment bound  $B$  satisfies

$$\frac{1}{L} \mathbb{E} [\|\nabla f(\mathbf{w})\|_2^2] \leq \mathcal{O} \left( \frac{\sqrt{L(f(\mathbf{w}_1) - f^*)}}{N} + \frac{B \min(n/s^2, \sqrt{n}/s)}{L} \right)$$

Moreover, the number of bits to communicate for each gradient transmission is the same as in the previous theorem

## CA6: Communication efficiency

Split the “MNIST” dataset to 10 random disjoint subsets, each for one worker, and consider SVM classifier in the form of  $\min_{\mathbf{w}} \frac{1}{N} \sum_{i \in [N]} f_i(\mathbf{w})$  with  $N = 10$ . An alternative approach to improve communication-efficiency is to compress the information message to be exchanged (usually gradients – either in primal or dual forms). Consider two compression/quantization methods for a vector: (Q1) keep only  $K$  values of a vector and set the rest to zero and (Q2) represent every element with fewer bits (e.g., 4 bits instead of 32 bits).

- a) Repeat parts a-b from CA5 using Q1 and Q2. Can you integrate Q1/Q2 to your solution in part c from CA5? Discuss.
- b) How do you make SVRG and SAG communication efficient for large-scale ML?

## Some references

- V. Smith, S. Forte, C. M. M. Takáč, M. I. Jordan, M. Jaggi, “CoCoA: A general Framework for Communication-Efficient Distributed Optimization”, JMLR, 2018.
- A. Reisizadeh, A. Mokhtari, H- Hassani, R. Pedarsani, “An Exact Quantized Decentralized Gradient Descent Algorithm,” arXiv, 2018.
- D. Alistarh, D. Grubic, J. Li, R. Tomioka, M. Vojnovic, “QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding”, NIPS, 2017.