COMPRESSED DISTRIBUTED REGRESSION OVER ADAPTIVE NETWORKS

Marco Carpentiero* *Vincenzo Matta** *Ali H. Sayed*†

* DIEM, University of Salerno, Fisciano (SA), Italy
† EPFL, School of Engineering, CH-1015 Lausanne, Switzerland

ABSTRACT

We examine the learning performance achievable by a network of agents that solve a distributed regression problem using the recently proposed ACTC (Adapt-Compress-Then-Combine) diffusion strategy. The agents operate under *communication constraints*: they are allowed to communicate only with their immediate neighbors, and the exchanged signals are encoded by using *randomized differential* compression operators. We show that the mean-square estimation error of each agent comprises the error that the agents would achieve without communication constraints plus a *compression loss*. Our results reveal the fundamental quantitative relationship existing between the compression loss and the peculiar attributes of the distributed regression problem. We show how these quantitative relationships can be used to optimize the allocation of communication resources across the agents and improve their learning performance as compared to a uniform allocation.

Index Terms— Distributed optimization, diffusion strategy, randomized quantizers, differential quantization.

1. INTRODUCTION AND RELATED WORK

A distributed learning system consists of a group of cognitive agents, linked by a graph, which cooperate to solve a common inferential task. Two major distributed paradigms are *federated learning* and *fully-decentralized learning* [1–6]. In this work, we focus on the latter paradigm [1–3], where there is no fusion center, and each agent accomplishes the learning task by means of local exchanges of information with its neighbors. We work under the online setting where the learning algorithm must be *adaptive* to track drifts in the streaming data. Common tools adopted in this setting are stochastic gradient algorithms with *constant* step-size [1], and distributed strategies like *consensus* [7, 8] or *diffusion* [1–3].

The communication burden due to information transmission among network agents represents a bottleneck for distributed learning. As a result, information compression strategies, e.g., quantization [9], need to be considered to cope with communication constraints. Data compression has already been successfully applied in distributed inferential systems [10–13], but distributed optimization, which is the core of this work, poses peculiar challenges. Two major issues are: i) the data distribution is unknown, preventing the use of traditional quantizer design [9]; and ii) the convergence of the iterative optimization algorithms can be impaired by the accumulation of compression errors. Recent works showed that the aforementioned issues can be alleviated using *randomized* [14–16] and *differential* [17] compression strategies. Randomized compression operators generate randomly coded outputs that exhibit properties useful for inferential purposes, such as unbiasedness and bounded variance. Examples of randomized compression operators applied in distributed optimization are randomized quantizers [14, 18] and randomized sparsifiers [15, 16]. Building on well-established techniques for the design of communication systems, e.g., Differential Pulse Code Modulation and Delta Modulation [9], differential compression can be used to exploit the dependence between consecutive samples in the optimization algorithm. By encoding only the difference between consecutive samples, the input range of the encoder can be considerably reduced, implying significant savings in terms of communication resources.

There already exist some useful works dealing with gradient-based algorithms in the presence of randomized and/or differential compression [19–23]. In these works, there are some restrictive design choices such as: i) non-adaptive implementations with diminishing step-size; ii) symmetric combination policies to model the network; iii) strong convexity at *all* agents; iv) consensus strategies to merge the information shared by the agents across the network, which lead to reduced stability ranges compared to diffusion strategies. Recent works [24–27] overcome these limitations and consider instead adaptive algorithms with constant step-size to track data drifts; non-symmetric and left-stochastic combination policies, or subspace constraints [26, 27] to represent a wide variety of network scenarios; strong convexity only at the *global level*, thus allowing for convex and non-convex cost functions at the local level; diffusion strategies, which have been shown to lead to superior convergence and mean-square-error performance under adaptive scenarios [1].

In [24, 25], a novel diffusion strategy nicknamed ACTC (Adapt-Compress-Then-Combine) is proposed. To meet communication constraints, the ACTC algorithm incorporates in the classical ATC (Adapt-Then-Combine) diffusion [1] an intermediate step of randomized differential compression. The *mean-square stability* and the *transient behavior* of the ACTC diffusion strategy were characterized in great detail in [24, 25]. It was shown that, by tuning a suitable design parameter, mean-square stability is guaranteed despite data compression. All agents converge to a small neighborhood of the desired solution with the same transient behavior of the ATC strategy, but spending significantly less transmission resources.

This work complements the aforementioned analysis of the ACTC strategy by characterizing its *steady-state performance*. The analysis focuses on the popular decentralized learning setting where distributed agents cooperate to solve a linear regression problem [1]. In the context of communication-constrained systems, this setting was examined in [28], where imperfect communication was modeled through noisy links. In comparison, in this work we take into account the compression mechanism (e.g., the quantizers) and the related budget (e.g., the bit-rates). Accounting for these elements in the analysis will be critical to perform an optimized allocation of the communication resources. Specifically: i) we prove the existence of an upper bound on the mean-square-error of each network agent, which is the sum of the error achieved without communication constraints and of a *compression loss*; ii) we derive useful quantitative relationships between the compression loss and the attributes of the distributed learning problem, e.g., the network topology and the different error sources such as gradient noise and bias; and iii) exploiting the obtained formulas, we show that communication resources can be allocated in an informed manner according to the features of each agent, improving the learning performance as compared to a uniform allocation.

Notation. We denote random variables with bold font and their realizations with normal font. All vectors are column vectors. I_M is the identity matrix of size M. For a matrix X, the notation $X > 0$ means that X is positive definite, whereas $\mathrm{Tr}(X)$ is the trace of X. The symbol $\mathbb{E}$ denotes the expectation operator.

2. BACKGROUND

We focus on a distributed linear regression problem tackled by a network of N agents, labeled $k = 1, 2, \dots, N$, which operate in an online setting by observing, over time epochs $i = 0, 1, \dots$, a streaming sequence of random variables $\boldsymbol{d}_{k,i} \in \mathbb{R}$ and random regression vectors $\boldsymbol{u}_{k,i} \in \mathbb{R}^M$ with covariance matrix $R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^\top]$. The processes $\{\boldsymbol{d}_{k,i}\}$ and $\{\boldsymbol{u}_{k,i}\}$ obey the linear regression model:

$$\boldsymbol{d}_{k,i} = \boldsymbol{u}_{k,i}^\top w_k^o + \boldsymbol{v}_{k,i}, \tag{1}$$

where $\boldsymbol{v}_{k,i}$ is a zero-mean additive noise process with $\mathbb{E}[\boldsymbol{v}_{k,i}^2] = \sigma_{v,k}^2$ and independent from the regressors, and $w_k^o \in \mathbb{R}^M$ is an unknown deterministic parameter vector. The processes $\{\boldsymbol{u}_{k,i}\}$ and $\{\boldsymbol{v}_{k,i}\}$ are independent over time and across the agents. Each agent is equipped with a local mean-square-error risk function:

$$J_k(w) = \mathbb{E}\left[\left(\boldsymbol{d}_{k,i} - \boldsymbol{u}_{k,i}^\top w\right)^2\right], \tag{2}$$

where the expectation is taken w.r.t. the random data $\{\boldsymbol{d}_{k,i}, \boldsymbol{u}_{k,i}\}$. By means of cooperation, the agents wish to minimize the following *global* cost function:

$$J(w) = \sum_{k=1}^{N} p_k J_k(w), \tag{3}$$

for some positive and convex (i.e., adding up to one) weights $\{p_k\}$.

Assumption 1 (Global Strong Convexity). *At least one agent has a positive definite covariance matrix $R_{u,k}$, which implies that the global cost function* (3) *is ν-strongly convex. Since $J(w)$ is twice differentiable, we have $\nabla^2 J(w) \geq \nu I_M$ for some $\nu > 0$.*

Under Assumption 1, the cost function in (3) has a unique minimizer $w^\star$, which can be shown to be a Pareto optimal solution relative to the individual cost functions $J_k(w)$ [1, 31]. By varying the weights $\{p_k\}$ in (3), we can attain different Pareto optimal solutions.

One popular tool to minimize (3) is a decentralized gradient-descent algorithm, which would rely on the *exact* gradient:

$$\nabla J_k(w) = 2\left(R_{u,k} w - r_{du,k}\right), \tag{4}$$

where $r_{du,k} = R_{u,k} w_k^o$. However, the exact gradient is not available in learning applications, as it depends on the data moments, which in turn depend on the same *unknown* parameters the inferential process is attempting to learn. For this reason, we will resort instead to the *stochastic* gradient-descent algorithm, were $\nabla J_k(w)$ is replaced by a *stochastic instantaneous* approximation thereof:

$$\boldsymbol{g}_{k,i}(w) = 2\,\boldsymbol{u}_{k,i}\left[\boldsymbol{u}_{k,i}^\top w - \boldsymbol{d}_{k,i}\right]. \tag{5}$$

Since the agents use a stochastic approximation of the actual gradient, we also introduce the *gradient noise*:

$$\boldsymbol{s}_{k,i}(w) \triangleq \boldsymbol{g}_{k,i}(w) - \nabla J_k(w), \tag{6}$$

and its covariance matrix evaluated at the minimizer $w^\star$, denoted by $R_{s,k}$, which can be shown to be [1]:

$$R_{s,k} = 4\,\sigma_{v,k}^2\, R_{u,k} + 4\,\mathbb{E}\left[\boldsymbol{U}_{k,i}(w^\star - w_k^o)(w^\star - w_k^o)^\top \boldsymbol{U}_{k,i}^\top\right], \tag{7}$$

where $\boldsymbol{U}_{k,i} = \boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^\top - R_{u,k}$.

Finally, we introduce the *bias* vector b, whose k-th entry b_k quantifies how far from 0 is the gradient of agent k evaluated at $w^\star$:

$$b_k \triangleq \nabla J_k(w^\star) = 2\left(R_{u,k} w^\star - r_{du,k}\right). \tag{8}$$

Note that the biases b_k are all equal to zero in the important case where all costs $J_k(w)$ are minimized at the same location, i.e., when the global minimizer $w^\star$ coincides with the unknown parameter vector w_k^o of each agent.

2.1. Network Graph and Combination Matrix

The agents are arranged in a network described by a *combination matrix* $A = [a_{\ell k}]$. When no link is present between two agents ℓ and k, the weights $a_{\ell k}$ and $a_{k\ell}$ must be equal to zero. Conversely, when information can flow only in one direction, say from ℓ to k, we have $a_{\ell k} > 0$ and $a_{k\ell} = 0$. Accordingly, the *directed* neighborhood of agent k (possibly including the self-loop $\ell = k$) is $\mathcal{N}_k \triangleq \{\ell = 1, 2, \dots, N : a_{\ell k} > 0\}$.

Assumption 2 (Strongly-Connected Network). *Given any pair of nodes (ℓ, k), a path with nonzero weights exists in both directions (i.e., from ℓ to k and vice versa), and at least one agent k in the entire network has a self-loop ($a_{kk} > 0$).* □

Assumption 3 (Left-Stochastic Combination Matrix). *For each agent $k = 1, ..., N$, the following conditions hold: $a_{\ell k} \geq 0$, $\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1$, $a_{\ell k} = 0$ for $\ell \notin \mathcal{N}_k$.* □

Under Assumptions 2 and 3, the combination matrix A is a primitive matrix and, from the Perron-Frobenius theorem, it has an eigenvector $\pi = [\pi_1, \pi_2, \dots, \pi_N]^\top$, the *Perron vector*, with all strictly positive entries such that: $\sum_{k=1}^{N} \pi_k = 1$ and $A\pi = \pi$ [1].

2.2. The ACTC Diffusion Strategy

In order to solve the distributed linear regression problem, the agents employ the ACTC strategy [24, 25], which consists of the following three steps, performed iteratively by all agents and for all time epochs $i > 0$.[1]

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\, \boldsymbol{g}_{k,i}(\boldsymbol{w}_{k,i-1}) \tag{9a}$$

$$\boldsymbol{q}_{\ell,i} = \boldsymbol{q}_{\ell,i-1} + \zeta\, \boldsymbol{Q}_\ell(\boldsymbol{\psi}_{\ell,i} - \boldsymbol{q}_{\ell,i-1}), \quad \forall \ell \in \mathcal{N}_k \tag{9b}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{q}_{\ell,i} \tag{9c}$$

— *Adaptation step* (9a): each agent k follows the descent direction $-\boldsymbol{g}_{k,i}(\cdot)$ in (5), weighted by the step-size $\mu > 0$, to update the past iterate $\boldsymbol{w}_{k,i-1}$, yielding the intermediate value $\boldsymbol{\psi}_{k,i}$.

[1] At time $i = 0$ each agent k is initialized with an arbitrary vector $\boldsymbol{q}_{k,0}$ (with finite second moment), receives the initial states $\{\boldsymbol{q}_{\ell,0}\}_{\ell \in \mathcal{N}_k}$ and computes an initial minimizer $\boldsymbol{w}_{k,0} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{q}_{\ell,0}$.

— *Compression step* (9b): each agent k compresses the difference, i.e., the *innovation*, between $\boldsymbol{\psi}_{k,i}$ and the previous quantized state $\boldsymbol{q}_{k,i-1}$, applying a suitable randomized operator $\boldsymbol{Q}_k(\cdot)$. All agents simultaneously transmit the encoded signals over the network. Then, each agent k receives from its neighbors $\ell \in \mathcal{N}_k$ the compressed differences $\boldsymbol{Q}_\ell(\boldsymbol{\psi}_{\ell,i} - \boldsymbol{q}_{\ell,i-1})$. The quantized states $\boldsymbol{q}_{\ell,i}$, for $\ell \in \mathcal{N}_k$, are recovered by adding to the previous states $\boldsymbol{q}_{\ell,i-1}$ the received compressed differences, scaled by a design parameter $\zeta \in (0,1)$ governing the stability of the ACTC strategy — see [24,25].

— *Combination step* (9c): each agent k computes a weighted combination of the updated quantized states $\{\boldsymbol{q}_{\ell,i}\}_{\ell\in\mathcal{N}_k}$, yielding the minimizer estimate $\boldsymbol{w}_{k,i}$.

2.3. Compression Operators

Following [14, 19, 21, 23–25], we implement the ACTC strategy in (9) relying on the following class of *randomized* operators.

Assumption 4 (Compression operators). *Given a constant $\omega > 0$ and a deterministic input $x \in \mathbb{R}^M$, the randomized compression operator $\boldsymbol{Q} : \mathbb{R}^M \to \mathbb{R}^M$ satisfies the following properties:*

$$\mathbb{E}\left[\boldsymbol{Q}(x) - x\right] = 0 \qquad \text{[unbiasedness]} \tag{10}$$

$$\mathbb{E}\,\|\boldsymbol{Q}(x) - x\|^2 \leq \omega\,\|x\|^2 \qquad \text{[non blow-up property]} \tag{11}$$

where expectations are evaluated w.r.t. the randomness of the operator. When the input is random, conditions (10) *and* (11) *are intended to hold conditionally on the input. Moreover, when applied in the middle of the ACTC strategy, given the past history $\{\{\boldsymbol{\psi}_{\ell,j}\}_{j=1}^{i}, \{\boldsymbol{q}_{\ell,j}\}_{j=0}^{i-1}\}_{\ell=1}^{N}$, the randomized compression mechanism at agent k depends only on the differential input $\boldsymbol{\psi}_{\ell,i} - \boldsymbol{q}_{\ell,i-1}$, and is independent across the agents.* □

From (11) we notice that small values of the *compression parameter* ω correspond to low distortion, i.e., to finely quantized data, while large values of ω correspond to coarsely quantized data.

3. STEADY-STATE PERFORMANCE

The next theorem characterizes the ACTC performance, in terms of an upper bound on the mean-square-error of each agent k in steady-state, i.e., as $i \to \infty$. Proofs are omitted for space limitations.

Theorem 1 (Steady-State Performance). *Let $w^\star$ be the minimizer of the function $J(w)$ in* (3) *with weights $p_k = \pi_k$. Under Assumptions 1-4, for sufficiently small values of μ and ζ such that the ACTC strategy is mean-square stable,*[2] *the mean-square-error of agent k is upper bounded as follows:*

$$\begin{aligned}
&\limsup_{i\to\infty} \mathbb{E}\|\boldsymbol{w}_{k,i} - w^\star\|^2 \\
&\leq \mu\,\zeta\cdot\Bigg\{\underbrace{\frac{1}{4}\mathrm{Tr}\left(\left(\sum_{k=1}^{N}\pi_k R_{u,k}\right)^{-1}\left(\sum_{k=1}^{N}\pi_k^2 R_{s,k}\right)\right)}_{\text{error without compression}} \\
&\quad + \underbrace{\underbrace{\frac{1}{2\nu}\left[\sum_{k=1}^{N}\pi_k^2\omega_k\Big(\mathrm{Tr}(R_{s,k}) + 2\|b_k\|^2\Big)\right]}_{\text{gradient noise and bias compression loss}} + \underbrace{\frac{\zeta c}{\nu}}_{\substack{\text{network error}\\ \text{compression loss}}}}_{\text{compression loss}}\Bigg\} + O(\mu^{3/2}),
\end{aligned} \tag{12}$$

where c is a constant (i.e., it is independent of μ) that can be computed from the system parameters (including $\{\omega_k\}$), whose expression is omitted for space limitations. ■

Theorem 1 allows us to decompose the steady-state error of the ACTC strategy in two main terms.

— *Error without compression.* This term embodies the estimation error of the distributed strategy when uncompressed information is shared. It has the same shape of the error achieved by the classical ATC diffusion strategy [1, 30].

— *Compression loss.* This term corresponds to the additional estimation error caused by the sharing of compressed information. It can be decomposed in two further components: the *gradient noise and bias compression loss* and the *network error component compression loss*. As regards the former term, observe that in classical quantization systems, the compression error scales with the variance of the quantizer input. In our setting, the role of this variance is played by two sources of variability that affect the quantizer input, namely, the gradient noise and the bias.

The network error component is instead due to the local discrepancies between the agents, i.e., to the difference between the individual agents' iterates and a coordinated, centralized evolution. In the classical ATC strategy, this is a higher-order term w.r.t. μ [1, 29, 30], while from (12) we see that in the presence of data compression it scales as μ and, compared with the other error terms, it is further weighted by the stability parameter ζ.

Inspecting the compression loss in (12), we see that it depends on structured interaction between compression resources, network topology and difficulty of the regression problem, encoded in the following main quantities: the local compression parameters $\{\omega_k\}$, the Perron weights $\{\pi_k\}$, the gradient noise covariances $\{R_{s,k}\}$, and the biases $\{b_k\}$. In the next section we will show how these features can be combined to provide an optimized allocation of the communication resources.

4. ILLUSTRATIVE EXAMPLES

We consider $N = 10$ agents, arranged according to the topology depicted in Fig. 1. Over this topology, we build a left-stochastic combination matrix using the uniform averaging rule [1]. The agents observe regressors with dimensionality $M = 30$ and diagonal covariance matrices $R_{u,k}$. The step-size is $\mu = 10^{-2}$ and the stability parameter is $\zeta = 10^{-1}$. In the forthcoming simulations, all agents share the same minimizer $w^\star = w_k^o$ for all k, and use the randomized quantizer proposed in [14], which belongs to the class defined by Assumption 4. We evaluate the performance of the ACTC strategy by means of the *network* mean-square-error, namely,

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|\boldsymbol{w}_{k,i} - w^\star\|^2. \tag{13}$$

Figure 1 shows that, as the bit-rate used by the agents increases, the performance of the ACTC strategy approaches the reference performance of the uncompressed ATC strategy [24, 25].

The quantitative relationships of the performance bound in Theorem 1 can be exploited to distribute the communication resources in a non-uniform manner, accounting for the peculiarities of distinct agents. Let $x = [x_1, x_2, \ldots, x_N]$, where x_k is the bit-rate assigned to agent k. In view of Theorem 1, we focus on optimization of the upper bound in (12). To facilitate the illustration of the main result, we neglect the network error component term that has a reduced impact for sufficiently small ζ, and we remove the integer constraint on

[2] The conditions on μ and ζ for the mean-square stability of the ACTC strategy are discussed in [24, 25].

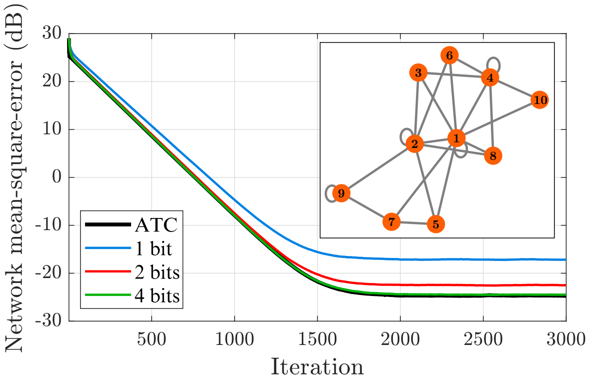

Fig. 1. ACTC network mean-square-error with *uniform* bit allocation, for different values of the bit-rate, for the setting in Sec. 4. Regressors and noises are sampled from Gaussian distributions. The covariance matrices $R_{u,k}$ and the noise variances $\sigma^2_{v,k}$ are set as $\{4I_M, 2I_M, I_M, 2I_M, I_M, I_M, I_M, I_M, I_M, I_M\}$ and $\{1, 0.3, 0.1, 0.3, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$, respectively. All errors are estimated by means of 10^3 Monte Carlo runs.

the bit-rates $\{x_k\}$, arriving at:

$$x^\star = \arg\min_{x\in\mathbb{R}^N} \sum_{k=1}^{N} \pi_k^2 \omega_k \left(\mathrm{Tr}(R_{s,k}) + 2\|b_k\|^2\right)$$
$$\text{s.t.} \quad \sum_{k=1}^{N} x_k = X, \quad x_{\min} \le x_k \le x_{\max} \quad \forall k, \tag{14}$$

where for the compression parameter of the randomized quantizer it is possible to find the relation [14, 18]:

$$\omega_k = \frac{1}{4}\frac{M}{(2^{x_k}-1)^2}. \tag{15}$$

Problem (14) can be solved exactly calling upon the Karush-Kuhn-Tucker (KKT) conditions [32], as shown in Fig. 2. It is also possible to gain insight on the rationale behind a given optimized allocation, by resorting to a common approach in bit-allocation problems [9], as we promptly show. Ignoring the box constraints $x_{\min} \le x_k \le x_{\max}$, using the *high-resolution approximation* $\omega_k \approx (M/4)2^{-2x_k}$, and solving (14) by the method of Lagrange multipliers (or by applying the arithmetic/geometric mean inequality) we get [9]:

$$x_k^\star = \bar{x} + \log_2 \frac{\pi_k}{\pi_{\rm av}} + \frac{1}{2}\log_2 \frac{d_k}{d_{\rm av}}, \tag{16}$$

where

$$\bar{x} = \frac{X}{N}, \quad d_k = \mathrm{Tr}(R_{s,k}) + 2\|b_k\|^2, \tag{17}$$

and $\pi_{\rm av}$, $d_{\rm av}$ are the geometric averages of $\{\pi_k\}$ and $\{d_k\}$.

Rule (16) states that the optimal allocation is a perturbation of the uniform average bit allocation, depending on the Perron weights π_k and the "distortion" term d_k. In particular, the rule prescribes that the assigned bit budget increases with the values of π_k and d_k, which has the following useful interpretation. Agents with many neighbors are very influential in assessing the network mean-square-error performance, as their compressed data is employed by many other agents to compute their minimizer — see steps (9b) and (9c) of the ACTC strategy. This explains why a higher Perron weight π_k favors the assignment of more bits. Likewise, equipping agents characterized by high distortion d_k with many communication resources, reduces the compression errors that propagate across all agents' estimates. Moreover, in the presence of conflicting requirements (e.g., high centrality and low distortion), the allocation rule (16) manages the trade-off. Figure 2 compares the uniform allocation, $x_k = \bar{x}$ for all k, against the optimized allocation obtained with the KKT conditions, when the overall budget is $X = 20$ bits. *Without any additional expense of communication resources*, the knowledge of the quantitative relationship between the network performance and the agents' attributes allows to push the learning performance closer to the reference ATC performance, by approximately 2 dB.

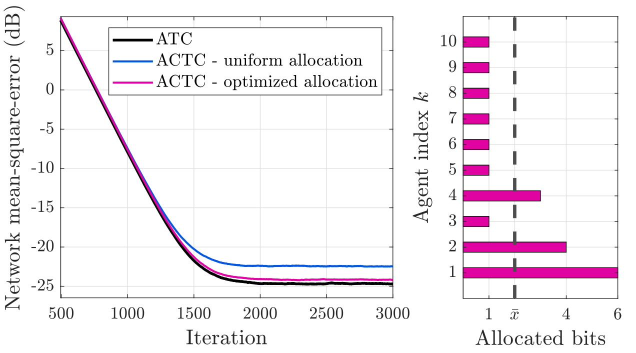

Fig. 2. ACTC network mean-square-error with uniform and optimized bit allocation, for the setting in Fig. 1. The allocation shown in the right plot is the solution (obtained using the KKT conditions) to problem (14) with $X = 20$, $x_{\min} = 1$, and $x_{\max} = 11$, rounded to the closest integer values meeting the constraints. Referring to Fig. 1, we see that agents 1, 2, 4 have wider neighborhoods (see the topology) and noisier data (see $R_{u,k}$ and $\sigma^2_{v,k}$ in the caption). According to (16), these features favor the assignment of more bits.

5. CONCLUSION

We considered the recent ACTC diffusion strategy to solve a distributed regression problem under communication constraints. We obtained an upper bound on the mean-square-error performance of the algorithm. This bound was used to optimize the allocation of the bit budget across the network agents, achieving improved learning performance as compared to a uniform bit allocation. Future extensions include the online estimation of the parameters necessary to optimize the quantizers, a decentralized resource allocation strategy, and the case of globally non-convex risks [33–35].

6. REFERENCES

[1] A. H. Sayed, "Adaptation, Learning, and Optimization over Networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4-5, pp. 311–801, 2014.

[2] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[3] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[4] V. Matta and A. H. Sayed, “Estimation and detection over adaptive networks,” in *Cooperative and Graph Signal Processing*, P. Djuric and C. Richard, Eds. Elsevier, 2018, pp. 69–106.

[5] J. B. Predd, S. B. Kulkarni, and H. V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, Jul. 2006.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[8] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[9] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer, New York, 2001.

[10] Wai-Man Lam and A. R. Reibman, “Design of quantizers for decentralized estimation systems,” *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1602–1605, Nov. 1993.

[11] J. A. Gubner, “Distributed estimation and quantization,” *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1456–1459, Jul. 1993.

[12] M. Longo, T. D. Lookabaugh, and R. M. Gray, “Quantization for decentralized hypothesis testing under communication constraints,” *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 241–255, Mar. 1990.

[13] V. Saligrama, M. Alanyali, and O. Savas, “Distributed detection in sensor networks with packet losses and finite capacity links,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4118–4132, Nov. 2006.

[14] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, “QSGD: communication-efficient SGD via gradient quantization and encoding” in *Proc. NIPS*, Long Beach, CA, USA, Dec. 2017, pp. 1707–1718.

[15] S. U. Stich, J-B. Cordonnier, and M. Jaggi, “Sparsified SGD with memory,” in *Proc. NIPS*, Montréal, Canada, Dec. 2018, pp. 4447–4458.

[16] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” in *Proc. NIPS*, Montréal, Canada, Dec. 2018, pp. 1306–1316.

[17] C.-Y. Lin, V. Kostina, and B. Hassibi, “Differentially quantized gradient descent,” in *Proc. IEEE ISIT*, Melbourne, Victoria, Australia, Jul. 2021, pp. 1200–1205.

[18] Z.-Q. Luo, “Universal decentralized estimation in a bandwidth constrained sensor network,” *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2210–2219, Jun. 2005.

[19] J. Wu, W. Huang, J. Huang, and T. Zhang, “Error compensated quantized SGD and its applications to large-scale distributed optimization,” in *Proc. ICML*, Stockholm, Sweden, Jul. 2018, pp. 5235–5333.

[20] T. Doan, S. T. Maguluri, and J. Romberg, “Convergence rates of distributed gradient methods under random quantization: A stochastic approximation approach,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4469–4484, Oct. 2021.

[21] A. Reisidazeh, A. Mokhtari, H. Hassani, and R. Pedarsani, “An exact quantized decentralized gradient descent algorithm,” *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934–4947, Aug. 2019.

[22] A. Koloskova, S. U. Stich, and M. Jaggi, “Decentralized stochastic optimization and gossip algorithms with compressed communication,” in *Proc. ICML*, Long Beach, CA, USA, Jun. 2019, pp. 3478–3487.

[23] D. Kovalev, A. Koloskova, M. Jaggi, P. Richtárik, and S. U. Stich, “A linearly convergent algorithm for decentralized optimization: sending less bits for free!,” in *Proc. AISTATS*, San Diego, CA, USA, Apr. 2021, pp. 4087–4095.

[24] M. Carpentiero, V. Matta and A. H. Sayed, “Adaptive diffusion with compressed communication,” in *Proc. IEEE ICASSP*, Singapore, May 2022, pp. 5672–5676.

[25] M. Carpentiero, V. Matta and A. H. Sayed, “Distributed adaptive learning under communication constraints,” *under review, available online at arXiv:2112.02129 [cs.LG].*

[26] R. Nassif, S. Vlaski, M. Antonini, M. Carpentiero, V. Matta, and A. H. Sayed, “Finite bit quantization for decentralized learning under subspace constraints,” in *Proc. EUSIPCO*, Belgrade, Serbia, Aug./Sep. 2022, pp. 1851–1855.

[27] R. Nassif, S. Vlaski, M. Carpentiero, V. Matta, M. Antonini, and A. H. Sayed, “Quantization for decentralized learning under subspace constraints,” *under review, available online at arXiv:2209.07821v1 [math.OC].*

[28] X. Zhao, S.-Y. Tu, and A. H. Sayed, “Diffusion adaptation over networks under imperfect information exchange and nonstationary data,” *IEEE Trans. Signal Process.*, vol. 60, no. 7, Apr. 2012, pp. 3460–3475.

[29] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks — part I: Transient analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3487–3517, Jun. 2015.

[30] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks — part II: Performance analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3518–3548, Jun. 2015.

[31] J. Chen and A. H. Sayed, “Distributed Pareto optimization via diffusion strategies,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[33] Y. Lu and C. De Sa, “Moniqua: Modulo quantized communication in decentralized SGD,” in *Proc. ICML*, Jul. 2020, pp. 6415–6425.

[34] T. Vogels, S. P. Karimireddy, and M. Jaggi, “Practical low-rank communication compression in decentralized deep learning,” in *Proc. NIPS*, Dec. 2020, pp. 14171–14181.

[35] H. Tang, X. Lian, C. Yu, T. Zhang, and J. Liu, “DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression,” in *Proc. ICML*, Long Beach, CA, USA, Jun. 2019, pp. 6155–6165.