Proceedings of Machine Learning Research vol 144:1–12, 2021

Linear Regression over Networks with Communication Guarantees

Konstantinos Gatsis KONSTANTINOS.GATSIS@ENG.OX.AC.UK
Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ

Abstract

A key functionality of emerging connected autonomous systems such as smart cities, smart transportation systems, and the industrial Internet-of-Things, is the ability to process and learn from data collected at different physical locations. This is increasingly attracting attention under the terms of distributed learning and federated learning. However, in connected autonomous systems, data transfer takes place over communication networks with often limited resources. This paper examines algorithms for communication-efficient learning for linear regression tasks by exploiting the informativeness of the data. The developed algorithms enable a tradeoff between communication and learning with theoretical performance guarantees and efficient practical implementations.

Keywords: Distributed Learning; Federated Learning; Learning over Networks

1. Introduction

Conventional machine learning approaches require data to be collected at a centralized location to be trained in a centralized manner. However, the emergence of new cyber-physical architectures that are distributed requires rethinking this approach. Examples of distributed cyber-physical architectures include the Industrial Internet-of-Things with sensors/actuators/robots connected to access points collecting data to jointly update system models and application operating conditions – see, for example, Fig. 1; or future transportation systems with connected vehicles collecting and communicating observations from the road; or large-scale sensing infrastructures in future Smart Cities. As a result, in distributed cyber-physical architectures there is a need to enable learning when data are collected by agents across different physical locations.

A concept relevant to address this need is federated learning, introduced by Google (Konečnỳ et al., 2016; Bonawitz et al., 2019), enabling multiple users to jointly solve a machine learning problem over a communication network from data collected from the users. A major challenge in federated learning is that data can be high-dimensional, making their communication costly and inefficient. To alleviate this communication bottleneck, one direction is based on communicating the machine learning model parameters as they are being trained, such as the weights of a Deep Neural Network, instead of the data itself, or communicate the gradients of the objective with respect to the parameters. In deep learning models with high dimensional weights, sparsification and quantization of the weights or the gradients is further introduced to limit the communication cost (Konečnỳ et al., 2016; Aji and Heafield, 2017; Sattler et al., 2019; Lin et al., 2020). Lazy updates are introduced in (Chen et al., 2018a,b), and combinations of non-periodic updates and quantization is explored in (Reisizadeh et al., 2020). Furthermore, when distributed learning is taking place over a wireless network, there is an interest in allocating the available network resources efficiently among the users holding the data (Gündüz et al., 2019; Ahn et al., 2020), such as power (Chen et al., 2020b) or rates (Chang and Tandon, 2020). The problem of scheduling gradient updates over multiple access channels has also received initial consideration, for example comparing time-based approaches

with approaches based on channel conditions (Yang et al., 2019), or including gradient information (Amiri et al., 2020; Chen et al., 2020a).

The present paper hinges on the idea that when model parameters are updated from noisy data, then not all updates are equally informative. Performing updates selectively can be beneficial, and we can evaluate the informativeness of the data by estimating the obtained gain in machine learning performance. Building upon this intuition the proposed algorithms aim for agents to update the machine learning task when their data are most informative, i.e., bring about the most gain. By prioritizing updates with more relevant information, agents can efficiently use communication resources and progress the learning task. This approach builds on recent work by the author, where centralized scheduling of multiple machine learning tasks was explored (Gatsis, 2021), while the present paper addresses the more challenging setup of decentralized communication schemes where agents decide to update independently. The technical methodology borrows ideas from the problem of scheduling control tasks over shared communication networks (Eisen et al., 2019; Gatsis et al., 2015; Ayan et al., 2019; Soleymani et al., 2016). The methodology is also related to event-triggered learning that tries to update only if necessary (Solowjow et al., 2018; Zhao et al., 2020).

The methodology is developed for the task of solving linear regression problems in Section 2 and the communication efficient learning problem is introduced. Section 3 introduces the proposed communication algorithm which prioritizes updates whose data carries the most information, i.e., that would lead to the highest performance gain. The approach is theoretically analyzed and guarantees are provided about both convergence and required communication resources. Importantly, the proposed approach allows to *provably tradeoff learning performance with communication efficiency*. Furthermore, as the method is developed ideally when the data distribution is known, special effort is placed on a practical communication algorithm that uses only the currently available data to estimate how informative the current update will be. Numerical evaluations in Section 4 validate the theoretical results and the improvements compared to approaches in the literature that treat the magnitude of the gradients as a measure of informativeness.

2. Problem Setup

The architecture examined in this paper, shown in Fig. 1, involves a single access-point/server interested in building a data-driven model by solving a machine learning task on data that are collected by multiple agents. The goal is to find a vector of weights (parameters) w of appropriate dimensions to minimize a performance metric (cost) $J(w)$. The aim will be to achieve this with communication efficiency. This is for example the case when an agent should not communicate all the time over a communication network to update the vector of parameters at the access point/server, e.g., due to capacity constraints.

Specifically we consider the machine learning task to be a linear regression problem (Shalev-Shwartz and Ben-David, 2014, Ch. 9). We are interested in finding a vector of weights w that explains the relationship between random variables $(x, y) \in \mathbb{R}^n \times \mathbb{R}$, i.e., $y \approx x^T w$. The random variables (x, y) follow in general a joint distribution denoted by μ. The desired choice for the weights is the one that minimizes the expected square prediction error, i.e.,

$$\min_w J(w) = \frac{1}{2}\mathbb{E}_{(x,y)\sim\mu}(y - x^T w)^2 \tag{1}$$

where the expectation is with respect to the data distribution μ – in the sequel we drop this notation when it is implied that expectation is with respect to this distribution.

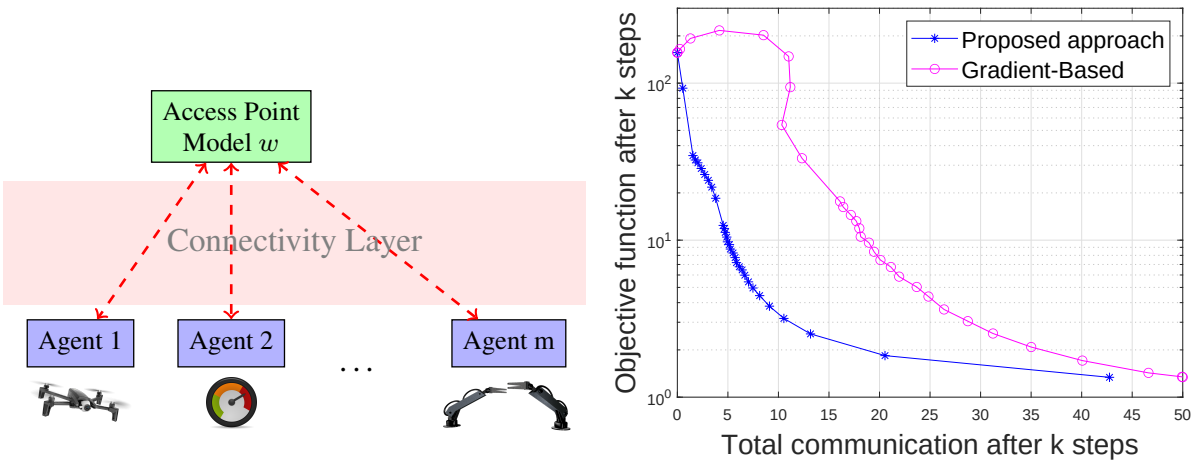

Figure 1: (Left) Architecture for solving machine learning tasks over networks of agents. Agents are collecting data and are communicating with an access point/server. Examples include Industrial IoT systems with sensors/actuators/robots connected to common access points, collecting data to jointly update system models and application operating conditions. (Right) Comparison between our communication efficient learning approach based on estimating the gain in (30) versus the approach in (31) based on the magnitude of the gradients.

The optimal solution w^* is given as the solution to the linear equations

$$\mathbb{E}xx^T w^* - \mathbb{E}xy = 0. \tag{2}$$

Towards finding an optimal set of weights, we would like to employ a gradient descent algorithm. Starting from some initial set of weights w_0 we would like to update the weights according to

$$w_{k+1} = w_k - \epsilon \nabla J(w_k) \tag{3}$$

where $\nabla J(w_k) = \mathbb{E}xx^T w_k - \mathbb{E}xy$, and $\epsilon > 0$ is a small positive stepsize. As will be illustrated later, choosing $\epsilon < 2/\lambda_{\max}(\mathbb{E}xx^T)$ guarantees convergence.

The distribution of the data is not a priori known, and hence as is common in machine leaning, e.g., in empirical risk minimization (Shalev-Shwartz and Ben-David, 2014, Ch. 2), we will attempt to minimize the empirical cost computed as an average over collected data. Specifically we assume that at each iteration k there are N new data points of the form

$$(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}, \qquad i = 1, \ldots, N. \tag{4}$$

We assume each data pair is independent and identically distributed according to a distribution μ.[1] Then we form the empirical cost

$$\hat{J}(w) = \frac{1}{2}\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i^T w)^2 \tag{5}$$

1. This setup arises either when an agent in Figure 1 collects N new independent samples at each iteration, or when it just maintains a large pool of samples and selects randomly N from them at each iteration as frequently done in stochastic gradient descent practice.

With this approximation, we follow a stochastic gradient vector

$$w_{k+1} = w_k - \epsilon g_k \tag{6}$$

computed over the data as

$$g_k = \nabla \hat{J}(w_k) = \frac{1}{N} \sum_{i=1}^{N} \left(x_i x_i^T w_k - x_i y_i \right) \tag{7}$$

After this update the prediction error becomes

$$J(w_{k+1}) = \frac{1}{2} \mathbb{E}(y - x^T w_{k+1})^2 \tag{8}$$

where the expectation is with respect to the distribution μ. We note that since the N data points are random, so is the constructed gradient direction g_k, the updated vector w_{k+1}, as well as the performance metric $J(w_{k+1})$. To evaluate how good is this updated prediction error, we would like to measure on average the quantity

$$\mathbb{E}[J(w_{k+1})|w_k] = \mathbb{E}_{data \sim \mu^N}[J(w_{k+1})|w_k] \tag{9}$$

It is important to note here that the expectation is over the N i.i.d. data that are collected at iteration k and used to construct the stochastic gradient g_k. In the paper, whenever an expectation over iterates w_k is taken, this is an expectation over the data collected until time k.

2.1. Communication-efficient learning problem

Given the above modeling for a machine learning task that needs to be solved, the communication problem is as follows. At each iteration k, the server broadcasts the current weights w_k to all agents. Then each agent i collects N local data points identically distributed (across time and across agents), computes a local stochastic gradient g_k^i from the available local data, and decides whether to transmit this gradient update over the communication network to the receiving server. The server maintains a current vector of weights w_k which will be updated depending on the information received from different agents. For simplicity of exposition the case of two agents is considered, leading to the update rule at the server

$$w_{k+1} = \begin{cases} w_k - \epsilon g_k^1 & \text{if agent 1 transmits} \\ w_k - \epsilon g_k^2 & \text{if agent 2 transmits} \\ w_k - \epsilon/2(g_k^1 + g_k^2) & \text{if both agents transmit} \\ w_k & \text{if no agent transmits} \end{cases} \tag{10}$$

We also denote with $\alpha_k^i \in \{1, 0\}$ the decision for each agent i to transmit or not. At the next iteration $k+1$ a new set of data is collected as in (4) at each agent, a new stochastic gradient direction g_{k+1}^i is computed at each agent, and the process repeats. The aim will be to *avoid sending updates all the time in order to limit the communication burden*.

Remark 1 (Scope of the setup) *The setup (linear regression, two agents) is chosen as a basis for theoretical analysis of convergence and communication utilization jointly, illustrating inefficiencies of approaches in the literature (Remark 3). A larger number of agents is consider in the numerical results in Section 4. Besides, linear regression forms the basis for relevant problems in the control systems and learning community, and extensions of the theoretical results to general convex learning problems and more agents is under investigation.*

3. Proposed communication-efficient learning

The approach is based on the notion of performance gain which can be thought as a measure of how informative are the data collected at each agent at each time step with respect to the machine learning problem. The gain at agent $i = 1, 2$ can be calculated by measuring how much will the objective change if the agent sends the update Whether this gain is negative or positive depends on the random direction of the update. The proposed approach then is to send a gradient update if the gain is large enough. Mathematically we write

$$\alpha_k^i = \begin{cases} 1 & \text{if } J(w_k - \epsilon g_k^i) - J(w_k) \leq -\lambda \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

for some scalar parameter $\lambda > 0$. Intuitively this approach saves up communication resources, because the larger the parameter value λ is, the more infrequent the updates will be. But then the question is what can be said about the progress of learning. We have then the following result.

Theorem 1 (Convergence) *Consider the optimization problem defined in (1) and let w^* be the optimal solution. Consider the update rule in (10). Suppose $g_k^i, i = 1, 2$, are independent random variables with mean equal to $\nabla J(w_k)$ and covariance G at each iteration k. Consider the communication strategy in (11). Then for any iteration N we have that*

$$\mathbb{E}J(w_N) \leq \rho^N J(w_0) + (1 - \rho^N)\left[J(w^*) + \frac{\epsilon^2 Tr(\Sigma_x G)}{1 - \rho}\right] + \lambda \sum_{\ell=0}^{N} \rho^{N-\ell} \frac{\sum_{i=1}^{2} \mathbb{E}(1 - \alpha_\ell^i)}{2} \tag{12}$$

where the expectation is with respect to the data collected until iteration N, and the parameters are $\Sigma_x = \mathbb{E}xx^T/2$ and $\rho = \max_i(1 - \epsilon\lambda_i(\mathbb{E}xx^T))^2$ and the stepsize $\epsilon > 0$ is chosen small enough so that $\rho < 1$.

Proof Note that by the dynamics in (10) we can write

$$\begin{aligned} J(w_{k+1}) = (1 - \alpha_k^1)(1 - \alpha_k^2)J(w_k) + \alpha_k^1(1 - \alpha_k^2)J(w_k - \epsilon g_k^1) \\ + (1 - \alpha_k^1)\alpha_k^2 J(w_k - \epsilon g_k^2) + \alpha_k^1\alpha_k^2 J(w_k - \epsilon/2 g_k^1 - \epsilon/2 g_k^2), \end{aligned} \tag{13}$$

depending on each of the four cases. Then due to the convexity of the problem we have for the last case the bound

$$J(w_k - \epsilon/2 g_k^1 - \epsilon/2 g_k^2) \leq 1/2 J(w_k - \epsilon g_k^1) + 1/2 J(w_k - \epsilon g_k^2). \tag{14}$$

Substituting this bound in (13) and after a rearrangement of terms we get

$$\begin{aligned} J(w_{k+1}) \leq \frac{1}{2}(1 - \alpha_k^2)\Big[(1 - \alpha_k^1)J(w_k) + \alpha_k^1 J(w_k - \epsilon g_k^1)\Big] + \frac{1}{2}\alpha_k^1 J(w_k - \epsilon g_k^1) \\ + \frac{1}{2}(1 - \alpha_k^1)\Big[(1 - \alpha_k^2)J(w_k) + \alpha_k^2 J(w_k - \epsilon g_k^2)\Big] + \frac{1}{2}\alpha_k^2 J(w_k - \epsilon g_k^1) \end{aligned} \tag{15}$$

Then the terms in the brackets can be bounded. Note that due to the choice in (11) the following inequality holds for all times (technically it holds almost surely as all the variables involved are random variables)

$$(1 - \alpha_k^i)J(w_k) + \alpha_k^i J(w_k - \epsilon g_k^i) \leq \lambda + J(w_k - \epsilon g_k^i). \tag{16}$$

This can be easily verified by examining the two cases $\alpha_k^i = 0$ or 1 separately. Substituting this inequality for agents $i = 1, 2$ in (15) we get

$$J(w_{k+1}) \leq \frac{1}{2}(1-\alpha_k^2)\Big[\lambda + J(w_k - \epsilon g_k^1)\Big] + \frac{1}{2}\alpha_k^1 J(w_k - \epsilon g_k^1) + \frac{1}{2}(1-\alpha_k^1)\Big[\lambda + J(w_k - \epsilon g_k^2)\Big] + \frac{1}{2}\alpha_k^2 J(w_k - \epsilon g_k^2) \tag{17}$$

Taking expectation over the stochastic gradients g_k^1 and g_k^2, conditioned on the current iterate w_k, and using the symmetry of the problem with respect to agents $i = 1, 2$ we get that

$$\mathbb{E}[J(w_{k+1}) \,\big|\, w_k] \leq \mathbb{E}[1-\alpha_k^i \,\big|\, w_k]\Big[\lambda + \mathbb{E}[J(w_k - \epsilon g_k^i) \,\big|\, w_k]\Big] + \mathbb{E}[\alpha_k^i J(w_k - \epsilon g_k^i) \,\big|\, w_k] \tag{18}$$

Then we have the following key fact, which *is shown separately in the Appendix,*

$$\mathbb{E}[\alpha_k^i J(w_k - \epsilon g_k^i) \,\big|\, w_k] \leq \mathbb{E}[\alpha_k^i \,\big|\, w_k]\; \mathbb{E}[J(w_k - \epsilon g_k^i) \,\big|\, w_k] \tag{19}$$

Substituting this bound in (18), we get

$$\mathbb{E}[J(w_{k+1}) \,\big|\, w_k] \leq \mathbb{E}[1-\alpha_k^i \,\big|\, w_k]\lambda + \mathbb{E}[J(w_k - \epsilon g_k^i) \,\big|\, w_k] \tag{20}$$

Then given the fact that the function $J(w)$ is quadratic, and the property of the stochastic gradient that the mean is unbiased $\mathbb{E}g_k^i = \nabla_{w_k} J(w_k)$ with a constant variance, we get that[2]

$$\mathbb{E}[J(w_k - \epsilon g_k) \,\big|\, w_k] \leq \rho J(w_k) + \epsilon^2 \mathrm{Tr}(\Sigma_x G) + (1-\rho)J(w^*) \tag{21}$$

Substituting this we get,

$$\mathbb{E}[J(w_{k+1}) \,\big|\, w_k] \leq \mathbb{E}[1-\alpha_k^i \,\big|\, w_k]\lambda + \rho J(w_k) + \epsilon^2 \mathrm{Tr}(\Sigma_x G) + (1-\rho)J(w^*) \tag{22}$$

Taking expectation on both sides with respect to the variable w_k, and iterating over time $k = 1, \ldots, N$, we get the desired result (12). ■

The result verifies that the update rule converges (in a stochastic sense) because $\rho < 1$ as can be confirmed by the appropriate choice of the stepsize $0 < \epsilon < 2/\lambda_{\max}(\mathbb{E}xx^T)$. Essentially the result follows because the function $J(w)$ can be thought as a Lyapunov function for the stochastic dynamics of the update in (10). A direct consequence of the above result is

$$\limsup_{N\to\infty} \mathbb{E}J(w_N) \leq J(w^*) + \frac{\lambda + \epsilon^2 \mathrm{Tr}(\Sigma_x G)}{1-\rho} \tag{23}$$

This means that eventually we get close to the optimal set of weights w^* subject to some overshoots. The latter are due to the stochastic gradient and its covariance G, which can be made small in practice by choosing the step size ϵ to be small – or by choosing a diminishing stepsize which will be analyzed in future work. Moreover, there is a penalty proportional to the parameter λ, introduced to save up on communication cost. It is also possible to choose a diminishing parameter λ to eliminate this effect.

2. We exploit the fact that $(I - \epsilon 2\Sigma_x)'\Sigma_x(I - \epsilon 2\Sigma_x) \preceq \rho\Sigma_x$

Remark 2 *In Theorem 1 we assumed for simplicity that the stochastic gradients have bounded covariances that are constant over time. In reality for the problem above the covariance of the stochastic gradient in (7) will depend on the current iterate w_k, but our choice can be justified in two ways. We can either consider these covariances to be uniformly bounded over time by some constant G. Or alternative if we consider the case close enough to the equilibrium $w_k \approx w^*$, then it follows that the covariances are indeed constant over time. A more detailed investigation will be explored in a follow up work.*

Furthermore, we can establish the following guarantee about the total communication rate of the proposed approach.

Theorem 2 (Communication guarantee) *Consider the same setup as in Theorem 1. The total communication rate satisfies*

$$\limsup_{N\to\infty} \sum_{k=0}^{N} \max\{\alpha_k^1, \alpha_k^2\} \leq \frac{J(w_0) - J(w^*)}{\lambda} \tag{24}$$

almost surely, with respect to the data collected as iterations $N \to \infty$.

Proof Due to the choice in (11) the following inequality holds for all times (technically it holds almost surely as all the variables involved are random variables)

$$\lambda \max\{\alpha_k^1, \alpha_k^2\} + J(w_{k+1}) \leq J(w_k). \tag{25}$$

This can be easily verified by examining the four cases for $\alpha_k^i = 0$ or 1 for $i = 1, 2$. Specifically, when both $\alpha_k^1 = \alpha_k^2 = 1$ we have that

$$\begin{aligned} J(w_{k+1}) &= J(w_k - \epsilon/2 g_k^1 - \epsilon/2 g_k^2) \leq 1/2 J(w_k - \epsilon g_k^1) + 1/2 J(w_k - \epsilon g_k^2) \\ &\leq 1/2(J(w_k) - \lambda) + 1/2(J(w_k) - \lambda). \end{aligned} \tag{26}$$

where the first inequality holds due to convexity and the second inequality holds due to the choice in (11).

Iterating (25) over time $k = 0, \ldots, N$, and summing up, we conclude that

$$\lambda \sum_{k=0}^{N} \max\{\alpha_k^1, \alpha_k^2\} + J(w_{N+1}) \leq J(w_0). \tag{27}$$

Moreover, since any value of the variable w_{N+1} is in general suboptimal, we have that $J(w_{N+1}) \geq J(w^*)$. From which we get the desired result (24). ■

This result counts communication as long as one agent transmits. It guarantees explicitly that increasing λ will decrease the resulting communication in an inversely proportional manner.

3.1. Practical communication scheme

Despite the above guarantee, implementing the proposed communication scheme in (11) would be practically challenging because it requires information that is not known. Specifically it would require knowledge of the data distribution in order to compute the actual performance gain. Since the true distribution is unknown, one approach is to *estimate the performance gain from the data*. In particular, since the objective function is quadratic, we can write the performance gain as

$$J(w_k - \epsilon g_k) - J(w_k) = -\epsilon g_k^T \nabla J(w_k) + \frac{1}{2}\epsilon^2 g_k^T \nabla^2 J(w_k) g_k \tag{28}$$

This is a quadratic function of the stochastic gradient g_k. Then we can approximate the quantities

$$\nabla J(w_k) \approx \frac{1}{N}\sum_{i=1}^{N}\left(x_i x_i^T w_k - x_i y_i\right) = g_k, \qquad \nabla^2 J(w_k) \approx \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T \tag{29}$$

where we note that the stochastic gradient direction g_k appears again. Hence, using the expression for the information gain in (28), we can approximate the gain as[3]

$$J(w_k - \epsilon g_k) - J(w_k) \approx -\epsilon g_k^T \left[I - \epsilon \frac{1}{2}\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T\right] g_k \tag{30}$$

It is crucial to emphasize that *this is no longer a simple quadratic function* of the data but a more complicated function - we note that the data appear both in the stochastic gradients g_k by (7) as well as the matrix in the middle. This approximate value of the gain may take again positive or negative values but it induces an approximation error/bias.

As a result, we can implement the communication decision in (11) with the approximation in (30). In this case we no longer have the performance guarantee in Theorem 1. In numerical evaluations however we see that despite the bias this mechanism performs very well.

Remark 3 (Other approaches in the literature) *A different perspective would be to treat the agents with the largest updates as the most important, and let an agent communicate if the norm of its (stochastic) gradient is large, i.e.,*

$$\alpha_k^i = \begin{cases} 1 & \text{if } \|g_k^i\|^2 \geq \mu \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

for some scalar parameter $\mu > 0$. From our expression on (30) we see that for small stepsizes ϵ the magnitude of the gradient may serve as a proxy for the performance gain. But in numerical comparisons we show that this scheme typically leads to worse performance. This may also be the case when the Hessian of the problem is further from an identity matrix. The idea of scheduling based on gradient magnitudes has been proposed in very recent works in federated learning over wireless channels (Amiri et al., 2020; Chen et al., 2020a), and in the context of sparsification and quantization for high-dimensional gradient updates (Aji and Heafield, 2017; Sattler et al., 2019). Our findings hence point to a novel and more communication- efficient approach for gradient updates. Finally, a different perspective is followed by (Chen et al., 2018a,b); when agents do not update their gradients at the server, the server just keeps a memory of past received gradients and uses them for gradient descent. A difference compared to the present paper is that here there is an explicit communication-learning tradeoff controlled by the parameter λ. A more detailed comparison between that approach and the one in the present paper will be considered in future work.

3. Overall at each agent these require $O(Nn)$ operations hence are scalable.

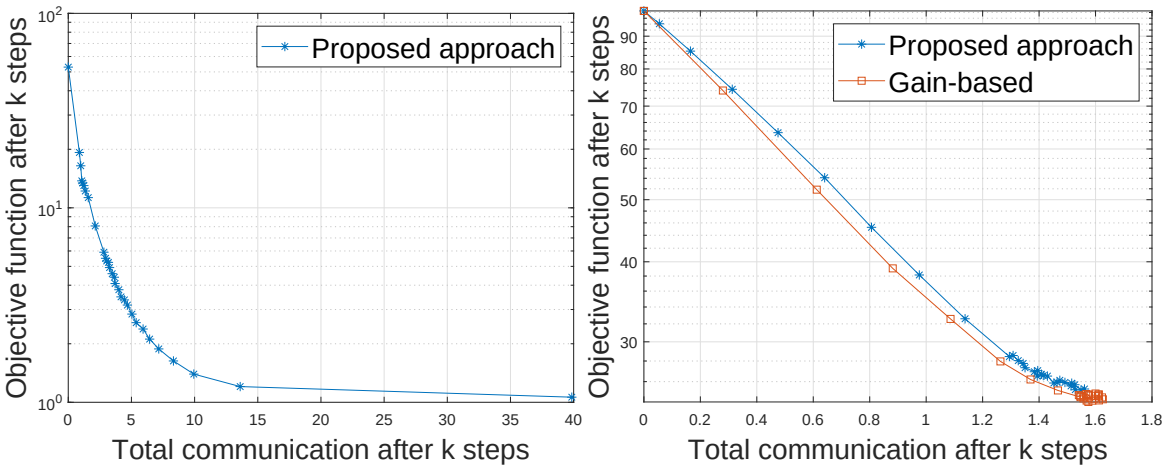

Figure 2: (Left) Evaluation of the tradeoff between communication rate and machine learning performance of the proposed algorithm in (11). (Right) Comparison between our communication approach in (11) requiring the data distribution to compute the gains by (28) versus estimating the gain by (30).

4. Numerical results

In this section we make an additional assumption about the data samples, that x_i are i.i.d. Gaussian random variables, while the points y_i are given as $y_i = x_i^T w^* + \eta_i$ where w^* is the true parameter and η_i are i.i.d. Gaussian measurement noises. These assumptions are not necessary for the theoretical analysis above.

We consider the communication algorithm in (11) with the performance gains estimated as in (30). We consider $m = 2$ agents. We consider a problem with dimensions $n = 2$, with covariances $\mathbb{E}xx^T = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$ (which affects the Hessian of the problem), the initial weights are $w_0 = 0$, and the true weights equal to $w^* = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$. First for stepsize $\epsilon = 0.1$ and $N = 5$ data points available at each iteration and at each agent (cf.(4)), we simulate algorithm (11) for varying values of the parameter λ. In Fig. 2(Left) we plot the observed mean learning performance after the $K = 10$ iterations ($J(w_K)$) versus the total communication rate ($\sum_{k=0}^{K} \sum_{i=1}^{2} \alpha_k^i$). We observe that the proposed communication approach indeed allows us to tradeoff communication rate with machine learning performance.

We would like to investigate how much bias is introduced by our practical scheme that is based on estimating the performance gain at each agent based on the currently available data. Hence we compare (11) when using the performance gains computed by (28) that requires knowledge of the data distributions, with the completely data-based scheme in (30). For the same linear regression setup as before, for $N = 5$ samples per agent, stepsize $\epsilon = 0.2$ and for a single time step, for varying value of the parameter λ the comparison is shown in Fig. 2(Right). In our numerical evaluations, we surprisingly do not observe a significant difference due to the estimation procedure. This was observed across different instances, reinforcing the usefulness of our scheme.

We finally compare our communication scheme (11) based on estimating the performance gain across tasks in (30) with the simple strategy based on the magnitude of the gradients at each agent

in (31). We consider $m = 5$ agents here. We consider a randomly chosen w^* of dimensions $n = 10$ and a covariance matrix $\mathbb{E}xx^T$ diagonal with randomly chosen coefficients. We assume $N = 20$ data points are available at each iteration per agent. We consider $K = 10$ steps in the algorithm. For stepsize $\epsilon = 0.2$ the comparisons are shown in Fig. 1(Right) for varying values of the parameters λ and μ in each of the schemes. We observe that our approach performs significantly better than the gradient-based one. The improvements get typically more significant as the setpsize increases. Our conclusion is that *the magnitude of the gradient is not a reliable measure for the informativeness of the data*. Our approach which is based on the more complex estimate of performance gain provides a more reliable and communication-efficient approach.

5. Concluding remarks

In this paper we examine the problem of solving machine learning tasks over a network. We consider the problem of selecting which updates to communicate to lower the communication rate. To exploit the informativeness of the data we examine the notion of performance gain and we illustrate numerically how this can be approximated from the data without further model knowledge. The approach is contrasted to other related works in the area of communication-efficient learning. Ongoing work explores the use of the approach in more complex networks of learning agents, as well as other machine learning tasks beyond linear regression.

Appendix A. Technical Results

Proof of (19). Let us consider the distribution of the gradient g denoted by $F(g)$. Then we can rewrite (19) as

$$\int \alpha(g)J(w-\epsilon g)dF(g) \leq \int \alpha(g)dF(g) \int J(w-\epsilon g)dF(g) \tag{32}$$

However, by definition of the communication rule (11) we have that $\alpha(g) = 1$ only when $J(w - \epsilon g) \leq J(w) - \lambda$ and zero otherwise. Let us define this set of values $S = \{g \in \mathbb{R}^n : J(w - \epsilon g) \leq J(w) - \lambda\}$. Then (19) is equivalent to

$$\int_S J(w-\epsilon g)dF(g) \leq \int_S dF(g) \left[\int_S J(w-\epsilon g)dF(g) + \int_{S^c} J(w-\epsilon g)dF(g)\right] \tag{33}$$

which is equivalent to

$$\int_{S^c} dF(g) \int_S J(w-\epsilon g)dF(g) \leq \int_S dF(g) \int_{S^c} J(w-\epsilon g)dF(g) \tag{34}$$

We can bound the left hand side because we can bound $J(w - \epsilon g)$ point wise on the set S as

$$\int_{S^c} dF(g) \int_S J(w-\epsilon g)dF(g) \leq \int_{S^c} dF(g) \int_S dF(g)\ [J(w) - \lambda] \tag{35}$$

Further we can bound the right hand side of (34) as

$$\int_S dF(g) \int_{S^c} J(w-\epsilon g)dF(g) \geq \int_S dF(g) \int_{S^c} dF(g)\ [J(w) - \lambda] \tag{36}$$

Combining (35) and (36) we verify (34) and conclude the proof.

References

Jin-Hyun Ahn, Osvaldo Simeone, and Joonhyuk Kang. Cooperative learning via federated distillation over fading channels. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8856–8860. IEEE, 2020.

Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 440–445, 2017.

Mohammad Mohammadi Amiri, Deniz Gündüz, Sanjeev R Kulkarni, and H Vincent Poor. Update aware device scheduling for federated learning at the wireless edge. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2598–2603. IEEE, 2020.

Onur Ayan, Mikhail Vilgelm, Markus Klügel, Sandra Hirche, and Wolfgang Kellerer. Age-of-information vs. value-of-information scheduling for cellular networked control systems. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, pages 109–117, 2019.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Wei-Ting Chang and Ravi Tandon. Communication efficient federated learning over multiple access channels. *arXiv preprint arXiv:2001.08737*, 2020.

Mingzhe Chen, H Vincent Poor, Walid Saad, and Shuguang Cui. Convergence time optimization for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 2020a.

Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 2020b.

Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2018a.

Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Başar. Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239*, 2018b.

Mark Eisen, Mohammad M. Rashid, Konstantinos Gatsis, Dave Cavalcanti, Nageen Himayat, and Alejandro Ribeiro. Control aware radio resource allocation in low latency wireless control systems. *IEEE Internet of Things Journal*, 6(5):7878–7890, 2019.

Konstantinos Gatsis. Adaptive scheduling for machine learning tasks over networks. In *American Control Conference (ACC)*, 2021. To Appear. Preprint available on Arxiv.

Konstantinos Gatsis, Miroslav Pajic, Alejandro Ribeiro, and George J. Pappas. Opportunistic control over shared wireless channels. *IEEE Transactions on Automatic Control*, 60(12):3140–3155, December 2015.

Deniz Gündüz, Paul de Kerret, Nicholas D Sidiropoulos, David Gesbert, Chandra R Murthy, and Mihaela van der Schaar. Machine learning in the air. *IEEE Journal on Selected Areas in Communications*, 37(10):2184–2199, 2019.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Chung-Yi Lin, Victoria Kostina, and Babak Hassibi. Achieving the fundamental convergence-communication tradeoff with differentially quantized gradient descent. *arXiv preprint arXiv:2002.02508*, 2020.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031, 2020.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Sparse binary compression: Towards distributed deep learning with minimal communication. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Touraj Soleymani, Sandra Hirche, and John S Baras. Optimal self-driven sampling for estimation based on value of information. In *2016 13th International Workshop on Discrete Event Systems (WODES)*, pages 183–188. IEEE, 2016.

Friedrich Solowjow, Dominik Baumann, Jochen Garcke, and Sebastian Trimpe. Event-triggered learning for resource-efficient networked control. In *2018 Annual American Control Conference (ACC)*, pages 6506–6512, 2018.

Howard H Yang, Zuozhu Liu, Tony QS Quek, and H Vincent Poor. Scheduling policies for federated learning in wireless networks. *IEEE Transactions on Communications*, 2019.

Zilong Zhao, Sophie Cerf, Bogdan Robu, and Nicolas Marchand. Event-based control for online training of neural networks. *IEEE Control Systems Letters*, 4(3):773–778, 2020.