Hindawi
Complexity
Volume 2020, Article ID 8830359, 10 pages
https://doi.org/10.1155/2020/8830359

Research Article

Online Supervised Learning with Distributed Features over Multiagent System

Xibin An, Bing He, Chen Hu, and Bingqi Liu

High-Tech Institute of Xi'an, Xi'an 710025, China

Correspondence should be addressed to Bing He; 861427055@qq.com

Received 31 August 2020; Revised 27 September 2020; Accepted 7 October 2020; Published 16 November 2020

Academic Editor: Ning Cai

Most current online distributed machine learning algorithms have been studied in a data-parallel architecture among agents in networks. We study online distributed machine learning from a different perspective, where the features about the same samples are observed by multiple agents that wish to collaborate but do not exchange the raw data with each other. We propose a distributed feature online gradient descent algorithm and prove that local solution converges to the global minimizer with a sublinear rate $O(\sqrt{2T})$. Our algorithm does not require exchange of the primal data or even the model parameters between agents. Firstly, we design an auxiliary variable, which implies the information of the global features, and estimate at each agent by dynamic consensus method. Then, local parameters are updated by online gradient descent method based on local data stream. Simulations illustrate the performance of the proposed algorithm.

1. Introduction

With the development of multiagent system, the observed data are being generated at anywhere, anytime, using different devices and technologies [1–3]. There is a lot of interest in extracting knowledge from this massive amount of data and using it to choose a suitable business strategy [4–6], to generate control command [7–9] or to make a decision [10–13]. Many applications are required to process incoming data in online way, e.g., a bank monitors the transactions of its clients to detect frauds [2], wireless sensor networks makes inference [14], and sensor network tracks the uncooperative target [15]. The study of online learning is becoming an important topic of research itself [16–18].

The success of online machine learning often depends on the entire data stream. In some applications, the observed data may be generated on and held by multiple agents [1, 13]. Collecting data to a central site for training incurs extra management and privacy concerns [1]. As a result, some distributed machine learning algorithms have been proposed to train a model by letting each agent perform local model updates and exchange some information between neighbors [19–22]. Most of the existing algorithms fall into the data-parallel computation [1], where each agent has its local data stream with the entire features. However, in network applications, multiple agents are used to monitor an environment, where agents are distributed over space and are used to collect different measurements. For example, the observation is generated by different observed models [8, 9]. It is urgent to develop some applicable algorithm to deal with data streams with distributed features over networks.

In batch learning settings, some algorithms have been proposed for distributed features, such as variance-reduced dynamic diffusion (VRD^2) [12], feature distributed machine learning (FDML) [1], and the ADMM (alternating direction method of multipliers) sharing [23]. VRD^2 and FDML obtain the optimal solution in primal domain, and the local model is trained in a distributed manner based on the local features. The ADMM sharing algorithm formulates distributed feature learning as a distributed primal-dual problem and then obtains the optimal solution by ADMM algorithm. These algorithms in [1, 12, 23] effectively deal with the batch distributed feature learning in a distributed form. However, these algorithms in [1, 12, 23] need to access the entire dataset and cannot be applied in online settings. As the observation is continuously arriving very fast in

networks, it is important to study online feature distributed machine learning.

In this paper, we consider the situation where the features are split across agents in online settings either due to privacy consideration or because they are already physically collected in a distributed manner by means of a networked architecture. We propose a distributed feature online gradient algorithm. Online supervised learning over networks is formulated as a "cost of sum" form. The procedure of the proposed algorithm requires two-scales: one scale is used to update the parameters by gradient descent and a second faster scale for running the consistency step multiple times to track an auxiliary term. The main contributions of this paper are summarized as follows.

(1) We propose a distributed feature online gradient (DFOG) descent algorithm. By exchanging some information between neighbors, local solution can approximate the global solution. Compared with VRD^2 [12], FDML [1], and the sharing ADMM algorithm [23], DFOG is applicable to online supervised learning with distributed features over networks.

(2) We firstly formulate the centralized cost as a "cost of sum" form. By dynamic consensus algorithm, each node can track the sum term, which implies the entire features of the sample at each round time. Then, with the help of online gradient descent algorithm, each node locally updates the parameters based on its data stream.

(3) We prove that the proposed algorithm achieves an $O(\sqrt{2T})$ regret bound. That is, local solution can approach to the global solution, which is the best decision trained based on the entire dataset. The only transmitted message is some parameters' information, and the proposed algorithm does not require the data of the total number times and does not exchange the raw data between neighbors.

The rest of this paper is organized as follows: the problem formulation is discussed in Section 2. In Section 3, we focus on our online optimization algorithm with distributed features over multiagent system, followed by the theoretical results in Section 4. In Section 5, simulations illustrate the effectiveness of our algorithm. Finally, we conclude the paper in Section 6.

Notation and terminology: let x be the feature space and y be the corresponding label. We denote the (i, j)th element of a matrix A by $a_{i,j}$. For $t \in \mathbb{N}^+$, the set $\{1, 2, \ldots, T\}$ is denoted by $[T]$. For a convex function f, its gradient at a point ω is denoted as $\nabla_\omega f(\omega)$. We denote N as the number of agents in the network. Let $\mathbb{R}^d$ be the d-dimensional vector space and $\|\omega\|_2^2$ is the Euclidean norm of a vector $\omega \in \mathbb{R}^d$.

2. Problem Formulation

We consider a multiagent system with N agents. The communication between agents is described by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [24], consisting of a set of nodes $\mathcal{V} = \{1, 2, \ldots, N\}$, a set of edges $\mathcal{E}$, and an adjacent matrix A [19]. For each agent $i \in \mathcal{V}$, we denote $\mathcal{E}_i = \{j \mid (j, i) \in \mathcal{E}\}$ as a set of neighbors of agent i (including agent i itself).

Assumption 1. The graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and the adjacent weighted matrix A satisfy the following [25]:

(i) A is a doubly stochastic matrix with positive diagonal, that is, $a_{ii} > 0$, $\sum_{j=1}^N a_{j,i} = 1$, and $\sum_{j=1}^N a_{i,j} = 1$;

(ii) There exists a scalar $\zeta > 0$ such that $a_{j,i} > \zeta$ if $(j, i) \in \mathcal{E}$;

(iii) There exists an integer $B \geq 1$ such that the graph $(\mathcal{V}, \mathcal{E}_{i(B+1)} \cup \cdots \cup \mathcal{E}_{(j+1)B})$ is strongly connected.

In this work, we focus on a binary online supervised learning with distributed features. The features are distributed over a collection of K agents, as illustrated in Figure 1.

At each time $t = 1, 2, \ldots, T$, network receives a labeled sample (x_t, y_t). For all the time T, we consider an empirical risk as follows:

$$L(\omega) = \frac{1}{T} \sum_{t=1}^{T} f\left(\omega^T x_t, y_t\right) + r(\omega), \tag{1}$$

where the parameters are denoted as $\omega \in \mathbb{R}^{d \times 1}$, d is the dimension of the features, and $y_t \in \{-1, +1\}$ is the corresponding scalar label of x_t at time t. Moreover, the cost $f(\omega)$ is convex and differentiable. In most problem of interest, the cost function is dependent on the inner product $\omega^T x$, such as the linear SVM cost $f = \max(0, 1 - y_t(w^T x_t))$ and the logistic regression cost $f = \log(1 + \exp(-y_t(\omega^T x_t)))$. The factor $r(\omega)$ represents the regularization term. Since the features of x_t are distributed across agents, we set ω and x_t to be column vector and formulate ω and x_t into N subvectors denoted by ω_i and $x_{t,i}$, respectively, that is,

$$x_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,N} \end{bmatrix}, \qquad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix}. \tag{2}$$

Each subfeature $x_{t,i}$ vector and subvector ω_i is located at agent i. Then, cost function (1) can be rewritten as

$$L = \frac{1}{T} \sum_{t=1}^{T} f\left(\sum_{i=1}^{N} \omega_i^T x_{t,i}; y_t\right) + \sum_{i=1}^{N} r\left(\omega_i\right), \tag{3}$$

where the regularization term is assumed to satisfy an additive form as

$$r(\omega) = \sum_{i=1}^{N} r\left(\omega_i\right). \tag{4}$$

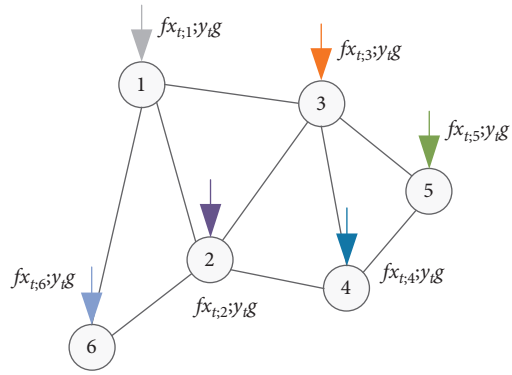

Figure 1: Distributing the features across agents.

This property holds for many popular regularization choices, such as l_2, l_1, and KL-divergence. Problem of this type has been studied before in the literature by using distributed optimization methods in [20, 21]. One common way is to formulate problem (3) into a constrained problem, that is,

$$L(\omega) = \min_{\omega} \frac{1}{T}\sum_{t=1}^{T} f(z_t; y_t) + \sum_{i=1}^{N} r(\omega_i) \text{s.t. } z_t = \sum_{i=1}^{N} \omega_i^T x_{t,i},$$
$$t = 1, 2, \ldots, T. \tag{5}$$

For all the time T, problem (5) is a classical "cost of sum" form [20]. An effective way is to design the Lagrangian function by introducing the dual variable γ [23], namely,

$$L(\gamma, z, \omega) = \frac{1}{T}\sum_{t=1}^{T} f(z_t; y_t) + \frac{1}{T}\sum_{t=1}^{T} \gamma_t z_t - \frac{1}{T}\sum_{t=1}^{T}\gamma_t \sum_{i=1}^{N} \omega_i^T x_{i,t} + \sum_{i=1}^{N} r(\omega_i). \tag{6}$$

Problem (6) can be solved in a number of distributed primal-dual methods, such as alternating direction method of multipliers (ADMM) [4, 22, 26] and primal-dual methods [27–29]. These techniques have good convergence properties but suffer from high computational costs and two-time scale communications.

The other way is studied in primal domain [12]. The algorithm in [12] requires a two-time scale operation: a faster time-scale for the consensus iterations and a slower time-scale for the data sampling and the gradient computing. First, we use a consensus strategy to obtain the sum term $\sum_{i=1}^{N} \omega_i^T x_{t,i}$, namely,

$$\widehat{z}_{n_k,i} = \sum_{j\in\mathcal{G}_i} a_{ij} N \omega_{k,j}^T x_{n_k,j}, \tag{7}$$

where n_k denotes the index of the sample selected uniformly at random from $\{1, 2, \ldots, T\}$. After sufficient iterations, it is well-known that $\widehat{z}_{n_k,i} \longrightarrow (1/N)\sum_{i=1}^{N} z_{n_k,i}$. Then, the stochastic-gradient step is used to update the parameters ω, where the gradient is evaluated by the gradient vector of the cost evaluated at some random data (x_{n_k}, y_{n_k}).

In online settings, since the data (x_t, y_t) is observed one by one, we cannot access to the total dataset $\{(x_t, y_t)\}_{t=1}^{T}$. These algorithms in [1, 12, 23] cannot be applied for data stream with distributed feature over networks. For each time $t = 1, 2, \ldots, T$, the multiagent system is endowed with a sequence of cost function $\{L_t\}_{t=1}^{T}$, and the goal is to minimize the sum of the cost function. Specifically, we want to minimize the difference between the total cost multiagent system has incurred and that of the best fixed decision in hindsight, which is called regret, and its definition is given as follows:

$$\text{Reg}^T = \sum_{t=1}^{T} L_t(\omega_t) - \sum_{t=1}^{T} L_t(\omega^*), \tag{8}$$

where ω^* is the best decision of problem (1), that is,

$$\omega^* = \arg\min_{\omega} \sum_{t=1}^{T} L_t(\omega). \tag{9}$$

Moreover, we consider the time-varying cost function L_t as

$$L_t(\omega_t) = Q\left(\sum_{i=1}^{N} \omega_{t,i}^T x_{t,i}; y_t\right) + \sum_{i=1}^{N} r\left(\omega_{t,i}\right). \tag{10}$$

Generally speaking, the cost $Q(\sum_{i=1}^{N} \omega_i^T x_{t,i}; y_t)$ satisfies Assumption 2.

Assumption 2. The loss function $Q(\cdot)$ is convex and differentiable, and the gradient $\nabla_\omega Q(\omega)$ is uniform boundedness, that is, $\|\nabla_\omega Q(\omega)\| \le C$ for some scalar $C > 0$.

Regret is the standard measure of the performance of online optimization algorithm [19]. An algorithm attains good performance if the regret is sublinear as a function of the total time T.

Remark 1. In the multiagent system, since the entries of the feature x_t are distributed over N agents, each agent just observes its own data stream. We face the following two challenges in solving problem (8):

(1) Distributed challenge: each agent only receives local data stream $(x_{t,i}, y_t)$ and does not access to the entire features (x_t, y_t). Under the condition that we do not exchange the raw data between neighbors, each agent needs to obtain some information on the entire features.

(2) Online challenge: at any time t_1, we only have observation for $t \le t_1$ and do not know L_t for $t_1 \le t \le T$. It is difficult to store all the observations due to the high-dimensional and high-velocity data stream. We need to update the parameters based on the current sample and the previous parameters and pursue a solution approximating to the global solution ω^*,

which is the best decision based on all the data $\{(x_t, y_t)\}_{t=1}^{t}$ as a prior in offline settings.

3. Distributed Feature Online Gradient Descent Algorithm

In this section, we first analyse a dynamic average consensus method for approximating the sum of $\omega_i^T x_{t,i}$ at agent i and propose an online convex optimization to update the parameters ω. The detailed framework is summarised in Figure 2.

Now, we consider the problem of minimizing (5) by means of an online convex optimization. Let $z_t = \sum_{i=1}^{N} \omega_{t,i}^T x_{t,i}$ denote the inner product that is available at time $t \in [T]$. The cost function L_t can be described as

$$L_t(\omega_t) = Q(z_t; y_t) + \sum_{i=1}^{N} r(\omega_{t,i}). \tag{11}$$

If each agent i can obtain the auxiliary variable z_t at any time t, the parameters $\omega_{t,i}$ can be obtained by minimizing the local cost $L_{t,i}$, which is defined as

$$L_{t,i} = Q(z_t; y_t) + r(\omega_{t,i}). \tag{12}$$

However, the computation of z_t needs to access to all the subfeatures $x_{t,i}$ and the subvectors $\omega_{t,i}$ over N agents. We denote the average of the local inner products as

$$\overline{z}_t = \frac{1}{N} \sum_{i=1}^{N} \omega_{t,i}^T x_{t,i}. \tag{13}$$

Motivated by works in [30–34], $\overline{z}_t$ can be approximated by a diffusion-based algorithm. Since the desired variable z_t is proportional to the average value $\overline{z}_t$, $z_t = N\overline{z}_t$, the consensus strategy can be used to approximate z_t. Specifically, for the total number of iterations M, each agent would repeat the following steps M times:

$$\widehat{z}_{t,i}^{m+1} = \sum_{j \in \mathrm{E}_i} a_{ij} \widehat{z}_{t,i}^{m}, \quad m = 0, 1, \ldots M-1, \tag{14}$$

where $\widehat{z}_{t,i}^{0} = N\omega_{t,i}^T x_{t,i}$. After each agent obtains the estimator of z_t denoted as $\widehat{z}_{t,i}$, problem (12) is converted into a differentiable dynamic problem. For online convex optimization problem, online gradient descent and its variants have been achieving optimal dynamic regret in many applications [35]. Recalling that ω_t and x_t are partitioned into N blocks, the gradient step can be performed in parallel over N agents. Specifically,

$$\omega_{t,i} = \omega_{t-1,i} - \mu_t \nabla_z Q(\widehat{z}_{t,i}; y_t) x_{t,i} - \mu_t \nabla_\omega r(\omega_{t,i}), \tag{15}$$

where the step-size μ_t should satisfy $\mu_t > 0$, $\sum_{t=1}^{\infty} \mu_t = \infty$, and $\sum_{t=1}^{\infty} \mu_t^2 < \infty$.

The full algorithm is summarized in Algorithm 1.

Remark 2. Compared with FDML [1], VRD^2 [12], and the ADMM sharing algorithm [23], DFOG is applicable for data stream with distributed features over multiagent system. At each round time, agents observe the same sample from different features. Each agent can obtain an auxiliary term, which implies the information on the entire features. Then, each agent locally runs a gradient descent step to update its local parameters. The procedure of Algorithm 1 is designed to update the parameters $\omega_{t,i}$ locally.

4. Algorithm Analysis

4.1. Convergency Analysis. In this section, we analyse the convergence of the proposed algorithm. We first show that the distance between $\widehat{z}_{t,i}$ and z_t is upper bounded by the difference between P^M and $1/N$, which is shown in Lemma 1 and proved in [25].

Lemma 1. *Let Assumption 1 holds, for all agents i, j; we have*

$$\left| [P^M]_{ij} - \frac{1}{N} \right| \le \left(1 - \frac{\zeta}{4N^2} \right)^{(M/B)-2}, \tag{16}$$

where N is total number of agents and M is the number of consensus steps in(14).

Then, we show that the regret of online gradient descent (OGD) is upper bounded by the cumulative difference between the loss of ω_t and ω_{t+1}, which is present in Lemma 2 and proved in [18].

Lemma 2. *Let $\{\omega_{t,i}\}_{t=1}^{T}$ denotes the sequence of parameters produced by OGD. Then, for any u, we have*

$$\mathrm{Reg}_i^T = \sum_{t=1}^{T} \left(L_{t,i}(\omega_{t,i}) - L_{t,i}(u) \right) \le r(u) - r(\omega_{1,i}) + \sum_{t=1}^{T} \left(Q(\omega_{t,i}) - Q(\omega_{t+1,i}) \right). \tag{17}$$

Because the features are distributed across agents, Reg_i^T mainly illustrates the difference between local parameters ω_i and the corresponding parameters ω_i^* in global solution. Based on the above lemma, we derive a regret bound of ω_i for DFOG with the regularization term $r(\omega_i) = (1/2)\mu\|\omega_i\|_2^2$.

Theorem 1. *Let Assumptions 1 and 2 hold, and consider running DFOG on a sequence of convex function, $Q(\omega_{t,i})$ for all t, with the regularization term $r(\omega_i) = (1/2)\mu\|\omega_i\|_2^2$. Let $\{\omega_{t,i}\}_{t=1}^{T}$ be the sequence of vectors produced by DFOG. If $\|u\| \le U$ and $\mu = (U/C)\sqrt{2T}$, the regret of ω_i satisfies*

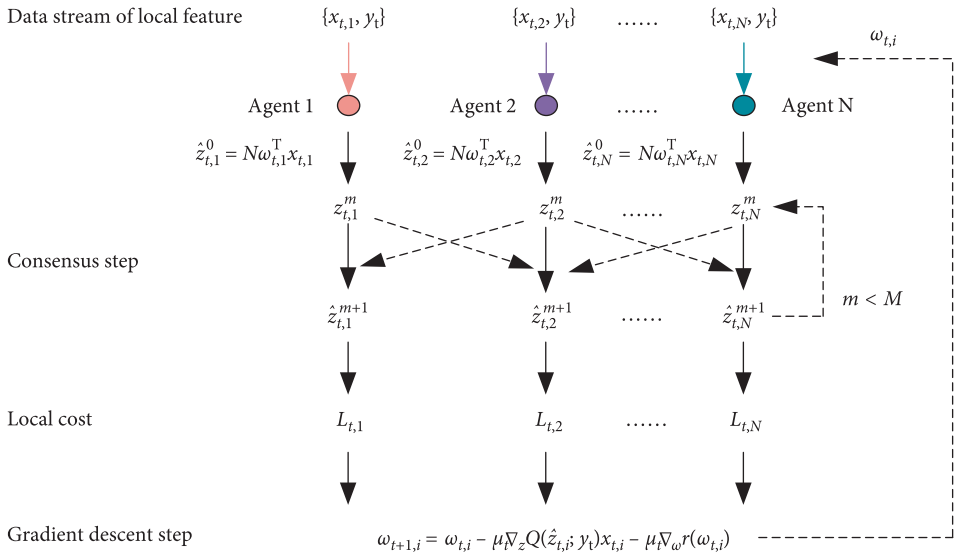

Figure 2: The framework of the proposed algorithm.

(1) Initialization: set $\omega_{0,i} = 0$.
(2) Repeat for $i = 1, 2, \ldots, N$:
(3) $\widehat{z}_{0,i} = N\omega_{0,i}^T x_{t,i}$
(4) For $m = 0, 1, 2, \ldots, M-1$
(5) $\widehat{z}_{t,i}^{m+1} = \sum_{j \in \mathscr{E}_i} a_{ij} \widehat{z}_{t,i}^m$
(6) End
(7) $\omega_{t,i} = \omega_{t-1,i} - \mu_t \nabla_z Q(\widehat{z}_{t,i}; y_t) x_{t,i} - \mu_t \nabla_\omega r(\omega_{t,i})$
(8) End

Algorithm 1: Distributed feature online gradient (DFOG) descent for agent i.

$$\mathrm{Reg}_i^T \leq \mathrm{CU}\sqrt{2T} + C_2 U\sqrt{2T} + U^2\sqrt{2T}, \tag{18}$$

where $C_2 = (C/2)(1 - (\zeta/4N^2))^{(M/B)-2}\|z_t\|_*\|x\|_*$. *The proof is presented in Appendix.*

Remark 3. This theorem indicates that the convergence rate of DFOG depends on the network topology through B and the number of consensus steps M. The larger the M is or the smaller the B is, the faster the convergence speed is. The theorem presents that the proposed algorithm converges to the global solution with sublinear rate. When the number of data samples increases, the difference between $\omega_{t,i}$ with ω_i^* will become closer.

4.2. Complexity Analysis

4.2.1. Time Complexity. There are two primary operations associated with learning for DFOG: (1) estimating the inner product $\widehat{z}_t$ for each sample at time t and (2) updating the parameters at gradient descent step. At any time t, each estimator $\widehat{z}_t$ computation requires $O(M)$ arithmetic operations. There is one gradient descent step to update the parameters, which requires $O(1)$ arithmetic operations. As for each time, each node will require $O(M)$ arithmetic operations. Hence, single node requires $O(\mathrm{TM})$ arithmetic operations for DFOG.

4.2.2. Space Complexity. At any time t, DFOG needs to store the parameters $\widehat{z}_t$ and ω_t, which are updated and time-varying. Hence, space complexity for DFOG is $O(1)$.

4.2.3. Communication Complexity. We denote the average degree of the communication graph as k. At each consensus step, each node requires to exchange $\widehat{z}_t$ (float type, 4 bytes) with its neighbors. Since the network topology is an undirected graph, it requires 8 kM bytes at any time t. Hence, DFOG requires communication traffic of DFOG is 8 kMT bytes for all the time T.

5. Simulation

In this section, we test our algorithm by minimizing norm regularized logistic regression on two public datasets, a9a and bank from UCI. Here, a multiagent system with 6 agents is considered, and the network is generated by the random geometric graph model. a9a dataset consists of 32561

Table 1: Parameter settings.

Parameter	Value
λ	0.1
N	5
M	30
B	10

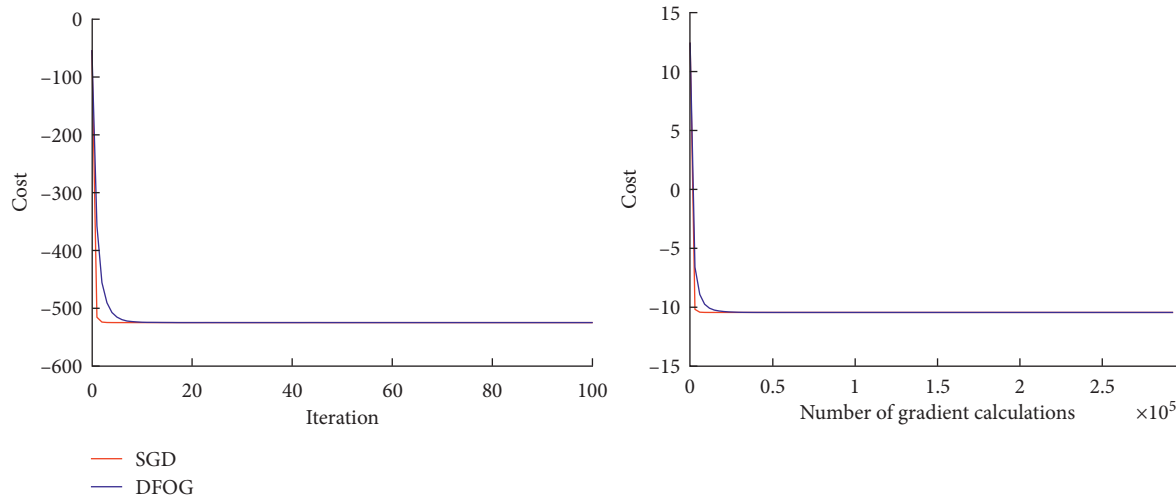

Figure 3: The evolution of cost for a9a dataset.

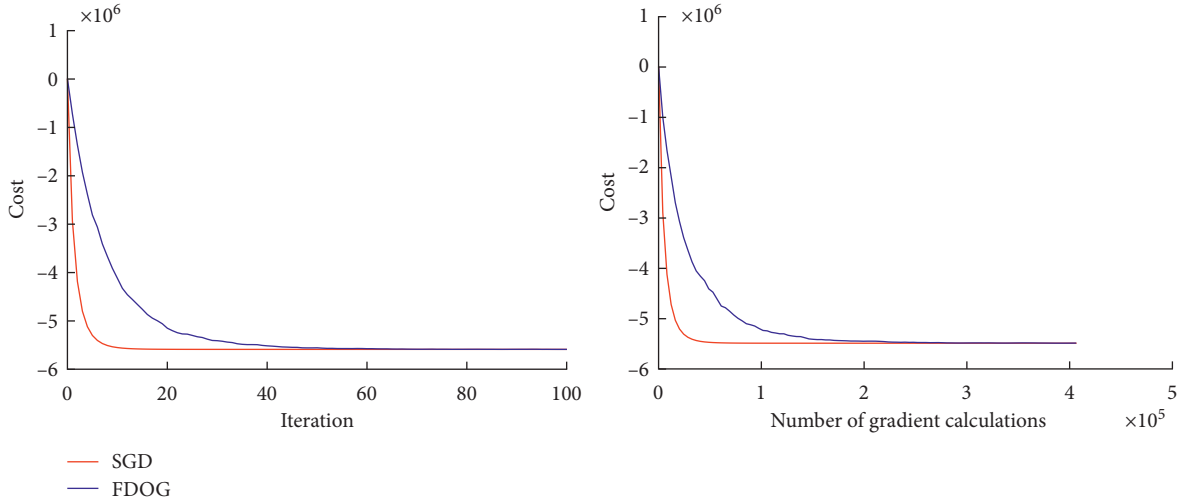

Figure 4: The evolution of cost for bank dataset.

training samples, 16281 testing samples, and 123 features. We separate the features into 6 parts, and each node obtains one part with 21, 21, 21, 20, 20, and 20 features as the local data, respectively. On the other hand, the bank dataset contains 4068 training samples, 453 testing samples, and 17 features. Similarly, we divide the features into 6 parts, each agent gets one part with 3, 3, 3, 3, 3, and 2 features as the local data, respectively. The loss function we use is

$$L(\omega) = \frac{1}{T}\sum_{t=1}^{T}\log\left(1+\exp\left(-y_t\sum_{i=1}^{N}\omega_i^T x_{t,i}\right)\right) + \lambda\sum_{i=1}^{N}\|\omega_i\|^2. \tag{19}$$

Generally, λ is a positive constant and $\lambda > 0$. The simulations are implemented in MATLAB to verify the proposed algorithm. Specifically, we use two optimization libraries, SGDLibrary [36] and DADAM [37], to minimize

Table 2: Testing error and the relative error of parameters.

Dataset	Testing error of SGD	Testing error of DFOG	$\sum_{i=1}^{N} \|\omega_i - \omega_i^*\| / \|\omega^*\|$
9a	0.7581	0.7581	0.0026
Bank	0.8455	0.8455	0.0064

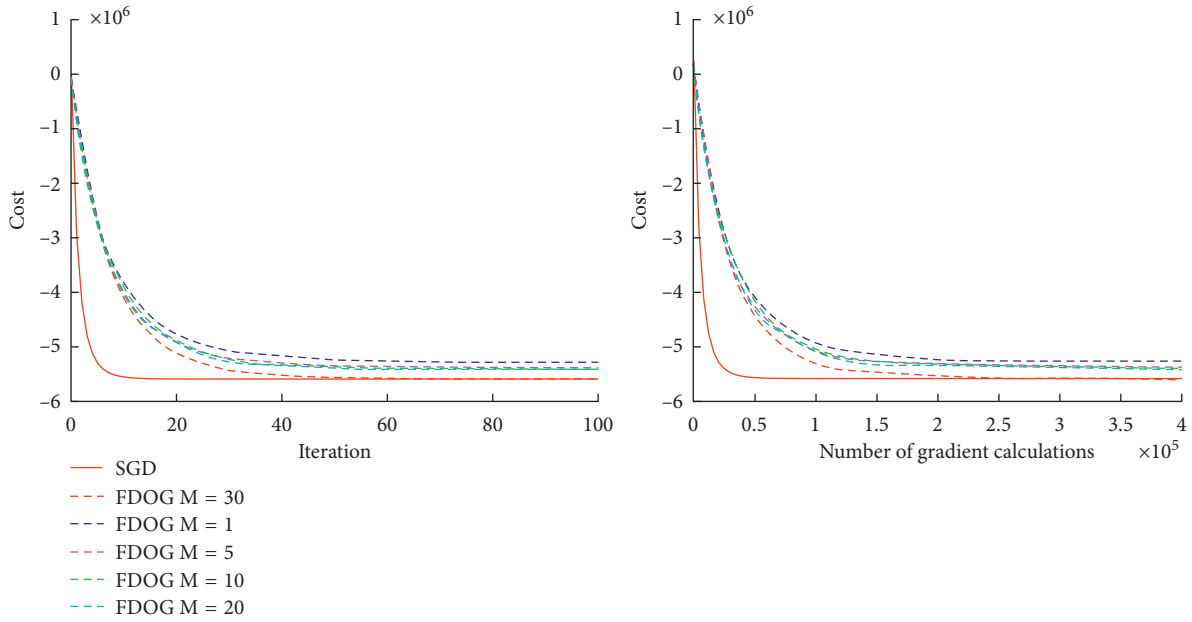

Figure 5: The convergence behavior with different M.

(19). In our simulation, the parameters are set as summarised in Table 1.

We adopt dynamic consensus method to obtain z_t in (14) and use online gradient descent algorithm to update the parameter ω_i locally in (15). In our simulations, we get the global model trained in a centralized manner if all the features were collected centrally by stochastic-gradient descent (SGD) algorithm. Next, we compare our algorithm against SGD algorithm proposed in [38] and keep track of the loss for different datasets and parameters settings. Figures 3 and 4 present the evolution of the cost during the training procedure for a9a and bank datasets, respectively. In addition, to make a fair comparison, we analyse the convergence curve based on the count of gradient calculated. Table 2 shows the testing error for different datasets and the error of parameters for DFOG and SGD. The results show that DFOG can converge to the centralized solution of SGD, while keeping local feature sets to the corresponding agent. That is, DFOG can deal with the online supervised learning problem caused by distributed features over networks.

We next show how the performance depends on different M. Note that when M is larger, we need to do more communication on the consistency step (14). Figure 5 shows the evolution of the cost with different M. It can be found that the larger the M we set, the faster the DFOG approaches to the centralized SGD algorithm.

6. Conclusions

In this paper, we considered an online supervised learning problem where the features are split across agents in online settings. We proposed an online supervised learning algorithm with distributed features over multiagent system. We first formulated the centralized cost as a "cost of sum" form. By dynamic consensus algorithm, each agent could effectively estimate the sum term, which is calculated based on the entire features at each round time. Then, with the help of online gradient descent algorithm, each agent locally updates the parameters. The algorithm designed does not require the data of the total number times and does not communicate the raw data between neighbors. We proved that local solution converges to the centralize minimizer, which is the best decision trained based on the entire dataset, and the proposed algorithm achieves an $\mathcal{O}(\sqrt{2T})$ regret bound. Simulations with real dataset verify the conclusion.

Distributed machining learning algorithms are worth of further studies due to their promising future, including distributed online boosting, distributed decision tree [39], the use of Big data-aided learning [40], and distributed learning over time-varying communication topology in networks.

Appendix

Proof of Theorem 1: let Assumption 2 holds, for each time t, then we have

$$Q\left(\omega_{t,i}\right) - Q\left(u\right) \leq < \omega_{t,i} - u, \quad \nabla_{\omega_{t,i}} Q >, \tag{A.1}$$

where $\langle \omega_i - u, \nabla_{\omega_{t,i}} Q\rangle$ is the inner product between vectors $\omega_i - u$ and $\nabla_{\omega_{t,i}} Q$. Moreover, we denote $\|\omega\| = \sqrt{\langle \omega, \omega\rangle}$.

Using Lemma 2, we obtain

$$\mathrm{Reg}_i^T \leq r\left(u\right) - r\left(\omega_1\right) + \sum_{t=1}^{T}\left(Q\left(\omega_{t,i}\right) - Q\left(\omega_{t+1,i}\right)\right) \leq \frac{1}{2\mu}\|u\|_2^2 + \sum_{t=1}^{T}\langle \omega_{t,i} - \omega_{t+1,i}, \nabla_{\omega_{t,i}} Q\rangle. \tag{A.2}$$

From equation (15), we have

$$\sum_{t=1}^{T}\langle \omega_{t,i} - \omega_{t+1,i}, \nabla_{\omega_{t,i}} Q\rangle = \sum_{t=1}^{T}\left(\nabla_{\widehat{z}_{t,i}} Q\left(\omega_{t,i}\right) x_{t,i} + \nabla_{\omega_{t,i}} r\left(\omega_{t,i}\right)\right)\nabla_{\omega_{t,i}} Q, \tag{A.3}$$

where $\widehat{z}_{t,i} = \sum_{i=1}^{N}[P^M]_{ij}\omega_{t,i}x_{t,i}$.

We derive the gradient of cost (12) as follows:

$$\nabla_{\omega_{t,i}} Q = \nabla_{z_t} Q\left(\omega_{t,i}\right) x_{t,i}. \tag{A.4}$$

Substituting (A.4) into (A.3),

$$\begin{aligned}\sum_{t=1}^{T}\langle \omega_{t,i} - \omega_{t+1,i}, \nabla_{\omega_{t,i}} Q\rangle &= \sum_{t=1}^{T}\left(\nabla_{\widehat{z}_{t,i}} Q \cdot x_{t,i} + \nabla_{\omega_{t,i}} r\right)\nabla_{\omega_{t,i}} Q \\ &= \sum_{t=1}^{T}\left\|\nabla_{\omega_{t,i}} Q\right\|_2^2 + \sum_{t=1}^{T}\left(\nabla_{\widehat{z}_{t,i}} Q - \nabla_{\overline{z}_{t,i}} Q\right) x_{t,i}\nabla_{\omega_{t,i}} Q + \sum_{t=1}^{T}\nabla_{\omega_{t,i}} r\nabla_{\omega_{t,i}} Q \\ &\leq TC^2 + \sum_{t=1}^{T} C\left\|\left(\nabla_{\widehat{z}_{t,i}} Q\left(\omega_{t,i}\right) - \nabla_{\overline{z}_{t,i}} Q\left(\omega_{t,i}\right)\right)\right\|\left\|x_{t,i}\right\| + 2\mathrm{TCU}.\end{aligned} \tag{A.5}$$

Let Assumption 2 holds such that $\|(\nabla_{\widehat{z}_{t,i}} Q(\omega_{t,i}) - \nabla_{\overline{z}_{t,i}} Q(\omega_{t,i}))\| \leq C\|\widehat{z}_{t,i} - \overline{z}_{t,i}\|$. Using Lemma 1, we have

$$\left\|\left(\nabla_{\widehat{z}_{t,i}} Q\left(\omega_{t,i}\right) - \nabla_{\overline{z}_{t,i}} Q\left(\omega_{t,i}\right)\right)\right\| \cdot \left\|x_{t,i}\right\| \leq C\left\|\widehat{z}_{t,i} - \overline{z}_{t,i}\right\| \leq C\left(1 - \frac{\zeta}{4N^2}\right)^{(M/B)-2}\left\|\sum_{i=1}^{N}\omega_{t,i}x_{t,i}\right\|. \tag{A.6}$$

Denoting $\|x\|_* = \max\|x_{t,i}\|$ and $\|z_t\|_* = \max\|\sum_{i=1}^{N}\omega_{t,i}x_{t,i}\|$ for $t = 1, \ldots, T$, we derive

$$\mathrm{Reg}_i^T \leq \frac{1}{2\mu}\|u\|_2^2 + \mu\mathrm{TC}^2 + \mu\mathrm{TC}\left(1 - \frac{\zeta}{4N^2}\right)^{(M/B)-2}\left\|z_t\right\|_*\|x\|_* + 2\mu\mathrm{CTU}. \tag{A.7}$$

If $\|u\| \leq U$ and $\mu_t = U/C\sqrt{2T}$, then

$$\mathrm{Reg}_i^T \leq \mathrm{CU}\sqrt{2T} + C_2U\sqrt{2T} + U^2\sqrt{2T}, \tag{A.8}$$

where $C_2 = (1/2)(1 - (\zeta/4N^2))^{(M/B)-2}\|z_t\|_*\|x\|_*$. Theorem 1 has been proved.

Data Availability

a9a dataset has been derived from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html. Bank dataset has been derived from https://archive.ics.uci.edu/ml/datasets/Bank+Marketing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML: a collaborative machine learning framework for distributed features," in *Proceedings of the Knowledge Discovery and Data Mining*, pp. 2232–2240, Anchorage, AK, USA, August 2019.

[2] A. T. Vu, G. D. F. Morales, J. Gama et al., "Distributed adaptive model rules for mining big data streams," in *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)*, IEEE, Washington DC, USA, October 2014.

[3] N. Cai, C. Diao, and M. J. Khan, "A novel clustering method based on quasi-consensus motions of dynamical multiagent systems," *Complexity*, vol. 2017, Article ID 4978613, , 2017.

[4] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2015.

[5] J. Xi, C. Wang, X. Yang, and B. Yang, "Limited-budget output consensus for descriptor multiagent systems with energy constraints," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, p. 4585, 2020.

[6] J. Xi, L. Wang, J. Zheng, and X. Yang, "Energy-constraint formation for multiagent systems with switching interaction topologies," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 7, pp. 2442–2454, 2020.

[7] L. Mo and S. Guo, "Consensus of linear multi-agent systems with persistent disturbances via distributed output feedback," *Journal of Systems Science and Complexity*, vol. 32, no. 3, pp. 835–845, 2019.

[8] Z. Ji, H. Lin, S. Cao, Q. Qi, and H. Ma, "The complexity in complete graphic characterizations of multiagent controllability," *IEEE Transactions on Cybernetics*, p. 1, 2020.

[9] S. Liu, Z. Ji, and H. Ma, "Jordan form-based algebraic conditions for controllability of multiagent systems under directed graphs," *Complexity*, vol. 2020, Article ID 7685460, 2020.

[10] L. Wang, J. Xi, M. He, and G. Liu, "Robust time-varying formation design for multiagent systems with disturbances: extended-state-observer method," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2796–2808, 2020.

[11] B. Ying and A. H. Sayed, "Diffusion gradient boosting for networked learning," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2512–2516, New Orleans, LA, USA, March 2017.

[12] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 977–992, 2019.

[13] N. Cai, M. He, Q. Wu, and M. J. Khan, "On almost controllability of dynamical complex networks with noises," *Journal of Systems Science and Complexity*, vol. 32, no. 4, pp. 1125–1139, 2019.

[14] D. Ciuonzo, P. S. Rossi, and P. Willett, "Generalized rao test for decentralized detection of an uncooperative target," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 678–682, 2017.

[15] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[16] P. Sharma, P. Khanduri, and L. Shen, "On distributed online convex optimization with sublinear dynamic regret and fit," 2020, https://arxiv.org/abs/2001.03166.

[17] A. Murdopo, "Distributed decision tree learning for mining big data streams," Master of Science Thesis, European Master in Distributed Computing, Dresden, Germany, 2013.

[18] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[19] D. Yuan, D. W. C. Ho, and G. P. Jiang, "An adaptive primal-dual subgradient algorithm for online distributed constrained optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3045–3055, 2017.

[20] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.

[21] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] Y. Hu, P. Liu, L. Kong, and D. Niu, "Learning privately over distributed features: an ADMM sharing approach," 2019, https://arxiv.org/abs/1907.07735.

[24] G. C. Rota, "Algebraic graph theory," *Graduate Texts in Mathematics*, vol. 207, no. 32, p. 298, 1994.

[25] A. Nedic, A. Olshevsky, and A. Ozdaglar, "Distributed subgradient methods and quantization effects," in *Proceedings of the Conference on Decision and Control*, pp. 4177–4184, Cancun, Mexico, December 2008.

[26] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[27] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[28] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: an exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[29] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning–part I: algorithm development," 2017, https://arxiv.org/abs/1702.05122.

[30] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[31] A. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[32] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[33] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental

limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.

[34] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proceedings of the Conference on Decision and Control*, pp. 338–343, San Diego, CA, USA, December 2006.

[35] T. Yang, L. Zhang, R. Jin, and J. Yi, "Tracking slowly moving clairvoyant: optimal dynamic regret of online learning with true and noisy gradient," in *Proceedings of The 33rd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research*, pp. 449–457, PMLR, New York, NY, USA, June 2016.

[36] H. Kasai, "SGDLibrary: a MATLAB library for stochastic optimization algorithms," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 7942–7946, 2017.

[37] P. Nazari, T. Davoud Ataee, and M. George, "Dadam: a consensus-based distributed adaptive gradient method for online optimization," 2019, https://arxiv.org/abs/1901.09109v6.

[38] D. Saad, "Online algorithms and stochastic approximations," *Online Learning*, vol. 5, pp. 6–13, 1998.

[39] J. Ye, "Stochastic gradient boosted distributed decision trees," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 2061–2064, Hong Kong, China, November 2009.

[40] G. Aceto, "Know your big data trade-offs when classifying encrypted mobile traffic with deep learning," in *Proceedings of the Traffic Monitoring and Analysis Conference*, pp. 121–128, Paris, France, June 2019.