Privacy-Preserving Distributed Learning with Nonsmooth Objective Functions

François Gauthier, Cristiano Gratton, Naveen K. D. Venkategowda, Stefan Werner
Department of Electronic Systems, NTNU – Norwegian University of Science and Technology, Trondheim

***Abstract*—This paper develops a fully distributed differentially-private learning algorithm based on the alternating direction method of multipliers (ADMM) to solve nonsmooth optimization problems. We employ an approximation of the augmented Lagrangian to handle nonsmooth objective functions. Furthermore, we perturb the primal update at each agent with a time-varying Gaussian noise with decreasing variance to provide zero-concentrated differential privacy. The developed algorithm has competitive privacy-accuracy trade-off and applies to nonsmooth and non necessarily strongly convex problems. Convergence and privacy-preserving properties are confirmed via both theoretical analysis and simulations.**

I. Introduction

Distributed machine learning algorithms have garnered significant research attention recently because of their capacity to process massive amounts of data over a network of agents [1], [2]. These methods have various applications including, monitoring of smart grids [3], statistical data analysis [4], and wireless sensor networks [5].

In many applications, the data treated by the agents is sensitive, and adversaries may try to extract private information from the network. It is, therefore, necessary to implement privacy mechanisms [6]. In this context, differential privacy provides a mechanism that protects each individual's privacy by ensuring minimal changes in the algorithm's output, whether the individual's information is present or not in the database [7], [8]. Introducing this type of privacy has the advantage of protecting from honest-but-curious agents unaffected by standard encryption techniques [6], [9]. Many interesting objectives can not be accurately modeled as strongly convex and smooth; for example, the least absolute deviation and generalized LASSO objectives [10], [11]. Therefore, it is necessary to develop algorithms applicable to these more challenging functions [12], [13].

In privacy-preserving learning, it is desirable to achieve good accuracy while providing high privacy guarantees. For this purpose, the idea of zero-concentrated differential privacy (zCDP) was introduced in [8] as an alternative to the standard (ϵ, δ)-differential privacy (DP). A comparison between the DP and zCDP metrics can be found in [7], [14], [15]. The zCDP has received much attention as it allows for better accuracy than DP while maintaining the same privacy level [6], [14], [16], [17].

This work was partly supported by the Research Council of Norway.

Several privacy-preserving distributed information processing techniques have been introduced in recent years [**?**], [18]–[26]. The algorithms in [22], [23] aim to limit an agent's privacy leakage at a single iteration. The work [24] extends the privacy leakage analysis to encompass the computation's whole length. The common assumption in these works is that the objective functions are smooth and strongly convex. In [**?**], [20], [21], smooth and non-strongly convex objectives are treated. In [19], the regularizer function can be nonsmooth, but the loss function is assumed smooth. All the above algorithms only offer solutions for problems with convex, smooth loss functions. In [27], the objective functions are not assumed to be smooth; however, the presented algorithm is not fully distributed since a central coordinator is required, and this algorithm uses an all-to-all communication protocol. Therefore, a fully distributed privacy-preserving signal processing technique that can accommodate nonsmooth and non-strongly convex loss functions is still lacking.

This paper proposes a fully distributed privacy-preserving algorithm that solves optimization problems with nonsmooth, not necessarily strongly convex, objective functions when data is distributed over a multi-agent network. The agents will use their local data-set and the received information to compute and update a local estimate of the solution. They will not share their private data but instead a noisy version of their estimate. We perturb the primal variable by adding Gaussian noise with a decreasing variance to ensure zero-concentrated differential privacy. Also, we bound the total privacy leakage of the agents throughout all the iterations. We use an approximation of the augmented Lagrangian with an l_2-norm prox-function to adapt our algorithm to the objective function's nonsmooth nature.

II. Problem Statement

We consider a connected network of $K \in \mathbb{N}$ agents modeled as an undirected graph $\mathcal{G}(\mathcal{K}, \mathcal{E})$ where the vertex set $\mathcal{K} = \{1, \dots, K\}$ corresponds to the agents and the edge set $\mathcal{E}$ contains the $|\mathcal{E}| = E$ communication links. In the fully distributed setting, an agent $k \in \mathcal{K}$ can only communicate with its neighbors whose indexes are in the set $\mathcal{N}_k$ with cardinality $|\mathcal{N}_k|$.

Each agent $k \in \mathcal{K}$ has a private data set $\mathcal{D}_k := \{(\mathbf{X}_k, \mathbf{y}_k) : \mathbf{X}_k = [\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,M_k}]^\mathsf{T} \in \mathbb{R}^{M_k \times P},\ \mathbf{y}_k = [y_{k,1}, \ldots, y_{k,M_k}]^\mathsf{T} \in \mathbb{R}^{M_k}\}$, where M_k is the number of data samples at agent k and P is the number of features in the data.

We consider the regularized empirical risk minimization (ERM) problem that is expressed as

$$\min_{\boldsymbol{\beta}} \sum_{k=1}^{K} \Big(\frac{1}{M_k} \sum_{j=1}^{M_k} \ell(\mathbf{x}_{k,j}, \mathbf{y}_{k,j}; \boldsymbol{\beta}) + \frac{\lambda}{K} R(\boldsymbol{\beta}) \Big), \tag{1}$$

where $\ell : \mathbb{R}^{M_k \times P} \times \mathbb{R}^{M_k} \times \mathbb{R}^P \to \mathbb{R}$ is the loss function, $R : \mathbb{R}^P \to \mathbb{R}$ is the regularizer function, and $\lambda > 0$ is the regularization parameter. We consider the learning problem where $\ell(\cdot)$ and $R(\cdot)$ are convex, but not necessarily strongly convex, and neither are they necessarily smooth.

To obtain a fully distributed solution for (1), we recast the above optimization problem as the following constrained minimization problem

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_k\}} \quad & \sum_{k=1}^{K} \Big(\frac{1}{M_k} \sum_{j=1}^{M_k} \ell(\mathbf{x}_{k,j}, \mathbf{y}_{k,j}; \boldsymbol{\beta}_k) + \frac{\lambda}{K} R(\boldsymbol{\beta}_k) \Big) \\ \text{s.t.} \quad & \boldsymbol{\beta}_k = \mathbf{z}_k^l,\ \boldsymbol{\beta}_l = \mathbf{z}_k^l, \quad l \in \mathcal{N}_k, \quad \forall k \in \mathcal{K}, \end{aligned} \tag{2}$$

where the primal variables $\mathcal{V} := \{\boldsymbol{\beta}_k\}_{k=1}^K$ represent local copies of $\boldsymbol{\beta}$ at the agents, and the equality constraints enforce consensus. The auxiliary variables $\mathcal{Z} := \{\mathbf{z}_k^l\}_{l \in \mathcal{N}_k}$ are only used to derive the local recursions, and are eventually eliminated.

III. Distributed Learning with Nonsmooth Objectives

This section introduces the proposed algorithm's core components and the necessary modifications to deal with nonsmooth objective functions.

To solve the minimization problem (2) with the ADMM in a distributed manner, we need to form the augmented Lagrangian, given by

$$\begin{aligned} \mathcal{L}_\rho(\mathcal{V}, \mathcal{M}, \mathcal{Z}) = & \sum_{k=1}^{K} \frac{\ell(\mathbf{X}_k, \mathbf{y}_k; \boldsymbol{\beta}_k)}{M_k} + \frac{\lambda R(\boldsymbol{\beta}_k)}{K} \\ & + \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} \Big[\boldsymbol{\mu}_k^{lT} (\boldsymbol{\beta}_k - \mathbf{z}_k^l) + \boldsymbol{\gamma}_k^{lT} (\boldsymbol{\beta}_l - \mathbf{z}_k^l) \Big] \\ & + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} \Big(||\boldsymbol{\beta}_k - \mathbf{z}_k^l||^2 + ||\boldsymbol{\beta}_l - \mathbf{z}_k^l||^2 \Big) \end{aligned} \tag{3}$$

where $\rho > 0$ is a penalty parameter and $\mathcal{M} := \{\{\boldsymbol{\mu}_k^l\}_{l \in \mathcal{N}_k}, \{\boldsymbol{\gamma}_k^l\}_{l \in \mathcal{N}_k}\}_{k=1}^K$ are the Lagrange multipliers associated with the constraints in (2).

Given that the Lagrange multipliers $\mathcal{M}$ are initialized to zero, by using the Karush-Kuhn-Tucker conditions of optimality for (2) and setting $\boldsymbol{\gamma}_k^t = 2 \sum_{l \in \mathcal{N}_k} (\boldsymbol{\gamma}_k^l)^t$, it can be shown that the Lagrange multipliers $\{\boldsymbol{\mu}_k^l\}_{l \in \mathcal{N}_k}$ and the auxiliary variables $\mathcal{Z}$ are eliminated [2], [28]. The resulting ADMM algorithm reduces to the following iterative updates at agent k

$$\begin{aligned} \boldsymbol{\beta}_k^{t+1} &= \arg\min_{\boldsymbol{\beta}_k} \left[f_k(\boldsymbol{\beta}_k) + \boldsymbol{\beta}_k^T \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left\| \boldsymbol{\beta}_k - \frac{\boldsymbol{\beta}_k^t + \boldsymbol{\beta}_l^t}{2} \right\|^2 \right] \\ \boldsymbol{\gamma}_k^{t+1} &= \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left(\boldsymbol{\beta}_k^{t+1} - \boldsymbol{\beta}_l^{t+1} \right) \end{aligned} \tag{4}$$

with

$$f_k(\boldsymbol{\beta}_k) = \frac{\ell(\mathbf{X}_k, \mathbf{y}_k; \boldsymbol{\beta}_k)}{M_k} + \frac{\lambda R(\boldsymbol{\beta}_k)}{K} \tag{5}$$

To handle nonsmooth $\ell(\cdot)$ and $R(\cdot)$ functions, we take the first-order approximation of f_k with an l_2-norm prox function, denoted as $\hat{f}_k$. Similarly, as in [27], such an approximation is given by

$$\begin{aligned} \hat{f}_k(\boldsymbol{\beta}_k; \mathcal{V}^t) = & \frac{\ell(\mathbf{X}_k, \mathbf{y}_k; \boldsymbol{\beta}_k^t)}{M_k} + \frac{\lambda R(\boldsymbol{\beta}_k^t)}{K} + \frac{\left\| \boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t \right\|^2}{2\eta_k^{t+1}} \\ & + \left(\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t \right)^\mathsf{T} \left(\frac{\ell'(\mathbf{X}_k, \mathbf{y}_k; \boldsymbol{\beta}_k^t)}{M_k} + \frac{\lambda R'(\boldsymbol{\beta}_k^t)}{K} \right) \end{aligned} \tag{6}$$

where $\mathcal{V}^t = \{\boldsymbol{\beta}_k^t, k \in \mathcal{K}\}$, η_k^t is a time-varying step-size, and $\ell'(\cdot)$ and $R'(\cdot)$ denote the subgradients of $\ell(\cdot)$ and $R(\cdot)$, respectively.

Finally, the steps of the algorithm at agent k are given by

- **Primal update**:
  $\boldsymbol{\beta}_k^{t+1} = \arg\min_{\boldsymbol{\beta}_k} \Big[\hat{f}_k(\boldsymbol{\beta}_k; \mathcal{V}^t) + \boldsymbol{\beta}_k^T \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left\| \boldsymbol{\beta}_k - \frac{\boldsymbol{\beta}_k^t + \boldsymbol{\beta}_l^t}{2} \right\|^2 \Big]$
- **Dual update** :
  $\boldsymbol{\gamma}_k^{t+1} = \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left(\boldsymbol{\beta}_k^{t+1} - \boldsymbol{\beta}_l^{t+1} \right)$

Taking the first-order approximation of f_k leads to an inexact update at a given iteration; however, the algorithm does not need to solve the problem with high precision at each iteration to guarantee overall accuracy [27]. In the end, considering $\hat{\mathcal{L}}_\rho$ instead of $\mathcal{L}_\rho$ in the primal update makes the algorithm capable of solving nonsmooth objectives with a minimal impact on overall accuracy.

IV. Privacy-Preserving Distributed Learning

This section introduces the privacy-preserving aspect of our algorithm and contains privacy and convergence analysis.

A. Distributed Algorithm with Primal Variable Perturbation

To prevent the leakage of private information of the participants, we introduce privacy in our algorithm via primal variable perturbation. That is, before sharing their local estimate $\boldsymbol{\beta}_k^t$ with their neighbors, the agents will perturb it with zero-mean Gaussian noise. The perturbed estimate of agent k

at iteration t will be denoted $\tilde{\boldsymbol{\beta}}_k^t$. Consequently, the local steps of the algorithm, for agent k at iteration t, are as follows:

$$\boldsymbol{\beta}_k^{t+1} = \arg\min_{\boldsymbol{\beta}_k} \Bigg[\hat{f}_k(\boldsymbol{\beta}_k; \tilde{\mathcal{V}}^t) + \boldsymbol{\beta}_k^T \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left\| \boldsymbol{\beta}_k - \frac{\tilde{\boldsymbol{\beta}}_k^t + \tilde{\boldsymbol{\beta}}_l^t}{2} \right\|^2 \Bigg] \tag{7}$$

$$\tilde{\boldsymbol{\beta}}_k^{t+1} = \boldsymbol{\beta}_k^{t+1} + \mathcal{N}(\mathbf{0}, \sigma_{k,t+1}^2 \mathbf{I}_P) \tag{8}$$

$$\boldsymbol{\gamma}_k^{t+1} = \boldsymbol{\gamma}_k^t + \rho \sum_{l \in \mathcal{N}_k} \left(\tilde{\boldsymbol{\beta}}_k^{t+1} - \tilde{\boldsymbol{\beta}}_l^{t+1} \right) \tag{9}$$

where $\tilde{\mathcal{V}}^t = \{\tilde{\boldsymbol{\beta}}_k^t, k \in \mathcal{K}\}$ is composed of the perturbed primal variables, $\tilde{\boldsymbol{\beta}}_k^t$, and every step can be implemented in a fully distributed manner as they only involve variables available within the agent's neighborhood.

We may consider different perturbation strategies to choose the value of the noise perturbation's variance in (8). Regardless of the chosen perturbation, the more messages are exchanged amongst agents, the easier it is for an adversary to extract information by aggregating the stalked messages [27]. Therefore, the total privacy of the algorithm decreases with the number of iterations.

Suppose the value of the noise perturbation's variance in (8) decreases slowly, at a rate of $1/\sqrt{t}$, t being the iteration index. In that case, the resulting algorithm can be seen as a fully distributed version of the DP-ADMM algorithm introduced in [27]. Both DP-ADMM and its distributed version, which we denote DDP-ADMM, use conventional (ϵ, δ)-differential privacy. In (ϵ, δ)-differential privacy, the privacy guarantee at each iteration decreases very slowly, most of the loss in privacy is due to the number of messages.

In contrast, if the noise perturbation's variance decreases at a linear rate $R < 1$ throughout the iterations, i.e., $\sigma_{k,t}^2 = R\sigma_{k,t-1}^2$, then we implement the novel dynamic zero-concentrated differential privacy. We denote this fully distributed algorithm implementing dynamic zero-concentrated differential privacy CDP-ADMM (concentrated DP-ADMM). In CDP-ADMM, the privacy loss due to the number of messages is of the same order as the privacy loss due to the decreasing variance. This method achieves better accuracy than (ϵ, δ)-differential privacy with the same level of privacy [14], [18].

The CDP-ADMM and DDP-ADMM algorithms are described in Algorithm 1 and solve (2) in a fully distributed fashion. They differ only in the variance of the noise added in (8). DP-ADMM solves (1) directly. In the simulation section, we will compare the performances of these three algorithms.

Algorithm 1 CDP-ADMM & DDP-ADMM

1: At all agents $k \in \mathcal{K}$, initialize $\boldsymbol{\beta}_k^0 = \mathbf{0}$, $\boldsymbol{\gamma}_k^0 = \mathbf{0}$,
   And run locally:
2: **for** $k = 1, 2, \ldots$ **do**
3: Update primal variable $\boldsymbol{\beta}_k^t$ as in (7)
4: Perturb $\boldsymbol{\beta}_k^t$ into $\tilde{\boldsymbol{\beta}}_k^t$ as in (8)
5: Share $\tilde{\boldsymbol{\beta}}_k^t$ with agents in $\mathcal{N}_k$
6: Update dual variable $\boldsymbol{\gamma}_k^t$ as in (9)
7: **end for**

B. Privacy Analysis

To analyze the privacy guarantee of CDP-ADMM in terms of differential privacy, we first need to measure the impact of an individual's absence in the database. This is done by computing the l_2-norm sensitivity. Then we can calibrate the magnitude of the noise added to $\boldsymbol{\beta}_k^t$ to achieve dynamic zero-concentrated differential privacy.

Definition I. *We define the l_2-norm sensitivity by*

$$\Delta_{k,2} = \max_{\mathcal{D}_k, \mathcal{D}_k'} \left\| \boldsymbol{\beta}_{k,\mathcal{D}_k}^t - \boldsymbol{\beta}_{k,\mathcal{D}_k'}^t \right\| \tag{10}$$

where $\boldsymbol{\beta}_{k,\mathcal{D}_k}^t$ and $\boldsymbol{\beta}_{k,\mathcal{D}_k'}^t$ denote the local primal variable updates from two neighboring data sets $\mathcal{D}_k$ and $\mathcal{D}_k'$ differing in only one data sample $(\mathbf{x}_{k,M_k}', y_{k,M_k}')$.

Two parameters govern the dynamic zero-concentrated differential privacy metric. The first one is the previously mentioned decrease rate of the variance, R. The second one, denoted φ_k^t, represents the privacy ensured for agent k at iteration t. A low value of φ_k^t ensures more privacy.

As in [27], we make the following necessary assumption.

Assumption 1. *There exists a constant c_1 such that $||\ell'(\cdot)|| \leqslant c_1$.*

Lemma I. *Under Assumption 1, the l_2-norm sensitivity in zero-concentrated differential privacy is given by*

$$\Delta_{k,2} = \max_{\mathcal{D}, \mathcal{D}'} ||\boldsymbol{\beta}_{k,\mathcal{D}}^t - \boldsymbol{\beta}_{k,\mathcal{D}'}^t|| = \frac{2c_1}{M_k(2\rho|\mathcal{N}_k| + \frac{1}{\eta^t})} \tag{11}$$

Proof. The proof follows from considering the explicit expressions of $\boldsymbol{\beta}_{k,D}^t$ and $\boldsymbol{\beta}_{k,D'}^t$, obtained via (6). The equality is obtained by upper-bounding the norm of their difference via the triangle inequality and the use of Assumption 1. □

Theorem I. *Under Assumption 1, the algorithm satisfies $\varphi_{k,t}$-CDP with the relation between $\varphi_{k,t}$ and $\sigma_{k,t}^2$ is given by*

$$\sigma_{k,t}^2 = \frac{\Delta_{k,2}^2}{2\varphi_{k,t}}. \tag{12}$$

Proof. We begin the proof by using [14, Lemma 2.5] on the distributions obtained for $\boldsymbol{\beta}_{k,D}^t$ and $\boldsymbol{\beta}_{k,D'}^t$ to obtain the α-Rényi divergence, necessary to establish zCDP. We then

apply Lemma I to the obtained α-Rényi divergence, replacing the distance between local estimates with its upper bound. Finally, we consider the probability of a given output in two neighboring datasets, and establish the formula by computing the ratio of these probabilities. □

Corollary. *For any $R \in (0,1)$ and $\delta \in (0,1)$, CDP-ADMM guarantees (ϵ, δ)-differential privacy with $\epsilon = \max_{0<k<K} \epsilon_k$, where $\epsilon_k = \varphi_{k,\infty} + 2\sqrt{\rho_k^{\infty} log\frac{1}{\delta}}$ and $\varphi_{k,\infty} = \varphi_{k,1} \frac{1-R^T}{R^{T-1}-R^T}$.*

Proof. We use [14, Lemma 1.7] and Theorem I to prove the privacy guarantee of CDP-ADMM in the (ϵ, δ)-DP metric for a given agent, and then use [14, Prop 1.3] to obtain the total privacy guarantee of the algorithm in the (ϵ, δ)-DP metric. □

Remark. Thanks to the result of the corollary, we can enforce all three algorithms to provide the same conventional (ϵ, δ)-differential privacy guarantees in the simulations.

C. Convergence Analysis

Under the basic assumption that the objective function is convex, the CDP-ADMM algorithm converges to the optimal solution for any $R \in (0,1)$. We give only the main ideas of the proof. We define

$$\hat{f}(\bar{\boldsymbol{\beta}}; \mathcal{V}^t) = \sum_{k=1}^{K} \hat{f}_k(\boldsymbol{\beta}_k; \mathcal{V}^t) \tag{13}$$

where $\bar{\boldsymbol{\beta}} = [\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, ..., \boldsymbol{\beta}_K^T]^T$ and $\hat{f}_k(\cdot)$ is the function defined in (6). Then, convergence follows from employing [29, Lemma 1,2,3] and [29, Th. 1] with $\hat{f}$ in place of the objective used in [29], and employing Jensen's inequality.

V. Simulations

To illustrate the performance of the CDP-ADMM algorithm, we consider the Lasso objective, i.e., we have $\ell(\mathbf{X}_k, \mathbf{y}_k; \boldsymbol{\beta}_k) = ||\mathbf{X}_k\boldsymbol{\beta}_k - \mathbf{y}_k||^2$ and $R(\boldsymbol{\beta}_k) = ||\boldsymbol{\beta}_k||_1$. Further, we consider a network of $K = 50$ agents, each with 50 local observations of the unknown parameter $\boldsymbol{\beta}$ of dimension $P = 8$. The observations of agent k are stored using the couple of matrices $(\mathbf{X}_k, \mathbf{y_k})$. $\mathbf{X}_k$'s entries are i.i.d. zero-mean unit-variance Gaussian random variables, and $\mathbf{y_k}$ contains the corresponding response vector obtained with $\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{w}$, with $\boldsymbol{\theta} \in \mathbb{R}^P$ and $\mathbf{w} \in \mathbb{R}^{M_k}$ chosen as random vectors with distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{N}(\mathbf{0}, 0.1\mathbf{I}_{M_k})$, respectively. The regularization parameter λ is set to $0.001||\mathbf{X}^{\mathsf{T}}\mathbf{y}||_\infty$ as in [30] and the penalty parameter ρ is set to 4.

Figure 1 shows the normalized error, defined as $\sum_{k=1}^{K} ||\boldsymbol{\beta}_k^t - \boldsymbol{\beta}_c||^2 / ||\boldsymbol{\beta}^c||^2$, versus iteration index t, $\boldsymbol{\beta}_c$ being the centralized solution obtained by the CVX toolbox [31]. The algorithms are tuned to provide the same total privacy guarantee. The initial faster convergence of DP-ADMM is due to its broadcast nature while the fully distributed algorithms converge at the same speed initially.

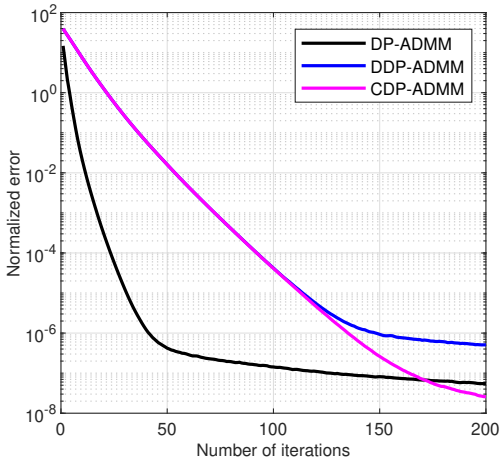

Fig. 1. Normalized error vs. iterations.

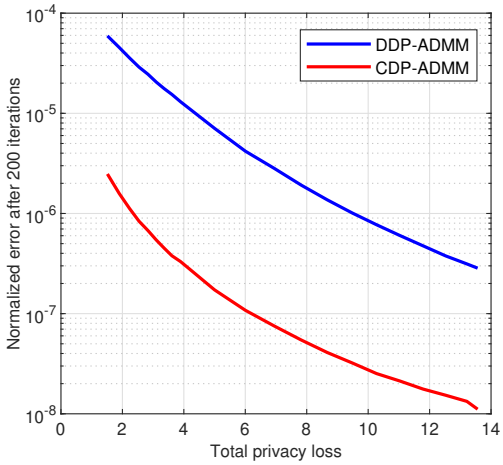

Fig. 2. Privacy-accuracy trade-off.

We notice that, after about 125 iterations, the convergence of DDP-ADMM drastically slows down, we can conjecture that the high level of noise does not allow better convergence. CDP-ADMM, however, continues to reach higher accuracy at each iteration and eventually reaches a higher accuracy than DP-ADMM just before 200 iterations. We observe that the use of zero-concentrated differential privacy allows for better accuracy given the same privacy constraints.

Figure 2 shows the normalized error obtained by the algorithms after 200 iterations versus their total privacy loss. We recognize the values obtained in Fig. 1 for a total privacy loss equal to 10. We consider a total privacy loss between 1 and 14 because it corresponds with an ϵ between 0 and 1 in (ϵ, δ)-differential privacy. We can see that the privacy-accuracy trade-off of both algorithms is very similar except that the one for CDP-ADMM is consistently lower than the one for DDP-ADMM. This means that for a given privacy guarantee, CDP-ADMM can achieve higher accuracy in 200 iterations.

VI. Conclusion

We developed a fully distributed differentially private learning algorithm based on the alternating direction method of multipliers to solve nonsmooth optimization problems. Our algorithm does not rely on any centralized processing and can handle nonsmooth loss and regularizer functions thanks to the first-order approximation of the objective functions. Furthermore, the application of zero-concentrated differential privacy via primal variable perturbation allows us to achieve a competitive privacy-accuracy trade-off.

References

[1] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 977–992, Feb. 2019.

[2] G. B. Giannakis, Q. Ling, G. Mateos, and I. D. Schizas, "Splitting Methods in Communication, Imaging, Science, and Engineering," in *Scientific Computation*. Springer Int. Publishing, 2016, pp. 461–497.

[3] S. Nabavi, J. Zhang, and A. Chakrabortty, "Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional PMU-PDC architectures," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2529–2538, Mar. 2015.

[4] D. Froelicher, J. R. Troncoso-Pastoriza, J. S. Sousa, and J.-P. Hubaux, "Drynx: Decentralized, secure, verifiable system for statistical queries and machine learning on distributed datasets," *IEEE Trans. Inf. Forensics and Secur.*, vol. 15, pp. 3035–3050, 2020.

[5] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed Clustering Using Wireless Sensor Networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 707–724, 2011.

[6] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[7] C. Dwork, N. Kohli, and D. Mulligan, "Differential Privacy in Practice: Expose your Epsilons!" *J. Privacy and Confidentiality*, vol. 9, no. 2, Oct. 2019.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Conf. Theory Cryptography*, 2006, pp. 265–284.

[9] A. Paverd, A. Martin, and I. Brown, "Modelling and automatically analysing privacy properties for honest-but-curious adversaries," *Tech. Rep.*, Oct. 2014.

[10] J. E. Gentle, "Least absolute values estimation: An introduction," *Commun. Statist.-Simul. Comput.*, vol. 6, no. 4, pp. 313–328, Jan. 1977.

[11] V. Roth, "The Generalized LASSO," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 16–28, Feb. 2004.

[12] F. H. Clarke, "Optimization and nonsmooth analysis." SIAM, 1990.

[13] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, "Nonsmooth Analysis and Control Theory," vol. 178. Springer Science & Business Media, 2008.

[14] M. Bun and T. Steinke, "Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds," in *Theory of Cryptography Conf.* Springer, 2016, pp. 635–658.

[15] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *28th USENIX Secur. Symp.*, 2019, pp. 1895–1912.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Theory of Cryptography*. Springer Berlin Heidelberg, 2006, pp. 265–284.

[17] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.

[18] J. Ding, Y. Gong, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," *Available at http://arxiv.org/abs/1901.02094*, Feb. 2019.

[19] C. Chen and J. Lee, "Rényi Differentially Private ADMM for Non-Smooth Regularized Optimization," *Proc. Tenth ACM Conf. Data and Appl. Secur. and Privacy*, pp. 319–328, 2020.

[20] J. Ding, X. Zhang, M. Chen, K. Xue, C. Zhang, and M. Pan, "Differentially Private Robust ADMM for Distributed Machine Learning," in *2019 IEEE Int. Conf. Big Data*, Dec. 2019, pp. 1302–1311.

[21] T. Zhang and Q. Zhu, "A Dual Perturbation Approach for Differential Private ADMM-Based Distributed Empirical Risk Minimization," *Proc. ACM Workshop Artif. Intell. Secur.*, p. 129–137, 2016.

[22] ——, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 148–161, Mar. 2018.

[23] ——, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forens. Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[24] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, Jul. 2018, pp. 5796–5805.

[25] Z. Huang and Y. Gong, "Differentially Private ADMM for Convex Distributed Learning: Improved Accuracy via Multi-Step Approximation," *arXiv preprint arXiv:2005.07890*, May 2020.

[26] Y. Ye, H. Chen, M. Xiao, M. Skoglund, and H. V. Poor, "Incremental ADMM with Privacy-Preservation for Decentralized Consensus Optimization," *IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 209–214, Aug. 2020.

[27] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forens. Security*, vol. 15, pp. 1002–1012, July 2019.

[28] C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner, "Distributed Ridge Regression with Feature Partitioning," *Proc. Asilomar Conf. Signals Syst. Comput.*, pp. 1423–1427, 2018.

[29] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust decentralized learning using ADMM with unreliable agents," *arXiv preprint arXiv:1710.05241*, Oct. 2017.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2010.

[31] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.