Rényi Differentially Private ADMM for Non-Smooth Regularized Optimization

Anonymous Author(s)

ABSTRACT

In this paper we consider the problem of minimizing composite objective functions consisting of a convex differentiable loss function plus a nonsmooth regularization term, such as L_1 norm or nuclear norm, under Rényi differential privacy (RDP). To solve the problem, we propose two stochastic alternating direction method of mulipliers (ADMM) algorithms: ssADMM based on gradient perturbation and mpADMM based on output perturbation. Both algorithms decompose the orignal problem into subproblems that have closed-form solutions. The first algorithm, ssADMM, applies the recent privacy amplification result for RDP to reduce the amount of noise to add. The second algorithm, mpADMM, numerically computes the sensitivity of ADMM variable updates and releases the updated parameter vector at the end of each epoch. We compare the performance of our algorithms with several baseline algorithms on both real and simulated datasets. Experimental results show that, in high privacy regimes (small ϵ), ssADMM and mpADMM outperform other baseline algorithms in terms of classification and feature selection performance, respectively.

KEYWORDS

differential privacy, convex optimization, ADMM, regularization

ACM Reference Format:

Anonymous Author(s). 2019. Rényi Differentially Private ADMM for Non-Smooth Regularized Optimization. In *CODASPY '20, Mar 16 - 18, 2020, New Orleans, LA,*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Concerns on privacy of individuals in the data used for training machine learning models have led to extensive research on private model building techniques [1, 3, 9, 10, 21, 34, 40], especially in the context of Empirical Risk Minimization (ERM). Let $D = (d_1, d_2, \ldots, d_n)$ be a dataset, where $d_i \in \mathcal{D}$. Many machine learning problems can be formulated as regularized optimization problems of the form:

$$\min_{x \in \mathbb{R}^p} F(x) := \frac{1}{n} \sum_{i=1}^{n} f(x, d_i) + \lambda h(x), \tag{1}$$

where $\lambda > 0$ is a regularization coefficient, $f : \mathbb{R}^p \times \mathcal{D} \to \mathbb{R}$ is a smooth convex loss function, and $h : \mathbb{R}^p \to \mathbb{R}$ is a simple convex *nonsmooth* regularizer such as L_1-norm or nuclear norm. This formulation has received substantial attention as it arises in many interesting applications of machine leraning such as generalized lasso [33], matrix recovery [24, 44], and a class of L_1 regularized problems. Despite recent advances in methods for differentially private ERM, many existing solutions are not directly applicable to the problem in (1) due to requirement for differentiability [1, 3, 10, 40] or strong convexity [9] of the regularization term $h(x)$. Alternating direction method of multipliers (ADMM) [18] has shown to be effective in solving optimization problems with complicated structure regularization.

In this paper, we propose two stochastic ADMM algorithms that satisfy Rényi Differential Privacy (RDP), namely subsampled stochastic ADMM (ssADMM) and model perturbation based ADMM (mpADMM). The first algorithm has the following key features. First, ssADMM is scalable and fast. The algorithm splits the composite objective function into differentiable and nonsmooth terms, $\sum_i f(x, d_i)$ and $h(x)$, using the ADMM framework. The differentiable term is further approximated by the first order Taylor expansion and linearization as in [27]. This approximated augmented Lagrangian function has a simple analytical solution. For the nonsmooth regularization term $h(x)$, ssADMM applies proximal mappings. For many nonsmooth regularization function popularly used in machine learning, such as L_1-norm, SCAD [17], and MCP [39], those proximal mappings yield closed form solutions. Therefore, both subproblems can be solved efficiently.

Second, ssADMM makes use of recently proposed *privacy amplification* lemma [36] to tightly bound the total privacy loss across many iterations. In the closed-form solution of the modified augmented Lagrangian function, the only data dependent term is the gradient $\nabla f(x^k)$, where x^k denotes the value of x at iteration k. The algorithm computes the gradient $\nabla f(x^k)$ using a randomly *subsampled* data and add Gaussian noise to ensure (α, ϵ_k)-RDP, which allows us to exploit the randomness in the subsampling and to introduce less noise to each iteration.

The second algorithm, mpADMM, takes the output perturbation approach but substantially differs from the original method. Unlike the original method which releases model parameters once only at the end, the proposed method releases the output after each epoch. For each epoch, we numerically compute the sensitivity of both primal and dual variable updates in ADMM and release the parameter vector using the Gaussian mechanism. The algorithm uses the released (noisy) output as the starting value for the next epoch.

Our contributions are summarized as follows:

- We propose two efficient Rényi differentially private algorithms, based on stochastic ADMM, for solving nonsmooth convex optimization problems. In our proposed ssADMM, each subproblem is solved exactly in closed form.
- We apply the recent privacy amplification result for RDP to stochastic ADMM and show that the inherent randomness in

CODASPY '20, Mar 16 - 18, 2020, New Orleans, LA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

subsampling process can be used to achieve stronger privacy protection.
- We empirically show the effectiveness of the proposed algorithms by performing extensive empirical evaluations on generalized linear models and comparing with other baseline algorithms. The results show that, in high privacy regimes (small ϵ), ssADMM and mpADMM outperform other baseline algorithms in terms of classification and feature selection performance, respectively.

The rest of this paper are organized as follow: Section 2 summarizes related work. In Section 3, we provide background on Rényi differential privacy and ADMM. Section 4 introduces the proposed Rényi differentially private ADMM algorithms. Section 5 provides the performance evaluations on both synthetic and real datasets. Section 6 concludes the paper.

2 RELATED WORK

Many works have been done to solve the empirical risk minimization problem under differential privacy. Generally, there are three types of algorithms proposed. Output perturbation algorithms perturb the model parameters based on sensitivity, for example, [9] analyzed the sensitivity of optimal solutions trained between neighboring databases; [40] tackled the case when full gradient descent is applied; and [37] and [10] analyzed the situation of applying stochastic gradient descent on permuting mini-batches. Objective perturbation algorithms perturb the training objective functions, and the privacy guarantee is subject to an exact solution of the ERM problem: [9] presented the first objective perturbation technique, and it is extended by [21]. Gradient perturbation algorithms perturb the (stochastic) gradients used for model updating by first-order optimization methods, and use a composition technique to quantify the overall privacy leak for multiple access of the data through gradient calculation. For example, [3] proposed "strong composition" theorem, then [1] proposed "moment accountant" method, which is also used in [34] and [22]. The Réyni differential privacy was introduced by [25], which can also be applied in gradient perturbation, especially after [36] proposed its amplification by subsampling results.

Alternating Direction Method of Multipliers (ADMM) is an old algorithm to solve optimization problems [5]. It has been extensively studied, and applied in many domains such as outlier recovery [31], image processing [7], and sensor detection [12]. In addition to its original version, many variations has been presented, such as [16, 38] and [27]. Several ADMM based differentially private algorithms have been presented, for example, [35] applied objective perturbation technique on the original ADMM problem, [42] and [43] applied output and objective perturbation technique, and [20] applied gradient perturbation technique on ADMM-based algorithms in distributed settings.

L_1 regularized ERM problem was first proposed for linear regression, that is least absolute shrinkage and selection operator (LASSO) [32]. Some variants of LASSO exists, such as [45] and [29]. It has been used for classification problems, and many algorithms for solving L_1 regularized generalized linear models were presented, such as [23], [28], and [4]. [26] and [19] has shown that L_1 regularized classification has good performance in feature selection. Limited to the assumption on the loss function, many differentially private ERM algorithms cannot be directly applied on L_1 regularized classification, with a few exceptions such as [1, 35], and [20].

3 PRELIMINARIES

In this section we introduce relative background of this paper. We will start with definitions and lemmas in differential privacy and Rényi differential privacy, the L_1-regularized classification problem we aim to solve, and then the ADMM algorithm based on which we proposed our algorithms.

We assume a dataset $D = \{d_1, ..., d_n\} \sim \mathcal{D}^n$ is a set collected from n individuals from an unknown population distribution $\mathcal{D}$, where $d_i = (s_i, l_i)$ for $i = 1, ..., n$ is a record of one individual, with s_i being a vector of features of dimension p, and $l_i \in \{-1, +1\}$ being its label. Two datasets D and D' are considered *neighboring*, if D' can be obtained by replacing one record with another one from $\mathcal{D}$, notated as $D \sim D'$. We use x, y, z to denote model parameters, and $\|\cdot\|_1$ (resp. $\|\cdot\|_2$) as L_1 (resp. L_2) norm of a vector.

3.1 Differential Privacy

Differential privacy is so far the standard standard for protecting the privacy of sensitive datasets. Its formal definition is stated as:

Definition 1 ((ϵ, δ)-Differential Privacy (DP)). *[15] [14] Given privacy parameters $\epsilon \geq 0, 0 \leq \delta \leq 1$, a randomized mechanism (algorithm) $\mathcal{M}$ satisfies (ϵ, δ)-DP if for every event $S \subseteq range(M)$, and for every pair of neighboring datasets $D \sim D'$,*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta \tag{2}$$

If $\delta = 0$, it is called *pure* differential privacy, and $\delta > 0$ is called *approximate* differential privacy.

With pure differential privacy, even the strongest attacker with arbitrary background information has limited ability to make inferences on the unknown record(s). With approximated differential privacy, this guarantee holds with a high chance, while failure of privacy preserving happens with probability at most δ (informally called "all-bets-are-off"). In practice, δ should be taken significantly small, such as $\Theta(n^{-2})$.

While approximate DP is a relaxation of pure DP, some other relaxations also exists, such as zero-concentrated differential privacy (zCDP) [6] and Rényi Differential Privacy (RDP) [25]. These relaxations do not have such semantic meanings as approximate DP, but they are shown to stand between pure and approximate DP: they provide weaker protection than pure DP, but stronger protection than approximated DP, for any given $\delta > 0$. In this paper, we will focus on Rényi Differential Privacy.

3.2 Rényi Differential Privacy

Define $Z = \frac{\Pr[\mathcal{M}(D) \in S]}{\Pr[\mathcal{M}(D') \in S]}$ as the privacy loss random variable, instead of requiring it always lies inside range $[-\epsilon, \epsilon]$ as pure DP, Rényi differential privacy (RDP) constraints its expectation by Rényi divergence.

Definition 2 ((α, ϵ)-Rényi Differential Privacy (RDP)). *[25] Given a real number $\alpha \in (1, +\infty)$ and privacy parameter $\epsilon \geq 0$, a randomized mechanism (algorithm) $\mathcal{M}$ satisfies (α, ϵ)-RDP if for every pair of neighboring datasets $D \sim D'$, the Rényi α-divergence*

between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ satisfies

$$D_\alpha[\mathcal{M}(D)\|\mathcal{M}(D')] \leq \epsilon \tag{3}$$

That is, the privacy parameter ϵ bounds the moment α of the *Rényi divergence* (D_α), which is defined as

Definition 3 (Rényi Divergence). *For probability distributions $\mathcal{M}(D)$ and $\mathcal{M}(D')$ over a set Ω, and let $\alpha \in (1, +\infty)$. Then Rényi α-divergence is*

$$D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) := \frac{1}{\alpha-1}\log \mathbb{E}_{x\sim\mathcal{M}(D')}\left[\left(\frac{P_{\mathcal{M}(D)}(x)}{P_{\mathcal{M}(D')}(x)}\right)^\alpha\right] \tag{4}$$

One method to achieve RDP is through the Gaussian mechanism: when a query $q(D)$ is taken over the dataset, the Gaussian mechanism adds a Gaussian noise $\gamma \sim \mathcal{N}(0, \sigma^2\mathbf{I}_k)$, and release perturbed $q(D) + \gamma$.

Lemma 1 (Gaussian Mechanism). *[25] Let $q : \mathcal{D}^n \to \mathbb{R}^k$ be a vector-valued function over datasets. Let $\mathcal{M}$ be a mechanism releasing $q(D) + \gamma$ where $\gamma \sim \mathcal{N}(0, \sigma^2\mathbf{I}_k)$, then for any $D \sim D'$ and any $\alpha \in (1, +\infty)$,*

$$D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \alpha\Delta_2^2(q)/(2\sigma^2) \tag{5}$$

Gaussian mechanism relies on the L_2 sensitivity:

Definition 4 (L_2 sensitivity). *Let $q : \mathcal{D}^n \to \mathbb{R}^k$ be a vector-valued function over datasets. The L_2 sensitivity of q, denoted as $\Delta_2(q)$, is defined as*

$$\Delta_2(q) = \sup_{D\sim D'} \|q(D) - q(D')\|_2 \tag{6}$$

Therefore, when scale the variance $\sigma^2 = \alpha\Delta_2^2(q)/(2\epsilon)$, then $\mathcal{M}$ satisfies (α, ϵ)-RDP.

Gaussian mechanism makes the mechanism $\mathcal{M}$ satisfy (α, ϵ)-RDP for a series of α, so we can use $\epsilon(\alpha)$ to denote the privacy ϵ under moment α. In empirical risk minimization algorithms, it is common that the mechanism is taken over a randomized subsample of the dataset B, instead of the whole dataset D. Then, application Gaussian Mechanism on the subsample B would satisfy $(\alpha, \epsilon(\alpha))$-RDP with respect to B. Due to the subsampling procedure, the mechanism would satisfy an amplified privacy with respect to the whole dataset D, as given by the following lemma:

Lemma 2 (RDP for subsampled mechanism). *[36] For a randomized mechanism $\mathcal{M}$ and a dataset $D \sim \mathcal{D}^n$, define $\mathcal{M} \circ$ subsample as (1) subsample without replacement m datapoints from the dataset (denote $q = m/n$ as sampling ratio); (2) apply $\mathcal{M}$ on the subsampled dataset as input, then if $\mathcal{M}$ satisfies $(\alpha, \epsilon(\alpha))$-RDP with respect to the subsample for all integers $\alpha > 2$, then the new randomized mechanism $\mathcal{M} \circ$ subsample satisfies $(\alpha, \epsilon'(\alpha))$-RDP with respect to D, where*

$$\epsilon'(\alpha) \leq \frac{1}{\alpha-1}\log\Big(1 + q^2\binom{\alpha}{2}\min\left\{4(e^{\epsilon(2)}-1), 2e^{\epsilon(2)}\right\} + \sum_{j=3}^{\alpha} q^j\binom{\alpha}{j}2e^{(j-1)\epsilon(j)}\Big) \tag{7}$$

Similar as DP, RDP has below composition properties:

Lemma 3 (RDP composition). *[25] For randomized mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ applied on dataset D, if $\mathcal{M}_1$ satisfies (α, ϵ_1)-RDP and M_2 satisfies (α, ϵ_2)-RDP, then their composition $\mathcal{M}_1 \circ \mathcal{M}_2$ satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

RDP is said to provide stronger protection than approximate DP, due to below conversion to (ϵ, δ)-DP:

Proposition 1 (RDP to (ϵ, δ)-DP). *[25] If $\mathcal{M}$ satisfies (α, ϵ)-RDP, then it satisfies $(\epsilon(\delta), \delta)$-DP for $\epsilon(\delta) \geq \epsilon + \frac{\log(1/\delta)}{\alpha-1}$.*

Therefore, when evaluating our proposed algorithms, to compare with other algorithms which satisfies (ϵ, δ)-DP, we keep track of (α, ϵ) pairs which our algorithm satisfies for a series of α values, then convert each pair into a $(\epsilon(\delta), \delta)$ pair it satisfies by Proposition 1, for a pre-defined small δ, and choose the smallest $\epsilon(\delta)$ as the (ϵ, δ)-DP it satisfies to compare with other algorithms.

3.3 Regularized Empirical Risk Minimization

Many problems in machine learning can be formulated as empirical risk minimization (ERM), which seek a solution $x^* \in \Theta$ that minimizes an empirical loss on the training data:

$$x^* = \arg\min_{x\in\Theta} F(x, D) := \arg\min_{x\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell(x, d_i), \tag{8}$$

where Θ is a parameter space, ℓ is a *loss* function. To prevent overfitting, it is common to add a (data-independent) regularization term into the objective function, i.e. $\ell(x, d_i) = f(x, d_i) + R(x)$. For L_1 regularization, $R(x) = \lambda\|x\|_1$. For example, L_1 regularized logistic regression, one can fit the model by solving

$$x^* = \arg\min_{x\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-l_i x^T s_i)) + \lambda\|x\|_1 \tag{9}$$

Recall that each datum $d_i = (s_i, l_i)$ as feature vector s_i and label l_i. However, due to that many optimization algorithms assume the loss function to be doubly differentiable, it cannot be directly used on L_1 regularization problems. In this paper, we make the following assumptions on the loss function:

- **Convexity** Both the data-dependent function f and regularization term R are convex.
- **Differentiability** The non-regularized data-dependent function f is continuously differentiable with respect to x.
- **Bounded gradient** There exists a constant $C > 0$ such that $\|\nabla f(x, d)\|_2 \leq C$ for all $x \in \Theta$ and $d \in \mathcal{D}$. Usually it is satisfied by preprocessing the data to ensure the feature s_i of each data d_i lies inside a ball of some radius r, or directly clip the L_2 norm of individual gradient by a threshold C.

3.4 Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) algorithm was proposed decades ago, and has recently been widely used to solve optimization problems in machine learning [5]. Consider the optimization problem

$$\begin{aligned} &\text{minimize} && f(x) + h(z) \\ &\text{subject to} && Ax + Bz = c \end{aligned} \tag{10}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, and $c \in \mathbb{R}^p$. ADMM forms the augmented Lagrangian of the problem:

$$L_\rho(x, z, y) := f(x) + h(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \quad (11)$$

where x, z are called the *primal* variables, $y \in \mathbb{R}^p$ is called the *dual* variable, and $\rho > 0$ is a pre-selected *penalty* parameter.

ADMM algorithm solves the optimization problem by alternating the iterations below

$$x\text{-minimization step: } x^{k+1} \leftarrow \arg\min_x L_\rho(x, z^k, y^k) \quad (12)$$

$$z\text{-minimization step: } z^{k+1} \leftarrow \arg\min_z L_\rho(x^{k+1}, z, y^k) \quad (13)$$

$$\text{dual variable update: } y^{k+1} \leftarrow y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \quad (14)$$

Therefore, x and z are updated in an alternating fashion, and separating minimization over x and z into two steps can make the otherwise hard-to-solve optimization problem solvable in a sequential manner.

3.5 Stochastic ADMM

One variant of ADMM, stochastic ADMM (sADMM), was proposed by [27] and tested on L_1 regularized linear regression (LASSO). This variant was proposed based on the observation that, for ADMM problems, usually one of $f(x)$ and $h(z)$ is data-dependent, and it is both expensive and unnecessary to exactly solve its minimization step for each iteration. To be specific, let f be data-dependent, and h be data-independent, then the optimization problem becomes $f(x, D) + h(z)$, and sADMM approximate L_ρ by *approximated* augmented Lagrangian $\hat{L}_\rho$, defined at iteration k as

$$\begin{aligned}\hat{L}_\rho(x, z, y) := f(x^k) + \langle \nabla f(x^k, B_k), x\rangle + \frac{\|x - x^k\|_2^2}{2\eta^k} \\ + h(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2\end{aligned} \quad (15)$$

where B_k is a portion of the data accessed at iteration k, and η^k is the learning rate at iteration k. After this approximation of L_ρ by $\hat{L}_\rho$, one can derive an exact solution for each x-minimization step in (12), instead of solving a computationally expensive ERM problem.

For L_1 regularized ERM, let $h(z)$ be the regularization term $R(z) = \lambda\|z\|_1$, the constraint $Ax + Bz = c$ reduces to $x = z$, then by taking derivative of $\hat{L}_\rho(x, z^k, y^k)$ and set to zero, one get

$$x^{k+1} \leftarrow \frac{1}{\rho + 1/\eta^k}(-\nabla f(x, B_k) - y^k + \rho z^k + x^k/\eta^k) \quad (16)$$

as the exact solution to minimize $\hat{L}_\rho(x, z^k, y^k)$, and

$$y^{k+1} \leftarrow y^k + \rho(x - z) \quad (17)$$

to update the dual variable y.

4 ALGORITHMS

In this section we propose the main algorithms. We propose two sADMM based L_1 regularized classification algorithms, both satisfies Rényi differential privacy. One achieves privacy by gradient perturbation relying on randomized subsampling; the other is through model perturbation after each epoch relying on sensitivity calculation. Both algorithms assume a centralized computing: all training data were collected in a center, which performs the computation locally. This is because we assume the data is small-to-median sized, where L_1 regularization are usually applied on.

4.1 Rényi differentially private subsampling algorithm

Our subsampling private sADMM algorithm (ssADMM) is presented in Algorithm 1. This algorithm is inspired by the gradient perturbation technique proposed in [1], on differentially private stochastic gradient descent (DP-SGD).

Similar as DP-SGD, our ssADMM algorithm perturbs the mini-batch gradient by Gaussian noise right after gradient evaluation in line 6. However, Algorithm 1 differs from DP-SGD for the following aspects: (i) By utilizing ADMM, we are able separate gradient descent and L_1 regularization into two steps, so that pure gradient can be computed and perturbed in x-minimization step; for DP-SGD, proximal gradient has to be used to handle L_1 regularization; (ii) while DP-SGD suggest using constant learning rate, we proved that using decreasing step size in Algorithm 1 help accelerate convergence, as in Theorem 2 and numerical experiments; (iii) authors of DP-SGD proposed the moment accountant (MA) method to analyze the privacy loss, and convert to (ϵ, δ)-DP; we use the most recent RDP for subsampling mechanism, which is a more advanced technique to analyze privacy loss, and also easier to implement.

Algorithm 1 RDP subsampling sADMM L_1 regularized ERM algorithm (ssADMM)

1: **Input**: Dataset $D = \{d_1, ..., d_n\}$. Penalty parameter ρ, mini-batch size m, total iterations T.
2: **Initialize**: primal variables x^0, z^0, dual variable y^0.
3: **for** iteration $k = 0, 1, ..., T-1$ **do**
4: Sample mini-batch B_k from D of size m.
5: $g_k \leftarrow \frac{1}{m}\sum_{d_i \in B_k} \nabla f(x^k, d_i)$ ▷ compute gradient
6: $\tilde{g}_k \leftarrow g_k + \gamma$ where $\gamma \sim N(0, \sigma^2\mathbf{I}_p)$ ▷ perturb gradient by Gaussian noise
7: Compute x^{k+1} by (16) using $\tilde{g}_k$ ▷ primal variable x
8: Compute z^{k+1} by (18) ▷ primal variable z
9: Compute y^{k+1} by (17) ▷ dual variable y
10: **end for**
11: **Output**: x^T

Since the regularization is data-independent, it does not cause any privacy leak. Therefore, any (non-) smooth regularizers are applicable for Algorithm 1, with the same privacy guarantee. Since in this paper we use L_1 regularization as an example, for the z-minimization step, we utilize soft-thresholding technique from [5] to acquire the solution to minimize $L_\rho(x^{k+1}, z, y^k)$:

$$z^{k+1} \leftarrow \mathcal{S}_{\frac{\lambda}{\rho}}(x^{k+1} + y^k/\rho) \quad (18)$$

where soft-thresholding operator is defined as

$$\mathcal{S}_t(x)_i = \begin{cases} x_i - t & \text{if } x_i > t \\ x_i + t & \text{if } x_i < -t \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Similar technique has been used in [27] and [35].

Another ADMM based algorithm proposed in [20] (DP-ADMM) also used gradient perturbation technique. Our method differed from theirs for the following aspects: (i) DP-ADMM is used for distributed learning, so that the training objective is assigned into multiple parties each holding a portion of the data, instead in ssADMM it is the data dependent loss and regularization that are separated; (ii) in DP-ADMM, each party is perturbing full gradient and transmit to the center, so that there is no privacy amplification effect, therefore although both algorithms solve optimization approximately, their privacy loss is higher than ours at each step. Our methods differ from the ADMM-objP method (DPLL in [35]) for the following aspect: (i) ADMM-objP perturb the training objective at each iteration, and use full gradient descent multiple times to acquire exact solution at each iteration, which is not as efficient as ours, since our method only access a portion of data once at each step; (ii) ADMM-objP guarantees privacy only if exact solution is acquired at each step, therefore the privacy guarantee is only theoretically true. The privacy guarantee of ssADMM is given by Theorem 1.

Theorem 1. *Algorithm 1 is (α, ϵ)-RDP.*

Proof. We first show the L_2 sensitivity of batch gradient g_k. Assume neighboring mini-batches B_i and B_i' differ by one record $d_s \in B$ and $d_s \in B'$, by Definition 4,

$$\begin{aligned} \Delta_2^k(g) =& \Delta_2[\frac{1}{m}\sum_{d_i \in B_k} \nabla f(x^k, d_i)] \\ =& \sup_{B_k \sim B_k'} \|\frac{1}{m}\sum_{d_i \in B_k} \nabla f(x^k, d_i) - \frac{1}{m}\sum_{d_i \in B_k'} \nabla f(x^k, d_i)\|_2 \\ =& \frac{1}{m}\sup \|\nabla f(x^k, d_s) - \nabla f(x^k, d_s')\|_2 \le \frac{2C}{m} \end{aligned} \tag{20}$$

Let $\epsilon_k(\alpha) = \alpha(\Delta_2^k(g))^2/2\sigma^2$. So each iteration is $(\alpha, \epsilon_k(\alpha))$-RDP by Lemma 1, with respect to the batch B_k. Since B_k is a randomized subsample of D, by Lemma 2, we can calculate $\epsilon_k'(\alpha)$ so that each iteration is $(\alpha, \epsilon_k'(\alpha))$-RDP with respect to D. Since the algorithm has run T iterations, let $\epsilon = \sum_{k=0}^{T-1} \epsilon_k'(\alpha)$, by Lemma 4, Algorithm 1 is (α, ϵ)-RDP. □

Theorem 2. *If we choose $\eta^k = O(1/\sqrt{k})$, and train for t iterations, then Algorithm 1 has the expected convergence rate of $O(1/\sqrt{t})$.*

Proof. See proof in appendix. □

4.2 Rényi differentially private model perturbation algorithm

Our model perturbation private sADMM algorithm (mpADMM) is presented in Algorithm 2. Different from perturbing the gradients, this algorithm use the unperturbed gradients to do model calculation for a whole step, and keep track of the L_2 sensitivity of all data-dependent model vectors. After each epoch, Gaussian noises are injected into model vectors x, y, z, and total privacy ϵ is updated, according to sensitivity and σ^2. Due to it is difficult to calculate the sensitivity over multiple epochs, we perform output perturbation after each epoch. Therefore, this algorithm can be considered as multiple-time output perturbation algorithm.

Algorithm 2 RDP model perturbation sADMM L_1 regularized ERM algorithm (mpADMM)

1: **Input**: Dataset $D = \{d_1, ..., d_n\}$. Penalty parameter ρ, total epochs T.
2: **Initialize**: primal variables x^0, z^0, dual variable y^0.
3: **for** epoch $k = 0, 1, ..., T-1$ **do**
4: $g_k \leftarrow \frac{1}{n}\sum_{d_i \in D} \nabla f(x^k, d_i)$ ▷ compute gradient
5: Compute x^{k+1} by (16) ▷ primal variable x
6: Compute z^{k+1} by (18) ▷ primal variable z
7: Compute y^{k+1} by (17) ▷ dual variable y
8: Sample $\gamma_1, \gamma_2, \gamma_3 \sim N(0, \sigma^2 \mathbf{I}_p)$
9: $x^{k+1} \leftarrow x^{k+1} + \gamma_1$, $y^{k+1} = y^{k+1} + \gamma_2$, $z^{k+1} = z^{k+1} + \gamma_3$ ▷ perturb the model
10: **end for**
11: **Output**: x^T

To calculate the sensitivity, since unperturbed batch gradient is used here, after one epoch, all primal and dual variables are data-dependent. Assume neighboring datasets D and D' differ at position s: $d_s \in D$ and $d_s' \in D'$. We define $\delta_x := x - (x')$ where x and (x') are primal variables evaluated on D and D', respectively, after one epoch. Also, define δ_z^k and δ_y^k similarly. Then, after epoch k,

$$\begin{aligned} \delta_x^{k+1} =& x^{k+1} - (x')^{k+1} \\ =& \frac{1}{\rho + 1/\eta^k}(-\frac{1}{n}\sum_{d_i \in D} \nabla f(x^k, d_i) - y^k + \rho z^k + x^k/\eta^k) - \\ & \frac{1}{\rho + 1/\eta^k}(-\frac{1}{n}\sum_{d_i \in D'} \nabla f(x^k, d_i) - y^k + \rho z^k + x^k/\eta^k) \\ =& (\nabla f(x^k, d_s') - \nabla f(x^k, d_s))/n(1 + \eta^{k+1}\rho) \end{aligned} \tag{21}$$

Consider when the soft-thresholding operator $\mathcal{S}_t$ (19) applied on two vectors w and w', and compare $\mathcal{S}_t(w) - \mathcal{S}_t(w')$ with $w - w'$ element-wise:

- If w_i and w_i' are of different signs, applying $\mathcal{S}$ on w_i and w_i' would bring them closer, therefore $|\mathcal{S}_t(w_i) - \mathcal{S}_t(w_i')| < |w_i - w_i'|$;
- If w_i and w_i' are of the same sign, without loss of generality, let $|w_i| \le |w_i'|$. One can easily observe that
  - If $t \le |w_i| \le |w_i'|$, then $|\mathcal{S}_t(w_i) - \mathcal{S}_t(w_i')| = |(|w_i| - t) - (|w_i'| - t)| = |w_i - w_i'|$;
  - If $|w_i| < t < |w_i'|$, then $|\mathcal{S}_t(w_i) - \mathcal{S}_t(w_i')| = |0 - (|w_i'| - t)| < |w_i - w_i'|$ since $t < |w_i'|$;
  - If $|w_i| \le |w_i'| \le t$, then $|\mathcal{S}_t(w_i) - \mathcal{S}_t(w_i')| = 0 \le |w_i - w_i'|$;

For vectors u, v, we can use $u \preccurlyeq v$ to represent $|u_i| < |v_i|$ and u_i, v_i have the same sign, for each index i. Obviously $u \preccurlyeq v$ indicates $\|u\|_2 \le \|v\|_2$. In either case above, we have $|\mathcal{S}_t(w_i) - \mathcal{S}_t(w_i')| \le$

$|w_i - w'_i|$, and sign preserves (or becomes zero), so $\mathcal{S}_t(w) - \mathcal{S}_t(w') \preccurlyeq w - w'$ for any threshold t. Therefore,

$$\begin{aligned}\delta_z^{k+1} =& z^{k+1} - (z')^{k+1} \\ =& \mathcal{S}_{\frac{\lambda}{\rho}}(x^{k+1} + y^k/\rho) - \mathcal{S}_{\frac{\lambda}{\rho}}((x')^{k+1} + y^k/\rho) \\ \preccurlyeq& x^{k+1} + y^k/\rho - ((x')^{k+1} + y^k/\rho) = \delta_x^{k+1}\end{aligned} \tag{22}$$

and

$$\begin{aligned}\delta_y^{k+1} &= y^{k+1} - (y')^{k+1} \\ &= y^k + \rho(x^{k+1} - z^{k+1}) - \left(y^k + \rho((x')^{k+1} - (z')^{k+1})\right) \\ &= \rho(\delta_x^{k+1} - \delta_z^{k+1}) \preccurlyeq \rho\delta_x^{k+1}\end{aligned} \tag{23}$$

The last $\preccurlyeq$ holds because $\delta_z^{k+1} \preccurlyeq \delta_x^{k+1}$, the subtraction by δ_z^{k+1} only pushes each element of δ_x^{k+1} towards zero. So we have below conclusions for sensitivities of x, z, y after epoch k:

$$\Delta_2^{k+1}(x) = \|\delta_x^{k+1}\|_2 \le \frac{2C}{n(1+\eta^{k+1}\rho)} \tag{24}$$

$$\Delta_2^{k+1}(z) = \|\delta_z^{k+1}\|_2 \le \|\delta_x^{k+1}\|_2 \le \frac{2C}{n(1+\eta^{k+1}\rho)} \tag{25}$$

$$\Delta_2^{k+1}(y) = \|\delta_y^{k+1}\|_2 \le \rho\|\delta_x^{k+1}\|_2 \le \frac{2\rho C}{n(1+\eta^{k+1}\rho)} \tag{26}$$

Theorem 3. *Algorithm 2 is (α, ϵ)-RDP.*

Proof. Let $\epsilon_{k+1,w}(\alpha) = \alpha(\Delta_2^{k+1}(w))^2/2\sigma^2$ for $w \in \{x, z, y\}$. By Lemma 1, each epoch is $(\alpha, \sum_{w\in\{x,z,y\}} \epsilon_{k+1,w}(\alpha))$-RDP, with respect to D. Since the algorithm has run T epochs, by Lemma 4, let $\epsilon = \sum_{k=1}^{T} \sum_{w\in\{x,z,y\}} \epsilon_{k,w}(\alpha))$, then Algorithm 2 is (α, ϵ)-RDP. □

5 EXPERIMENTAL RESULTS

In this section we will present our experimental results on both real and simulated datasets. We will first show performance of classification on two real datasets, then show performance of both classification and feature selection on a synthetic dataset.

5.1 ERM models

We perform our experiments on L_1 regularized logistic regression and huberized SVM. The objective function of logistic regression is in (9). For huberized SVM, the objection function is

$$F(x, D) := \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{huber}}(l_i x^T s_i) + \lambda\|x\|_1 \tag{27}$$

where

$$\ell_{\text{huber}}(z) := \begin{cases} 0 & \text{if } z > 1 + h \\ \frac{1}{4h}(1 + h - z)^2 & \text{if } |1 - z| \le h \\ 1 - z & \text{otherwise} \end{cases} \tag{28}$$

is the huberized hinge loss (we set $h = 0.5$ in all experiments).

5.2 Baselines

Many differentially private ERM algorithms cannot be applied to L_1 regularized classification, such as ObjPert [9], [21], OutPert [41], PVP and DVP [40], PSGD [37], and RSGD [10]. Therefore, we compare our proposed algorithms with these baselines: DP-SGD [1], DP-ADMM [35], ADMM-objP [20], and Non-Private approach.

DP-SGD performs stochastic gradient descent with Gaussian perturbation. (Although their paper proposed moment accountant approach to analyze the privacy leak, we use Lemma 2 to analyze as we do on ssADMM, since it gives tighter bound on ϵ.) For DP-SGD, when the algorithm requires taking gradient on $f(x^k, B_k) + \lambda\|x^k\|_1$, we use the proximal gradient technique

$$x^{k+1} \leftarrow \mathcal{S}_{\lambda\eta^k}[x^k - \eta^k \nabla f(x^k, B_k)] \tag{29}$$

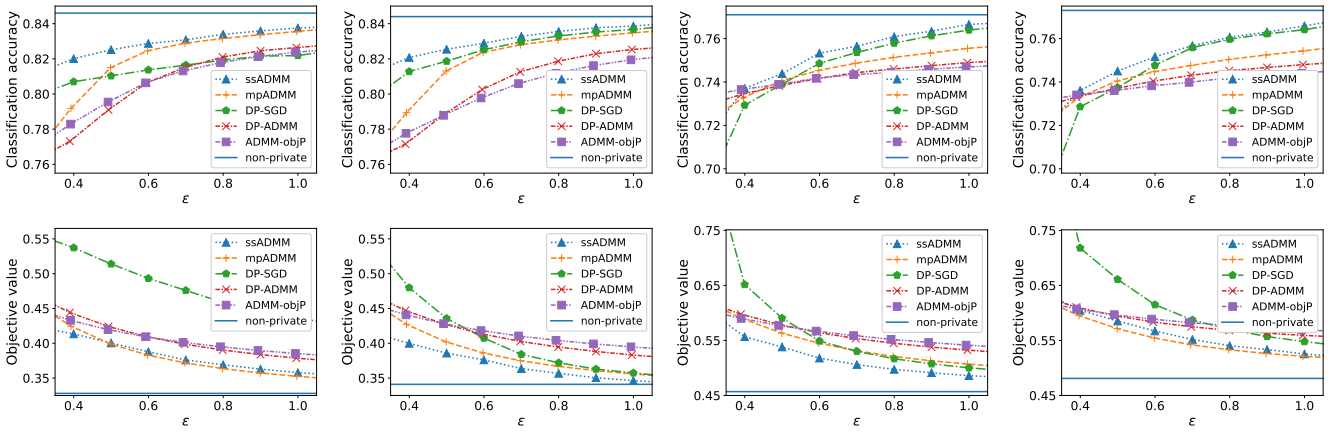
to update x^{k+1}, as suggested in [13] and [11]. DP-ADMM is a distributed learning version of ADMM, where each party transfers perturbed primal variables to the center, and the center draw a consensus of the parties then transfer primal and dual variable back to each party. ADMM-objP is an ADMM version of the objective perturbation algorithm. At each iteration, the trainer optimize a perturbed unregulated objective function, therefore although the algorithm satisfies pure ϵ-DP, in practice it is not really differentially private due to the objective function can only be approximately solved. According to their paper, we apply gradient descent enough times and assume the optimization problem is exactly solved at each iteration.

The DP-SVRG algorithm presented in [34] can also be applied on non-smooth regularizers, but we have implemented and found that, due to the extra privacy budget required to spent on perturbing the full gradient, with the high privacy range ($\epsilon \le 1$), if we choose a large σ^2, the perturbed full gradient cannot help as a control variant to fasten the training, but actually slows down the minimization of empirical loss; if we choose a small σ^2, the privacy budget accumulates too fast and exceed our range in a few iterations. Therefore we have dropped this algorithm in our comparisons.

5.3 Datasets and Pre-proessing

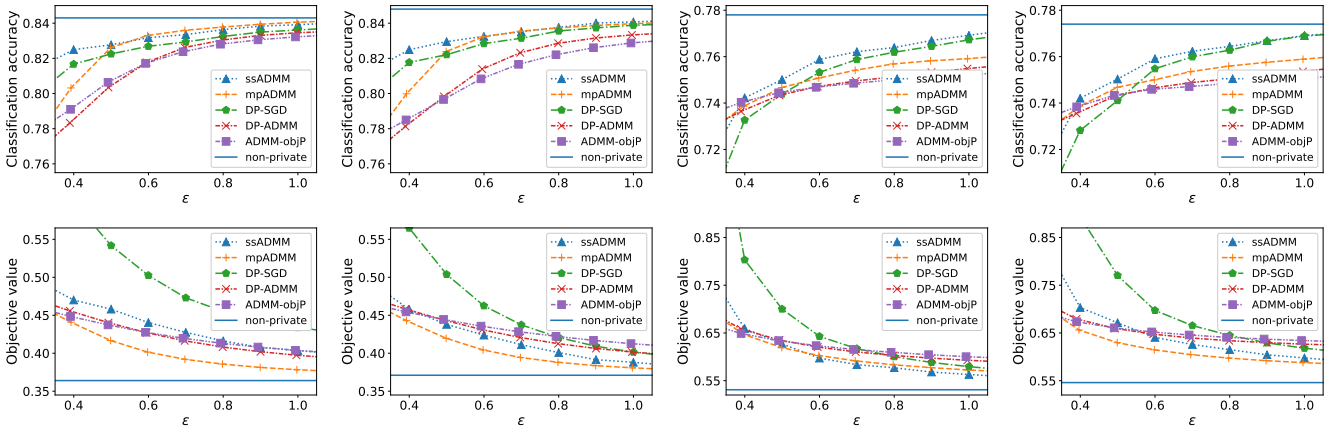
Two real datasets on human subjects were used in our study: (i) the Adult dataset [8] was generated from 1994 US Census, with $n = 48,842$, $p = 124$, and the frequency of the majority label is 0.761; (ii) the IPUMS-BR dataset [30] was extracted from IPUMS data, with $n = 38,000$, $p = 53$, and the frequency of the majority label is 0.507.

To test the performance on feature selection, we created a synthetic dataset with many irrelevant features, using similar strategy as in [35]. To be specific, we generate a 100-dimension data $s_i \sim \mathcal{N}(0_{100}, \Sigma)$ where $\Sigma_{i,j} = 0.5^{|i-j|}$. Let x be the true model, defined as $x_{1:10} = (0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5)$, $x_{11:20} = -x_{1:10}$, and $x_{21:100} = (0, ..., 0)$. For the label of each row l_i, we sample the Bernoulli distribution with $P(l_i = 1) = 1/(1+\exp(-x^T s_i + \iota))$, where $\iota \sim \mathcal{N}(0, 1)$ is a random noise. Therefore, to predict l_i, s_i contains 20 relevant features and 80 irrelevant features. We generate 40,000 samples to constitute one dataset, the frequency of the majority label is 0.500. We only perform logistic regression on simulated data, since it is usually used for attribute selection.



(a) Adult $\lambda = 0.0001$ **(b) Adult $\lambda = 0.001$** **(c) IPUMS-BR $\lambda = 0.0001$** **(d) IPUMS-BR $\lambda = 0.001$**

Figure 1: Logistic regression result by ϵ (Top: Classification accuracy; Bottom: Objective value)



(a) Adult $\lambda = 0.0001$ **(b) Adult $\lambda = 0.001$** **(c) IPUMS-BR $\lambda = 0.0001$** **(d) IPUMS-BR $\lambda = 0.001$**

Figure 2: Huberized SVM result by ϵ (Top: Classification accuracy; Bottom: Objective value)

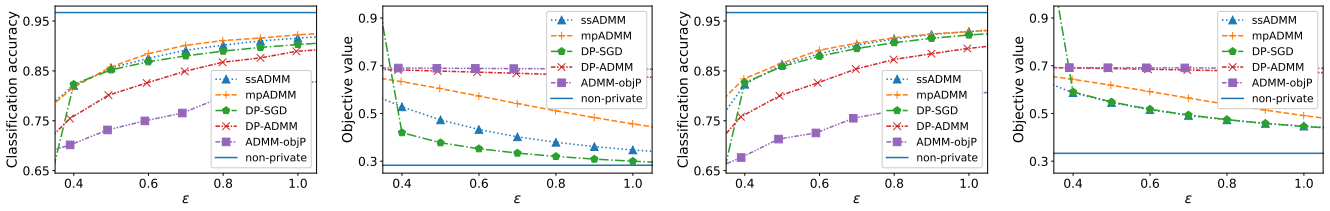
We did 10-fold cross validation on each experiment for each algorithm, and due to randomness from noisy injection, we repeat each fold 10 times and report average classification accuracy and objective value on testing data. For the simulated data, we generated 10 datasets using the simulation strategy, and report the average performance.

An intercept is added into each dataset. All numerical attributes are re-scaled into [0, 1] by Min-Max scalar. For the algorithms requiring feature vector to have bounded L_2 norm, we normalize to make $\|x_i\| \leq 1$ for $i = 1, ..., n$.

5.4 Parameter setting

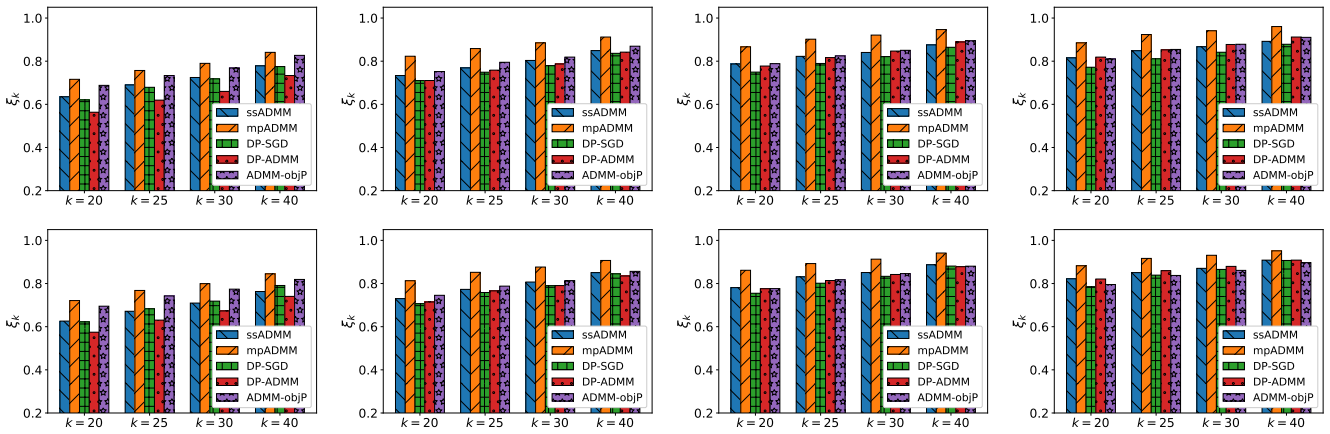
We keep $\delta = 10^{-8}$ for all experiments. For those algorithms satisfying RDP, we choose the best conversion to (ϵ, δ)-DP. In non-private settings, model users usually train a series models with different candidates of regularization coefficient λ, and select the one with highest testing performance. However, this process is data-dependent, therefore in private settings we cannot take a "best performing" coefficient for granted. Instead, we performed two group of experiments by two frequently using coefficients: low regularization with $\lambda = 0.0001$ and high regularization with $\lambda = 0.001$.

For ssADMM and DP-SGD, we set mini-batch size $m = \sqrt{n}$. We choose $\eta^k = \eta^0/h$ where h is the current expected epoch (we consider every n/m iterations as one expected epoch), since we find this schedule has the best performance for both algorithms, compare to a constant learning rate, or a decreasing one at a rate of $O(1/\sqrt{k})$. After tuning on the simulated data, we set penalty term $\rho = 0.25$ for ssADMM and $\rho = 0.5$ for mpADMM. For mpADMM, we use a constant learning rate η. For DP-ADMM, we assume there are 2 parties, each holding half of the data. (If there is only one party, DP-ADMM will reduce to DP-SGD with sampling ratio=1.) For ADMM-objP, at each iteration we optimize the perturbed objective function by full gradient descent running 20 epochs. Other parameters for DP-ADMM and ADMM-objP are set according to their paper.



(a) Accuracy $\lambda = 0.0001$ **(b) Objective value** $\lambda = 0.0001$ **(c) Accuracy** $\lambda = 0.001$ **(d) Objective value** $\lambda = 0.001$

Figure 3: Classification performance on simulated data



(a) $\epsilon = 0.4$ **(b)** $\epsilon = 0.6$ **(c)** $\epsilon = 0.8$ **(d)** $\epsilon = 1.0$

Figure 4: Attribute selection performance on simulated data (Top: $\lambda = 0.0001$**; Bottom:** $\lambda = 0.001$**)**

5.5 Classification Performance on Real Data

Figure 1 and Figure 2 plots the testing data accuracy (top) and objective values (bottom) of the algorithms trading off with privacy parameter ϵ, for L_1 regularized logistic regression and huberized SVM, respectively. We can see for classification accuracy, ssADMM outperforms other algorithms in most cases. This is in accordance with the experiment in [27] that sADMM outperforms proximal gradient in non-private setting. [2] also show that ADMM based algorithms are more robust to noisy data with outliers. Although DP-SGD has better classification accuracy than mpADMM in some cases, its objective value is usually higher. DP-ADMM and ADMM-objP can achieve high utility when ϵ gets high, but in our testing range of ϵ, they cannot perform as good as other algorithms. mpADMM performs better in adult dataset than in IPUMS-BR dataset, probably because Adult dataset is more sparse compare to IPUMS-BR, due to it is binary transferred through one-hot encoding. And that model perturbation are more robust to data with irrelevant attributes is in accordance with our observations on the simulated data.

5.6 Performance on Simulated Data

To measure the attribute selection performance, we test how many relevant attributes are selected by each algorithm for L_1 regularized logistic regression. Since the dataset is standardized, we can use the magnitude of the coefficient to rank the attributes, due to that noisy perturbation might cause the coefficients of irrelevant attributes slightly differ from zero.

We define a criterion ξ_k to measure the coverage of relevant attributes if top k attributes suggested by the algorithm were selected. For example, since we know there are 20 relevant attributes in the simulated data, if we select $k = 30$ attributes by magnitude of coefficient, 16 of them are the true relevant ones (i.e. among $x_1, ..., x_{20}$), then $\xi_{30} = 16/20 = 0.8$. This make sense because in real case, the number of attributes we choose to select from an attribute ranker depends on the budget we can spend to collect data. We test all algorithms for $k = 20$, 25, 30, and 40.

Figure 3 shows the classification performance of each algorithm on the simulated data. For non-private performance, we assume the true model is known. We can see that ssADMM, mpADMM, and DP-SGD have similar performance in classification. Figure 4 shows the performance of attribute selection. Although classification accuracy are close, we can see that mpADMM can detect more relevant attributes, especially in the lower ϵ range. ADMM-objP, which was originally proposed for feature selection, can outperform ssADMM and DP-SGD for feature selection in low ϵ while its classification accuracy is behind ssADMM and DP-SGD. However, ADMM-objP usually require much more epochs in training compare to the other algorithms. Therefore, if we know the data is sparse and the major goal is focused on attribute selection, mpADMM is more preferable.

6 CONCLUSIONS

We present two privatizations of stochastic ADMM under Rényi differential privacy. One algorithm combines gradient perturbation technique with privacy amplification result to reduce the total privacy loss throughout the execution. The other algorithm uses the output perturbation (with numerical computation of sensitivity) to privately release the solution at the end of each training epoch. These algorithms can be used to solve optimization problems with complex structural regularization that induces sparsity.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.

[2] Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. 2015. Auxiliary image regularization for deep cnns with noisy labels. *arXiv preprint arXiv:1511.07069* (2015).

[3] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE, 464–473.

[4] Yatao An Bian, Xiong Li, Yuncai Liu, and Ming-Hsuan Yang. 2019. Parallel Coordinate Descent Newton Method for Efficient L_1-Regularized Loss Minimization. *IEEE transactions on neural networks and learning systems* (2019).

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

[6] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[7] Stanley H Chan, Xiran Wang, and Omar A Elgendy. 2016. Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging* 3, 1 (2016), 84–98.

[8] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 27.

[9] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, Mar (2011), 1069–1109.

[10] Chen Chen, Jaewoo Lee, and Dan Kifer. 2019. Renyi Differentially Private ERM for Smooth Objectives. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2037–2046.

[11] Patrick L Combettes and Jean-Christophe Pesquet. 2011. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 185–212.

[12] Neil K Dhingra, Mihailo R Jovanović, and Zhi-Quan Luo. 2014. An ADMM algorithm for optimal sensor and actuator selection. In *53rd IEEE Conference on Decision and Control*. IEEE, 4039–4044.

[13] John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10, Dec (2009), 2899–2934.

[14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[16] Ernie Esser. 2009. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM report* 9 (2009), 31.

[17] Jianqing Fan and Runze Li. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association* 96, 456 (2001), 1348–1360.

[18] Daniel Gabay and Bertrand Mercier. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 2, 1 (1976), 17–40.

[19] Joshua Goodman. 2004. Exponential priors for maximum entropy models. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. 305–312.

[20] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. 2019. DP-ADMM: ADMM-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security* (2019).

[21] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. 2012. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*. 25–1.

[22] Antti Koskela and Antti Honkela. 2018. Learning rate adaptation for differentially private stochastic gradient descent. *arXiv preprint arXiv:1809.03832* (2018).

[23] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. 2006. Efficient l¯ 1 regularized logistic regression. In *AAAI*, Vol. 6. 401–408.

[24] Guangcan Liu, Qingshan Liu, and Ping Li. 2016. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE transactions on pattern analysis and machine intelligence* 39, 1 (2016), 47–60.

[25] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 263–275.

[26] Andrew Y Ng. 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 78.

[27] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. 2013. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*. 80–88.

[28] Mee Young Park and Trevor Hastie. 2007. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69, 4 (2007), 659–677.

[29] Trevor Park and George Casella. 2008. The bayesian lasso. *J. Amer. Statist. Assoc.* 103, 482 (2008), 681–686.

[30] Steven Ruggles, Katie Genadek, Ronald Goeken, Josiah Grover, and Matthew Sobek. 2015. Integrated public use microdata series: Version 6.0 [dataset]. *Minneapolis: University of Minnesota* 23 (2015), 56.

[31] Huachun Tan, Jianshuai Feng, Guangdong Feng, Wuhong Wang, and Yu-Jin Zhang. 2013. Traffic volume data outlier recovery via tensor model. *Mathematical Problems in Engineering* 2013 (2013).

[32] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.

[33] Ryan J Tibshirani, Jonathan Taylor, et al. 2011. The solution path of the generalized lasso. *The Annals of Statistics* 39, 3 (2011), 1335–1371.

[34] Di Wang, Minwei Ye, and Jinhui Xu. 2017. Differentially Private Empirical Risk Minimization Revisited: Faster and More General. In *Advances in Neural Information Processing Systems*. 2722–2731.

[35] Puyu Wang and Hai Zhang. 2019. Differential Privacy for Sparse Classification Learning. *arXiv preprint arXiv:1908.00780* (2019).

[36] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. 2018. Subsampled R\'enyi Differential Privacy and Analytical Moments Accountant. *arXiv preprint arXiv:1808.00087* (2018).

[37] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. 2017. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 1307–1322.

[38] Junfeng Yang and Xiaoming Yuan. 2013. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation* 82, 281 (2013), 301–329.

[39] Cun-Hui Zhang et al. 2010. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics* 38, 2 (2010), 894–942.

[40] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. 2017. Efficient private ERM for smooth objectives. *arXiv preprint arXiv:1703.09947* (2017).

[41] Tao Zhang and Quanyan Zhu. 2016. Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security* 12, 1 (2016), 172–187.

[42] Tao Zhang and Quanyan Zhu. 2017. Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security* 12, 1 (2017), 172–187.

[43] Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2018. Improving the privacy and accuracy of ADMM-based distributed algorithms. *arXiv preprint arXiv:1806.02246* (2018).

[44] Xiao Zhang, Lingxiao Wang, Yaodong Yu, and Quanquan Gu. 2018. A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *International conference on machine learning*. 5857–5866.

[45] Hui Zou. 2006. The adaptive lasso and its oracle properties. *Journal of the American statistical association* 101, 476 (2006), 1418–1429.

A PROOF OF THEOREM 2

The proof is done by applying similar technique for Theorem 1 in [27], plus considering the Gaussian noise term added. Define

$$u := \begin{pmatrix} x \\ z \end{pmatrix}, \overline{u}^k := \begin{pmatrix} \frac{1}{k}\sum_{i=1}^{k-1} x^i \\ \frac{1}{k}\sum_{i=1}^{k-1} z^i \end{pmatrix}, \theta(u) := f(x) + h(z),$$

and define

$$w := \begin{pmatrix} x \\ z \\ y \end{pmatrix}, \overline{w}^k := \begin{pmatrix} \frac{1}{k}\sum_{i=1}^{k-1} x^i \\ \frac{1}{k}\sum_{i=1}^{k-1} z^i \\ \frac{1}{k}\sum_{i=1}^{k-1} y^i \end{pmatrix}, F(w) := \begin{pmatrix} -y \\ y \\ x - z \end{pmatrix}$$

Denote $u^* := \begin{pmatrix} x^* \\ z^* \end{pmatrix}$ as the optimal solution, and $\delta_{k+1} := \nabla f(x^k, B_k) - \nabla f(x^k, D)$, $d_{\mathcal{X}} := \sup_{x_a, x_b \in \mathcal{X}} \|x_a - x_b\|$, $d_{y^*} := \|y^0 - y^*\|$.

Therefore, consider the expectation of $\theta(\overline{u}^t) - \theta(u^*)$ after t iterations,

$$\begin{aligned}
&\mathbb{E}\Big[\theta(\overline{u}^t) - \theta(u^*) + (\overline{w}^t - w^*)^T F(\overline{w}^t)\Big] \\
=&\mathbb{E}\Big[\theta(\overline{u}^t) - \theta(u^*) + (\overline{x}^t - x^*)^T(-\overline{y}^t) + (\overline{z}^t - z^*)^T(\overline{y}^t) \\
&+ (\overline{y} - y)^T(\overline{x}^t - \overline{z}^t)\Big] \\
\le&\mathbb{E}\Big[\frac{1}{t}\sum_{k=0}^{t-1}\Big[\frac{\eta^k}{2}\|\nabla f(x^k, B_k) + \gamma^k\|^2 + \frac{1}{2\eta^k}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2) \\
&+ \langle \delta_{k+1}, x^* - x^k\rangle\Big] + \frac{1}{t}\Big(\frac{\rho}{2}\|x^* - z^0\|^2 + \frac{1}{2\rho}\|y - y^0\|^2\Big)\Big] \\
\le&\mathbb{E}\Big[\frac{1}{t}\sum_{k=0}^{t-1}\Big[\frac{\eta^k(C^2 + p\sigma^2)}{2} + \langle \delta_{k+1}, x^* - x^k\rangle\Big] \\
&+ \frac{1}{t}\Big(\frac{d_{\mathcal{X}}^2}{2\eta^{t-1}} + \frac{\rho}{2}d_{y^*}^2 + \frac{1}{2\rho}\|y - y^0\|^2\Big)\Big] \\
=&\mathbb{E}\Big[\frac{1}{t}\sum_{k=0}^{t-1}\Big[\frac{\eta^k(C^2 + p\sigma^2)}{2}\Big] + \frac{1}{t}\Big(\frac{d_{\mathcal{X}}^2}{2\eta^{t-1}} + \frac{\rho}{2}d_{y^*}^2 + \frac{1}{2\rho}\|y - y^0\|^2\Big)\Big]
\end{aligned} \tag{30}$$

while the first inequality holds by applying an expected version of Lemma 2 in [27], note that since noisy perturbation $\gamma \sim \mathcal{N}(0, \sigma^2\mathbf{I}_p)$, $\mathbb{E}[\nabla f(x^k, B_k) + \gamma] = \nabla f(x^k, B_k)$, and $\mathbb{E}[\|\nabla f(x^k, B_k) + \gamma^k\|^2] \le \mathbb{E}[\|\nabla f(x^k, B_k)\|^2] + \mathbb{E}[\|\gamma\|^2] + 2\mathbb{E}[\|\nabla f(x^k, B_k)\|]\mathbb{E}[\gamma] \le C^2 + p\sigma^2$. The last equality holds because we assume x^k is independent of B_k (which was used to calculate x^{k+1}) is independent of x^k, hence $\mathbb{E}_{B_k|B_{[0:k-1]}}\langle \delta_{k+1}, x^* - x^k\rangle = 0$.

The above holds for all dual variable y, hence it holds for y in a ball $\mathcal{B}_0 = \{y : \|y\|_2 \le \beta\}$. According to (33) in [27],

$$\max_{y \in \mathcal{B}_0}\{\theta(\overline{u}^t) - \theta(u^*) + (\overline{w}^t - w^*)^T F(\overline{w}^t)\} = \theta(\overline{u}^t) - \theta(u^*) + \beta\|\overline{x}_t - \overline{z}_t\| \tag{31}$$

Therefore, continue on (30), we can have

$$\begin{aligned}
&\mathbb{E}\Big[\theta(\overline{u}^t) - \theta(u^*) + \beta\|\overline{x}_t - \overline{z}_t\|\Big] \\
\le&\mathbb{E}\Big[\frac{1}{t}\sum_{k=0}^{t-1}\Big[\frac{\eta^k(C^2 + p\sigma^2)}{2}\Big] + \frac{1}{t}\Big(\frac{d_{\mathcal{X}}^2}{2\eta^{t-1}} + \frac{\rho}{2}d_{y^*}^2 + \frac{1}{2\rho}\|y - y^0\|^2\Big)\Big] \\
\le&\mathbb{E}\Big[\frac{1}{t}\sum_{k=0}^{t-1}\Big[\frac{\eta^k(C^2 + p\sigma^2)}{2}\Big] + \frac{1}{t}\Big(\frac{d_{\mathcal{X}}^2}{2\eta^{t-1}} + \frac{\rho}{2}d_{y^*}^2\Big)\Big] \\
&+ \mathbb{E}\Big[\max_{y \in \mathcal{B}_0}\Big\{\frac{1}{2\rho t}\|y - y_0\|^2\Big\}\Big] \\
\le&\frac{1}{t}\Big(\frac{C^2 + p\sigma^2}{2}\sum_{k=1}^{t}\eta^k + \frac{d_{\mathcal{X}}^2}{2\eta^{t-1}}\Big) + \frac{\rho d_{y^*}^2}{2t} + \frac{\beta^2}{2\rho t}
\end{aligned} \tag{32}$$

So if we choose $\eta^k = \frac{d_{\mathcal{X}}}{\sqrt{2(C^2 + p\sigma^2)k}} = O(1/\sqrt{k})$, $\mathbb{E}\big[\theta(\overline{u}^t) - \theta(u^*) + \beta\|\overline{x}_t - \overline{z}_t\|\big] \le \frac{d_{\mathcal{X}}\sqrt{2(C^2 + p\sigma^2)}}{\sqrt{t}} + \frac{\rho d_{y^*}^2}{2t} + \frac{\beta^2}{2\rho t} = O(1/\sqrt{t})$.