Federated learning with convex global and local constraints

Chuan He *he000233@umn.edu*
Department of Computer Science and Engineering
University of Minnesota

Le Peng *peng0347@umn.edu*
Department of Computer Science and Engineering
University of Minnesota

Ju Sun *jusun@umn.edu*
Department of Computer Science and Engineering
University of Minnesota

Abstract

This paper considers federated learning (FL) with constraints, where the central server and all local clients collectively minimize a sum of convex local objective functions subject to convex global and local constraints. To train the model without moving local data from clients to the central server, we propose an FL framework in which each local client performs multiple updates using the local objective and local constraint, while the central server handles the global constraint and performs aggregation based on the updated local models. In particular, we develop a proximal augmented Lagrangian (AL) based algorithm for FL with convex global and local constraints. The subproblems arising in this algorithm are solved by an inexact alternating direction method of multipliers (ADMM) in a federated fashion. Under a local Lipschitz condition and mild assumptions, we establish the worst-case complexity bounds of the proposed algorithm for finding an approximate KKT solution. To the best of our knowledge, this work proposes the first algorithm for FL with global and local constraints. Our numerical experiments demonstrate the practical advantages of our algorithm in performing Neyman-Pearson classification and enhancing model fairness in the context of FL.

1 Introduction

Federated learning (FL) has emerged as a prominent distributed machine learning paradigm, finding extensive application across diverse domains. FL aims to train a learning model with good performance for all local data while ensuring the privacy of local clients by minimizing the disclosure of sensitive local information. A common FL strategy is that each local client independently performs certain iterations using local data to update its local parameters. Subsequently, the central server collects the updated local parameters (and sometimes derivatives of local models) from local clients and performs an aggregation to compute a new global parameter. This iterative FL strategy, with multiple local updates, saves the communication efforts between the central server and local clients. Also, as the raw data stored on the local clients is never shared directly with the central server, this FL strategy ensures the privacy of sensitive local information held by each client.

On the other hand, recent years have seen a proliferation of deep learning tasks that are framed as constrained optimization problems. These constraints typically encode prior knowledge and essential properties pivotal to the learning tasks. Particularly, the learning tasks that fall within constrained optimization span various areas, including robustness evaluation (Goodfellow et al., 2014), fairness-aware learning (Agarwal et al., 2018), addressing label imbalance (Saito & Rehmsmeier, 2015), neural architecture search (Zoph et al., 2018), topology optimization (Christensen & Klarbring, 2008), knowledge-aware machine learning (McClenny

& Braga-Neto, 2020), etc. We will now describe two specific examples that motivate our research in this paper.

Neyman-Pearson classification: Consider a binary classification problem where the primary concern is the risk of misclassifying one specific class more than the other, as often occurs in medical diagnosis. To address this problem, the Neyman-Pearson classification model is proposed as follows (e.g., see (Tong et al., 2016)):

$$\min_w \frac{1}{n_0}\sum_{i=1}^{n_0}\varphi(w, z_{i,0}) \quad \text{s. t.} \quad \frac{1}{n_1}\sum_{i=1}^{n_1}\varphi(w, z_{i,1}) \le r,$$

where w is the weight parameter, φ is a loss function, $\{z_{i,0}\}_{i=1}^{n_0}$ and $\{z_{i,1}\}_{i=1}^{n_1}$ are the training data from two separate classes 0 and 1, respectively, and $r > 0$ controls the training error for class 1. The Neyman-Pearson classification model is introduced as a widely studied statistical learning model that has been commonly used for handling asymmetric training error priorities (Scott, 2007; Rigollet & Tong, 2011; Tong et al., 2018).

Fairness-aware learning: Incorporating fairness constraints into the training of machine learning models is widely recognized as an important approach to ensure the models' trustworthiness (Agarwal et al., 2018; Celis et al., 2019; Mehrabi et al., 2021). Training a model with fairness constraints is usually formulated as follows:

$$\min_w \frac{1}{n}\sum_{i=1}^{n}\varphi(w, z_i) \quad \text{s. t.} \quad \min_{1\le i\le k} p_j(w, \{z_i\}_{i=1}^n) \ge \rho \max_{1\le j\le k} p_j(w, \{z_i\}_{i=1}^n),$$

where w is the weight parameter, φ is a loss function, p_j, $1 \le j \le k$, are performance metrics, $\rho \in [0, 1]$ is the targeted fairness level, and $\{z_i\}$ is the training data set.

In the FL literature, many efforts have been devoted to mitigating class imbalance (Shen et al., 2021) and improving model fairness (Du et al., 2021; Chu et al., 2021; Gálvez et al., 2021) through the application of constrained optimization models. Nevertheless, these algorithms are often specialized to particular use cases and suffer from a lack of computational complexity guarantees for achieving consensus, optimality, and feasibility in their solutions. The main goals of this paper are twofold: (1) to investigate a general optimization problem with convex constraints in an FL setting; (2) to develop an FL algorithm with complexity guarantees for finding its solution. Specifically, we consider the following general optimization formulation of FL problems with convex global and local constraints [1]:

$$\min_w \left\{\sum_{i=1}^{n} f_i(w) + h(w)\right\} \quad \text{s. t.} \quad \underbrace{c_0(w) \le 0}_{\text{global constraint}}, \quad \underbrace{c_i(w) \le 0, \;\; 1 \le i \le n}_{\text{local constraints}}, \tag{1}$$

where the functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $1 \le i \le n$, and the mappings $c_i : \mathbb{R}^d \to \mathbb{R}^{m_i}$, $0 \le i \le n$, are convex and continuously differentiable, and $h : \mathbb{R}^d \to (-\infty, \infty]$ is a simple closed convex function. The convexity assumption is necessary for our initial theoretical exploration of FL with constraints. We also explore the applicability of our FL algorithm to classification tasks with nonconvex fairness constraints in Section 5.3.

The global constraint in (1), namely $c_0(w) \le 0$, refers to a constraint that can be directly accessed by the central server. The local constraints in (1), namely $c_i(w) \le 0$ for $1 \le i \le n$, refer to constraints that depend on the local data that clients used for training the model. Throughout this paper, we assume that

for each $1 \le i \le n$, the local objective f_i and local constraint c_i are handled solely by the local client i, and the central server has access to the global constraint c_0.

This assumption generalizes the one commonly imposed for unconstrained FL, where each local objective function is solely handled by one local client. In addition, our model (1) is tailored for scenarios where local clients have enough amount of reliable data points to establish their own local constraints. Meanwhile, to enhance generalization property, the central server forms a global constraint by incorporating certain public or external data points. Additionally, it is noteworthy that solving an FL problem with n local constraints,

[1] Distributed optimization with global and local constraints has been studied before in the literature (e.g., see Zhu & Martínez (2011); Nedic et al. (2010)).

such as $c_i(w) \leq r$, $1 \leq i \leq n$, can yield a feasible solution for the coupled constraints involving data points from all local clients, such as $1/n \sum_{i=1}^{n} c_i(w) \leq r$.

Due to the sophistication of the constraints in problem (1), existing FL algorithms face challenges when attempting to apply or extend them directly to solve (1). For example, a natural approach for this problem is to adopt existing FL algorithms to minimize the quadratic penalty function associated with (1). However, to ensure global convergence to a solution for (1), it is often necessary to minimize a sequence of penalty functions with sufficiently large penalty parameters, rendering the solution process highly unstable and inefficient (e.g., see Nocedal & Wright (2006)). Moreover, in the centralized setting, Lagrangian methods are frequently employed for constrained optimization in deep learning (e.g., see Cotter et al. (2019)). However, these methods often require careful tuning of initial multipliers and step-sizes for the multipliers. In contrast, we propose an FL algorithm grounded in the proximal augmented Lagrangian (AL) method. This algorithm efficiently and robustly finds an (ϵ_1, ϵ_2)-KKT solution of (1) (see Definition 1 for its definition). At each iteration of this algorithm, a fixed penalty parameter is employed, and an approximate solution to a proximal AL subproblem associated with (1) is computed by an inexact alternating direction method of multipliers (ADMM) in a federated manner. We study the worst-case complexity of this algorithm under a *locally Lipschitz* assumption on ∇f_i, $1 \leq i \leq n$, and ∇c_i, $0 \leq i \leq n$. Our main contributions are highlighted below.

- We propose a proximal AL based FL algorithm (Algorithm 1) for seeking an approximate KKT solution of problem (1). The proposed algorithm naturally generalizes the current FL algorithms designed for unconstrained finite-sum optimization (see problem (2) below). Under a *locally Lipschitz* condition and mild assumptions, we establish the worst-case complexity for finding an (ϵ_1, ϵ_2)-KKT solution of problem (1). To the best of our knowledge, the proposed algorithm is the first one for FL with global and local constraints, and its complexity results are entirely new in the literature.
- We conduct numerical experiments by comparing our proximal AL based FL algorithm with existing FL algorithms on several real-world constrained learning tasks including binary classification with specified recall and classification with nonconvex fairness constraints. Our numerical results validate that our FL algorithm can achieve solution quality comparable to the centralized algorithm.
- We propose an inexact ADMM based FL algorithm (Algorithm 2) for solving an unconstrained finite-sum optimization problem (see problem (13) below). Equipped with a newly introduced verifiable termination criterion, Algorithm 2 serves as a subproblem solver for Algorithm 1. We establish a global linear convergence rate for this algorithm under the assumptions of strongly convex local objectives and *locally* Lipschitz continuous gradients.

1.1 Related works

FL algorithms for unconstrained optimization: Federated learning has emerged as a cornerstone technique for privacy-preserved distributed learning since Google proposed the seminal work (McMahan et al., 2017). Unlike traditional centralized learning methods, FL enables the training of models with distributed edge clients, ranging from small mobile devices like phones (Mills et al., 2019) to large data providers such as hospitals and banks (Long et al., 2020). This inherent property of privacy preservation aligns seamlessly with the principles upheld by various critical domains, including healthcare (Rieke et al., 2020; Peng et al., 2023a;b), finance (Long et al., 2020), IoT (Mills et al., 2019), and transportation Liu et al. (2020), where safeguarding data privacy is essential.

FedAvg, introduced by McMahan et al. (2017), is the first and also the most widely applied FL algorithm. It was proposed for solving the unconstrained finite-sum optimization problem:

$$\min_{w} f(w) = \sum_{i=1}^{n} f_i(w). \tag{2}$$

Since then, many variants have been proposed to tackle various practical issues, such as data heterogeneity (Karimireddy et al., 2020; Li et al., 2021c; Zhang et al., 2021), system heterogeneity (Li et al., 2020; Wang et al., 2020; Gong et al., 2022), fairness (Li et al., 2021b), efficiency (Sattler et al., 2019; Konečnỳ

et al., 2016), and incentives (Travadi et al., 2023). For example, Li et al. (2020) proposed FedProx by adding a proximal term in the local objective to handle clients with different computation capabilities. Karimireddy et al. (2020) proposed Scallfold to address the issue of date heterogeneity where local data is non-independent and identically distributed (non-iid). Additionally, ADMM based FL algorithms have been proposed in Acar et al. (2021); Gong et al. (2022); Zhang et al. (2021); Wang et al. (2022); Zhou & Li (2023), and these methods have been shown to be inherently resilient to heterogeneity. Reddi et al. (2020) extended FedAvg by introducing adaptive optimizers for server aggregation, significantly reducing communication costs and improving FL scalability. Li et al. (2021b) proposed Ditto, a personalized FL framework that demonstrates improved client fairness and robustness. More variants of FL algorithms and their applications can be found in the survey (Li et al., 2021a). Despite the numerous FL algorithms proposed previously, they primarily focus on unconstrained FL problems, leaving a gap between constrained optimization and FL.

Centralized algorithms for constrained optimization: Recent years have witnessed fruitful algorithmic developments for constrained optimization in the centralized setting. In particular, there has been a rich literature on inexact AL methods for solving convex constrained optimization problems (e.g., see Aybat & Iyengar (2013); Necoara et al. (2019); Patrascu et al. (2017); Xu (2021); Lan & Monteiro (2016); Lu & Zhou (2023); Lu & Mei (2023)). In addition, AL methods and variants have also been extended to solve nonconvex constrained optimization problems (e.g., see Hong et al. (2017); Grapiglia & Yuan (2021); Birgin & Martínez (2020); Kong et al. (2023); Li et al. (2021d); He et al. (2023a;b); Lu (2022)). Moreover, sequential quadratic programming methods (Boggs & Tolle, 1995; Curtis & Overton, 2012), trust-region methods (Byrd et al., 1987; Powell & Yuan, 1991), interior point method (Wächter & Biegler, 2006), and extra-point method Huang et al. (2022) have also been proposed for solving constrained optimization problems. Furthermore, there have been many recent works on algorithms for finding second-order stationary points of nonconvex constrained optimization problems (e.g., see He & Lu (2023); He et al. (2023a;b); Lu et al. (2020); Xie & Wright (2021); Mokhtari et al. (2018); Haeser et al. (2019); O'Neill & Wright (2021); Cartis et al. (2012; 2013; 2015); Agarwal et al. (2021); Nouiehed et al. (2018)).

Distributed algorithms for constrained optimization: In another line of research, many algorithms have been developed for distributed optimization with global and local constraints. An early work Nedic et al. (2010) introduced a distributed projected subgradient algorithm for distributed optimization with local constraints. This work has been extended to handle scenarios involving time-varying directed graphs in Lin et al. (2016); Wang et al. (2017). Yet, these methods require each node to compute a projection on the local constraint set, which is applicable only to relatively simple constraints. To address more complicated constraints, distributed primal-dual algorithms were developed in Aybat & Hamedani (2016; 2019) for distributed convex optimization with conic constraints. In addition, primal-dual projected subgradient algorithms Zhu & Martínez (2011); Yuan et al. (2011) have been developed for distributed optimization with global and local constraints. For an overview of algorithmic developments in distributed optimization with constraints, we refer to Yang et al. (2019). We emphasize that the existing algorithms for constrained distributed optimization do not follow the common FL framework where clients perform multiple local updates before aggregating the global model. The algorithm development in this paper follows a distinct trajectory compared to them.

FL algorithms for constrained learning problems: Some studies have combined constrained optimization techniques with FL algorithms to tackle complex learning tasks, such as addressing label imbalances and promoting model fairness. For example, Shen et al. (2021) proposes an FL algorithm aimed at handling class imbalances, using primal-dual updates through the Lagrangian function. Similarly, Du et al. (2021); Chu et al. (2021) propose FL algorithms designed for fairness-constrained learning, also incorporating primal-dual updates using the Lagrangian function. In addition, Gálvez et al. (2021) proposes an FL algorithm for fairness-constrained learning, implementing primal-dual updates with the augmented Lagrangian function. Nonetheless, these studies are tailored to specific applications and do not establish convergence guarantees regarding constraint feasibility, stationarity, or consensus.

In contrast with the aforementioned algorithms, this paper focuses on FL problems with convex global and local constraints. To the best of our knowledge, this work provides the first exploration in such a setting.

2 Notation and preliminaries

Throughout this paper, we let $\mathbb{R}^d$ and $\mathbb{R}^d_+$ denote the d-dimensional Euclidean space and nonnegative orthant, respectively. Let $\langle\cdot,\cdot\rangle$ denote the standard inner product. Let $\|\cdot\|$ stand for the Euclidean norm of a vector or the spectral norm of a matrix, and $\|\cdot\|_\infty$ stand for the ℓ_∞-norm of a vector. Let $[v]_+$ denote the nonnegative part of a vector v. In addition, $\widetilde{\mathcal{O}}(\cdot)$ represents $\mathcal{O}(\cdot)$ with logarithmic terms omitted.

Given a closed convex function $h:\mathbb{R}^d\to(-\infty,\infty]$, ∂h and $\mathrm{dom}(h)$ denote the subdifferential and domain of h, respectively. The proximal operator associated with h is denoted by prox_h, that is, $\mathrm{prox}_h(u)=\arg\min_w\{\|w-u\|^2/2+h(w)\}$ for all $u\in\mathbb{R}^d$. Given a continuously differentiable mapping $\phi:\mathbb{R}^d\to\mathbb{R}^p$, let the transpose of its Jacobian be denoted as $\nabla\phi(w)=[\nabla\phi_1(w)\ \cdots\ \nabla\phi_p(w)]\in\mathbb{R}^{d\times p}$. We say that $\nabla\phi$ is L-Lipschitz continuous on a set Ω for some $L>0$ if $\|\nabla\phi(u)-\nabla\phi(v)\|\le L\|u-v\|$ for all $u,v\in\Omega$. In addition, we say that $\nabla\phi$ is locally Lipschitz continuous on Ω if for any $w\in\Omega$, there exist $L_w>0$ and an open set $\mathcal{U}_w$ containing w such that $\nabla\phi$ is L_w-Lipschitz continuous on $\mathcal{U}_w$.

Given a nonempty closed convex set $\mathcal{C}\subseteq\mathbb{R}^d$, let $\mathrm{dist}(u,\mathcal{C})$ and $\mathrm{dist}_\infty(u,\mathcal{C})$ stand for the Euclidean distance and the Chebyshev distance from u to $\mathcal{C}$, respectively. That is, $\mathrm{dist}(u,\mathcal{C})=\min_{v\in\mathcal{C}}\|u-v\|$ and $\mathrm{dist}_\infty(u,\mathcal{C})=\min_{v\in\mathcal{C}}\|u-v\|_\infty$. The normal cone of $\mathcal{C}$ at $u\in\mathcal{C}$ is denoted by $\mathcal{N}_{\mathcal{C}}(u)$. The Minkowski sum of two sets of vectors $\mathcal{B}$ and $\mathcal{C}$ in Euclidean space is denoted as $\mathcal{B}+\mathcal{C}=\{b+c:b\in\mathcal{B},c\in\mathcal{C}\}$.

For ease of presentation, we let $m=\sum_{i=0}^n m_i$ and adopt the following notations throughout this paper:

$$f(w)=\sum_{i=1}^n f_i(w),\qquad c(w)=[c_0(w)^T\ \cdots\ c_n(w)^T]^T,\qquad \mu=[\mu_0^T\ \cdots\ \mu_n^T]^T. \tag{3}$$

We make the following assumption on problem (1) throughout this paper.

Assumption 1. *The strong duality holds for problem (1) and its dual problem*

$$\sup_{\mu\ge0}\inf_w\left\{f(w)+h(w)+\langle\mu,c(w)\rangle\right\}. \tag{4}$$

That is, both problems have optimal solutions, and moreover, their optimal values coincide.

Under Assumption 1, it is known that $(w,\mu)\in\mathrm{dom}(h)\times\mathbb{R}^m_+$ is a pair of optimal solutions of (1) and (4) if and only if it satisfies (e.g., see Lu & Zhou (2023))

$$0\in\begin{pmatrix}\nabla f(w)+\partial h(w)+\nabla c(w)\mu\\ c(w)-\mathcal{N}_{\mathbb{R}^m_+}(\mu)\end{pmatrix}. \tag{5}$$

In general, it is hard to find an exact optimal solution of (1) and (4). Therefore, we are instead interested in seeking an approximate KKT solution of problems (1) and (4) defined as follows.

Definition 1. *Given any $\epsilon_1,\epsilon_2>0$, we say $(w,\mu)\in\mathrm{dom}(h)\times\mathbb{R}^m_+$ is an (ϵ_1,ϵ_2)-KKT solution of problems (1) and (4) if $\mathrm{dist}_\infty\left(0,\nabla f(w)+\partial h(w)+\nabla c(w)\mu\right)\le\epsilon_1$ and $\mathrm{dist}_\infty(c(w),\mathcal{N}_{\mathbb{R}^m_+}(\mu))\le\epsilon_2$.*

This definition is consistent with the ϵ-KKT solution considered in Lu & Zhou (2023) except that Definition 1 uses the Chebyshev distance rather than the Euclidean distance, and two different tolerances ϵ_1,ϵ_2 are used for measuring stationarity and feasibility violation, respectively.

3 A proximal AL based FL algorithm for solving (1)

In this section we propose a proximal AL based FL algorithm for solving (1). Specifically, we describe this algorithm in Section 3.1, and then analyze its complexity results in Section 3.2.

We first make the following assumptions throughout this section.

Assumption 2. (a) *The proximal operator for h can be exactly evaluated.*

(b) *The functions f_i, $1\le i\le n$, and mappings c_i, $0\le i\le n$, are continuously differentiable, and ∇f_i, $1\le i\le n$, and ∇c_i, $0\le i\le n$, are locally Lipschitz continuous on $\mathbb{R}^d$.*

Assumption 1(b) holds if ∇f_i, $1 \le i \le n$, and ∇c_i, $0 \le i \le n$, are globally Lipschitz continuous on $\mathbb{R}^d$. This assumption holds for a broader class of functions. For example, the gradient of the quadratic penalty term associated with (1), namely $\|c(\cdot)\|^2$, is merely locally Lipschitz continuous on $\mathbb{R}^d$ even if ∇c is globally Lipschitz continuous on $\mathbb{R}^d$ (see Remark 3.1). Additionally, the gradient of a convex high-degree polynomial function is merely locally Lipschitz continuous on $\mathbb{R}^d$, but not globally Lipschitz continuous on $\mathbb{R}^d$.

3.1 Algorithm description

In this subsection we describe the proximal AL based FL algorithm (Algorithm 1) for finding an (ϵ_1, ϵ_2)-KKT solution of problem (1) for any prescribed $\epsilon_1, \epsilon_2 \in (0, 1)$. This algorithm follows a similar framework to a centralized proximal AL method (see Appendix C). At each iteration, it applies an inexact ADMM (see Algorithm 2 below) to find an approximate solution w^{k+1} to the proximal AL subproblem associated with problem (1):

$$\min_w \left\{ \ell_k(w) := \sum_{i=1}^n f_i(w) + h(w) + \frac{1}{2\beta} \sum_{i=0}^n \left(\|[\mu_i^k + \beta c_i(w)]_+\|^2 - \|\mu_i^k\|^2 \right) + \frac{1}{2\beta}\|w - w^k\|^2 \right\}. \tag{6}$$

Then the multiplier estimates are updated according to the classical scheme:

$$\mu_i^{k+1} = [\mu_i^k + \beta c_i(w^{k+1})]_+, \quad 0 \le i \le n.$$

Algorithm 1 A proximal AL based FL algorithm for solving problem (1)

Input: tolerances $\epsilon_1, \epsilon_2 \in (0, 1)$, $w^0 \in \mathrm{dom}(h)$, $\mu_i^0 \ge 0$ for $0 \le i \le n$, $\bar{s} > 0$, and $\beta > 0$.

for $k = 0, 1, 2, \ldots$ **do**

Set $\tau_k = \bar{s}/(k+1)^2$.

Call Algorithm 2 (see Section 4 below) with $(\tau, \tilde{w}^0) = (\tau_k, w^k)$ to find an approximate solution w^{k+1} to (9) in a federated manner such that

$$\mathrm{dist}_\infty(0, \partial \ell_k(w^{k+1})) \le \tau_k. \tag{7}$$

Server update: The central server updates $\mu_0^{k+1} = [\mu_0^k + \beta c_0(w^{k+1})]_+$.
Communication (broadcast): Each local client i, $1 \le i \le n$, receives w^{k+1} from the central server.
Client update (local): Each local client i, $1 \le i \le n$, updates $\mu_i^{k+1} = [\mu_i^k + \beta c_i(w^{k+1})]_+$.
Communication: Each local client i, $1 \le i \le n$, sends $\|\mu_i^{k+1} - \mu_i^k\|_\infty$ to the central server.
Termination (server side): Output (w^{k+1}, μ^{k+1}) and terminate the algorithm if

$$\|w^{k+1} - w^k\|_\infty + \beta\tau_k \le \beta\epsilon_1, \qquad \max_{0 \le i \le n} \{\|\mu_i^{k+1} - \mu_i^k\|_\infty\} \le \beta\epsilon_2. \tag{8}$$

end for

Notice that the subproblem (6) can be rewritten as

$$\min_w \left\{ \ell_k(w) = \sum_{i=0}^n P_{i,k}(w) + h(w) \right\}, \tag{9}$$

where $P_{i,k}$, $0 \le i \le n$, are defined as

$$P_{0,k}(w) := \frac{1}{2\beta} \left(\|[\mu_0^k + \beta c_0(w)]_+\|^2 - \|\mu_0^k\|^2 \right) + \frac{1}{2(n+1)\beta}\|w - w^k\|^2, \tag{10}$$

$$P_{i,k}(w) := f_i(w) + \frac{1}{2\beta} \left(\|[\mu_i^k + \beta c_i(w)]_+\|^2 - \|\mu_i^k\|^2 \right) + \frac{1}{2(n+1)\beta}\|w - w^k\|^2, \quad \forall 1 \le i \le n. \tag{11}$$

When Algorithm 2 (see Section 4) is applied to solve problem (9), the local merit function $P_{i,k}$, constructed from the local objective f_i and local constraint c_i, is handled by the respective local client i, while the merit

function $P_{0,k}$ is handled by the central server. Hence, Algorithm 2 is well-suited for the FL framework that the local objective f_i and local constraint c_i are handled by the local client i, and the central server performs aggregation and handles the global constraint c_0.

In addition, the following lemma shows that $\nabla P_{i,k}$, $0 \le i \le n$, are locally Lipschitz continuous on $\mathbb{R}^d$. Its proof is deferred to Appendix B.1.

Lemma 3.1 (**Local Lipschitz continuity of $\nabla P_{i,k}$**). *Suppose that Assumption 1 holds. Then the gradients $\nabla P_{i,k}$, $0 \le i \le n$, are locally Lipschitz continuous on $\mathbb{R}^d$.*

Remark 3.1. *It is worth noting that $\nabla P_{i,k}$, $0 \le i \le n$, are typically not globally Lipschitz continuous on $\mathbb{R}^d$ even if ∇f_i, $1 \le i \le n$, and ∇c_i, $0 \le i \le n$, are globally Lipschitz continuous on $\mathbb{R}^d$. For example, consider the case where $c_0(w) = \|w\|^2 - 1$. By (10), one has that*

$$\nabla P_{0,k}(w) = 2[\mu_0^k + \beta(\|w\|^2 - 1)]_+ w + \frac{1}{(n+1)\beta}(w - w^k).$$

In this case, it is not hard to verify that ∇c_0 is globally Lipschitz continuous on $\mathbb{R}^d$, but $\nabla P_{0,k}$ is not.

For ease of later reference, we refer to the update from w^k to w^{k+1} as one outer iteration of Algorithm 1, and call one iteration of Algorithm 2 for solving (6) one inner iteration of Algorithm 1. In the rest of this section, we study the following measures of complexity for Algorithm 1.

- *Outer iteration complexity*, which measures the number of outer iterations of Algorithm 1;
- *Total inner iteration complexity*, which measures the total number of iterations of Algorithm 2 that are performed in Algorithm 1.

The following theorem concerns the output of Algorithm 1, whose proof is deferred to Appendix B.2.

Theorem 3.1 (**Output of Algorithm 1**). *If Algorithm 1 successfully terminates, its output (w^{k+1}, μ^{k+1}) is an (ϵ_1, ϵ_2)-KKT solution of problem (1).*

3.2 Complexity analysis

In this subsection we establish the outer and total inner iteration complexity for Algorithm 1. To proceed, we let (w^*, μ^*) be any pair of optimal solutions of problems (1) and (4) and fixed throughout this section.

Below, we establish a lemma to show that all the iterates generated by Algorithm 1 are bounded. Its proof can be found in Appendix B.3.

Lemma 3.2 (**Bounded iterates of Algorithm 1**). *Suppose that Assumption 1 holds. Let $\{w^k\}_{k\in\mathbb{K}}$ be all the iterates generated by Algorithm 1, where $\mathbb{K}$ is a subset of consecutive nonnegative integers starting from 0. Then we have $w^k \in \mathcal{Q}_1$ for all $k \in \mathbb{K}$, where*

$$\mathcal{Q}_1 = \{w \in \mathbb{R}^d : \|w - w^*\| \le r_0 + 2\bar{s}\beta\}, \quad r_0 = \|(w^0, \mu^0) - (w^*, \mu^*)\|, \tag{12}$$

and w^0, μ^0, $\bar{s}$, and β are inputs of Algorithm 1.

The following theorem states the worst-case complexity results of Algorithm 1, whose proof is relegated to Appendix B.4.

Theorem 3.2 (**Complexity results of Algorithm 1**). *Suppose that Assumption 1 holds. Then the number of outer iteration of Algorithm 1 is at most $\mathcal{O}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$, and the total number of inner iterations of Algorithm 1 is at most $\widetilde{\mathcal{O}}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$.*

To the best of our knowledge, Theorem 3.2 provides the first worst-case complexity results for finding an approximate KKT solution of problem (1) in an FL framework.

3.3 Communication overheads

In this subsection we discuss the communication overheads in Algorithm 1. In its outer loop, after solving a proximal AL subproblem, a single communication round occurs. During this round, the central server sends the current weights w^{k+1} to all local clients, and each client sends back the maximum change in their respective multipliers, measured by $\|\mu_i^{k+1} - \mu_i^k\|_\infty$, to the central server.

In addition, Algorithm 1 invokes an inexact ADMM based FL algorithm (Algorithm 2) as a subroutine for solving the proximal AL subproblem (9). At each iteration of Algorithm 2, there is a single communication round between the clients and the central server. During this round, the central server updates the global weights through aggregation and enforces global constraints, after which it broadcasts the updated global weights to all clients. Subsequently, each local client solves a strongly convex subproblem based on their individual local objectives and constraints, and then sends the updated local weights back to the central server. The communication approach employed by Algorithm 2 is similar to the commonly used FL framework for unconstrained problems (Li et al., 2020; Zhang et al., 2021).

4 An inexact ADMM for FL

In this section we propose an inexact ADMM based FL algorithm for solving a class of finite-sum optimization problems. This algorithm is used as a subproblem solver for the proximal AL based FL algorithm (Algorithm 1). In particular, we consider the following regularized unconstrained finite-sum optimization problem:

$$\min_w \left\{ F_h(w) := \sum_{i=0}^{n} F_i(w) + h(w) \right\}, \tag{13}$$

where $F_i : \mathbb{R}^d \to \mathbb{R}$, $0 \le i \le n$, are continuously differentiable and convex functions. Throughout this section, we assume that the central server has access to F_0, and for each $1 \le i \le n$, the local objective F_i is handled solely by the local client i.

4.1 Algorithm description

In this subsection we propose an inexact ADMM based FL algorithm (Algorithm 2) for solving problem (13). Since each participating client i handles one local objective F_i independently, we obtain the following equivalent consensus reformulation for problem (13):

$$\min_{w,u_i} \left\{ \sum_{i=1}^{n} F_i(u_i) + F_0(w) + h(w) \right\} \quad \text{s.t.} \quad u_i = w, \quad 1 \le i \le n, \tag{20}$$

which allows each local client i to handle the local variable u_i and the local objective function F_i while imposing consensus constraints that force clients' local parameters u_i equal to the global parameter w. This reformulation enables the applicability of an inexact ADMM that solves problem (20) in a federated manner. At each iteration, an ADMM optimizes the AL function associated with (20):

$$\mathcal{L}_F(w, u, \lambda) := \sum_{i=1}^{n} \left[F_i(u_i) + \langle \lambda_i, u_i - w \rangle + \frac{\rho_i}{2} \|u_i - w\|^2 \right] + F_0(w) + h(w) \tag{21}$$

with respect to the variables w, u, and λ alternately, where $u = [u_1^T, \ldots, u_n^T]^T$ and $[\lambda_1^T, \ldots, \lambda_n^T]^T$ collect all the local parameters and the multipliers associated with the consensus constraints, respectively. Specifically, at the iteration t, one performs

$$w^{t+1} \approx \arg\min_w \mathcal{L}_F(w, u^t, \lambda^t), \tag{22}$$

$$u^{t+1} \approx \arg\min_u \mathcal{L}_F(w^{t+1}, u, \lambda^t), \tag{23}$$

$$\lambda_i^{t+1} = \lambda_i^t + \rho_i(u_i^{t+1} - w^{t+1}), \quad \forall 1 \le i \le n.$$

Algorithm 2 An inexact ADMM based FL algorithm for solving problem (13)

Input: tolerance $\tau \in (0,1]$, $q \in (0,1)$, $\tilde{w}^0 \in \mathrm{dom}(h)$, and $\rho_i > 0$ for $1 \le i \le n$;

Set $w^0 = \tilde{w}^0$, and $(u_i^0, \lambda_i^0, \tilde{u}_i^0) = (\tilde{w}^0, -\nabla F_i(\tilde{w}^0), \tilde{w}^0 - \nabla F_i(\tilde{w}^0)/\rho_i)$ for $1 \le i \le n$.

for $t = 0, 1, 2, \ldots$ **do**

Set $\varepsilon_{t+1} = q^t$;

Server update: The central server finds an approximate solution w^{t+1} to

$$\min_w \left\{ \varphi_{0,t}(w) = F_0(w) + h(w) + \sum_{i=1}^n \left[\frac{\rho_i}{2} \|\tilde{u}_i^t - w\|^2 \right] \right\} \tag{14}$$

such that $\mathrm{dist}_\infty(0, \partial\varphi_{0,t}(w^{t+1})) \le \varepsilon_{t+1}$.

Communication (broadcast): Each local client i, $1 \le i \le n$, receives w^{t+1} from the server.

Client update (local): Each local client i, $1 \le i \le n$, finds an approximate solution u_i^{t+1} to

$$\min_{u_i} \left\{ \varphi_{i,t}(u_i) = F_i(u_i) + \langle \lambda_i^t, u_i - w^{t+1} \rangle + \frac{\rho_i}{2} \|u_i - w^{t+1}\|^2 \right\} \tag{15}$$

such that $\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty \le \varepsilon_{t+1}$, and then updates

$$\lambda_i^{t+1} = \lambda_i^t + \rho_i(u_i^{t+1} - w^{t+1}), \tag{16}$$

$$\tilde{u}_i^{t+1} = u_i^{t+1} + \lambda_i^{t+1}/\rho_i, \tag{17}$$

$$\tilde{\varepsilon}_{i,t+1} = \|\nabla\varphi_{i,t}(w^{t+1}) - \rho_i(w^{t+1} - u_i^t)\|_\infty. \tag{18}$$

Communication: Each local client i, $1 \le i \le n$, sends $(\tilde{u}_i^{t+1}, \tilde{\varepsilon}_{i,t+1})$ back to the central server.

Termination (server side): Output w^{t+1} and terminate this algorithm if

$$\varepsilon_{t+1} + \sum_{i=1}^n \tilde{\varepsilon}_{i,t+1} \le \tau. \tag{19}$$

end for

By the definition of $\mathcal{L}_F$ in (21), one can verify that the step (22) is equivalent to (14), and also the step (23) can be computed in parallel, which corresponds to (15). Therefore, the updates of an ADMM naturally suit the FL framework, as the separable structure in (21) over the pairs $\{(u_i, \lambda_i)\}$ enables the local update of (u_i, λ_i) at each client i while w is updated by the central server.

We now make some remarks about Algorithm 2. Since F_i, $0 \le i \le n$, are convex, the subproblems (14) and (15) are strongly convex. Consequently, their approximate solutions w^{t+1} and u_i^{t+1}, $1 \le i \le n$, can be found by a gradient-based algorithm with a global linear convergence rate. Furthermore, the value $\tilde{\varepsilon}_{i,t+1}$ in (18) serves as a measure for local optimality and consensus for client i. By summing up $\tilde{\varepsilon}_{i,t+1}$ for $1 \le i \le n$ and including ε_{t+1}, one can obtain a stationarity measure for the current iterate (see (19)), as presented in the following theorem. Its proof can be found in Appendix A.1.

Theorem 4.1 (**output of Algorithm 2**). *If Algorithm 2 terminates at some iteration t, then its output w^{t+1} satisfies* $\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) \le \tau$.

As seen from Theorem 4.1, Algorithm 2 outputs a point that approximately satisfies the first-order optimality condition of problem (1).

4.2 Complexity analysis

In this subsection we establish the iteration complexity for the inexact ADMM, namely, Algorithm 2. We now make the following additional assumptions on problem (13) throughout this section.

Assumption 3. (a) *The functions F_i, $0 \le i \le n$, are continuously differentiable, and moreover, ∇F_i, $0 \le i \le n$, are locally Lipschitz continuous on $\mathbb{R}^d$.*

(b) *The functions F_i, $0 \le i \le n$, are strongly convex on $\mathbb{R}^d$, that is, there exists some $\sigma > 0$ such that*

$$\langle \nabla F_i(u) - \nabla F_i(v), u - v \rangle \ge \sigma \|u - v\|^2, \quad \forall u, v \in \mathbb{R}^d, \ 0 \le i \le n.$$

Recall from (10 and 11) that $P_{i,k}$, $0 \le i \le n$, are strongly convex with modulus $1/[(n+1)\beta]$. Using this and the discussions in Section 3.1, we see that problem (9) satisfies Assumption 3, and therefore, Algorithm 2 is applicable to (9). Moreover, it follows from Theorem 4.1 that Algorithm 2 with $(\tau, \tilde{w}^0) = (\tau_k, w^k)$ is capable of finding an approximate solution w^{k+1} to (9) such that (7) holds.

Notice that the local Lipschitz continuity assumption for ∇F_i, $0 \le i \le n$, in Assumption 3(a) is weaker compared to the prevalent assumption on global Lipschitz continuity. In addition, under Assumption 3(b), problem (13) is strongly convex and thus has a unique optimal solution. We refer to this optimal solution of (13) as $\tilde{w}^*$ throughout this section.

The following lemma shows that all the iterates generated by Algorithm 2 lie in a compact set. Its proof can be found in Appendix A.2.

Lemma 4.1 (**bounded iterates of Algorithm 2**). *Suppose that Assumption 3 holds. Let $\{u_i^{t+1}\}_{1 \le i \le n, t \in \mathbb{T}}$ and $\{w^{t+1}\}_{t \in \mathbb{T}}$ be all the iterates generated by Algorithm 2, where $\mathbb{T}$ is a subset of consecutive nonnegative integers starting from 0. Then we have $w^{t+1} \in \mathcal{Q}$ and $u_i^{t+1} \in \mathcal{Q}$ for all $1 \le i \le n$ and $t \in \mathbb{T}$, where*

$$\mathcal{Q} = \left\{ v : \|v - \tilde{w}^*\|^2 \le \frac{n+1}{\sigma^2(1-q^2)} + \frac{1}{\sigma} \sum_{i=1}^{n} \left(\rho_i \|\tilde{w}^* - \tilde{w}^0\|^2 + \frac{1}{\rho_i} \|\nabla F_i(\tilde{w}^*) - \nabla F_i(\tilde{w}^0)\|^2 \right) \right\}. \tag{24}$$

The iteration complexity of Algorithm 2 is established in the following theorem, whose proof is relegated to Appendix A.3.

Theorem 4.2 (**iteration complexity of Algorithm 2**). *Suppose that Assumption 3 holds. Then Algorithm 2 terminates in at most $\mathcal{O}(|\log \tau|)$ iterations.*

As seen from Theorem 4.2, Algorithm 2 enjoys a global linear convergence rate for solving problem (1) under Assumption 3. This contrasts with the results in the existing literature, where the assumptions on strong convexity and global Lipschitz continuous gradients are typically required to establish a global linear convergence rate (e.g., see Lin et al. (2015)). To our knowledge, this provides the first study on the global linear convergence rate of an inexact ADMM under the *local Lipschitz* continuity of ∇F_i, $0 \le i \le n$.

5 Numerical experiments

In this section we conducted some experiments to test the performance of our proposed proximal AL based FL algorithm (Algorithm 1). Specifically, we compare Algorithm 1 with a centralized proximal AL method (abbreviated as cProx-AL) for solving a simulated linear equality constrained quadratic programming problem in Section 5.1. In Sections 5.2 and 5.3, we evaluate the performance of Algorithm 1 and cProx-AL on real-world datasets for a Neyman-Pearson classification problem and a classification problem with nonconvex fairness constraints. Additional numerical results for comparing unconstrained and classification models are presented in Appendix D.2. All the experiments are conducted in a system environment with an AMD EPYC 7763 64-core processor, using Python for execution.

5.1 Linear equality constrained quadratic programming

In this subsection we consider the linear equality constrained quadratic programming problem:

$$\min_{w} \ \sum_{i=1}^{n} \left(\frac{1}{2} w^T A_i w + b_i^T w \right) \quad \text{s. t.} \quad C_i w + d_i = 0, \quad 0 \le i \le n, \tag{25}$$

Table 1: Numerical results for problem (25)

			objective value			feasibility violation	
n	d	$\tilde{m}$	Algorithm 1	cProx-AL	relative difference	Algorithm 1	cProx-AL
1	100	1	-0.23 (4.65e-6)	-0.23 (2.38e-5)	1.63e-3 (1.01e-4)	3.33e-4 (1.14e-5)	5.68e-4 (2.82e-5)
	300	3	-0.37 (2.74e-6)	-0.37 (1.32e-6)	1.01e-3 (3.51e-5)	3.52e-4 (1.44e-5)	4.45e-4 (1.70e-5)
	500	5	-0.30 (1.36e-5)	-0.30 (7.54e-6)	1.34e-3 (4.62e-5)	4.38e-4 (5.36e-5)	3.85e-4 (1.05e-5)
5	100	1	9.81 (7.18e-5)	9.80 (1.46e-5)	1.09e-3 (7.96e-6)	1.34e-4 (9.02e-6)	8.03e-4 (1.57e-6)
	300	3	8.47 (8.12e-5)	8.45 (1.30e-5)	1.36e-3 (9.62e-6)	1.09e-4 (1.31e-5)	8.28e-4 (1.98e-6)
	500	5	9.92 (4.43e-5)	9.91 (4.87e-6)	8.26e-4 (4.27e-6)	1.33e-4 (9.68e-6)	3.73e-4 (2.43e-7)
10	100	1	49.40 (9.02e-5)	49.37 (5.82e-6)	5.59e-4 (1.67e-5)	7.31e-5 (7.54e-6)	5.88e-4 (1.34e-7)
	300	3	41.49 (7.04e-5)	41.44 (5.48e-6)	1.14e-3 (1.77e-6)	8.56e-5 (2.27e-7)	8.73e-4 (7.26e-6)
	500	5	41.45 (2.25e-5)	41.41 (5.30e-6)	9.39e-4 (4.94e-7)	9.29e-4 (2.55e-6)	7.66e-4 (1.37e-7)

where $A_i \in \mathbb{R}^{d\times d}$, $1 \le i \le n$, are positive semidefinite, $b_i \in \mathbb{R}^d$, $1 \le i \le n$, $C_i \in \mathbb{R}^{\tilde{m}\times d}$, $0 \le i \le n$, and $d_i \in \mathbb{R}^{\tilde{m}}$, $0 \le i \le n$.

For each $(d, n, \tilde{m})$, we randomly generate an instance of problem (25). In particular, for each $1 \le i \le n$, we first generate a random matrix A_i by letting $A_i = U_i D_i U_i^T$, where $D_i \in \mathbb{R}^{d\times d}$ is a diagonal matrix. The diagonal entries of D_i are generated randomly from a uniform distribution over $[0.5, 1]$, and $U_i \in \mathbb{R}^{d\times d}$ is a randomly generated orthogonal matrix. We then randomly generate C_i, $0 \le i \le n$, with all entries drawn from a normal distribution with mean zero and a standard deviation of $1/\sqrt{d}$. Finally, we generate b_i for $1 \le i \le n$ and d_i for $0 \le i \le n$ as random vectors uniformly selected from the unit Euclidean sphere.

Our aim is to apply Algorithm 1 and cProx-AL to find a $(10^{-3}, 10^{-3})$-KKT solution of problem (25), and compare their performance. In particular, we exactly solve the convex quadratic programming subproblems (14) and (15) arising in Algorithm 1. In addition, cProx-AL is presented in Algorithm 3, where w^{k+1} is obtained by solving (9) using a centralized method. We run 10 trials of Algorithm 1 and cProx-AL, where for each run, both algorithms have the same initial point w^0, randomly chosen from the unit Euclidean sphere. We set the other parameters for Algorithm 1 and cProx-AL as $\mu_i^0 = (0, \ldots, 0)^T$ $\forall 0 \le i \le n$, $\bar{s} = 0.1$ and $\beta = 10$. We also set $\rho_i = 1$ $\forall 1 \le i \le n$ for Algorithm 2.

The computational results of Algorithm 1 and the cProx-AL for solving the instances generated above are presented in Table 1. In detail, the values of d, n, and $\tilde{m}$ are listed in the first three columns, respectively. For each triple $(d, n, \tilde{m})$, the average objective value of Algorithm 1 and cProx-AL, the relative difference between these two algorithms, and the average feasibility violation of Algorithm 1 and cProx-AL over 10 trial runs are given in the remaining columns. The respective standard deviations are listed in parentheses. It is observed that both Algorithm 1 and cProx-AL are capable of finding nearly feasible solutions with similar objective value. In addition, given the small standard deviation, we observe that the convergence behavior of Algorithm 1 remains stable across 10 trial runs.

5.2 Neyman-Pearson classification

In this subsection we consider the Neyman-Pearson binary classification problem:

$$\min_w \frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_{i0}} \sum_{j=1}^{m_{i0}} \phi(w; (x_j^{(i0)}, 0)) \quad \text{s.t.} \quad \frac{1}{m_{i1}} \sum_{j=1}^{m_{i1}} \phi(w; (x_j^{(i1)}, 1)) \le r_i, \quad 1 \le i \le n, \tag{26}$$

where $\{x_j^{(i0)}\}_{1\le j\le m_{i0}}$ and $\{x_j^{(i1)}\}_{1\le j\le m_{i1}}$ are the sets of samples in client i associated with labels 0 and 1, respectively. In our experiment, we set ϕ as the binary logistic loss (see Hastie et al. (2009)):

$$\phi(w; (x_j^{(i)}, y_j^{(i)})) = -y_j^{(i)} w^T x_j^{(i)} + \log(1 + e^{w^T x_j^{(i)}}), \quad y_j^{(i)} \in \{0, 1\}. \tag{27}$$

Table 2: Numerical results for problem (26).

dataset	n	loss for class 0			loss for class 1 (≤ 0.2)			
		Algorithm 1	cProx-AL	relative difference	Algorithm 1		cProx-AL	
					mean	max	mean	max
breast-cancer-wisc	1	0.27 (1.52e-04)	0.27 (3.02e-05)	7.09e-04 (2.02e-04)	0.20 (1.80e-07)	0.20 (1.80e-07)	0.20 (1.84e-08)	0.20 (1.84e-08)
	5	0.34 (4.50e-02)	0.33 (4.55e-02)	1.15e-02 (5.17e-03)	0.19 (7.33e-06)	0.20 (1.08e-06)	0.19 (1.13e-04)	0.20 (1.72e-05)
	10	0.37 (1.08e-01)	0.37 (1.08e-01)	3.92e-04 (2.76e-04)	0.17 (1.15e-05)	0.20 (6.05e-09)	0.17 (1.14e-05)	0.20 (2.95e-08)
	20	0.46 (2.12e-01)	0.45 (2.12e-01)	3.43e-02 (2.91e-02)	0.16 (3.52e-05)	0.20 (3.76e-06)	0.16 (7.03e-06)	0.20 (7.70e-08)
adult-a	1	0.73 (2.19e-04)	0.73 (1.25e-04)	2.24e-04 (3.46e-04)	0.20 (6.30e-07)	0.20 (6.30e-07)	0.20 (1.73e-06)	0.20 (1.73e-06)
	5	0.74 (1.03e-02)	0.74 (1.03e-02)	4.25e-03 (7.44e-04)	0.20 (2.14e-04)	0.20 (2.80e-04)	0.20 (1.21e-05)	0.20 (2.28e-06)
	10	0.77 (1.98e-02)	0.77 (1.98e-02)	2.69e-03 (3.24e-03)	0.19 (6.41e-05)	0.20 (9.76e-05)	0.19 (2.00e-05)	0.20 (1.23e-05)
	20	0.78 (2.86e-02)	0.79 (2.81e-02)	1.13e-02 (4.11e-03)	0.18 (6.40e-04)	0.20 (6.59e-05)	0.18 (1.96e-05)	0.20 (3.19e-06)
monks-1	1	1.58 (7.61e-05)	1.58 (7.50e-05)	1.39e-05 (1.09e-05)	0.20 (1.09e-07)	0.20 (1.09e-07)	0.20 (3.01e-07)	0.20 (3.01e-07)
	5	1.65 (8.39e-02)	1.65 (8.41e-02)	2.08e-04 (1.84e-04)	0.19 (6.39e-05)	0.20 (5.39e-05)	0.19 (5.04e-06)	0.20 (5.60e-07)
	10	1.71 (1.18e-01)	1.71 (1.18e-01)	4.59e-04 (3.32e-04)	0.18 (3.98e-05)	0.20 (4.46e-05)	0.18 (6.44e-06)	0.20 (1.60e-06)
	20	1.81 (1.49e-01)	1.79 (1.60e-01)	1.78e-02 (1.38e-02)	0.17 (1.68e-04)	0.20 (2.24e-04)	0.17 (4.60e-06)	0.20 (1.62e-06)

We consider three real-world datasets, namely 'breast-cancer-wisc', 'adult-a', and 'monks-1', from the UCI repository.[2] The dataset description can be found in Table 4 in Appendix D.1. For each dataset, we conducted an imbalanced classification task that minimizes the binary classification loss while ensuring the loss for class 1 (minority) was less than a threshold $r = 0.2$. To simulate the FL setting, we divided each dataset into n folds, mimicking distributed clients each holding the same amount of data with equal imbalanced ratios.

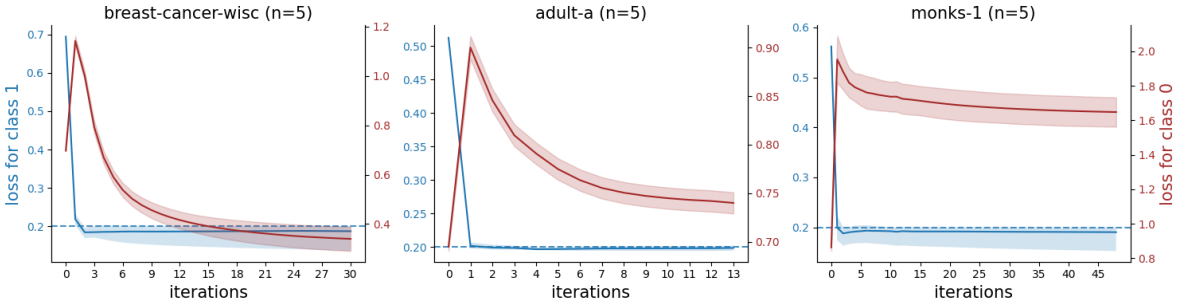

Figure 1: Convergence behavior of losses for classes 0 and 1 across all local clients in one random trial, over the outer iterations of Algorithm 1 on three real-world datasets. The solid blue and brown lines indicate the convergence behavior of the average loss for classes 0 and 1 over all clients, respectively. The blue and brown shaded areas indicate the regions between the maximum loss and minimum loss for classes 0 and 1 over all clients, respectively.

We apply Algorithm 1 and cProx-AL to find an $(10^{-3}, 10^{-3})$-KKT solution of problem (26). cProx-AL is presented in Algorithm 3, where w^{k+1} is obtained by applying L-BFGS method built in scipy.optimize.minimize to solve the subproblem. We run 10 trials of Algorithm 1 and cProx-AL, where for each run, both algorithms have the same initial point w^0, randomly chosen from the unit Euclidean sphere. We set the other parameters for Algorithm 1 and cProx-AL as $\mu_i^0 = (0, \ldots, 0)^T \ \forall 0 \leq i \leq n$, $\bar{s} = 0.001$ and $\beta = 300$. We also set $\rho_i = 0.01 \ \forall 1 \leq i \leq n$ for Algorithm 2.

The computational results for solving problems (26) using three real-world datasets are presented in Table 2. In detail, the first two columns of Table 2 represent the names of the dataset and the number of clients. In the last two columns, we present the losses for class 0 and class 1, respectively. By computing the average of 10 random trials, we include the relative difference of the objective value between Algorithm 1 and cProx-AL, and also the mean and max loss values for class 1 among all local clients. The respective standard deviations are listed in parentheses. Comparing the losses for both classes achieved by Algorithm 1 and cProx-AL in Table 2, we observe that both algorithms can yield solutions of similar quality. Given the small standard deviation, we observe that the convergence behavior of Algorithm 1 remains stable across 10 trial runs.

[2]see `https://archive.ics.uci.edu/datasets`

These observations demonstrate the ability of Algorithm 1 to solve the problem in an FL framework stably without compromising solution quality.

Figure 1 shows the convergence behavior of losses for classes 0 and 1 across all local clients over the outer iterations of Algorithm 1. From this figure, we observe that Algorithm 1 consistently reduces the losses for class 0 (local constraint) over all clients to a level below a threshold (≤ 0.2) while also consistently minimizing the losses for class 1 (local objective) over all local clients.

5.3 Classification with fairness constraints

In this subsection we consider the classification with global and local fairness constraints:

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\phi(w;(x_j^{(i)},y_j^{(i)})) \tag{28a}$$

$$\text{s.t. } -r_i \leq \frac{1}{\tilde{m}_i}\sum_{j=1}^{\tilde{m}_i}\phi(w;(\tilde{x}_j^{(i)},\tilde{y}_j^{(i)})) - \frac{1}{\hat{m}_i}\sum_{j=1}^{\hat{m}_i}\phi(w;(\hat{x}_j^{(i)},\hat{y}_j^{(i)})) \leq r_i, \quad 0 \leq i \leq n. \tag{28b}$$

where ϕ is the logistic loss defined as in (27), $(x_j^{(i)},y_j^{(i)}) \in \mathbb{R}^d \times \{0,1\}$, $1 \leq j \leq m_i$, are the feature-label pairs at client i. For each client i, the local dataset $\{(x_j^{(i)},y_j^{(i)})\}_{1\leq j\leq m_i}$ is divided into two sensitive groups $\{(\tilde{x}_j^{(i)},\tilde{y}_j^{(i)})\}_{1\leq j\leq \tilde{m}_i}$ and $\{(\hat{x}_j^{(i)},\hat{y}_j^{(i)})\}_{1\leq j\leq \hat{m}_i}$. The global dataset at the central server also includes two sensitive groups of samples $\{(\tilde{x}_j^{(0)},\tilde{y}_j^{(0)})\}_{1\leq j\leq \tilde{m}_0}$ and $\{(\hat{x}_j^{(0)},\hat{y}_j^{(0)})\}_{1\leq j\leq \hat{m}_0}$.

We consider the real-world dataset named 'adult-b' consisting of a training set and a testing set.[3] Each sample in this dataset has 39 features and a binary label. We conducted a binary classification task with fairness constraints that control the loss disparity between two sensitive groups of samples. We allocate 22,654 samples from the training set to the local dataset at clients, and 5,659 samples from the testing set to form the global dataset at the central server. To simulate an FL setting, we partitioned each dataset into n folds, ensuring an equal number of samples at each client.

We apply Algorithm 1 and cProx-AL to find an $(10^{-3},10^{-3})$-KKT solution of problem (28). cProx-AL is presented in Algorithm 3, where w^{k+1} is obtained by applying L-BFGS method built in scipy.optimize.minimize to solve the subproblem. We run 10 trials of Algorithm 1 and cProx-AL, where for each run, both algorithms have the same initial point w^0, randomly chosen from the unit Euclidean sphere. We set the other parameters for Algorithm 1 and the cProx-AL method as $\mu_i^0 = (0,\ldots,0)^T$ $\forall 0 \leq i \leq n$, $\bar{s} = 0.001$ and $\beta = 10$. We also set $\rho_i = 10^8$ $\forall 1 \leq i \leq n$ for Algorithm 2.

The computational results for solving problems (28) are presented in Table 3. In detail, the first column of Table 3 represents the number of clients. In the last two columns, we present the classification loss and loss disparity, respectively, which include results computed from the classification model with fairness constraints in (28). By computing the average of 10 random trials, we include the relative difference of the objective value between Algorithm 1 and cProx-AL, and also the mean and max loss disparity (absolute difference of losses for two sensitive groups) among all clients and the central server. The respective standard deviations are listed in parentheses. Comparing the classification loss and loss disparity of Algorithm 1 and cProx-AL in Table 3 reveals that both Algorithm 1 and cProx-AL can yield solutions of similar quality. Given the small standard deviation, we observe that the convergence behavior of Algorithm 1 remains stable across 10 trial runs. These observations demonstrate the ability of Algorithm 1 to solve the problem in an FL framework stably without compromising solution quality, and it also implies the potential of our algorithm in solving FL problems with particular nonconvex constraints.

Figure 2 shows the convergence behavior of loss disparity and classification loss across all local clients in one random trial, over the outer iterations of Algorithm 1. From this figure, we see that our proposed method consistently relegates the loss disparities (local/global constraints) on all clients and the central server to a

[3]This dataset can be found in `https://github.com/heyaudace/ml-bias-fairness/tree/master/data/adult`.

Table 3: Numerical results for problem (28).

n	objective value			loss disparity (≤ 0.1)			
	Algorithm 1	cProx-AL	relative difference	Algorithm 1		cProx-AL	
				mean	max	mean	max
1	0.37 (9.83e-05)	0.37 (4.14e-05)	1.97e-03 (2.53e-04)	0.10 (1.14e-04)	0.10 (1.36e-04)	0.10 (3.69e-06)	0.10 (5.38e-06)
5	0.37 (3.99e-03)	0.37 (4.05e-03)	1.86e-03 (4.69e-04)	0.09 (5.34e-05)	0.10 (7.51e-05)	0.09 (3.68e-05)	0.10 (4.36e-06)
10	0.37 (6.39e-03)	0.37 (6.52e-03)	2.39e-03 (8.40e-04)	0.08 (1.68e-04)	0.10 (2.15e-05)	0.08 (1.52e-04)	0.10 (6.56e-06)
20	0.38 (9.46e-03)	0.37 (9.86e-03)	4.61e-03 (2.43e-03)	0.08 (9.75e-05)	0.10 (1.01e-04)	0.08 (4.90e-05)	0.10 (6.06e-06)

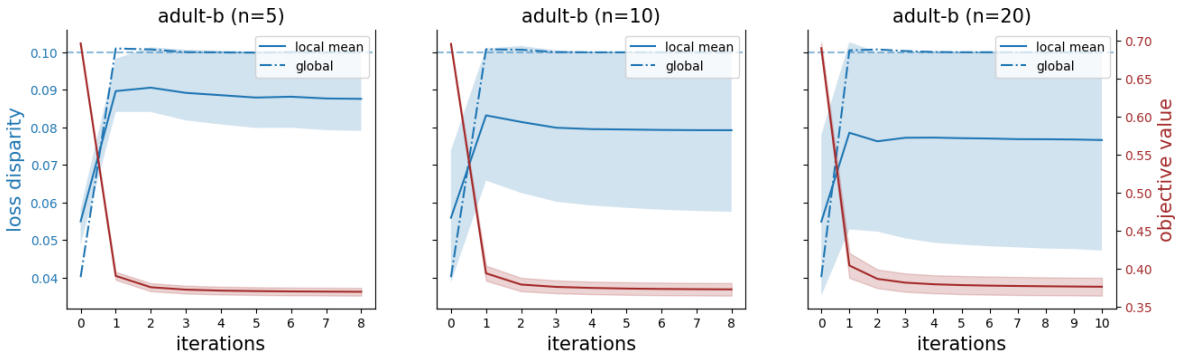

Figure 2: Convergence behavior of loss disparity and classification loss across all local clients in one random trial, over the outer iterations of Algorithm 1 on the adult dataset. The solid blue and brown lines indicate the convergence behavior of the average loss disparity and classification loss over all clients, respectively. The blue and brown shaded areas indicate thregions between the maximum value and minimum value of loss disparity and classification loss over all clients, respectively. The blue dashdot line indicates the convergence behavior of the global loss disparity in the central server.

level below a threshold (≤ 0.1) while also consistently minimizing the classification losses (local objectives) on all local clients.

6 Concluding remarks

In this paper we proposed a proximal AL based algorithm for solving FL problems with convex global and local constraints. We then analyzed its worst-case iteration complexity under mild assumptions. Finally, we performed numerical experiments using real-world datasets to assess the performance of our proposed algorithm for Neyman-Pearson classification and classification with fairness constraints in the FL setting. The numerical results clearly demonstrate the practical efficacy of our proposed algorithm.

There are several possible extensions of this work. First, the development of FL algorithms allowing partial client participation has been a popular research direction. Paricularly, recent works have shown that ADMM based FL algorithms allow partial participation (e.g., see Zhou & Li (2023); Wang et al. (2022)). It would be very interesting to extend our algorithm to incorporate this feature. Second, in scenarios where local datasets are large, clients may need stochastic gradient-based methods. Therefore, developing a stochastic variant of our proposed algorithm is an intriguing research direction. Third, extending our algorithm to the nonconvex setting would make our algorithm applicable to a broader class of problems.

Acknowledgments

C. He is partially supported by the NIH fund R01CA287413 and the UMN Research Computing Seed Grant. L. Peng is partially supported by the CISCO Research fund 1085646 PO USA000EP390223. J. Sun is partially supported by the NIH fund R01CA287413 and the CISCO Research fund 1085646 PO USA000EP390223. The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported in this article. The research reported in

this publication is partially supported by the National Cancer Institute of the National Institutes of Health under Award Number R01CA287413. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69. PMLR, 2018.

Naman Agarwal, Nicolas Boumal, Brian Bullins, and Coralia Cartis. Adaptive regularization with cubics on manifolds. *Mathematical Programming*, 188:85–134, 2021.

Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. A primal-dual method for conic constrained distributed optimization problems. *Advances in Neural Information Processing Systems*, 29, 2016.

Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. A distributed ADMM-like method for resource sharing over time-varying networks. *SIAM Journal on Optimization*, 29(4):3036–3068, 2019.

Necdet Serhat Aybat and Garud Iyengar. An augmented Lagrangian method for conic convex programming. *arXiv preprint arXiv:1302.6322*, 2013.

Ernesto G Birgin and José Mario Martínez. Complexity and performance of an augmented Lagrangian algorithm. *Optimization Methods and Software*, 35(5):885–920, 2020.

Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

Richard H Byrd, Robert B Schnabel, and Gerald A Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24(5):1152–1170, 1987.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA Journal of Numerical Analysis*, 32(4):1662–1695, 2012.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM Journal on Optimization*, 23(3):1553–1574, 2013.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods. *SIAM Journal on Numerical Analysis*, 53(2):836–851, 2015.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 319–328, 2019.

Peter W Christensen and Anders Klarbring. *An introduction to structural optimization*, volume 153. Springer Science & Business Media, 2008.

Lingyang Chu, Lanjun Wang, Yanjie Dong, Jian Pei, Zirui Zhou, and Yong Zhang. Fedfair: Training fair models in cross-silo federated learning. *arXiv preprint arXiv:2109.05662*, 2021.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pp. 300–332. PMLR, 2019.

Frank E Curtis and Michael L Overton. A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. *SIAM Journal on Optimization*, 22(2):474–500, 2012.

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 181–189. SIAM, 2021.

Borja Rodríguez Gálvez, Filip Granqvist, Rogier van Dalen, and Matt Seigel. Enforcing fairness in private federated learning via the modified method of differential multipliers. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

Yonghai Gong, Yichuan Li, and Nikolaos M Freris. FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2575–2587. IEEE, 2022.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Geovani Nunes Grapiglia and Ya-xiang Yuan. On the complexity of an augmented Lagrangian method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 41(2):1546–1568, 2021.

Gabriel Haeser, Hongcheng Liu, and Yinyu Ye. Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary. *Mathematical Programming*, 178: 263–299, 2019.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, New York, 2nd edition, 2009.

Chuan He and Zhaosong Lu. A Newton-CG based barrier method for finding a second-order stationary point of nonconvex conic optimization with complexity guarantees. *SIAM Journal on Optimization*, 33 (2):1191–1222, 2023.

Chuan He, Heng Huang, and Zhaosong Lu. A Newton-CG based barrier-augmented Lagrangian method for general nonconvex conic optimization. *arXiv preprint arXiv:2301.04204*, 2023a.

Chuan He, Zhaosong Lu, and Ting Kei Pong. A Newton-CG based augmented Lagrangian method for finding a second-order stationary point of nonconvex equality constrained optimization with complexity guarantees. *SIAM Journal on Optimization*, 33(3):1734–1766, 2023b.

Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pp. 1529–1538. PMLR, 2017.

Kevin Huang, Nuozhou Wang, and Shuzhong Zhang. An accelerated variance reduced extra-point approach to finite-sum VI and optimization. *arXiv preprint arXiv:2211.03269*, 2022.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

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

Weiwei Kong, Jefferson G Melo, and Renato DC Monteiro. Iteration complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical Lagrangian function. *SIAM Journal on Optimization*, 33(1):181–210, 2023.

Guanghui Lan and Renato DC Monteiro. Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Mathematical Programming*, 155(1-2):511–547, 2016.

Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021b.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021c.

Zichong Li, Pin-Yu Chen, Sijia Liu, Songtao Lu, and Yangyang Xu. Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2170–2178. PMLR, 2021d.

Peng Lin, Wei Ren, and Yongduan Song. Distributed multi-agent optimization subject to nonidentical constraints and communication delays. *Automatica*, 65:120–131, 2016.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. On the global linear convergence of the ADMM with multi-block variables. *SIAM Journal on Optimization*, 25(3):1478–1497, 2015.

Yi Liu, JQ James, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, 7(8):7751–7763, 2020.

Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pp. 240–254. Springer, 2020.

Songtao Lu. A single-loop gradient descent and perturbed ascent algorithm for nonconvex functional constrained optimization. In *International Conference on Machine Learning*, pp. 14315–14357. PMLR, 2022.

Songtao Lu, Meisam Razaviyayn, Bo Yang, Kejun Huang, and Mingyi Hong. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. *Advances in Neural Information Processing Systems*, 33:2811–2822, 2020.

Zhaosong Lu and Sanyou Mei. Accelerated first-order methods for convex optimization with locally Lipschitz continuous gradient. *SIAM Journal on Optimization*, 33(3):2275–2310, 2023.

Zhaosong Lu and Zirui Zhou. Iteration-complexity of first-order augmented Lagrangian methods for convex conic programming. *SIAM Journal on Optimization*, 33(2):1159–1190, 2023.

Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019.

Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Ion Necoara, Andrei Patrascu, and Francois Glineur. Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2):305–335, 2019.

Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

Maher Nouiehed, Jason D Lee, and Meisam Razaviyayn. Convergence to second-order stationarity for constrained non-convex optimization. *arXiv preprint arXiv:1810.02024*, 2018.

Michael O'Neill and Stephen J Wright. A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees. *IMA Journal of Numerical Analysis*, 41(1):84–121, 2021.

Andrei Patrascu, Ion Necoara, and Quoc Tran-Dinh. Adaptive inexact fast augmented Lagrangian methods for constrained convex optimization. *Optimization Letters*, 11:609–626, 2017.

Le Peng, Gaoxiang Luo, Andrew Walker, Zachary Zaiman, Emma K Jones, Hemant Gupta, Kristopher Kersten, John L Burns, Christopher A Harle, Tanja Magoc, et al. Evaluation of federated learning variations for COVID-19 diagnosis using chest radiographs from 42 US and European hospitals. *Journal of the American Medical Informatics Association*, 30(1):54–63, 2023a.

Le Peng, Sicheng Zhou, Jiandong Chen, Rui Zhang, Ziyue Xu, and Ju Sun. A systematic evaluation of federated learning on biomedical natural language processing. In *International Workshop on Federated Learning for Distributed Data Mining*, 2023b. URL `https://openreview.net/forum?id=pLEQFXACNA`.

MJD Powell and ya-xiang Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49(1):189–211, 1991.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ Digital Medicine*, 3(1):119, 2020.

Philippe Rigollet and Xin Tong. Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, 2011.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-IID data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019.

Clayton Scott. Performance measures for Neyman–Pearson classification. *IEEE Transactions on Information Theory*, 53(8):2852–2863, 2007.

Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2021.

Xin Tong, Yang Feng, and Anqi Zhao. A survey on Neyman-Pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(2):64–81, 2016.

Xin Tong, Yang Feng, and Jingyi Jessica Li. Neyman-pearson classification algorithms and np receiver operating characteristics. *Science Advances*, 4(2):eaao1659, 2018.

Yash Travadi, Le Peng, Xuan Bi, Ju Sun, and Mochen Yang. Welfare and fairness dynamics in federated learning: A client selection perspective. *Statistics and Its Interface*, 2023.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

Han Wang, Siddartha Marella, and James Anderson. Fedadmm: A federated primal-dual algorithm allowing partial participation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 287–294. IEEE, 2022.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33:7611–7623, 2020.

Peng Wang, Peng Lin, Wei Ren, and Yongduan Song. Distributed subgradient-based multiagent optimization with more general step sizes. *IEEE Transactions on Automatic Control*, 63(7):2295–2302, 2017.

Yue Xie and Stephen J Wright. Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints. *Journal of Scientific Computing*, 86:1–30, 2021.

Yangyang Xu. Iteration complexity of inexact augmented Lagrangian methods for constrained convex programming. *Mathematical Programming*, 185:199–244, 2021.

Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

Deming Yuan, Shengyuan Xu, and Huanyu Zhao. Distributed primal-dual subgradient method for multiagent optimization via consensus algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(6):1715–1724, 2011.

Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. FedPD: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021.

Shenglong Zhou and Geoffrey Ye Li. Federated learning via inexact ADMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Minghui Zhu and Sonia Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2011.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

Appendix

In Appendices A and B, we provide proofs of main results in Sections 4 and 3, respectively. We present some additional numerical results in Appendix D.

A Proof of the main results in Section 4

Throughout this section, we let $(\tilde{w}^*, u^*)$ be the optimal solution of (20), and λ^* be the associated Lagrangian multiplier. Recall from the definition of $\tilde{u}_i^0$ in Algorithm 2 and (17) that

$$\tilde{u}_i^t = u_i^t + \lambda_i^t/\rho_i, \quad \forall 1 \le i \le n, t \ge 0. \tag{29}$$

A.1 Proof of Theorem 4.1

Proof of Theorem 4.1. By the definition of F_h in (13), one has that

$$\partial F_h(w^{t+1}) = \sum_{i=0}^n \nabla F_i(w^{t+1}) + \partial h(w^{t+1}). \tag{30}$$

In addition, notice from (14), (15), and (29) that

$$\begin{aligned}
\partial \varphi_{0,t}(w^{t+1}) &= \nabla F_0(w^{t+1}) + \sum_{i=1}^n \rho_i(w^{t+1} - \tilde{u}_i^t) + \partial h(w^{t+1}) \\
&= \nabla F_0(w^{t+1}) + \sum_{i=1}^n [\rho_i(w^{t+1} - u_i^t) - \lambda_i^t] + \partial h(w^{t+1}), \\
\nabla \varphi_{i,t}(w^{t+1}) &= \nabla F_i(w^{t+1}) + \lambda_i^t, \quad \forall 1 \le i \le n.
\end{aligned}$$

Combining these with (30), we obtain that

$$\partial F_h(w^{t+1}) = \partial \varphi_{0,t}(w^{t+1}) + \sum_{i=1}^n [\nabla \varphi_{i,t}(w^{t+1}) - \rho_i(w^{t+1} - u_i^t)],$$

which together with $\mathrm{dist}_\infty(0, \partial \varphi_{0,t}(w^{t+1})) \le \varepsilon_{t+1}$ (see Algorithm 2) and (18) implies that

$$\begin{aligned}
\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) &\le \mathrm{dist}_\infty(0, \partial \varphi_{0,t}(w^{t+1})) + \sum_{i=1}^n \|\nabla \varphi_{i,t}(w^{t+1}) - \rho_i(w^{t+1} - u_i^t)\|_\infty \\
&\le \varepsilon_{t+1} + \sum_{i=1}^n \tilde{\varepsilon}_{i,t+1}.
\end{aligned}$$

Using this and the termination criterion in (19), we obtain that $\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) \le \tau$ holds as desired. □

A.2 Proof of Lemma 4.1

Proof of Lemma 4.1. Recall from Assumption 3 that F_i, $0 \le i \le n$, are strongly convex with modulus $\sigma > 0$. In addition, by (16), (29), and the fact that $\mathrm{dist}_\infty(0, \partial \varphi_{0,t}(w^{t+1})) \le \varepsilon_{t+1}$, one can obtain that there exist some $h^{t+1} \in \partial h(w^{t+1})$ and $\|e_0^{t+1}\|_\infty \le \varepsilon_{t+1}$ such that

$$\begin{aligned}
e_0^{t+1} &= \nabla F_0(w^{t+1}) + h^{t+1} + \sum_{i=1}^n \rho_i(w^{t+1} - \tilde{u}_i^t) \overset{(29)}{=} \nabla F_0(w^{t+1}) + h^{t+1} + \sum_{i=1}^n [\rho_i(w^{t+1} - u_i^t) - \lambda_i^t] \\
&\overset{(16)}{=} \nabla F_0(w^{t+1}) + h^{t+1} + \sum_{i=1}^n [\rho_i(u_i^{t+1} - u_i^t) - \lambda_i^{t+1}].
\end{aligned} \tag{31}$$

Using (16) and the fact that $\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty \le \varepsilon_{t+1}$, one can see that there exists $\|e_i^{t+1}\|_\infty \le \varepsilon_{t+1}$ such that

$$e_i^{t+1} = \nabla\varphi_{i,t}(u_i^{t+1}) \overset{(15)}{=} \nabla F_i(u_i^{t+1}) + \lambda_i^t + \rho_i(u_i^{t+1} - w^{t+1}) \overset{(16)}{=} \nabla F_i(u_i^{t+1}) + \lambda_i^{t+1}, \quad \forall 1 \le i \le n, \tag{32}$$

Recall that $\tilde{w}^*$ and u^* are the optimal solution of (20), and $\lambda^* \in \mathbb{R}^m$ is the associated Lagrangian multiplier. Then there exists $h^* \in \partial h(\tilde{w}^*)$ such that

$$\nabla F_i(u_i^*) + \lambda_i^* = 0, \quad \nabla F_0(\tilde{w}^*) + h^* - \sum_{i=1}^n \lambda_i^* = 0, \quad u_i^* = \tilde{w}^*, \quad \forall 1 \le i \le n. \tag{33}$$

In view of this, (32), and the strong convexity of F_i, one can deduce that

$$\begin{aligned}\sigma\|u_i^{t+1} - \tilde{w}^*\|^2 &\le \langle u_i^{t+1} - \tilde{w}^*, \nabla F_i(u_i^{t+1}) - \nabla F_i(\tilde{w}^*)\rangle = \langle u_i^{t+1} - \tilde{w}^*, -\lambda_i^{t+1} + \lambda_i^* + e_i^{t+1}\rangle \\ &\le \langle u_i^{t+1} - \tilde{w}^*, -\lambda_i^{t+1} + \lambda_i^*\rangle + \frac{\sigma}{2}\|u_i^{t+1} - \tilde{w}^*\|^2 + \frac{1}{2\sigma}\|e_i^{t+1}\|^2,\end{aligned}$$

where the equality is due to $\tilde{w}^* = u_i^*$, $\nabla F_i(u_i^*) = \lambda_i^*$, and (32), and the last inequality follows from $\langle a, b\rangle \le t/2\|a\|^2 + 1/(2t)\|b\|^2$ for all $a, b \in \mathbb{R}^d$ and $t > 0$. By (31), (33), and the strong convexity of F_0, one has that

$$\begin{aligned}\sigma\|w^{t+1} - \tilde{w}^*\|^2 &\le \langle w^{t+1} - \tilde{w}^*, \nabla F_0(w^{t+1}) + h^{t+1} - \nabla F_0(\tilde{w}^*) - h^*\rangle \\ &= \langle w^{t+1} - \tilde{w}^*, \sum_{i=1}^n [\lambda_i^{t+1} - \lambda_i^* - \rho_i(u_i^{t+1} - u_i^t)] + e_0^{t+1}\rangle, \\ &\le \langle w^{t+1} - \tilde{w}^*, \sum_{i=1}^n [\lambda_i^{t+1} - \lambda_i^* - \rho_i(u_i^{t+1} - u_i^t)]\rangle + \frac{\sigma}{2}\|w^{t+1} - \tilde{w}^*\|^2 + \frac{1}{2\sigma}\|e_0^{t+1}\|^2,\end{aligned}$$

where the first inequality is due to the strong convexity of F_0 and the convexity of h, the equality is due to (31) and the second relation in (33), and the last inequality follows from $\langle a, b\rangle \le t/2\|a\|^2 + 1/(2t)\|b\|^2$ for all $a, b \in \mathbb{R}^d$ and $t > 0$. Summing up these inequalities and rearranging the terms, we obtain that

$$\begin{aligned}&\frac{\sigma}{2}(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^n \|u_i^{t+1} - \tilde{w}^*\|^2) \\ &\le \langle w^{t+1} - \tilde{w}^*, \sum_{i=1}^n [\lambda_i^{t+1} - \lambda_i^* - \rho_i(u_i^{t+1} - u_i^t)]\rangle + \frac{1}{2\sigma}\|e_0^{t+1}\|^2 + \sum_{i=1}^n (\langle u_i^{t+1} - \tilde{w}^*, -\lambda_i^{t+1} + \lambda_i^*\rangle + \frac{1}{2\sigma}\|e_i^{t+1}\|^2) \\ &\le \sum_{i=1}^n \langle w^{t+1} - u_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^n \rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \\ &\overset{(16)}{=} \sum_{i=1}^n \frac{1}{\rho_i}\langle \lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^n \rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \frac{n+1}{2\sigma}\varepsilon_{t+1}^2,\end{aligned} \tag{34}$$

where the second inequality is due to $\|e_i^{t+1}\| \le \varepsilon_{t+1}$ for all $0 \le i \le n$ and $t \ge 0$. Notice that the following well-known identities hold:

$$\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle = \frac{1}{2}(\|w^{t+1} - u_i^{t+1}\|^2 - \|w^{t+1} - u_i^t\|^2 + \|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2), \tag{35}$$

$$\langle \lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle = \frac{1}{2}(\|\lambda_i^* - \lambda_i^t\|^2 - \|\lambda_i^* - \lambda_i^{t+1}\|^2 - \|\lambda_i^t - \lambda_i^{t+1}\|^2). \tag{36}$$

These along with (16) and (34) imply that

$$\begin{aligned}&\frac{\sigma}{2}(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^n \|u_i^{t+1} - \tilde{w}^*\|^2) + \sum_{i=1}^n \frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2 - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \\ &\overset{(34)}{\le} \sum_{i=1}^n \frac{1}{\rho_i}\langle \lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^n \rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \sum_{i=1}^n \frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2\end{aligned}$$

$$
\begin{aligned}
&\overset{(35)}{\leq} \sum_{i=1}^{n} \frac{1}{\rho_i}\langle \lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^* \rangle + \sum_{i=1}^{n} \frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2 + \|w^{t+1} - u_i^{t+1}\|^2) \\
&\overset{(16)}{=} \sum_{i=1}^{n} \frac{1}{\rho_i}\langle \lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^* \rangle + \sum_{i=1}^{n} \frac{1}{2\rho_i}\|\lambda_i^{t+1} - \lambda_i^t\|^2 + \sum_{i=1}^{n} \frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2) \\
&\overset{(36)}{=} \sum_{i=1}^{n} \frac{1}{2\rho_i}(\|\lambda_i^* - \lambda_i^t\|^2 - \|\lambda_i^* - \lambda_i^{t+1}\|^2) + \sum_{i=1}^{n} \frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2) \\
&= \sum_{i=1}^{n}[(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2) - (\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2)]. \qquad (37)
\end{aligned}
$$

Summing up this inequality over $t = 0, \ldots, \bar{t}$, we obtain that

$$
\begin{aligned}
&\sum_{t=0}^{\bar{t}} \left[\frac{\sigma}{2} \left(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^{n} \|u_i^{t+1} - \tilde{w}^*\|^2 \right) + \sum_{i=1}^{n} \frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2 - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \right] \\
&\leq \sum_{i=1}^{n} \left[\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^0\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^0\|^2 \right) - \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{\bar{t}+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{\bar{t}+1}\|^2 \right) \right]. \qquad (38)
\end{aligned}
$$

Recall from Algorithm 2 that $\varepsilon_{t+1} = q^t$, $u_i^0 = \tilde{w}^0$, and $\lambda_i^0 = -\nabla F_i(\tilde{w}^0)$. Also, notice from (33) that $\tilde{w}^* = u_i^*$ and $\lambda_i^* = -\nabla F_i(u_i^*)$. By these and (38), one can deduce that

$$
\begin{aligned}
&\frac{\sigma}{2}(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^{n} \|u_i^{t+1} - \tilde{w}^*\|^2) \leq \frac{n+1}{2\sigma}\sum_{t=0}^{\infty} q^{2t} + \sum_{i=1}^{n} \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^0\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^0\|^2 \right) \\
&\leq \frac{n+1}{2\sigma(1-q^2)} + \sum_{i=1}^{n} \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^0\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^0\|^2 \right) \\
&= \frac{n+1}{2\sigma(1-q^2)} + \sum_{i=1}^{n} \left(\frac{\rho_i}{2}\|\tilde{w}^* - \tilde{w}^0\|^2 + \frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*) - \nabla F_i(\tilde{w}^0)\|^2 \right).
\end{aligned}
$$

In view of this and the definition of $\mathcal{Q}$ in (24), we can observe that $w^{t+1} \in \mathcal{Q}$ and $u_i^{t+1} \in \mathcal{Q}$ for all $t \in \mathbb{T}$ and $1 \leq i \leq n$. Hence, the conclusion of this lemma holds as desired. □

A.3 Proof of Theorem 4.2

Lemma A.1. *Assume that $r, c > 0$ and $q \in (0,1)$. Let $\{a_t\}_{t\geq 0}$ be a sequence satisfying*

$$
(1+r)a_{t+1} \leq a_t + cq^{2t}, \quad \forall t \geq 0. \qquad (39)
$$

Then we have

$$
a_{t+1} \leq \max\left\{q, \frac{1}{1+r}\right\}^{t+1} \left(a_0 + \frac{c}{1-q} \right), \quad \forall t \geq 0. \qquad (40)
$$

Proof. It follows from (39) that

$$
\begin{aligned}
a_{t+1} &\leq \frac{1}{1+r}a_t + \frac{1}{1+r}cq^{2t} \leq \frac{1}{(1+r)^2}a_{t-1} + \frac{cq^{2(t-1)}}{(1+r)^2} + \frac{cq^{2t}}{1+r} \\
&\leq \cdots \leq \frac{1}{(1+r)^{t+1}}a_0 + \sum_{i=0}^{t} \frac{cq^{2i}}{(1+r)^{t+1-i}} = \frac{1}{(1+r)^{t+1}}a_0 + c\sum_{i=0}^{t} \frac{q^i}{(1+r)^{t+1-i}}q^i \\
&\leq \frac{1}{(1+r)^{t+1}}a_0 + c\max\left\{q, \frac{1}{1+r}\right\}^{t+1} \sum_{i=0}^{t} q^i
\end{aligned}
$$

$$\leq \frac{1}{(1+r)^{t+1}}a_0 + \frac{c}{1-q}\max\left\{q, \frac{1}{1+r}\right\}^{t+1} \leq \max\left\{q, \frac{1}{1+r}\right\}^{t+1}\left(a_0 + \frac{c}{1-q}\right),$$

where the fifth inequality is due to $q^i \leq \max\{q, 1/(1+r)\}^i$ and $1/(1+r)^{t+1-i} \leq \max\{q, 1/(1+r)\}^{t+1-i}$. Hence, the relation (40) holds as desired. □

Lemma A.2. *Let $\mathcal{Q}$ be defined in (24). Then there exists some $L_{\nabla F} > 0$ such that*

$$\|\nabla F_i(u) - \nabla F_i(v)\| \leq L_{\nabla F}\|u - v\|, \quad \forall u, v \in \mathcal{Q}, 0 \leq i \leq n. \tag{41}$$

Proof. Notice from (24) that the set $\mathcal{Q}$ is convex and compact. By this and the local Lipschitz continuity of ∇F_i on $\mathbb{R}^d$, one can verify that there exists some constant $L_{\nabla F} > 0$ such that (41) holds (see also Lemma 1 in Lu & Mei (2023)). □

Lemma A.3. *Suppose that Assumption 3 holds. Let $\{w^{t+1}\}_{t\in\mathbb{T}}$ and $\{u_i^{t+1}\}_{1\leq i\leq n, t\in\mathbb{T}}$ be all the iterates generated by Algorithm 2, where $\mathbb{T}$ is defined in Lemma 4.1. Then we have*

$$S_t \leq q_r^t\left[S_0 + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right], \quad \forall t \geq 0, \tag{42}$$

where σ and $L_{\nabla F}$ are given in Assumption 3(ii) and Lemma A.2, respectively, q and ρ_i, $1 \leq i \leq n$, are inputs of Algorithm 2, and

$$q_r = \max\left\{q, \frac{1}{1+r}\right\}, \quad r = \min_{1\leq i\leq n}\left\{\frac{\sigma\rho_i}{\rho_i^2 + 2L_{\nabla F}^2}\right\}, \tag{43}$$

$$S_t = \sum_{i=1}^n\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2\right), \quad \forall t \geq 0. \tag{44}$$

Proof. Recall from (37) that

$$\begin{aligned}
&\sum_{i=1}^n\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2\right)\\
&\geq \sum_{i=1}^n\left(\frac{\rho_i+\sigma}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2 + \frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2\right) + \frac{\sigma}{2}\|w^{t+1} - \tilde{w}^*\|^2 - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2\\
&\geq \sum_{i=1}^n\left(\frac{\rho_i+\sigma}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2\right) - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2.
\end{aligned} \tag{45}$$

Also, notice from (32), (33), and (41) that

$$\|\lambda_i^* - \lambda_i^{t+1}\|^2 \overset{(32)(33)}{\leq} (\|\nabla F_i(\tilde{w}^*) - \nabla F_i(u_i^{t+1})\| + \|e_i^{t+1}\|)^2 \overset{(41)}{\leq} 2L_{\nabla F}^2\|\tilde{w}^* - u_i^{t+1}\|^2 + 2\varepsilon_{t+1}^2,$$

which implies that

$$\|\tilde{w}^* - u_i^{t+1}\|^2 \geq \frac{2\rho_i}{\rho_i^2 + 2L_{\nabla F}^2}\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2\right) - \frac{2\varepsilon_{t+1}^2}{\rho_i^2 + 2L_{\nabla F}^2}. \tag{46}$$

By this, the definition of S_t in (44), and (45), one has that

$$\begin{aligned}
&S_t + \left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)q^{2t}\\
&= \sum_{i=1}^n\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2\right) + \left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\varepsilon_{t+1}^2
\end{aligned}$$

$$
\begin{aligned}
&\overset{(45)}{\geq} \sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^{t+1}\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^{t+1}\|^2\right)+\frac{\sigma}{2}\sum_{i=1}^{n}\|\tilde{w}^*-u_i^{t+1}\|^2+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\varepsilon_{t+1}^2\\
&\overset{(46)}{\geq} \sum_{i=1}^{n}\left(1+\frac{\sigma\rho_i}{\rho_i^2+2L_{\nabla F}^2}\right)\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^{t+1}\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^{t+1}\|^2\right)\\
&\geq (1+r)S_{t+1}.
\end{aligned}
$$

When $t=0$, (42) holds clearly. When $t\geq 1$, by the above inequality, (43), and Lemma A.1 with $(a_t,c)=(S_t,\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2})$, we obtain that

$$
\begin{aligned}
S_t &\leq \max\left\{q,\frac{1}{1+r}\right\}^t\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right]\\
&= q_r^t\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right].
\end{aligned}
$$

Hence, the conclusion of this lemma holds as desired. □

Proof of Theorem 4.2. By the definition of $\varphi_{i,t}$ in (15), one has that for each $1\leq i\leq n$,

$$
\|\nabla\varphi_{i,t}(w^{t+1})-\nabla\varphi_{i,t}(u_i^{t+1})\|\leq\|\nabla F_i(w^{t+1})-\nabla F_i(u_i^{t+1})\|+\rho_i\|w^{t+1}-u_i^{t+1}\|\leq(L_{\nabla F}+\rho_i)\,\|w^{t+1}-u_i^{t+1}\|,
$$

where the second inequality is due to (41) and the fact that $w^{t+1}\in\mathcal{Q}$ and $u_i^{t+1}\in\mathcal{Q}$ for all $1\leq i\leq n$ (see Lemma 4.1). By the above inequality, (18), and the fact that $\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty\leq\varepsilon_{t+1}$ (see Algorithm 2), one can obtain that

$$
\begin{aligned}
\varepsilon_{t+1}+\sum_{i=1}^{n}\tilde{\varepsilon}_{i,t+1} &\overset{(18)}{=} \varepsilon_{t+1}+\sum_{i=1}^{n}\|[\nabla\varphi_{i,t}(w^{t+1})-\rho_i(w^{t+1}-u_i^t)]\|_\infty\\
&\leq \varepsilon_{t+1}+\sum_{i=1}^{n}\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty+\sum_{i=1}^{n}\|\nabla\varphi_{i,t}(w^{t+1})-\nabla\varphi_{i,t}(u_i^{t+1})\|+\sum_{i=1}^{n}\rho_i\|w^{t+1}-u_i^t\|\\
&\leq (n+1)\varepsilon_{t+1}+\sum_{i=1}^{n}(L_{\nabla F}+\rho_i)\|w^{t+1}-u_i^{t+1}\|+\sum_{i=1}^{n}\rho_i\|w^{t+1}-u_i^t\|, \qquad (47)
\end{aligned}
$$

where the first inequality is due to $\|u\|_\infty\leq\|u\|$ for all $u\in\mathbb{R}^d$ and the triangle inequality. Also, by (37), (42), and (44), one can see that

$$
\begin{aligned}
\frac{\sigma}{4}\|w^{t+1}-u_i^{t+1}\|^2\leq\frac{\sigma}{2}\|w^{t+1}-\tilde{w}^*\|^2+\frac{\sigma}{2}\|u_i^{t+1}-\tilde{w}^*\|^2 &\overset{(37)}{\leq} \sum_{i=1}^{n}(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^t\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^t\|^2)\\
&\overset{(44)}{=} S_t \overset{(42)}{\leq} q_r^t\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right]. \qquad (48)
\end{aligned}
$$

Using again (37), (42), and (44), we obtain that

$$
\begin{aligned}
\frac{1}{2}(\sum_{i=1}^{n}\rho_i\|w^{t+1}-u_i^t\|)^2\leq(\sum_{i=1}^{n}\rho_i)(\sum_{i=1}^{n}\frac{\rho_i}{2}\|w^{t+1}-u_i^t\|^2) &\overset{(37)}{\leq} (\sum_{i=1}^{n}\rho_i)\sum_{i=1}^{n}(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^t\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^t\|^2)\\
&\overset{(44)}{=} (\sum_{i=1}^{n}\rho_i)S_t \overset{(42)}{\leq} (\sum_{i=1}^{n}\rho_i)q_r^t\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right], \qquad (49)
\end{aligned}
$$

where the first inequality is due to the Cauchy-Schwarz inequality. Combining (47) with (48) and (49), we obtain that

$$
\varepsilon_{t+1}+\sum_{i=1}^{n}\tilde{\varepsilon}_{i,t+1}
$$

$$\leq (n+1)q^t + \left(\frac{2}{\sqrt{\sigma}}\sum_{i=1}^n (L_{\nabla F}+\rho_i) + \sqrt{2\sum_{i=1}^n \rho_i}\right)\left[S_0 + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right]^{1/2} q_r^{t/2}. \quad (50)$$

Recall from Algorithm 2 and (33) that $(u_i^0, \lambda_i^0) = (\tilde{w}^0, -\nabla F_i(\tilde{w}^0))$ and $\lambda_i^* = -\nabla F_i(\tilde{w}^*)$. By these and (44), one has

$$S_0 = \sum_{i=1}^n \left(\frac{\rho_i}{2}\|\tilde{w}^* - \tilde{w}^0\|^2 + \frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*) - \nabla F_i(\tilde{w}^0)\|^2\right). \quad (51)$$

For convenience, denote

$$b = \left(\frac{2}{\sqrt{\sigma}}\sum_{i=1}^n (L_{\nabla F}+\rho_i) + \sqrt{2\sum_{i=1}^n \rho_i}\right) \times \left[\sum_{i=1}^n \left(\frac{\rho_i}{2}\|\tilde{w}^* - \tilde{w}^0\|^2 + \frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*) - \nabla F_i(\tilde{w}^0)\|^2\right) + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right]^{1/2}.$$

Using this, (50), and (51), we obtain that

$$\varepsilon_{t+1} + \sum_{i=1}^n \tilde{\varepsilon}_{i,t+1} \leq (n+1)q^t + bq_r^{t/2} \leq (n+1+b)q_r^{t/2}.$$

where the last inequality is due to $q \leq q_r < 1$. This along with the termination criterion in (19) implies that the number of iterations of Algorithm 2 is bounded above by

$$\left\lceil \frac{2\log(\tau/(n+1+b))}{\log q_r} \right\rceil + 1 = \mathcal{O}(|\log \tau|). \quad (52)$$

Hence, the conclusion of this theorem holds as desired. □

We observe from the proof of Theorem 4.2 that under Assumption 3, the number of iterations of Algorithm 2 is bounded by the quantity in (52).

B Proof of the main results in Section 3

With the abbreviations in (3), we define the Lagrangian function associated with problems (1) and (4) as

$$l(w,\mu) = \begin{cases} f(w) + h(w) + \langle \mu, c(w)\rangle & \text{if } w \in \mathrm{dom}(h) \text{ and } \mu \geq 0, \\ -\infty & \text{if } w \in \mathrm{dom}(h) \text{ and } \mu \ngeq 0, \\ \infty & \text{if } w \notin \mathrm{dom}(h), \end{cases}$$

Then one can verify that (e.g., see equation (17) in Lu & Zhou (2023))

$$\partial l(w,\mu) = \begin{cases} \begin{pmatrix} \nabla f(w) + \partial h(w) + \nabla c(w)\mu \\ c(w) - \mathcal{N}_{\mathbb{R}^m_+}(\mu) \end{pmatrix} & \text{if } w \in \mathrm{dom}(h) \text{ and } \mu \geq 0, \\ \emptyset & \text{otherwise.} \end{cases} \quad (53)$$

We also define the set-valued operator associated with problems (1) and (4) as

$$\mathcal{T}_l : (w,\mu) \to \{(u,\nu) \in \mathbb{R}^d \times \mathbb{R}^m : (u,-\nu) \in \partial l(w,\mu)\}, \quad \forall (w,\mu) \in \mathbb{R}^d \times \mathbb{R}^m. \quad (54)$$

In view of (53), (54), and the definition of KKT solution in (5), we observe that finding an KKT solution of problems (1) and (4) is equivalent to solving the inclusion problem (see (Lu & Zhou (2023))):

$$\text{Find} \quad (w,\mu) \in \mathbb{R}^d \times \mathbb{R}^m \quad \text{such that} \quad (0,0) \in \mathcal{T}_l(w,\mu). \quad (55)$$

Let $f_0(w) \equiv 0$ throughout this section. From Lemma 1 in Lu & Zhou (2023), one can observe that

$$\nabla P_{i,k}(w) = \nabla f_i(w) + \nabla c_i(w)[\mu_i^k + \beta c_i(w)]_+ + \frac{1}{(n+1)\beta}(w - w^k), \quad \forall 0 \leq i \leq n. \quad (56)$$

B.1 Proof of Lemma 3.1

Proof of Lemma 3.1. Fix an arbitrary $w \in \mathbb{R}^d$ and a bounded open set $\mathcal{U}_w$ containing w. We suppose that ∇f_i is $L_{w,1}$-Lipschitz continuous on $\mathcal{U}_w$, and ∇c_i is $L_{w,2}$-Lipschitz continuous on $\mathcal{U}_w$. Also, let $U_{w,1} = \sup_{w\in\mathcal{U}_w}\|c_i(w)\|$ and $U_{w,2} = \sup_{w\in\mathcal{U}_w}\|\nabla c_i(w)\|$. By (10), (11), and (56) one has for each $0 \le i \le n$ and $u, v \in \mathcal{U}_w$ that

$$\begin{aligned}\|\nabla P_{i,k}(u) - \nabla P_{i,k}(v)\| &\overset{(56)}{\le} \|\nabla f_i(u) - \nabla f_i(v)\| + \|\nabla c_i(u) - \nabla c_i(v)\|\|[\mu_i^k + \beta c_i(u)]_+\| \\ &\quad + \|[\mu_i^k + \beta c_i(u)]_+ - [\mu_i^k + \beta c_i(v)]_+\|\|\nabla c_i(v)\| + \frac{1}{(n+1)\beta}\|u - v\| \\ &\le L_{w,1}\|u - v\| + (\|\mu_i^k\| + \beta U_{w,1})L_{w,2}\|u - v\| \\ &\quad + \beta\|c_i(u) - c_i(v)\|\|\nabla c_i(v)\| + \frac{1}{(n+1)\beta}\|u - v\| \\ &\le \left[L_{w,1} + (\|\mu_i^k\| + \beta U_{w,1})L_{w,2} + \beta U_{w,2}^2 + \frac{1}{(n+1)\beta}\right]\|u - v\|.\end{aligned}$$

Therefore, $\nabla P_{i,k}(u)$ is locally Lipschitz continuous on $\mathbb{R}^d$, and the conclusion holds as desired. □

B.2 Proof of Theorem 3.1

Proof of Theorem 3.1. Notice from (3) and (6) that

$$\ell_k(w) = f(w) + h(w) + \frac{1}{2\beta}\left(\|[\mu^k + \beta c(w)]_+\|^2 - \|\mu^k\|^2\right) + \frac{1}{2\beta}\|w - w^k\|^2.$$

By this, (53), and the fact that $\mu^{k+1} = [\mu^k + \beta c(w^{k+1})]_+$, one has

$$\begin{aligned}\partial\ell_k(w^{k+1}) - \frac{1}{\beta}(w^{k+1} - w^k) &= \nabla f(w^{k+1}) + \partial h(w^{k+1}) + \nabla c(w^{k+1})[\mu^k + \beta c(w^{k+1})]_+ \\ &= \nabla f(w^{k+1}) + \partial h(w^{k+1}) + \nabla c(w^{k+1})\mu^{k+1} = \partial_w l(w^{k+1}, \mu^{k+1}). \qquad (57)\end{aligned}$$

Using similar arguments as for the second relation of equation (52) in Lu & Zhou (2023), we obtain that

$$\frac{1}{\beta}(\mu^{k+1} - \mu^k) \in \partial_\mu l(w^{k+1}, \mu^{k+1}). \qquad (58)$$

In view of this, (7), (8), and (57), one can see that

$$\begin{aligned}&\mathrm{dist}_\infty(0, \partial_w l(w^{k+1}, \mu^{k+1})) \overset{(57)}{\le} \mathrm{dist}_\infty(0, \partial\ell_k(w^{k+1})) + \frac{1}{\beta}\|w^{k+1} - w^k\|_\infty \overset{(7)}{\le} \tau_k + \frac{1}{\beta}\|w^{k+1} - w^k\|_\infty \overset{(8)}{\le} \epsilon_1, \\ &\mathrm{dist}_\infty(0, \partial_\mu l(w^{k+1}, \mu^{k+1})) \overset{(58)}{\le} \frac{1}{\beta}\|\mu^{k+1} - \mu^k\|_\infty \overset{(8)}{\le} \epsilon_2.\end{aligned}$$

These along with (53) and Definition 1 imply that (w^{k+1}, μ^{k+1}) is an (ϵ_1, ϵ_2)-KKT solution of problem (1), which proves this theorem as desired. □

B.3 Proof of Lemma 3.2

Recall that $\mathbb{K}$ is a subset of nonnegative integers defined in Lemma 3.2. We define $\mathbb{K} - 1 = \{k - 1 : k \in \mathbb{K}\}$, and for any $0 \le k \in \mathbb{K} - 1$, define

$$w_*^k = \arg\min_w \ell_k(w), \qquad \mu_*^k = [\mu^k + \beta c(w_*^k)]_+. \qquad (59)$$

The following lemma shows that the update from (w^k, μ^k) to (w^{k+1}, μ^{k+1}) can be viewed as applying an inexact proximal point algorithm (PPA) to the inclusion problem (55). Its proof can be found in Lemma 5 in Lu & Zhou (2023).

Lemma B.1. *Let $\{(w^k,\mu^k)\}_{k\in\mathbb{K}}$ be generated by Algorithm 1, where $\mathbb{K}$ is defined in Lemma 3.2. Then for any $k\in\mathbb{K}$, we have*

$$\|(w^{k+1},\mu^{k+1})-\mathcal{J}_\beta(w^k,\mu^k)\|\le\beta\tau_k,$$

where $\mathcal{J}_\beta=(\mathcal{I}+\beta\mathcal{T}_l)^{-1}$, $\mathcal{I}$ is the identity mapping, and $\mathcal{T}_l$ is defined in (54).

The following lemma establishes some properties of (w^k,μ^k) and (w_*^k,μ_*^k). Its proof can be found in Lemma 14 in Lu & Mei (2023).

Lemma B.2. *Let $\{(w^k,\mu^k)\}_{k\in\mathbb{K}}$ be generated by Algorithm 1, where $\mathbb{K}$ is defined in Lemma 3.2. Let (w_*^k,μ_*^k) be defined in (59) for all $0\le k\le\mathbb{K}-1$. Then the following relations hold.*

$$\|(w^k,\mu^k)-(w_*^k,\mu_*^k)\|^2+\|(w_*^k,\mu_*^k)-(w^*,\mu^*)\|^2\le\|(w^k,\mu^k)-(w^*,\mu^*)\|^2,\quad\forall 0\le k\le\mathbb{K}-1,$$
$$\|(w^k,\mu^k)-(w^*,\mu^*)\|\le\|(w^0,\mu^0)-(w^*,\mu^*)\|+\beta\sum_{j=0}^{k-1}\tau_j,\quad\forall 0\le k\in\mathbb{K}.$$

Notice from Algorithm 1 that $\tau_k=\bar{s}/(k+1)^2$ for all $k\ge0$. Therefore, one has $\sum_{j=0}^{\infty}\tau_j\le2\bar{s}$. In view of this, (12), and Lemma B.2, we observe that

$$\|w^k-w^*\|\le r_0+2\bar{s}\beta,\quad\|\mu^k-\mu^*\|\le r_0+2\bar{s}\beta,\quad\forall 0\le k\in\mathbb{K},\tag{60}$$
$$\|w^k-w_*^k\|\le r_0+2\bar{s}\beta,\quad\|w_*^k-w^*\|\le r_0+2\bar{s}\beta,\quad\forall 0\le k\in\mathbb{K}-1.\tag{61}$$

where r_0 is defined in (12), and β and $\bar{s}$ are inputs of Algorithm 1. The first relation in (60) leads to the conclusion that $w^k\in\mathcal{Q}_1$ for all $k\in\mathbb{K}$, which immediately implies that Lemma 3.2 holds.

B.4 Proof of Theorem 3.2

We provide a technical lemma concerning the convergence rate of an inexact PPA applied to a monotone inclusion problem. Its proof can be found in Lemma 3 in Lu & Zhou (2023).

Lemma B.3. *Let $\mathcal{T}:\mathbb{R}^p\rightrightarrows\mathbb{R}^q$ be a maximally monotone operator and $z^*\in\mathbb{R}^p$ such that $0\in\mathcal{T}(z^*)$. Let $\{z^k\}$ be a sequence generated by an inexact PPA, starting with z^0 and obtaining z^{k+1} be approximately evaluating $\mathcal{J}_\beta(z^k)$ such that*

$$\|z^{k+1}-\mathcal{J}_\beta(z^k)\|\le e_k$$

for some $\beta>0$ and $e_k\ge0$, where $\mathcal{J}_\beta=(\mathcal{I}+\beta\mathcal{T})^{-1}$ and $\mathcal{I}$ is the identity operator. Then, for any $K\ge1$, we have

$$\min_{K\le k\le2K}\|z^{k+1}-z^k\|\le\frac{\sqrt{2}\left(\|z^0-z^*\|+2\sum_{k=0}^{2K}e_k\right)}{\sqrt{K+1}}.$$

Recall from (12) that $\mathcal{Q}_1$ is a compact set. We let

$$U_{\nabla f}=\sup_{w\in\mathcal{Q}_1}\max_{1\le i\le n}\|\nabla f_i(w)\|,\quad U_{\nabla c}=\sup_{w\in\mathcal{Q}_1}\max_{0\le i\le n}\|\nabla c_i(w)\|,\quad U_c=\sup_{w\in\mathcal{Q}_1}\max_{0\le i\le n}\|c_i(w)\|.\tag{62}$$

Lemma B.4. *Suppose that Assumption 1 holds, and let $\{w^{k,t+1}\}_{t\in\mathbb{T}_k}$ and $\{u_i^{k,t+1}\}_{1\le i\le n,t\in\mathbb{T}_k}$ be all the iterates generated by Algorithm 2 for solving the subproblem (9) at the kth iteration of Algorithm 1, where $\mathbb{T}_k$ is a consecutive nonnegative integers starting from 0. Then we have $w^{k,t+1}\in\mathcal{Q}_2$ and $u_i^{k,t+1}\in\mathcal{Q}_2$ for all $t\in\mathbb{T}_k$ and $1\le i\le n$, where*

$$\mathcal{Q}_2=\left\{v:\|v-u\|^2\le\frac{(n+1)^3\beta^2}{(1-q^2)}+(n+1)\beta\sum_{i=1}^{n}\left[\rho_i(r_0+2\bar{s}\beta)^2+\frac{4}{\rho_i}U_{\nabla P}^2\right],u\in\mathcal{Q}_1\right\},\tag{63}$$
$$U_{\nabla P}=U_{\nabla f}+\frac{2(r_0+2\bar{s}\beta)}{(n+1)\beta}+U_{\nabla c}\left(\|\mu^*\|+r_0+2\bar{s}\beta+\beta U_c\right).\tag{64}$$

Proof. By (56) and the definition of $P_{i,k}$ in (11), one has for all $w \in \mathcal{Q}_1$ and $1 \le i \le n$ that

$$\begin{aligned}\|\nabla P_{i,k}(w)\| &\overset{(56)}{=} \|\nabla f_i(w) + \nabla c_i(w)[\mu_i^k + \beta c_i(w)]_+ + \frac{1}{(n+1)\beta}(w - w^k)\| \\ &\le \|\nabla f_i(w)\| + \|\nabla c_i(w)\|\|[\mu_i^k + \beta c_i(w)]_+\| + \frac{1}{(n+1)\beta}\|w - w^k\| \\ &\overset{(62)}{\le} U_{\nabla f} + U_{\nabla c}(\|\mu_i^*\| + \|\mu_i^k - \mu_i^*\| + \beta U_c) + \frac{1}{(n+1)\beta}(\|w - w^*\| + \|w^k - w^*\|) \\ &\overset{(12)(60)}{\le} U_{\nabla f} + U_{\nabla c}\left(\|\mu^*\| + r_0 + 2\bar{s}\beta + \beta U_c\right) + \frac{2(r_0 + 2\bar{s}\beta)}{(n+1)\beta} \overset{(64)}{=} U_{\nabla P}. \end{aligned} \tag{65}$$

Recall from Section 3 that Algorithm 2 with $(\tilde{w}^0, \tau) = (w^k, \tau_k)$ is applicable to the subproblem (9). In addition, recall that the subproblem (9) has an optimal solution w_*^k (see (59)), $P_{i,k}$, $0 \le i \le n$, are strongly convex with modulus $1/[(n+1)\beta]$. By Lemma 4.1 with $(F_i, \tilde{w}^*, \tilde{w}^0, \sigma) = (P_{i,k}, w_*^k, w^k, 1/[(n+1)\beta])$, we obtain that $w^{k,t+1} \in \widetilde{\mathcal{Q}}$ and $u_i^{k,t+1} \in \widetilde{\mathcal{Q}}$ for all $t \in \mathbb{T}_k$ and $1 \le i \le n$, where

$$\widetilde{\mathcal{Q}} = \left\{ v : \|v - w_*^k\|^2 \le \frac{(n+1)^3\beta^2}{(1-q^2)} + (n+1)\beta \sum_{i=1}^n \left(\rho_i \|w_*^k - w^k\|^2 + \frac{1}{\rho_i}\|\nabla P_{i,k}(w_*^k) - \nabla P_{i,k}(w^k)\|^2 \right) \right\}. \tag{66}$$

Notice from (61) that $\|w_*^k - w^k\| \le r_0 + 2\bar{s}\beta$. It follows from (12), (60), and (61) that $w^k, w_*^k \in \mathcal{Q}_1$. By these, (65), and (66), one has that

$$\widetilde{\mathcal{Q}} \subseteq \left\{ v : \|v - w_*^k\|^2 \le \frac{(n+1)^3\beta^2}{(1-q^2)} + (n+1)\beta \sum_{i=1}^n \left[\rho_i (r_0 + 2\bar{s}\beta)^2 + \frac{4}{\rho_i} U_{\nabla P}^2 \right] \right\}.$$

This along with (63) and the fact that $w_*^k \in \mathcal{Q}_1$ implies that the conclusion of this lemma holds as desired. □

Let $L_{\nabla f,2}$ be the Lipschitz constant of ∇f_i, $1 \le i \le n$, on $\mathcal{Q}_2$, and $L_{\nabla c,2}$ be the Lipschitz constant of ∇c_i, $0 \le i \le n$, on $\mathcal{Q}_2$. Also, we let

$$U_{\nabla c,2} = \sup_{w \in \mathcal{Q}_2} \max_{0 \le i \le n} \|\nabla c_i(w)\|, \quad U_{c,2} = \sup_{w \in \mathcal{Q}_2} \max_{0 \le i \le n} \|c_i(w)\|. \tag{67}$$

We define

$$L_{\nabla P,2} = L_{\nabla f,2} + (\|\mu^*\| + r_0 + 2\bar{s}\beta + \beta U_{c,2}) L_{\nabla c,2} + \beta U_{\nabla c,2}^2 + \frac{1}{(n+1)\beta}. \tag{68}$$

By the local Lipschitz continuity of ∇f_i, $1 \le i \le n$, and ∇c_i, $0 \le i \le n$, and a similar argument as in the proof of Lemma A.2, one can observe that $L_{\nabla f,2}$, $L_{\nabla c,2}$, and $L_{\nabla P,2}$ are well-defined.

To proceed, we next show that $\nabla P_{i,k}$, $0 \le i \le n$, are $L_{\nabla P,2}$-Lipschitz continuous on $\mathcal{Q}_2$. By the definitions of $L_{\nabla f,2}$ and $L_{\nabla c,2}$, (10), (11), (56), (60), (67), and (68), one has that for all $u, v \in \mathcal{Q}_2$ and $0 \le i \le n$,

$$\begin{aligned}\|\nabla P_{i,k}(u) - \nabla P_{i,k}(v)\| &\overset{(56)}{\le} \|\nabla f_i(u) - \nabla f_i(v)\| + \|[\mu_i^k + \beta c_i(u)]_+\|\|\nabla c_i(u) - \nabla c_i(v)\| \\ &\quad + \|[\mu_i^k + \beta c_i(u)]_+ - [\mu_i^k + \beta c_i(v)]_+\|\|\nabla c_i(v)\| + \frac{1}{(n+1)\beta}\|u - v\| \\ &\overset{(67)}{\le} L_{\nabla f,2}\|u - v\| + (\|\mu_i^*\| + \|\mu_i^k - \mu_i^*\| + \beta U_{c,2}) L_{\nabla c,2}\|u - v\| \\ &\quad + \beta U_{\nabla c,2}^2\|u - v\| + \frac{1}{(n+1)\beta}\|u - v\| \\ &\overset{(60)}{\le} \left[L_{\nabla f,2} + (\|\mu^*\| + r_0 + 2\bar{s}\beta + \beta U_{c,2}) L_{\nabla c,2} + \beta U_{\nabla c,2}^2 + \frac{1}{(n+1)\beta} \right] \|u - v\| \\ &\overset{(68)}{=} L_{\nabla P,2}\|u - v\|. \end{aligned} \tag{69}$$

Proof of Theorem 3.2. We first derive an upper bound for the number of outer iterations of Algorithm 1. Recall that $\sum_{j=0}^{\infty}\tau_j = 2\bar{s}$. It follows from Lemmas B.1 and B.3 that

$$\begin{aligned}\min_{K\le k\le 2K}\frac{1}{\beta}\|(w^{k+1},\mu^{k+1})-(w^k,\mu^k)\| &\le \frac{\sqrt{2}\left(\|(w^0,\mu^0)-(w^*,\mu^*)\|+2\beta\sum_{j=0}^{\infty}\tau_j\right)}{\beta\sqrt{K+1}}\\ &\le \frac{\sqrt{2}\left(\|(w^0,\mu^0)-(w^*,\mu^*)\|+4\bar{s}\beta\right)}{\beta\sqrt{K+1}} = \frac{\sqrt{2}\,(r_0+4\bar{s}\beta)}{\beta\sqrt{K+1}},\end{aligned}$$

which then implies that

$$\begin{aligned}&\min_{K\le k\le 2K}\left\{\tau_k+\frac{1}{\beta}\|w^{k+1}-w^k\|_\infty\right\} \le \frac{\bar{s}}{(K+1)^2}+\frac{\sqrt{2}\,(r_0+4\bar{s}\beta)}{\beta\sqrt{K+1}} \le \left[\bar{s}+\frac{\sqrt{2}\,(r_0+4\bar{s}\beta)}{\beta}\right]\frac{1}{\sqrt{K+1}},\\ &\min_{K\le k\le 2K}\frac{1}{\beta}\|\mu^{k+1}-\mu^k\|_\infty \le \frac{\sqrt{2}\,(r_0+4\bar{s}\beta)}{\beta\sqrt{K+1}}.\end{aligned}$$

We see from these and the termination criterion in (8) that the number of outer iterations of Algorithm 1 is at most

$$K_{\epsilon_1,\epsilon_2} = 2\left[\bar{s}+\frac{\sqrt{2}(r_0+4\bar{s}\beta)}{\beta}\right]^2\max\{\epsilon_1^{-2},\epsilon_2^{-2}\} = \mathcal{O}(\max\{\epsilon_1^{-2},\epsilon_2^{-2}\}).$$

We next derive an upper bound for the total number of inner iterations of Algorithm 1. Recall from (10) and (11) that $P_{i,k}$, $0\le i\le n$, are strongly convex with modulus $1/[(n+1)\beta]$. In addition, notice from Lemma 3.1 that $P_{i,k}$, $0\le i\le n$, are locally Lipschitz continuous on $\mathbb{R}^d$. Therefore, Algorithm 2 is applicable to the subproblem (9).

From Lemma B.4, we see that all iterates generated by Algorithm 2 for solving (9) lie in $\mathcal{Q}_2$. Also, in view of (69), we see that $\nabla P_{i,k}$, $1\le i\le n$, are $L_{\nabla P,2}$-Lipschitz continuous on $\mathcal{Q}_2$. Therefore, by Theorem 4.2 with $(\tau, F_i, \sigma, L_{\nabla F}, \tilde{w}^*, \tilde{w}^0) = (\tau_k, P_{i,k}, 1/[(n+1)\beta], L_{\nabla P,2}, w_*^k, w^k)$ and the discussion at the end of Appendix A.3, one can see that the number of iterations of Algorithm 2 for solving (9) is no more than

$$T_k = \left\lceil\frac{2\log(\tau_k/(n+1+b_k))}{\log\tilde{q}_r}\right\rceil+1 \tag{70}$$

where

$$\begin{aligned}&\tilde{q}_r = \max\left\{q,\frac{1}{1+\tilde{r}}\right\},\quad \tilde{r} = \min_{1\le i\le n}\left\{\frac{\rho_i}{(n+1)\beta(\rho_i^2+2L_{\nabla P,2}^2)}\right\},\\ &b_k = \left(2\sqrt{(n+1)\beta}\sum_{i=1}^n(L_{\nabla P,2}+\rho_i)+\sqrt{2(\sum_{i=1}^n\rho_i)}\right)\\ &\times\left[\sum_{i=1}^n\left(\frac{\rho_i}{2}\|w_*^k-w^k\|^2+\frac{1}{2\rho_i}\|\nabla P_{i,k}(w_*^k)-\nabla P_{i,k}(w^k)\|^2\right)+\frac{1}{1-q}\left(\frac{(n+1)^2\beta}{2}+\frac{1}{(n+1)\beta}\sum_{i=1}^n\frac{1}{\rho_i^2+2L_{\nabla P,2}^2}\right)\right]^{1/2}.\end{aligned}$$

Recall from (60), (61), and the definitions of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ that $w_*^k, w^k\in\mathcal{Q}_1\subseteq\mathcal{Q}_2$. It then follows that

$$\frac{\rho_i}{2}\|w_*^k-w^k\|^2+\frac{1}{2\rho_i}\|\nabla P_{i,k}(w_*^k)-\nabla P_{i,k}(w^k)\|^2 \overset{(69)}{\le} \frac{\rho_i^2+L_{\nabla P,2}^2}{2\rho_i}\|w_*^k-w^k\|^2 \overset{(61)}{\le} \frac{\rho_i^2+L_{\nabla P,2}^2}{2\rho_i}(r_0+2\bar{s}\beta)^2.$$

Then one has $b_k\le\bar{b}$, where

$$\bar{b} = \left(2\sqrt{(n+1)\beta}\sum_{i=1}^n(L_{\nabla P,2}+\rho_i)+\sqrt{2(\sum_{i=1}^n\rho_i)}\right)$$

$$\times \left[\sum_{i=1}^{n} \frac{\rho_i^2 + L_{\nabla P,2}^2}{2\rho_i}(r_0 + 2\bar{s}\beta)^2 + \frac{1}{1-q}\left(\frac{(n+1)^2\beta}{2} + \frac{1}{(n+1)\beta}\sum_{i=1}^{n}\frac{1}{\rho_i^2 + 2L_{\nabla P,2}^2}\right)\right]^{1/2}.$$

By $b_k \leq \bar{b}$, $\tau_k = \bar{s}/(k+1)^2$, $k \leq K_{\epsilon_1,\epsilon_2}$, and (70), one has that

$$T_k \leq \left\lceil \frac{2\log((n+1+\bar{b})(K_{\epsilon_1,\epsilon_2}+1)^2/\bar{s})}{\log(\tilde{q}_r^{-1})} \right\rceil + 1.$$

Therefore, by $K_{\epsilon_1,\epsilon_2} = \mathcal{O}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$, one can see that the total number of inner iterations of Algorithm 1 is at most

$$\sum_{k=0}^{K_{\epsilon_1,\epsilon_2}} T_k \leq (K_{\epsilon_1,\epsilon_2}+1)\left(\left\lceil \frac{2\log((n+1+\bar{b})(K_{\epsilon_1,\epsilon_2}+1)^2/\bar{s})}{\log(\tilde{q}_r^{-1})} \right\rceil + 1\right) = \widetilde{\mathcal{O}}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\}).$$

This completes the proof as desired. □

C A centralized proximal AL method

In this part, we present a centralized proximal AL method (see Algorithm 2 in Lu & Zhou (2023)).

Algorithm 3 A centralized proximal AL method for solving problem (1)

Input: tolerances $\epsilon_1, \epsilon_2 \in (0,1)$, $w^0 \in \mathrm{dom}(h)$, $\mu^0 \geq 0$, and $\beta > 0$.

for $k = 0, 1, 2, \ldots$ **do**

Find an approximate solution w^{k+1} to the proximal AL subproblem:

$$\min_w \left\{ \ell_k(w) = f(w) + \frac{1}{2\beta}\left(\|[\mu^k + \beta c(w)]_+\|^2 - \|\mu^k\|^2\right) + \frac{1}{2\beta}\|w - w^k\|^2 \right\}$$

such that

$$\mathrm{dist}_\infty(0, \partial\ell_k(w^{k+1})) \leq \tau_k.$$

Update the Lagrangian multiplier:

$$\mu^{k+1} = [\mu^k + \beta c(w^{k+1})]_+.$$

Output (w^{k+1}, μ^{k+1}) and terminate the algorithm if

$$\|w^{k+1} - w^k\|_\infty + \beta\tau_k \leq \beta\epsilon_1, \qquad \|\mu^{k+1} - \mu^k\|_\infty \leq \beta\epsilon_2.$$

end for

D Additional numerical results

D.1 Dataset description for Neyman-Pearson classification

In this part, we describe the datasets for Neyman-Pearson classification in Section 5.2. 'breast-cancer-wisc', 'adult-a', and 'monks-1' are three binary classification datasets. We present the total number of samples for class 0 and class 1 and the number of features.

D.2 Comparison between constrained and unconstrained classification

In this part, we present a comparison between the performance of constrained and unconstrained classification over all local clients.

Table 4: Datasets for Neyman-Pearson classification

dataset	class 0/class 1	feature dimension
breast-cancer-wisc	455/240	20
adult-a	24715/7840	21
monks-1	275/275	21

In particular, we first compare the Neyman-Pearson classification and the unconstrained classification model:

$$\min_w \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_{i0}+m_{i1}}\left[\sum_{j=1}^{m_{i0}}\phi(w;(x_j^{(i0)},0))+\sum_{j=1}^{m_{i1}}\phi(w;(x_j^{(i1)},1))\right], \tag{71}$$

where ϕ is the logistic loss defined in (27). The first three subplots in Figure 3 summarize the losses for class 0 and class 1 across all clients for three datasets: 'breast-cancer-wisc', 'adult-a', and 'monks-1' with client numbers of 5, 10, and 20. We observe that when solving the unconstrained classification problem (71), the loss for class 1 is significantly higher than the loss for class 0, and the max loss for class 1 can be much higher than the average loss for class 1. By solving the constrained classification problem (26), we can regulate the loss for class 1 across all local clients to remain below a predefined threshold.

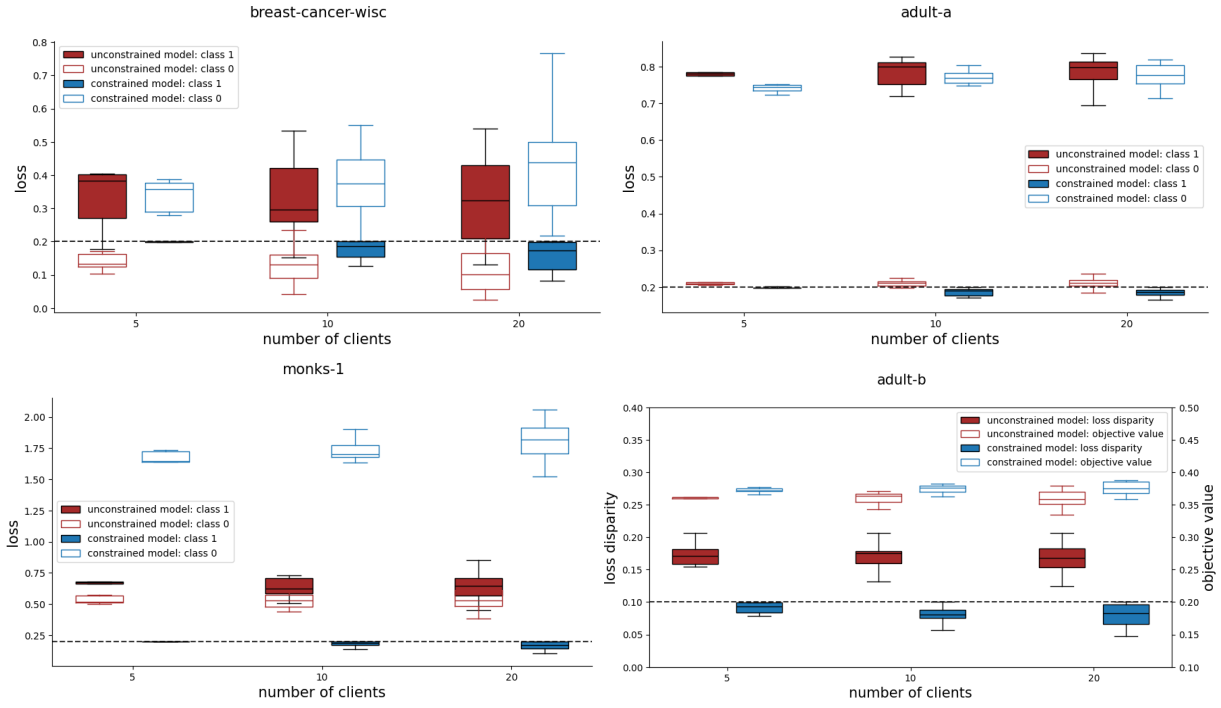

Figure 3: Constrained classification vs. Unconstrained classification

Next, we compare the classification with fairness constraints and the unconstrained classification model:

$$\min_w \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\phi(w;(x_j^{(i)},y_j^{(i)})), \tag{72}$$

where ϕ is the logistic loss defined in (27). The last subplot in Figure 3 summarizes the classification loss and loss disparity across all clients and the central server for the dataset 'adult-b' with client numbers of 5, 10, and 20. We observe that that by solving the classification problem with fairness constraints in (28), we can effectively mitigate the loss disparity among all clients and the central server. This can substantially improve the results obtained from solving the unconstrained classification problem (72).