

Decentralized Feature-Distributed Optimization for Generalized Linear Models

Anonymous authors
Paper under double-blind review

Abstract

We consider the “all-for-one” decentralized learning problem for generalized linear models. The features of each sample are partitioned among several collaborating agents in a connected network, but only one agent observes the response variables. To solve the regularized empirical risk minimization in this distributed setting, we apply the Chambolle–Pock primal–dual algorithm to an equivalent saddle-point formulation of the problem. The primal and dual iterations are either in closed-form or reduce to coordinate-wise minimization of scalar convex functions. We establish convergence rates for the empirical risk minimization under two different assumptions on the loss function (Lipschitz and square root Lipschitz), and show how they depend on the characteristics of the design matrix and the Laplacian of the network.

1 Introduction

Let $\ell: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ denote a given *sample loss function* that is convex and, for simplicity, differentiable in its first argument. Given data points $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ and a convex regularization function $r(\cdot)$, we consider the minimization of regularized empirical risk in generalized linear models, i.e.,

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{x}_i^\top \boldsymbol{\theta}, y_i) + r(\boldsymbol{\theta}),$$

in a *non-standard* distributed setting where the data features, rather than samples, are distributed among m agents that communicate through a connected network.

The problem can be formally stated as follows. With $\mathcal{A}_1, \dots, \mathcal{A}_m$ denoting a partition of $[d] \stackrel{\text{def}}{=} \{1, \dots, d\}$ into m disjoint blocks, each agent $j \in [m]$ observes the *local features* $\mathbf{x}_{j,i} \stackrel{\text{def}}{=} (\mathbf{x}_i)_{\mathcal{A}_j} \in \mathbb{R}^{d_j}$ for every $i \in [n]$, where $(\mathbf{u})_{\mathcal{A}}$ denotes the restriction of \mathbf{u} to the coordinates enumerated by the index set \mathcal{A} . Without loss of generality we may assume that each \mathcal{A}_j is a set of d_j consecutive indices and simply write¹

$$\mathbf{x}_i = [\mathbf{x}_{1,i}; \ \cdots; \ \mathbf{x}_{m,i}].$$

We also denote the $n \times d_j$ *local design matrix* for agent $j \in [m]$ by

$$\mathbf{X}_j = [\mathbf{x}_{j,1} \ \cdots \ \mathbf{x}_{j,n}]^\top,$$

and the full $n \times d$ design matrix by

$$\mathbf{X} = [\mathbf{X}_1 \ \cdots \ \mathbf{X}_m] = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_n]^\top.$$

We assume that only one of the agents, say the first agent, observes the response $(y_i)_{i=1}^n$ and the other agents only have access to their local features. There is an underlying communication network which can be abstracted by a *connected* undirected graph G over the vertex set $\mathcal{V} = [m]$. If distinct agents j and

¹We denote the vertical concatenations using semicolons as the delimiters.

j' can communicate directly, then they are adjacent in G and we write $j \sim_G j'$. The *Laplacian* of the communication graph, which is central in the distributed computations of the optimization algorithms, is denoted by \mathbf{L} .

Using the shorthand

$$\ell_i(\cdot) \stackrel{\text{def}}{=} \ell(\cdot, y_i)$$

that we use henceforth to simplify the notation, we seek an approximation to the (regularized) *empirical risk minimizer*

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i(\mathbf{x}_i^\top \boldsymbol{\theta}) + r(\boldsymbol{\theta}). \quad (1)$$

where the regularizer $r(\cdot)$ is typically used to induce a certain structure (e.g., sparsity) in $\hat{\boldsymbol{\theta}}$.

To solve this optimization in our distributed setting, we use a primal–dual formulation that accommodates local calculations. Specifically, with $\ell_i^*: \mathbb{R} \rightarrow \mathbb{R}$ denoting the *convex conjugate* of the function $\ell_i(\cdot)$, the minimization in (1) can be formulated as the saddle-point problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{\lambda}_1 \in \mathbb{R}^n} \frac{1}{n} \boldsymbol{\lambda}_1^\top \mathbf{X} \boldsymbol{\theta} - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) + r(\boldsymbol{\theta}),$$

where $\boldsymbol{\lambda}_1 = [\lambda_{1,1}; \dots; \lambda_{1,n}]$ is the dual variable. The regularizer $r(\boldsymbol{\theta})$ might also be represented using its conjugate, making the objective of the resulting saddle-point problem linear in the primal variable $\boldsymbol{\theta}$. However, to avoid the need for the “dualization” of the regularizer, we focus on the special but important case that the regularizer is *separable* with respect to the agents. Partitioning the coordinates of the primal variable $\boldsymbol{\theta}$ according to the partitioning of the features among the agents as

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_1; \dots; \boldsymbol{\theta}_m],$$

with $\boldsymbol{\theta}_j \in \mathbb{R}^{d_j}$, we assume that the regularizer takes the form

$$r(\boldsymbol{\theta}) = \sum_{j=1}^m r_j(\boldsymbol{\theta}_j), \quad (2)$$

where for each $j \in [m]$ the convex functions $r_j(\cdot)$ have a simple *proximal mapping* that is available to the j th agent. Giving each agent its own version of the dual variable denoted by $\boldsymbol{\lambda}_j \in \mathbb{R}^n$, we can express (1) in a form which is amenable to distributed computations as

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m \in \mathbb{R}^n} \sum_{j=1}^m r_j(\boldsymbol{\theta}_j) + \frac{1}{n} \boldsymbol{\lambda}_j^\top \mathbf{X}_j \boldsymbol{\theta}_j - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) \\ \text{subject to } \mathbf{L} [\boldsymbol{\lambda}_1 \ \dots \ \boldsymbol{\lambda}_m]^\top = \mathbf{0}. \end{aligned} \quad (3)$$

The constraint involving the Laplacian simply enforces $\boldsymbol{\lambda}_j = \boldsymbol{\lambda}_{j'}$ for all $j \sim_G j'$. With $\langle \cdot, \cdot \rangle$ denoting the usual (Frobenius) inner product henceforth, we can use the Lagrangian form of the inner optimization to express (3) equivalently as

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m \in \mathbb{R}^n} \min_{\mathbf{V} \in \mathbb{R}^{n \times m}} \frac{1}{n} \langle \mathbf{V}^\top, \mathbf{L} [\boldsymbol{\lambda}_1 \ \dots \ \boldsymbol{\lambda}_m]^\top \rangle + \sum_{j=1}^m r_j(\boldsymbol{\theta}_j) + \frac{1}{n} \boldsymbol{\lambda}_j^\top \mathbf{X}_j \boldsymbol{\theta}_j - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) \\ = \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \min_{\mathbf{V} \in \mathbb{R}^{n \times m}} \max_{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m \in \mathbb{R}^n} \sum_{j=1}^m r_j(\boldsymbol{\theta}_j) + \frac{1}{n} \boldsymbol{\lambda}_j^\top \mathbf{X}_j \boldsymbol{\theta}_j - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) + \frac{1}{n} \langle \mathbf{V} \mathbf{L}, [\boldsymbol{\lambda}_1 \ \dots \ \boldsymbol{\lambda}_m] \rangle \\ = \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} \max_{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) + \sum_{j=1}^m r_j(\boldsymbol{\theta}_j) + \frac{1}{n} \boldsymbol{\lambda}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j), \end{aligned} \quad (4)$$

where the second line follows from strong duality.

In [Section 2](#) we describe the iterations based on the Chambolle–Pock primal–dual algorithm ([Chambolle & Pock, 2016](#)) to solve the saddle-point problem (4). Our main result and the assumptions under which it holds are provided in [Section 3](#). Some numerical experiments are also provided in [Section 4](#). Proofs of the main result can be found in [Appendix A](#).

1.1 Related work

Minimization of a sum of (convex) functions is the most studied problem in distributed optimization due to its prevalence in machine learning. The most commonly considered setting in the literature is by far the *sample-distributed* setting, where each agent merely has access to one of the summands of the objective function that can be computed using the locally available samples. The literature primarily considers two different communication models. *Centralized* first-order methods have a main computing agent that aggregates the local (sub)gradient evaluations of the other agents, updates the iterate and sends it back to the other agents. Therefore, the communication time for these methods grows linearly with the *diameter* of the underlying network. In contrast, *decentralized* first-order methods do not rely on a single aggregating agent; every agent maintains and updates a local copy of the candidate minimizer through local computations and communications with its immediate neighbors, and consistency of the solution across agents is achieved either through local averaging or consensus constraints. Due to the diffusion-style nature of the iterations, the convergence rate of these methods depends on a certain notion of *spectral gap* of the communication graph. Many algorithms have been introduced for sample-distributed decentralized convex optimization; surveys of the literature can be found in ([Yang et al., 2019](#); [Gorbunov et al., 2020](#)), and prominent references include ([Johansson et al., 2008](#); [Nedić & Ozdaglar, 2009](#); [Wang & Elia, 2011](#); [Zhu & Martinez, 2012](#); [Duchi et al., 2012](#); [Scaman et al., 2017](#)). In general, the computation and communication complexity of these algorithms to find an ϵ -accurate solution range from the “slow rate” of $O(\epsilon^{-2}) + O(\epsilon^{-1})$ for Lipschitz-continuous convex functions, to the “linear rate” of $O(\log(1/\epsilon))$ for smooth and strongly convex functions. Lower bounds and (nearly) optimal algorithms for a few common objective classes are established in ([Scaman et al., 2019](#)).

The “feature-distributed” setting that we consider is studied to a lesser extent, but has found important applications such as *sensor fusion* ([Sasiadek, 2002](#)) and *cross-silo federated learning* ([Kairouz et al., 2021](#)). This setting is also relevant in *parallelized computing* to amplify the performance of resource limited computing agents in large-scale problems.

Centralized federated learning protocols, in which the agents communicate with a server, with distributed features are proposed in ([Hu et al., 2019](#)) and ([Chen et al., 2020](#)). [Hu et al. \(2019\)](#) proposed the FDML method and, under convexity and smoothness of the objective, established a regret bound for SGD that decays with the number of iterations T at the rate of $O(1/\sqrt{T})$. It is also assumed in this result that the iterates never exit a neighborhood of the true parameter, basically imposing the strong convexity on the objective in an implicit form. [Chen et al. \(2020\)](#) proposed a method called VAFL, in which a server maintains a global parameter and each client operates on local features and parameters that determine the client’s corresponding predictor. The clients and the server communicate in an asynchronous fashion and exchange the value of clients’ predictors and the gradients of the sample loss with respect to these predictors. Under certain models of the communication delays that impose the asynchrony, a variant of stochastic gradient descent is shown to converge at a rate $O(1/T)$ under strong convexity. The performance of VAFL in the case of smooth nonconvex objectives and nonlinear predictors that are separable across the agents is also considered in ([Chen et al., 2020](#)). However, in this general setting where the guarantees are inevitably weaker, only the temporal average of the squared norm of the gradients (in expectation with respect to the SGD samples) are shown to converge at a rate $O(1/\sqrt{T})$.

The CoLa algorithm of [He et al. \(2018\)](#) considers a ubiquitous class of convex minimization problems in machine learning and statistics that involve *linear predictors*, in the decentralized distributed setting. Following the formulation of ([Smith et al., 2018](#)), a pair of convex programs that are dual to each other are considered in ([He et al., 2018](#)) depending on whether the data is split across the samples, or across the features. This latter setting is the closest related work in the literature to the present paper. The main step in each iteration of the CoLa algorithm is a regularized convex quadratic minimization. This minimiza-

tion step is generally nontrivial and needs to be performed by a dedicated subroutine, though the analysis accommodates subroutines that compute inexact solutions. In contrast, our convex-concave saddle point formulation of the problem leads to iterations in which every agent evaluates either a closed-form expression or a simple proximal operator, except for one agent whose computations are as simple as performing one-dimensional strongly convex minimization for each dual coordinate. Furthermore, while our algorithm achieves an accuracy of $O(1/T)$ after T iterations similar to the CoLa (in the general convex setting), our convergence analysis applies to the broader class of square root Lipschitz loss functions, defined below in [Section 3](#), that includes the usual smooth loss functions as special case ([Srebro et al., 2010](#), Lemma 2.1).

[Arablouei et al. \(2015\)](#); [Gratton et al. \(2018\)](#) present algorithms based on ADMM for solving decentralized least-squares problems with distributed features, and establish asymptotic convergence. A feature-decentralized algorithm for logistic regression is presented in ([Slavković et al., 2007](#)), though no convergence guarantees are given.

Finally, the primal-dual algorithm we present in the next section is related to an application of the distributed saddle point algorithm of [Mateos-Núñez & Cortés \(2017\)](#) where the goal is minimizing a sum of functions of independent variables subject to linear inequality constraints (see Remark III.1 in that paper). The general algorithm considered in ([Mateos-Núñez & Cortés, 2017](#)) is a (projected) gradient descent ascent method. Consequently, its convergence analysis relies on certain boundedness assumptions on the iterates and the corresponding gradients. Furthermore, while being applicable to a broader set of saddle point problems than our method, this gradient descent ascent method is only shown to converge at the rate $1/\sqrt{T}$.

1.2 Contributions

Using the dual representation of the loss functions $\ell_i(\cdot)$ in (1), we convert the corresponding minimization problem to a saddle-point problem that enables us to perform the decentralized minimization in the unconventional feature-distributed setting.

Using the Chambolle–Pock primal–dual algorithm as a natural method to solve the resulting saddle-point problem, we provide convergence guarantees for the algorithm in terms of the primal objective (rather than the primal–dual gap). In particular, we show convergence to the minimum primal value at a rate of $1/T$, assuming that the loss functions $\ell_i(\cdot)$ are either Lipschitz or square root Lipschitz smooth. The square root Lipschitz smooth functions include the more common Lipschitz gradient functions.

In each iteration, the updates to the primal variables are either in closed-form or involve evaluation of a simple proximal mapping. Updating the dual variable corresponding to the “main agent” among the n agents, only requires computing the Moreau envelope of n univariate functions that is highly parallelizable and can be solved efficiently. Updating the dual variables for the rest of the agents is even simpler and is expressed in closed-form.

2 The primal–dual algorithm

Let f and g be convex functions such that f is smooth and has a tractable first-order oracle, and the possibly nonsmooth g admits a tractable proximal mapping. Furthermore, let h be a convex function whose *convex conjugate*, denoted by h^* , admits a tractable proximal mapping. The Chambolle–Pock primal–dual algorithm ([Chambolle & Pock, 2016](#)) solves the saddle-point problem

$$\min_{\mathbf{z}} \max_{\boldsymbol{\lambda}} f(\mathbf{z}) + g(\mathbf{z}) + \boldsymbol{\lambda}^\top \mathbf{K} \mathbf{z} - h^*(\boldsymbol{\lambda}),$$

for a given matrix \mathbf{K} . Denoting the columns of \mathbf{V} by $\mathbf{v}_1, \dots, \mathbf{v}_m$, and the Kronecker product by \otimes , the optimization problem (4) fits into the above formulation by choosing

$$\begin{aligned} \mathbf{z} &= [\boldsymbol{\theta}_1; \cdots; \boldsymbol{\theta}_m; \mathbf{v}_1; \cdots; \mathbf{v}_m], \\ \boldsymbol{\lambda} &= [\boldsymbol{\lambda}_1; \cdots; \boldsymbol{\lambda}_m], \end{aligned}$$

$$\mathbf{K} = \frac{1}{n} \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_m \end{bmatrix} \mathbf{L} \otimes \mathbf{I},$$

$$f \equiv 0,$$

$$g(\mathbf{z}) = r(\boldsymbol{\theta}) = \sum_{j=1}^m r_j(\boldsymbol{\theta}_j),$$

and

$$h^*(\boldsymbol{\lambda}) = \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}).$$

The update rule of the Chambolle–Pock algorithm can be summarized as

$$\mathbf{z}_{t+1} = \underset{\mathbf{z} \in \mathbb{R}^{d+mn}}{\operatorname{argmin}} f(\mathbf{z}_t) + \langle \nabla f(\mathbf{z}_t), \mathbf{z} - \mathbf{z}_t \rangle + g(\mathbf{z}) + \boldsymbol{\lambda}_t^\top \mathbf{K} \mathbf{z} + \frac{1}{2\tau} \|\mathbf{z} - \mathbf{z}_t\|_2^2$$

$$\boldsymbol{\lambda}_{t+1} = \underset{\boldsymbol{\lambda} \in \mathbb{R}^{mn}}{\operatorname{argmin}} h^*(\boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top \mathbf{K} (2\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{1}{2\sigma} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t\|_2^2,$$

for appropriately chosen parameters $\tau, \sigma > 0$. Writing this update explicitly for our special case, we have

$$\mathbf{z}_{t+1} = \underset{\mathbf{z} \in \mathbb{R}^{d+mn}}{\operatorname{argmin}} r((\mathbf{z})_{[d]}) + \boldsymbol{\lambda}_t^\top \mathbf{K} \mathbf{z} + \frac{1}{2\tau} \|\mathbf{z} - \mathbf{z}_t\|_2^2$$

$$\boldsymbol{\lambda}_{t+1} = \underset{\boldsymbol{\lambda} \in \mathbb{R}^{mn}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) - \boldsymbol{\lambda}^\top \mathbf{K} (2\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{1}{2\sigma} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t\|_2^2.$$

Expanding the linear term in the primal update, the equivalent local primal update for each agent $j \in [m]$ can be written as

$$\boldsymbol{\theta}_{j,t+1} = \underset{\boldsymbol{\theta}_j \in \mathbb{R}^{d_j}}{\operatorname{argmin}} r_j(\boldsymbol{\theta}_j) + \frac{1}{n} \boldsymbol{\lambda}_{j,t}^\top \mathbf{X}_j \boldsymbol{\theta}_j + \frac{1}{2\tau} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j,t}\|_2^2, \quad (5)$$

$$\mathbf{v}_{j,t+1} = \underset{\mathbf{v}_j \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n} \left(\sum_{j' \in [m]: j \sim_G j'} \boldsymbol{\lambda}_{j',t} - \boldsymbol{\lambda}_{j,t} \right)^\top \mathbf{v}_j + \frac{1}{2\tau} \|\mathbf{v}_j - \mathbf{v}_{j,t}\|_2^2. \quad (6)$$

Similarly, the equivalent local dual update for each agent $j \in [m] \setminus \{1\}$ is

$$\boldsymbol{\lambda}_{j,t+1} = \underset{\boldsymbol{\lambda}_j \in \mathbb{R}^n}{\operatorname{argmin}} -\frac{1}{n} \boldsymbol{\lambda}_j^\top \left(\sum_{j' \in [m]: j \sim_G j'} 2(\mathbf{v}_{j',t+1} - \mathbf{v}_{j',t}) - \mathbf{v}_{j,t} + \mathbf{v}_{j',t} \right) - \frac{1}{n} \boldsymbol{\lambda}_j^\top \mathbf{X}_j (2\boldsymbol{\theta}_{j,t+1} - \boldsymbol{\theta}_{j,t}) + \frac{1}{2\sigma} \|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j,t}\|_2^2. \quad (7)$$

The fact that $h^*(\cdot)$ depends entirely on $\boldsymbol{\lambda}_1$ makes the local dual update for the first agent (i.e., $j = 1$) different and in the form

$$\boldsymbol{\lambda}_{1,t+1} = \underset{\boldsymbol{\lambda}_1 \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}) - \frac{1}{n} \boldsymbol{\lambda}_1^\top \left(\sum_{j' \in [m]: 1 \sim_G j'} 2(\mathbf{v}_{1,t+1} - \mathbf{v}_{j',t+1}) - \mathbf{v}_{1,t} + \mathbf{v}_{j',t} \right) - \frac{1}{n} \boldsymbol{\lambda}_1^\top \mathbf{X}_1 (2\boldsymbol{\theta}_{1,t+1} - \boldsymbol{\theta}_{1,t}) + \frac{1}{2\sigma} \|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_{1,t}\|_2^2, \quad (8)$$

where the scalars $(\lambda_{1,i})_i$ denote the coordinates of $\boldsymbol{\lambda}_1$ and should not be confused with the vectors $(\boldsymbol{\lambda}_{1,t})_t$. The primal update (5) is simply an evaluation of the proximal mapping of τr_j denoted by $\operatorname{prox}_{\tau r_j}(\mathbf{u}) =$

$\operatorname{argmin}_{\mathbf{u}'} \tau r_j(\mathbf{u}') + \|\mathbf{u}' - \mathbf{u}\|_2^2/2$. The updates (6) and (7) can also be solved in closed-form. While (8) does not admit a similar closed-form expression, it can be equivalently written in terms of the functions $\ell_1(\cdot), \dots, \ell_n(\cdot)$ using the separability of the objective function and the relation between the *Moreau envelope* of a function and its convex conjugate (Bauschke & Combettes, 2011, Proposition 13.24). Therefore, we can summarize the iterations as

$$\boldsymbol{\theta}_{j,t+1} = \operatorname{prox}_{\tau r_j} \left(\boldsymbol{\theta}_{j,t} - \frac{\tau}{n} \mathbf{X}_j^\top \boldsymbol{\lambda}_{j,t} \right), \quad \text{for } j \in [m], \quad (9)$$

$$\mathbf{v}_{j,t+1} = \mathbf{v}_{j,t} - \frac{\tau}{n} \sum_{j' \in [m]: j \sim_G j'} \boldsymbol{\lambda}_{j,t} - \boldsymbol{\lambda}_{j',t}, \quad \text{for } j \in [m], \quad (10)$$

$$\boldsymbol{\lambda}_{j,t+1} = \boldsymbol{\lambda}_{j,t} + \frac{\sigma}{n} \mathbf{X}_j (2\boldsymbol{\theta}_{j,t+1} - \boldsymbol{\theta}_{j,t}) + \frac{\sigma}{n} \sum_{j' \in [m]: j \sim_G j'} 2(\mathbf{v}_{j,t+1} - \mathbf{v}_{j',t+1}) - \mathbf{v}_{j,t} + \mathbf{v}_{j',t}, \quad \text{for } j \in [m] \setminus \{1\}, \quad (11)$$

$$\boldsymbol{\lambda}_{1,t+1} = \operatorname{argmin}_{\boldsymbol{\lambda}_1 \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell_i \left(\frac{n}{\sigma} (\boldsymbol{\lambda}_{1,t+1/2} - \boldsymbol{\lambda}_1)_i \right) + \frac{1}{2\sigma} \|\boldsymbol{\lambda}_1\|_2^2, \quad (12)$$

where $(\mathbf{u})_i$ denotes the i th coordinate of a vector \mathbf{u} , and the ‘‘intermediate dual iterate’’ $\boldsymbol{\lambda}_{1,t+1/2}$ is defined as

$$\boldsymbol{\lambda}_{1,t+1/2} = \boldsymbol{\lambda}_{1,t} + \frac{\sigma}{n} \mathbf{X}_1 (2\boldsymbol{\theta}_{1,t+1} - \boldsymbol{\theta}_{1,t}) + \frac{\sigma}{n} \sum_{j' \in [m]: 1 \sim_G j'} 2(\mathbf{v}_{1,t+1} - \mathbf{v}_{j',t+1}) - \mathbf{v}_{1,t} + \mathbf{v}_{j',t}. \quad (13)$$

Interestingly, (12) is a separable optimization with respect to the coordinates of $\boldsymbol{\lambda}_1$, i.e., for each $i \in [n]$ we have

$$(\boldsymbol{\lambda}_{1,t+1})_i = \operatorname{argmin}_{\lambda \in \mathbb{R}} \frac{1}{n} \ell_i \left(\frac{n}{\sigma} (\boldsymbol{\lambda}_{1,t+1/2})_i - \lambda \right) + \frac{1}{2\sigma} \lambda^2.$$

Therefore, (12) admits efficient and parallelizable solvers.

3 Convergence guarantees

We begin by stating a few assumptions that will be used to provide convergence guarantees for the primal iterates $(\boldsymbol{\theta}_{j,t})_{t \geq 1}$. Recall the assumptions that the loss function $\ell(\cdot, \cdot)$ is *nonnegative* and the regularizer is separable as in (2). We will provide convergence rates for two different classes of loss functions. First, the Lipschitz loss functions, for which there exists a constant $\rho \geq 0$ such that

$$|\ell(u, w) - \ell(v, w)| \leq \rho |u - v|, \quad \text{for all } u, v, w \in \mathbb{R}.$$

By differentiability of $\ell(\cdot, \cdot)$ in its first argument, the condition above is equivalent to

$$\left| \frac{d\ell(u, v)}{du} \right| \leq \rho, \quad \text{for all } u, v \in \mathbb{R}. \quad (\text{Lip.})$$

Examples of the Lipschitz loss functions are the absolute loss, the Huber loss, and the logistic loss. Second, the *square root Lipschitz* loss functions, for which there exists a constant $\rho \geq 0$ such that

$$|\sqrt{\ell(u, w)} - \sqrt{\ell(v, w)}| \leq \frac{\rho}{2} |u - v|, \quad \text{for all } u, v, w \in \mathbb{R}.$$

Again, invoking differentiability of $\ell(\cdot, \cdot)$ we can equivalently write

$$\left| \frac{d\ell(u, v)}{du} \right| \leq \rho \sqrt{\ell(u, v)}. \quad (\sqrt{\cdot}\text{-Lip.})$$

Examples of the square root Lipschitz loss functions are the squared loss, the Huber loss.

Furthermore, we assume that for some known constant $R > 0$ the empirical risk minimizer $\hat{\boldsymbol{\theta}}$ is bounded as

$$\left\| \hat{\boldsymbol{\theta}} \right\|_2 \leq R. \quad (\text{minimizer bound})$$

We also assume that the agents are provided with the constant χ that bounds the usual operator norm of the design matrix as

$$\|\mathbf{X}\| \leq \chi. \quad (\text{design bound})$$

The constants $\delta > 0$ that bounds the spectral gap of the network as

$$\|\mathbf{L}^\dagger\| \leq \delta^{-1}, \quad (\text{spectral gap})$$

with \mathbf{M}^\dagger denoting the Moore-Penrose pseudoinverse of the matrix \mathbf{M} , as well as the constant $D > 0$ that bounds the operator norm of the Laplacian as

$$\|\mathbf{L}\| \leq D, \quad (\text{Laplacian bound})$$

are also provided to the agents. Because $n\|\mathbf{K}\| \leq \max_{j \in [m]} \|\mathbf{X}_j\| + \|\mathbf{L} \otimes \mathbf{I}\| \leq \|\mathbf{X}\| + \|\mathbf{L}\|$, instead of assuming an additional bound for $\|\mathbf{K}\|$, we will use the bound $\|\mathbf{K}\| \leq (\chi + D)/n$.

Theorem 1. *Suppose that the m agents are given the positive constants R , χ , δ and D that respectively satisfy (minimizer bound), (design bound), (spectral gap), and (Laplacian bound), so that they can choose $\sigma = m^{1/2}n^{3/2}\rho / \left((\chi + D)R\sqrt{1 + 2\chi^2/\delta^2} \right)$ and $\tau = n^2 / \left((\chi + D)^2\sigma \right)$. Denote the temporal average of the vectors $\boldsymbol{\theta}_{j,t}$ over the first $T \geq 1$ iterations by*

$$\bar{\boldsymbol{\theta}}_j = \frac{1}{T} \sum_{t=1}^T \boldsymbol{\theta}_{j,t}, \quad \text{for } j \in [m], \quad (14)$$

and let $\bar{\boldsymbol{\theta}} = [\bar{\boldsymbol{\theta}}_1; \dots; \bar{\boldsymbol{\theta}}_m]$. Under the Lipschitz loss model (Lip.) we have

$$\frac{1}{n} \sum_{i=1}^n \ell_i(\bar{\boldsymbol{\theta}}) + r(\bar{\boldsymbol{\theta}}) \leq \frac{1}{n} \sum_{i=1}^n \ell_i(\hat{\boldsymbol{\theta}}) + r(\hat{\boldsymbol{\theta}}) + \frac{2(\chi + D)R\rho}{(n/m)^{1/2}T} \sqrt{1 + \frac{2\chi^2}{\delta^2}} \quad (15)$$

Similarly, under the square root Lipschitz loss model ($\sqrt{\cdot}$ -Lip.) and for $T \geq 2mn\rho^2/\sigma$ we have an “isomorphic convergence” given by

$$\frac{1}{n} \sum_{i=1}^n \ell_i(\bar{\boldsymbol{\theta}}) + r(\bar{\boldsymbol{\theta}}) \leq \left(1 + \frac{2(\chi + D)R\rho}{m^{1/2}n^{3/2}T} \sqrt{1 + \frac{2\chi^2}{\delta^2}} \right) \times \left(\frac{1}{n} \sum_{j=1}^n \ell_i((\mathbf{X}\hat{\boldsymbol{\theta}})_i) + r(\hat{\boldsymbol{\theta}}) + \frac{(\chi + D)R\rho}{(n/m)^{1/2}T} \sqrt{1 + \frac{2\chi^2}{\delta^2}} \right). \quad (16)$$

The prescribed τ and σ are “optimized” for the Lipschitz model. The well-tuned choice of τ and σ under the square root Lipschitz model is slightly different and depends on the minimum value of the objective. For simplicity, we used the former in the theorem for both models.

For a better understanding of the convergence bounds (15) and (16), it is worth considering more interpretable approximations of the quantities D and δ . With $\Delta(G)$ denoting the maximum degree of the graph G , we have an elementary bound $\|\mathbf{L}\| \leq 2\Delta(G)$, so it suffices to choose $D \geq 2\Delta(G)$. Furthermore, for a connected graph, $\|\mathbf{L}^\dagger\|$ is reciprocal to the second smallest eigenvalue of \mathbf{L} , and we can invoke an inequality due to Mohar (1991, Theorem 2.3) that relates the spectral gap, diameter, and the maximum degree of a graph, we have $\|\mathbf{L}^\dagger\| \geq 2(\text{diam}(G) - 1 - \log(m - 1)) / \Delta(G)$ which can provide a general bound on how large δ can possibly be. Another inequality (Mohar, 1991, Theorem 4.2), attributed to Brendan McKay, also provides the bound $\|\mathbf{L}^\dagger\| \leq m \text{diam}(G) / 4$ which implies a conservative choice of $\delta \leq 4 / (m \text{diam}(G))$. The networks that

are (*spectral*) *expanders* are more favorable as they typically have larger spectral gap and smaller maximum degree simultaneously. For instance, for k -regular *Ramanujan graphs* we can choose $\delta = k - 2\sqrt{k-1}$ (Lubotzky et al., 1988; Mohar, 1992).

The algorithm can be generalized by assigning weights to the edges of the network and choosing \mathbf{L} to be the Laplacian of the weighted network. The effect of using weighted edges on the algorithm is that the simple summation iterates of the neighboring agents in (10), (11), and (13) (thereby (12)), will become weighted summations. Using weighted edges allows us, in principle, to optimize bounds (15) and (16) by adjusting the edge weights.

We have shown that we can solve (1) in the feature-distributed setting and achieve a convergence rate of $O(1/T)$ under relatively simple assumptions. The iterations each agent has to solve is rather simple, including (12) thanks to its separability. However, there are a few limitations in the proposed framework that have to be considered. First, the agents cannot rely only on local information to choose τ and σ ; in general they can obtain the required global information at the cost of extra communications. Second, the scope of the algorithm is limited by the fact that the loss function acts on linear predictors $\mathbf{x}_i^\top \boldsymbol{\theta}$. It is worth mentioning, however, that this limitation is basically necessary to stay in the realm of convex optimization; we are not aware of any widely used nonlinear predictor whose composition with standard loss functions is convex. Third, the considered saddle-point formulation incurs a significant communication and computation cost associated with the iterates $(\boldsymbol{\lambda}_{j,t})$ and $(\mathbf{v}_{j,t})$; it is not clear if this is inherent to the problem.

4 Numerical Experiments

We provide several numerical experiments to illustrate the behavior of the proposed algorithm with varying quantities of agents and communication graphs. In the case where computation is of greater cost than communication, we find that our algorithm can make use of parallelism to improve performance.

We solve the least squares problem

$$\text{minimize}_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$$

for a synthetic dataset of $2^{14} = 16384$ samples and $2^{11} = 2048$ features so that \mathbf{X} is a 16384×2048 matrix. To construct the synthetic dataset, the design matrix \mathbf{X} , the ground truth vector $\boldsymbol{\theta}_*$, and the noise vector \mathbf{e} are all populated by i.i.d. samples of the standard normal distribution. The corresponding noisy response vector \mathbf{y} is then computed as $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}_* + \mathbf{e}$. In all experiments, the features are partitioned equally among the agents, i.e., each agent has access to exactly d/m features.

We explore the following communication graph structures:

- *Complete Graph*: All agents are connected to all other agents.
- *Star Graph*: All agents are connected only to the first agent.
- *Erdős-Rényi Graph*: Each of the possible $\binom{m}{2}$ pairs of agents are connected with probability $p \in \{0.1, 0.5\}$ independent of the other connections. To avoid violating the connectivity requirement of the communication graph (with high probability), we only consider graphs of 8 or more agents in the case $p = 0.5$, and graphs of 32 or more agents in the case of $p = 0.1$.
- *2D Lattice Graph*: The agents are arranged in 2D space as a square lattice. Each agent is connected to its cardinal and diagonal neighbors. The first agent is located at one of the four center-most lattice points.
- *Random Geometric Graph*: Agents are assigned positions in the 2D unit square uniformly at random. A pair of agents are connected if the Euclidean distance between their positions is less than 0.3. Again, to avoid violating the connectivity requirement of the communication graph (with high probability), we only consider 32 agents or more.

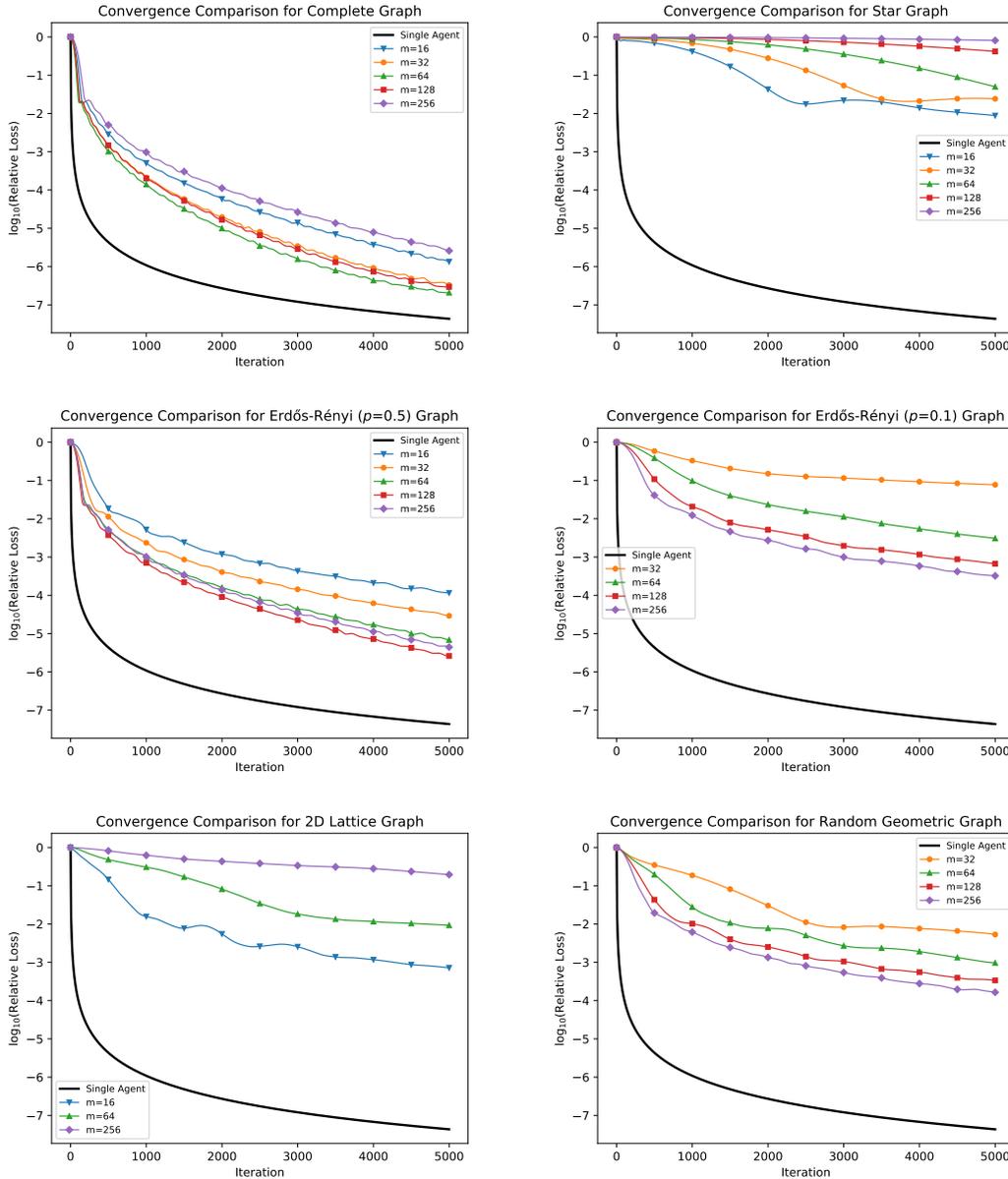


Figure 1: Plots depicting algorithm progress for varying communication graph structures and number of agents. The single agent progress is included in all plots for reference. With \mathcal{L}_t denoting the objective (i.e., the regularized empirical risk) at $\bar{\theta}_t$, and \mathcal{L}_* denoting the minimum value of the objective, the vertical axis represents the base-10 logarithm of the relative error defined as $\log_{10} \left(\frac{\mathcal{L}_t - \mathcal{L}_*}{\mathcal{L}_0 - \mathcal{L}_*} \right)$. The horizontal axis represents number of iterations completed.

As a baseline, we solve the single agent problem using the proposed primal-dual algorithm but with the Lagrange multiplier v terms fixed at zero, however we recognize that the problem choice could also be solved by other algorithms, e.g. gradient descent. (For the single agent case, the Laplacian constraints of (3) are trivially satisfied and can be omitted.) Figure 1 shows the convergence behavior of the proposed algorithm for each of the aforementioned communication graph structures. The complete graph tends to converge faster than any other graph for a fixed number of agents, and performs best at 64 agents (with 32 features per agent) instead of continually improving with increasing quantity of agents. Similarly, the Erdős-Rényi

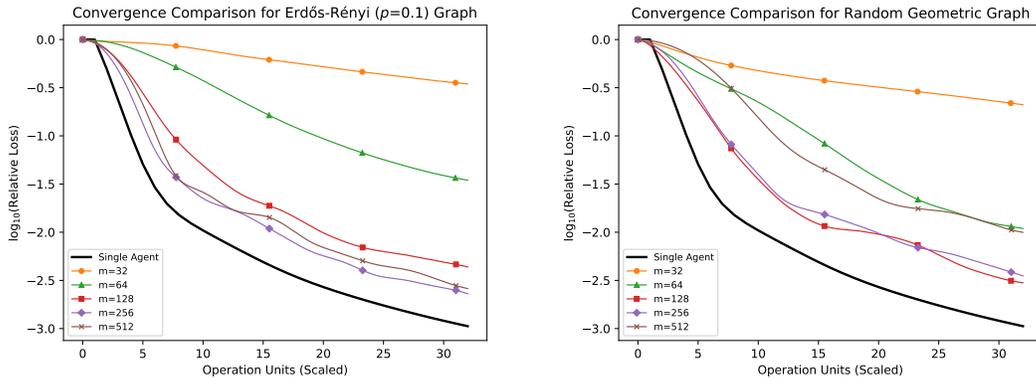


Figure 2: Plots depicting algorithm progress for Erdős-Rényi ($p = 0.1$) and random geometric graphs under the given cost paradigm. With \mathcal{L}_t denoting the objective (i.e., the regularized empirical risk) at $\bar{\theta}_t$, and \mathcal{L}_* denoting the minimum value of the objective, the vertical axis represents the base-10 logarithm of the relative error defined as $\log_{10} \left(\frac{\mathcal{L}_t - \mathcal{L}_*}{\mathcal{L}_0 - \mathcal{L}_*} \right)$. The horizontal axis represents units of operations per agent completed (not iteration) normalized such that the single agent case completes one iteration per unit of operation (i.e. the single agent completes 32 iterations). Explicitly, iteration t corresponds to $\frac{n(4(d/m)+2\Delta(G)+7)+5(d/m)}{n(4d+1)+5d}t$ on the horizontal axis (except for the single agent case, where iteration t corresponds to t on the horizontal axis). In short, settings with fewer operations per agent per iteration complete more iterations.

graphs perform best at 128 and 256 agents for $p = 0.5$ and $p = 0.1$, respectively. Convergence degrades as p decreases. The random geometric graph performs very similarly to the Erdős-Rényi graph for $p = 0.1$. Both the star and 2D lattice graphs perform increasingly worse as the quantity of agents increases. We speculate this is caused by a large quantity of comparatively small eigenvalues for the associated Laplacian matrices.

If we assume a situation where cost is dominated by computation rather than communication, the proposed algorithm can achieve comparable performance to the single agent case even under relatively sparse graphs. Recall that n , m , and d represent the number of samples, agents, and features, respectively, and that $\Delta(G)$ denotes the maximum degree of the communication graph G . One can show that each iteration of the proposed algorithm requires each agent complete $n(4(d/m) + 2\Delta(G) + 7) + 5(d/m)$ floating point operations.² In the single agent case, one can show $n(4d + 1) + 5d$ floating point operations are needed per iteration.³

We also compare scenarios for a fixed number of operations per agent. As the number of agents increases \mathbf{X} and θ are increasingly split over more agents, effectively parallelizing the problem. This leads to a decrease in the number of operations per agent for the matrix-vector multiplies in (9) and (11) which dominate the operation cost. Figure 2 illustrates how, under this cost paradigm, the relatively sparse Erdős-Rényi ($p = 0.1$) and random geometric graphs with 256 agents achieve performance comparable to that of the single agent case. This speaks to the promise of the proposed algorithm for very large problem sizes over relatively sparse graphs.

References

Reza Arablouei, Kutluyil Doğançay, Stefan Werner, and Yih-Fang Huang. Model-distributed solution of regularized least-squares problem over sensor networks. In *2015 IEEE International Conference on Acoustics*,

²On a per-iteration per-agent basis, updating θ according to (9) equates to $2n(d/m) + 4(d/m)$ operations, updating v according to (10) equates to $n(\Delta(G) + 3)$ operations, and updating λ according to (11) equates to $n(2(d/m) + \Delta(G) + 4) + (d/m)$ operations. We omit the presumed negligible cost of the first agent solving (12), which for the specific case of least squares would be an extra $2n$ operations. For the specific case of *non-regularized* least squares, we could also omit $3(d/m)$ operations from the θ updates.

³To compute the required operations in the single agent case, a similar calculation is performed to that of Footnote 2 with caveats. The v quantities are absent, leading to a reduction of $n(\Delta(G) + 3)$ from the updates in (10) as well as $n(\Delta(G) + 3)$ from the updates in (11).

- Speech and Signal Processing (ICASSP)*, pp. 3821–3825, 2015. doi: 10.1109/ICASSP.2015.7178686.
- Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer New York, New York, NY, 2011. ISBN 978-1-4419-9467-7. doi: 10.1007/978-1-4419-9467-7. URL <https://doi.org/10.1007/978-1-4419-9467-7>.
- Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal—dual algorithm. *Math. Program.*, 159(1–2):253–287, September 2016. ISSN 0025-5610. doi: 10.1007/s10107-015-0957-3.
- Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. VAFL: a method of vertical asynchronous federated learning. arXiv preprint [arXiv: 2007.06081 \[cs.LG\]](https://arxiv.org/abs/2007.06081), 2020.
- J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Trans. Auto. Control*, 57(3):592–606, 2012.
- Eduard Gorbunov, Alexander Rogozin, Aleksandr Beznosikov, Darina Dvinskikh, and Alexander Gasnikov. Recent theoretical advances in decentralized distributed convex optimization. arXiv preprint [arXiv: 2011.13259 \[math.OC\]](https://arxiv.org/abs/2011.13259), 2020.
- Cristiano Gratton, Naveen K.D. Venkategowda, Reza Arablouei, and Stefan Werner. Distributed ridge regression with feature partitioning. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1423–1427, 2018. doi: 10.1109/ACSSC.2018.8645549.
- Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/05a70454516ecd9194c293b0e415777f-Paper.pdf>.
- Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, pp. 2232–2240, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330765. URL <https://doi.org/10.1145/3292500.3330765>.
- B. Johansson, T. Kevieczky, M. Johansson, and K. H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Proc. IEEE Conf. Decision and Control*, pp. 4185–4190, Cancun, Mexico, 2008.
- Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D’Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. In Peter Kairouz and H. Brendan McMahan (eds.), *Foundations and Trends in Machine Learning*, volume 14. 2021. doi: 10.1561/22000000083. URL <http://dx.doi.org/10.1561/22000000083>.
- Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Comb.*, 8(3):261–277, 1988. doi: 10.1007/BF02126799. URL <https://doi.org/10.1007/BF02126799>.
- David Mateos-Núñez and Jorge Cortés. Distributed saddle-point subgradient algorithms with laplacian averaging. *IEEE Transactions on Automatic Control*, 62(6):2720–2735, 2017. doi: 10.1109/TAC.2016.2616646.

- Bojan Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graph. Comb.*, 7(1):53–64, March 1991. ISSN 0911-0119. doi: 10.1007/BF01789463. URL <https://doi.org/10.1007/BF01789463>.
- Bojan Mohar. Laplace eigenvalues of graphs—a survey. *Discrete Mathematics*, 109(1):171–183, 1992. ISSN 0012-365X. doi: [https://doi.org/10.1016/0012-365X\(92\)90288-Q](https://doi.org/10.1016/0012-365X(92)90288-Q). URL <https://www.sciencedirect.com/science/article/pii/0012365X9290288Q>.
- A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Auto. Control*, 54(1):48–61, 2009.
- J.Z. Sasiadek. Sensor fusion. *Annual Reviews in Control*, 26(2):203–228, 2002. ISSN 1367-5788. doi: [https://doi.org/10.1016/S1367-5788\(02\)00045-7](https://doi.org/10.1016/S1367-5788(02)00045-7). URL <https://www.sciencedirect.com/science/article/pii/S1367578802000457>.
- K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proc. Int. Conf. Machine Learning*, volume 70, pp. 3027–3036, 2017.
- Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal convergence rates for convex distributed optimization in networks. *Journal of Machine Learning Research*, 20(159):1–31, 2019. URL <http://jmlr.org/papers/v20/19-543.html>.
- Aleksandra B. Slavković, Yuval Nardi, and Matthew M. Tibbits. “secure” logistic regression of horizontally and vertically partitioned distributed databases. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pp. 723–728, 2007. doi: 10.1109/ICDMW.2007.114.
- Virginia Smith, Simone Forte, Chenxin Ma, Martin Takáč, Michael I. Jordan, and Martin Jaggi. CoCoA: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18(230):1–49, 2018. URL <http://jmlr.org/papers/v18/16-512.html>.
- Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL <https://proceedings.neurips.cc/paper/2010/file/76cf99d3614e23eabab16fb27e944bf9-Paper.pdf>.
- J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *Proc. IEEE Conf. Decision and Control*, pp. 3800–3805, Orlando, FL, 2011.
- 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. ISSN 1367-5788. doi: <https://doi.org/10.1016/j.arcontrol.2019.05.006>. URL <https://www.sciencedirect.com/science/article/pii/S1367578819300082>.
- M. Zhu and S. Martinez. On distributed convex optimization under inequality and equality constraints. *IEEE Trans. Auto. Control*, 57(1):151–164, 2012.

A Proof of Theorem 1

As the dual parameters are not important for our purposes, our goal is to convert the established saddle-point convergence rates of the Chambolle–Pock algorithm (Chambolle & Pock, 2016) into primal convergence rates. Similar to (14), define the temporal average of the other iterates over the first T iterations as

$$\bar{\mathbf{v}}_j = \frac{1}{T} \sum_{t=1}^T \mathbf{v}_{j,t}$$

$$\bar{\lambda}_j = \frac{1}{T} \sum_{t=1}^T \lambda_{j,t},$$

for $j \in [m]$, and let $\bar{\mathbf{V}} = [\bar{\mathbf{v}}_1 \ \cdots \ \bar{\mathbf{v}}_m]$ and $\bar{\boldsymbol{\lambda}} = [\bar{\boldsymbol{\lambda}}_1; \ \cdots; \ \bar{\boldsymbol{\lambda}}_n]$. Furthermore, denote the objective of the saddle-point problem (4) by

$$\mathcal{E}(\boldsymbol{\theta}, \mathbf{V}, \boldsymbol{\lambda}) = \underbrace{\sum_{j=1}^m r_j(\boldsymbol{\theta}_j)}_{=r(\boldsymbol{\theta})} + \frac{1}{n} \boldsymbol{\lambda}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j) - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\lambda_{1,i}). \quad (17)$$

With the iterates initialized at zero (i.e., $\boldsymbol{\theta}_{j,0} = \mathbf{0}$, $\mathbf{v}_{j,0} = \mathbf{0}$, and $\boldsymbol{\lambda}_{j,0} = \mathbf{0}$ for all $j \in [m]$), and observing that

$$\tau\sigma \|\mathbf{K}\|^2 \leq 1,$$

we can apply the convergence rate established in (Chambolle & Pock, 2016, Theorem 1, and Remark 2) to obtain

$$\begin{aligned} & \mathcal{E}(\bar{\boldsymbol{\theta}}, \bar{\mathbf{V}}, \boldsymbol{\lambda}) - \mathcal{E}(\boldsymbol{\theta}, \mathbf{V}, \bar{\boldsymbol{\lambda}}) \\ & \leq \frac{1}{T} \sum_{j=1}^m \left(\frac{1}{2\tau} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j,0}\|_2^2 + \frac{1}{2\tau} \|\mathbf{v}_j - \mathbf{v}_{j,0}\|_2^2 + \frac{1}{2\sigma} \|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j,0}\|_2^2 \right. \\ & \quad \left. - \frac{1}{n} (\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j,0})^\top \left(\mathbf{X}_j (\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j,0}) + \sum_{j' \in [m]: j \sim_G j'} \mathbf{v}_j - \mathbf{v}_{j,0} - \mathbf{v}_{j'} + \mathbf{v}_{j',0} \right) \right) \\ & \leq \frac{1}{T} \sum_{j=1}^m \left(\frac{1}{\tau} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j,0}\|_2^2 + \frac{1}{\tau} \|\mathbf{v}_j - \mathbf{v}_{j,0}\|_2^2 + \frac{1}{\sigma} \|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j,0}\|_2^2 \right) \\ & = \frac{1}{T} \left(\frac{1}{\tau} \|\boldsymbol{\theta}\|_2^2 + \frac{1}{\tau} \|\mathbf{V}\|_{\text{F}}^2 + \frac{1}{\sigma} \|\boldsymbol{\lambda}\|_2^2 \right), \end{aligned}$$

for all $\boldsymbol{\theta}$, \mathbf{V} , and $\boldsymbol{\lambda}$. Rearranging the terms, we equivalently have

$$\mathcal{E}(\bar{\boldsymbol{\theta}}, \bar{\mathbf{V}}, \boldsymbol{\lambda}) - \frac{1}{T\sigma} \|\boldsymbol{\lambda}\|_2^2 \leq \mathcal{E}(\boldsymbol{\theta}, \mathbf{V}, \bar{\boldsymbol{\lambda}}) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right).$$

Recalling (17), taking the maximum of the left-hand side with respect to $\boldsymbol{\lambda}$, and applying Lemma 1 to the part corresponding to $\boldsymbol{\lambda}_1$, we have

$$\begin{aligned} & r(\bar{\boldsymbol{\theta}}) + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) - \frac{1}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i))^2 + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2 \\ & \leq \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} r(\boldsymbol{\theta}) + \frac{1}{n} \sum_{j=1}^m \bar{\boldsymbol{\lambda}}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j) - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\bar{\lambda}_{1,i}) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right). \end{aligned} \quad (18)$$

Next we establish a few more inequalities depending on the characteristics of the loss function, that together with (18) yield the desired convergence rates.

A.1 Lower bound for the left-hand side of (18)

A.1.1 Lipschitz loss

We first consider the case of Lipschitz loss functions (Lip.). Using convexity of $\ell_i(\cdot)$, we can write

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) & \geq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) - \ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) \sum_{j=2}^m (\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j)_i \\ & \geq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) - \frac{m-1}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i))^2 - \frac{T\sigma}{4n^2} \sum_{j=2}^m \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2, \end{aligned}$$

where the second inequality is an application of the basic inequality $2ab \leq a^2 + b^2$. By construction, we have

$$\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1 + \sum_{j=2}^m \mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j = \mathbf{X} \bar{\boldsymbol{\theta}}.$$

Therefore, in view of (Lip.), what we have shown is

$$\begin{aligned} r(\bar{\boldsymbol{\theta}}) + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) - \frac{1}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i))^2 + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2 \\ \geq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) + r(\bar{\boldsymbol{\theta}}) - \frac{mn\rho^2}{T\sigma}. \end{aligned} \quad (19)$$

A.1.2 Square root Lipschitz loss

The second case we consider is that of the square root Lipschitz loss functions ($\sqrt{\cdot}$ -Lip.). It follows from ($\sqrt{\cdot}$ -Lip.) that

$$\sum_{i=1}^n (\ell'_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i))^2 \leq \rho^2 \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i).$$

For sufficiently large T we have $\gamma \stackrel{\text{def}}{=} n\rho^2/(T\sigma) < 1/m$, and we can lower bound the left-hand side of (18), excluding the term $r(\bar{\boldsymbol{\theta}})$, as

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) - \frac{1}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i))^2 + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2 \\ \geq (1-\gamma) \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2 \\ \geq (1-\gamma) \left(\frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) - \ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) \sum_{j=2}^m (\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j)_i \right) + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2, \end{aligned} \quad (20)$$

where we used the convexity of the function $\ell_i(\cdot)$ in the second line. Again using the basic inequality $2ab \leq a^2 + b^2$, we have

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i) \sum_{j=2}^m (\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j)_i \\ \leq \sum_{i=1}^n \frac{(1-\gamma)(m-1)}{T\sigma} (\ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i))^2 + \frac{T\sigma}{4(1-\gamma)n^2} \sum_{j=2}^m (\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j)_i^2 \\ = \frac{(1-\gamma)(m-1)}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i))^2 + \frac{T\sigma}{4(1-\gamma)n^2} \sum_{j=2}^m \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2. \end{aligned} \quad (21)$$

By ($\sqrt{\cdot}$ -Lip.) we also have

$$\sum_{i=1}^n (\ell'_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i))^2 \leq \rho^2 \sum_{i=1}^n \ell_i((\mathbf{X} \bar{\boldsymbol{\theta}})_i),$$

which together with (20) and (21), and by adding back the term $r(\bar{\boldsymbol{\theta}})$, yields

$$r(\bar{\boldsymbol{\theta}}) + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i) - \frac{1}{T\sigma} \sum_{i=1}^n (\ell'_i((\mathbf{X}_1 \bar{\boldsymbol{\theta}}_1 + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_1)_i))^2$$

$$\begin{aligned}
& + \sum_{j=2}^m \frac{T\sigma}{4n^2} \|\mathbf{X}_j \bar{\boldsymbol{\theta}}_j + \bar{\mathbf{V}} \mathbf{L} \mathbf{e}_j\|_2^2 \\
& \geq r(\bar{\boldsymbol{\theta}}) + (1-\gamma)(1-\gamma(1-\gamma)(m-1)) \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\bar{\boldsymbol{\theta}})_i) \\
& \geq (1-m\gamma) \left(\frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\bar{\boldsymbol{\theta}})_i) + r(\bar{\boldsymbol{\theta}}) \right)
\end{aligned} \tag{22}$$

A.2 Upper bound for the right-hand side of (18)

Furthermore, the right-hand side of the inequality (18) can be bounded as

$$\begin{aligned}
& \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} r(\boldsymbol{\theta}) + \frac{1}{n} \sum_{j=1}^m \bar{\boldsymbol{\lambda}}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j) - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\bar{\lambda}_{1,i}) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right) \\
& \leq \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} r(\boldsymbol{\theta}) + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \boldsymbol{\theta}_1 + \mathbf{V} \mathbf{L} \mathbf{e}_1)_i) + \frac{1}{n} \sum_{j=2}^m \bar{\boldsymbol{\lambda}}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right).
\end{aligned}$$

Imposing the constraints $\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j = \mathbf{0}$ for $j = 2, \dots, m$, can only increase the value of minimum on the right-hand side. Namely, we have

$$\begin{aligned}
& \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} r(\boldsymbol{\theta}) + \frac{1}{n} \sum_{j=1}^m \bar{\boldsymbol{\lambda}}_j^\top (\mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j) - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\bar{\lambda}_{1,i}) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right) \\
& \leq \min_{\boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{n \times m}} r(\boldsymbol{\theta}) + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}_1 \boldsymbol{\theta}_1 + \mathbf{V} \mathbf{L} \mathbf{e}_1)_i) + \frac{1}{T\tau} \left(\|\boldsymbol{\theta}\|_2^2 + \|\mathbf{V}\|_{\text{F}}^2 \right) \\
& \quad \text{subject to } \mathbf{X}_j \boldsymbol{\theta}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j = \mathbf{0}, \text{ for } j \in [m] \setminus \{1\} \\
& \leq \min_{\mathbf{V} \in \mathbb{R}^{n \times m}} \frac{1}{T\tau} \|\mathbf{V}\|_{\text{F}}^2 + \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\hat{\boldsymbol{\theta}})_i) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \|\hat{\boldsymbol{\theta}}\|_2^2 \\
& \quad \text{subject to } \mathbf{X}_j \hat{\boldsymbol{\theta}}_j + \mathbf{V} \mathbf{L} \mathbf{e}_j = \mathbf{0}, \text{ for } j \in [m] \setminus \{1\} \\
& \leq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\hat{\boldsymbol{\theta}})_i) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \|\hat{\boldsymbol{\theta}}\|_2^2 + \frac{1}{T\tau} \|(\mathbf{L} \otimes \mathbf{I})^\dagger\|^2 \left\| \left[\sum_{j=2}^m \mathbf{X}_j \hat{\boldsymbol{\theta}}_j; \quad -\mathbf{X}_2 \hat{\boldsymbol{\theta}}_2; \quad \dots; \quad -\mathbf{X}_m \hat{\boldsymbol{\theta}}_m \right] \right\|_2^2 \\
& \leq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\hat{\boldsymbol{\theta}})_i) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \|\hat{\boldsymbol{\theta}}\|_2^2 + \frac{2}{T\tau} \|(\mathbf{L} \otimes \mathbf{I})^\dagger\|^2 \|\mathbf{X}\|^2 \|\hat{\boldsymbol{\theta}}\|_2^2,
\end{aligned} \tag{23}$$

where $\hat{\boldsymbol{\theta}}$ is the empirical risk minimizer given by (1), and we used the bound

$$\left\| \left[\sum_{j=2}^m \mathbf{X}_j \hat{\boldsymbol{\theta}}_j; \quad -\mathbf{X}_2 \hat{\boldsymbol{\theta}}_2; \quad \dots; \quad -\mathbf{X}_m \hat{\boldsymbol{\theta}}_m \right] \right\|_2^2 \leq \|\mathbf{X}\|^2 \|\hat{\boldsymbol{\theta}}\|_2^2 + \max_{j \in [m] \setminus \{1\}} \|\mathbf{X}_j\|^2 \|\hat{\boldsymbol{\theta}}\|_2^2.$$

A.3 Convergence of the regularized empirical risk

We are now ready to derive the convergence rates under the loss models (Lip.) and ($\sqrt{\cdot}$ -Lip.).

A.3.1 Lipschitz loss

In the case of Lipschitz loss model (Lip.), the bounds (18), (19), and (23) guarantee that

$$\frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\bar{\boldsymbol{\theta}})_i) + r(\bar{\boldsymbol{\theta}}) \leq \frac{1}{n} \sum_{i=1}^n \ell_i((\mathbf{X}\hat{\boldsymbol{\theta}})_i) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \left(1 + 2\|(\mathbf{L} \otimes \mathbf{I})^\dagger\|^2 \|\mathbf{X}\|^2 \right) \|\hat{\boldsymbol{\theta}}\|_2^2 + \frac{mn\rho^2}{T\sigma}$$

$$\leq \frac{1}{n} \sum_{i=1}^n \ell_i \left((\mathbf{X}\hat{\boldsymbol{\theta}})_i \right) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \left(1 + \frac{2\chi^2}{\delta^2} \right) R^2 + \frac{mn\rho^2}{T\sigma}.$$

Using the values of σ and τ prescribed by [Theorem 1](#), we get [\(15\)](#).

A.3.2 Square root Lipschitz loss

Similarly, for square root Lipschitz losses ($\sqrt{\cdot}$ -Lip.), it follows from [\(18\)](#), [\(22\)](#), and [\(23\)](#) that for $T \geq 2mn\rho^2/\sigma$ we have

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \ell_i \left((\mathbf{X}\bar{\boldsymbol{\theta}})_i \right) &\leq \left(1 - \frac{mn\rho^2}{T\sigma} \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^n \ell_i \left((\mathbf{X}\hat{\boldsymbol{\theta}})_i \right) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \left(1 + 2\|(\mathbf{L} \otimes \mathbf{I})^\dagger\|^2 \|\mathbf{X}\|^2 \right) \|\hat{\boldsymbol{\theta}}\|_2^2 \right) \\ &\leq \left(1 + \frac{2mn\rho^2}{T\sigma} \right) \left(\frac{1}{n} \sum_{i=1}^n \ell_i \left((\mathbf{X}\hat{\boldsymbol{\theta}})_i \right) + r(\hat{\boldsymbol{\theta}}) + \frac{1}{T\tau} \left(1 + \frac{2\chi^2}{\delta^2} \right) R^2 \right). \end{aligned}$$

Using the values of σ and τ prescribed by [Theorem 1](#) yields [\(16\)](#).

A.4 Auxiliary Lemma

Lemma 1. *Let f_1 and f_2 be differentiable (closed) convex functions defined over a linear space \mathcal{X} . Denote their corresponding convex conjugate functions defined on the dual space \mathcal{X}^* respectively by f_1^* and f_2^* . For all $\mathbf{u} \in \mathcal{X}$ we have*

$$\max_{\mathbf{v} \in \mathcal{X}^*} \langle \mathbf{u}, \mathbf{v} \rangle - (f_1^*(\mathbf{v}) + f_2^*(\mathbf{v})) \geq f_1(\mathbf{u}) - f_2^*(\nabla f_1(\mathbf{u})).$$

Proof. The result follows from the duality of summation and *infimal convolution* ([Bauschke & Combettes, 2011](#), Proposition 13.24), that is

$$\begin{aligned} \max_{\mathbf{v} \in \mathcal{X}^*} \langle \mathbf{u}, \mathbf{v} \rangle - (f_1^*(\mathbf{v}) + f_2^*(\mathbf{v})) &= \min_{\mathbf{w} \in \mathcal{X}} f_1(\mathbf{u} - \mathbf{w}) + f_2(\mathbf{w}) \\ &\geq \min_{\mathbf{w} \in \mathcal{X}} f_1(\mathbf{u}) - \langle \nabla f_1(\mathbf{u}), \mathbf{w} \rangle + f_2(\mathbf{w}) \\ &= f_1(\mathbf{u}) - f_2^*(\nabla f_1(\mathbf{u})). \end{aligned}$$

□

B Supplementary Code and Figures

All code may be found in the supplementary materials file, along with a read-me file which details how to reproduce the results.

In an attempt to mimic the results of [Figure 1](#), 101 trials were run for each setting. The random number generator was given a unique seed in each trial to generate different values. As a result, all of the problem tensors (\mathbf{X} , $\boldsymbol{\theta}_*$, \mathbf{e} and by consequence \mathbf{y}) along with graph structure (for random graphs) were re-sampled in each trial. [Figure 3](#) shows the statistic results from these trials. It appears that convergence is most influenced by choice of graph structure as the random graphs tend to have a higher variance. However, intuitively this influence diminishes as the number of agents increases. A non-random graph structure yields nearly no perceivable spread, implying that the problem description is sufficiently robust for our purposes.

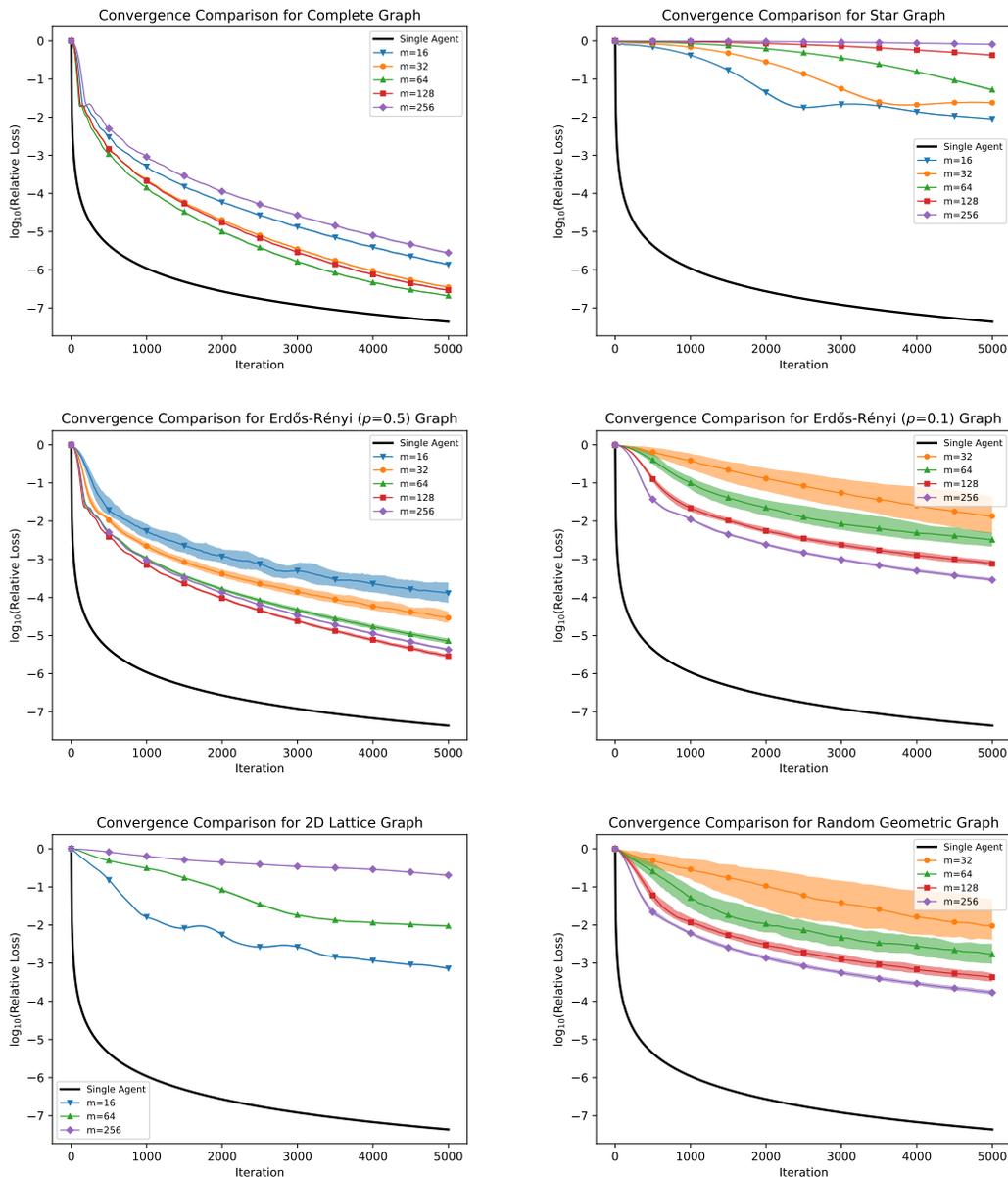


Figure 3: Plots highlighting the effect of randomness on convergence results. All plots are analogous to those in Figure 1, however the series' values now portray the median loss while the shaded regions portray the 10th to 90th percentile over 101 trials iteration-wise. For the results of non-random graphs (i.e. the single agent, complete graph, star graph, and 2D lattice graph) the difference between 10th and 90th percentile is not readily visible, but has indeed been plotted.