Efficient Distributed Learning with Sparsity

Jialei Wang [1] **Mladen Kolar** [1] **Nathan Srebro** [2] **Tong Zhang** [3]

Abstract

We propose a novel, efficient approach for distributed sparse learning with observations randomly partitioned across machines. In each round of the proposed method, worker machines compute the gradient of the loss on local data and the master machine solves a shifted ℓ_1 regularized loss minimization problem. After a number of communication rounds that scales only logarithmically with the number of machines, and independent of other parameters of the problem, the proposed approach provably matches the estimation error bound of centralized methods.

1. Introduction

We consider learning a sparse linear regressor β minimizing the population objective:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \mathbb{E}_{\mathbf{X},Y\sim\mathcal{D}} \left[\ell(Y, \langle \mathbf{X}, \boldsymbol{\beta}\rangle)\right], \qquad (1)$$

where $(\mathbf{X}, Y) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathcal{Y}$ are drawn from an unknown distribution $\mathcal{D}$ and $\ell(\cdot,\cdot)$ is a convex loss function, based on N i.i.d. samples $\{\mathbf{x}_i, y_i\}_{i=1}^N$ drawn from $\mathcal{D}$, and when the support $S := \text{support}(\boldsymbol{\beta}^*) = \{j \in [p] \mid \beta_j^* \neq 0\}$ of β^* is small, $|S| \leqslant s$. In a standard single-machine setting, a common empirical approach is to minimize the ℓ_1 regularized empirical loss (see, e.g., (2) below). Here we consider a setting where data are distributed across m machines, and, for simplicity, assume[1] that $N = nm$, so that each machine j has access to n i.i.d. observations (from the same source $\mathcal{D}$) $\{\mathbf{x}_{ji}, y_{ji}\}_{i=1}^n$ (equivalently, that $N = nm$ samples are randomly partitioned across machines).

The main contribution of the paper is a novel algorithm for estimating $\boldsymbol{\beta}^*$ in a distributed setting. Our estimator is able to achieve the performance of a centralized procedure that has access to all data, while keeping computation and communication costs low. Compared to the existing one-shot estimation approach (Lee et al., 2015b), our method can achieve the same statistical performance without performing the expensive debiasing step. As the number of communication rounds increases, the estimation accuracy improves until matching the performance of a centralized procedure, which happens after the logarithm of the total number of machines rounds. Furthermore, our results can be achieved under weak assumptions on the data generating procedure.

We assume that the communication occurs in rounds. In each round, machines exchange messages with the master machine. Between two rounds, each machine only computes based on its local information, which includes local data and previous messages (Zhang et al., 2013b; Shamir & Srebro, 2014; Arjevani & Shamir, 2015). In a non-distributed setting, efficient estimation procedures need to balance statistical efficiency with computation efficiency (runtime). In a distributed setting, the situation is more complicated and we need to balance two resources, local runtime and number of rounds of communication, with the statistical error. The local runtime refers to the amount of work each machine needs to do. The number of rounds of communication refers to how often do local machines need to exchange messages with the master machine. We compare our procedure to other algorithm using the aforementioned metrics.

We consider the following two baseline estimators of $\boldsymbol{\beta}^*$: the local estimator uses data available only on the master (first) machine and ignores data available on other machines. In particular, it computes

$$\widehat{\beta}_{\text{local}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_{1i}, \langle \mathbf{x}_{1i}, \boldsymbol{\beta}\rangle) + \lambda||\boldsymbol{\beta}||_1 \qquad (2)$$

using locally available data. The local procedure is efficient in both communication and computation, however, the resulting estimation error is large compared to an estimator that uses all of the available data. The other idealized baseline is the centralized estimator

$$\widehat{\beta}_{\text{centralize}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n} \ell(y_{ji}, \langle \mathbf{x}_{ji}, \boldsymbol{\beta}\rangle) + \lambda||\boldsymbol{\beta}||_1.$$

[1]University of Chicago, USA [2]Toyota Technological Institute at Chicago, USA [3]Tencent AI Lab, China. Correspondence to: Jialei Wang <jialei@uchicago.edu>, Mladen Kolar <mkolar@chicagobooth.edu>, Nathan Srebro <nati@ttic.edu>, Tong Zhang <tongzhang@tongzhang-ml.org>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[1]Results in the paper easily generalize to a setting where each machine has a different number of observations.

Approach	$n \gtrsim ms^2 \log p$		$ms^2 \log p \gtrsim n \gtrsim s^2 \log p$	
	Communication	Computation	Communication	Computation
Centralize	$n \cdot p$	$\mathtt{T}_{\mathtt{lasso}}(mn, p)$	$n \cdot p$	$\mathtt{T}_{\mathtt{lasso}}(mn, p)$
Avg-Debias	p	$p \cdot \mathtt{T}_{\mathtt{lasso}}(n, p)$	$\times$	$\times$
This paper (EDSL)	p	$2 \cdot \mathtt{T}_{\mathtt{lasso}}(n, p)$	$\log m \cdot p$	$\log m \cdot \mathtt{T}_{\mathtt{lasso}}(n, p)$

Table 1. Comparison of resources required for matching the centralized error bound of various approaches for high-dimensional distributed sparse linear regression problems, where $\mathtt{T}_{\mathtt{lasso}}(n, p)$ is the runtime for solving a generalized lasso problem of size $n \times p$.

Unfortunately, due to data being huge and communication expensive, we cannot compute the centralized estimator, even though it achieves the optimal statistical error.

In a related setting, Lee et al. (2015b) studied a one-shot approach to learning $\boldsymbol{\beta}^*$, called *Avg-Debias*, that is based on averaging the debiased lasso estimators (Zhang & Zhang, 2013). Under strong assumptions on the data generating procedure, their approach matches the centralized error bound after one round of communication. While an encouraging result, there are limitations to this approach, that we list below.

- The debiasing step in Avg-Debias is computationally heavy as it requires each local machine to estimate a $p \times p$ matrix. For example, Javanmard (2014) (section 5.1) transforms the problem of estimating the debiasing matrix Θ into p generalized lasso problems. This is computationally prohibitive for high-dimensional problems (Zhang & Zhang, 2013; Javanmard & Montanari, 2014). In comparison, our procedure requires only solving one ℓ_1 penalized objective in each iteration, which has the same time complexity as computing $\widehat{\beta}_{\text{local}}$ in (2). See Section 2 for details.
- Avg-Debias procedure only matches the statistical error rate of the centralized procedure when the sample size per machine satisfies $n \gtrsim ms^2 \log p$. Our approach improves this sample complexity to $n \gtrsim s^2 \log p$.
- Avg-Debias procedure requires strong conditions on the data generating process. For example, the data matrix is required to satisfy the generalized coherence condition for debiasing to work[2]. As we show here, such a condition is not needed for consistent high-dimensional estimation in a distributed setting. Instead, we only require standard restricted eigenvalue condition that are commonly assumed in the high-dimensional estimation literature.

Our method (EDSL) addresses the aforementioned issues of Avg-Debias. Table 1 summarizes the resources required for the approaches discussed above to solve the distributed sparse linear regression problems.

Parallel Work In parallel work (publicly announced on arXiv simultaneously with the results in this contribution), Jordan et al. (2016) present a method which is equivalent to the first iteration of our method, and thus achieves the same computational advantage over Avg-Debias as depicted in the left column of Table 1 and discussed in the first and third bullet points above. Jordan et al. extend the idea in ways different and orthogonal to this submission, by considering also low-dimensional and Bayesian inference problems. Still, for high-dimensional problems, they only consider a one-shot procedure, and so do not achieve statistical optimality in the way our method does, and do not allow using $n \lesssim ms^2 \log p$ samples per machine (see right half of Table 1). The improved one-shot approach is thus a parallel contribution, made concurrently by Jordan et al. and by us, while the multi-step approach and accompanied reduction in required number of samples (discusse in the second bullet point above) and improvement in statistical accuracy is a distinct contribution of this this submission.

Other Related Work A large body of literature exists on distributed optimization for modern massive data sets (Dekel et al., 2012; Duchi et al., 2012; 2014; Zhang et al., 2013b; Zinkevich et al., 2010; Boyd et al., 2011; Balcan et al., 2012; Yang, 2013; Jaggi et al., 2014; Ma et al., 2015; Shamir & Srebro, 2014; Zhang & Xiao, 2015; Lee et al., 2015a; Arjevani & Shamir, 2015). A popular approach to distributed estimation is averaging estimators formed locally by different machines (Mcdonald et al., 2009; Zinkevich et al., 2010; Zhang et al., 2012; Huang & Huo, 2015). Divide-and-conquer procedures also found applications in statistical inference (Zhao et al., 2014a; Cheng & Shang, 2015; Lu et al., 2016). Shamir & Srebro (2014) and Rosenblatt & Nadler (2014) showed that averaging local estimators at the end will have bad dependence on either condition number or dimension of the problem. Yang (2013), Jaggi et al. (2014) and Smith et al. (2016) studied distributed optimization using stochastic (dual) coordinate descent, these approaches try to find a good balance between computation and communication, however, their communication com-

[2]The generalized coherence states that there exists a matrix Θ, such that $||\widehat{\Sigma}\Theta - I_p||_\infty \lesssim \sqrt{\frac{\log p}{n}}$, where $\widehat{\Sigma}$ is the empirical covariance matrix.

plexity depends badly on the condition number. As a result, they are not better than first-order approaches, such as (proximal) accelerated gradient descent (Nesterov, 1983), in terms of communication. Shamir et al. (2014) and Zhang & Xiao (2015) proposed truly communication-efficient distributed optimization algorithms. They leveraged the local second-order information and, as a result, obtained milder dependence on the condition number compared to the first-order approaches (Boyd et al., 2011; Shamir & Srebro, 2014; Ma et al., 2015). Lower bounds were studied in Zhang et al. (2013a), Braverman et al. (2015), and Arjevani & Shamir (2015). However, it is not clear how to extend these existing approaches to problems with non-smooth objectives, including the ℓ_1 regularized problems.

Most of the above mentioned work is focused on estimators that are (asymptotically) linear. Averaging at the end reduces the variance of the these linear estimators, resulting in an estimator that matches the performance of a centralized procedure. Zhang et al. (2013c) studied averaging local estimators obtained by the penalized kernel ridge regression, with the ℓ_2 penalty was chosen smaller than usual to avoid the large bias problem. The situation in a high-dimensional setting is not so straightforward, since the sparsity inducing penalty introduces the bias in a non-linear way. Zhao et al. (2014b) illustrated how averaging debiased composite quantile regression estimators can be used for efficient inference in a high-dimensional setting. Averaging debiased high-dimensional estimators was subsequently used in Lee et al. (2015b) for distributed estimation, multi-task learning (Wang et al., 2015), and statistical inference (Battey et al., 2015).

Notation. We use $[n]$ to denote the set $\{1, \ldots, n\}$. For a vector $a \in \mathbb{R}^n$, we let $\text{support}(a) = \{j \ : \ a_j \neq 0\}$ be the support set, $||a||_q$, $q \in [1, \infty)$, the ℓ_q-norm defined as $||a||_q = (\sum_{i\in[n]} |a_i|^q)^{1/q}$, and $||\mathbf{a}||_\infty = \max_{i\in[n]} |a_i|$. For a matrix $A \in \mathbb{R}^{n_1 \times n_2}$, we use the following element-wise ℓ_∞ matrix norms $||A||_\infty = \max_{i\in[n_1], j\in[n_2]} |a_{ij}|$. Denote $\mathbf{I}_n$ as $n \times n$ identity matrix. For two sequences of numbers $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$, we use $a_n = \mathcal{O}(b_n)$ to denote that $a_n \leqslant C b_n$ for some finite positive constant C, and for all n large enough. If $a_n = \mathcal{O}(b_n)$ and $b_n = \mathcal{O}(a_n)$, we use the notation $a_n \asymp b_n$. We also use $a_n \lesssim b_n$ for $a_n = \mathcal{O}(b_n)$ and $a_n \gtrsim b_n$ for $b_n = \mathcal{O}(a_n)$.

Paper Organization. We describe our method in Section 2, and present the main results in the context of sparse linear regression in Section 3, and provide a generalized theory in Section 4. We demonstrate the effectiveness of the proposal via experiments in Section 5, and conclude the paper with discussions in Section 6. In Appendix, in Section A we illustrate some concrete examples of the general results in Section 4, and all proofs are deferred in Section B. More experimental results are presented in Section C.

Algorithm 1 Efficient Distributed Sparse Learning (EDSL).

Input: Data $\{\mathbf{x}_{ji}, y_{ji}\}_{j\in[m], i\in[n]}$, loss function $\ell(\cdot, \cdot)$.
Initialization: The master obtains $\widehat{\boldsymbol{\beta}}_0$ by minimizing (3), and broadcast $\widehat{\boldsymbol{\beta}}_0$ to every worker.
for $t = 0, 1, \ldots$ **do**
  Workers:
  for $j = 2, 3, \ldots, m$ **do**
    if *Receive* $\widehat{\boldsymbol{\beta}}_t$ *from the master* **then**
      Calculate gradient $\nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)$ and send it to the master.
    end
  end
  Master:
  if *Receive* $\{\nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)\}_{j=2}^m$ *from all workers* **then**
    Obtain $\widehat{\boldsymbol{\beta}}_{t+1}$ by solving the shifted ℓ_1 regularized problem in (4).
    Broadcast $\widehat{\boldsymbol{\beta}}_{t+1}$ to every worker.
  end
end

2. Methodology

In this section, we detail our procedure for estimating $\boldsymbol{\beta}^*$ in a distributed setting. Algorithm 1 provides an outline of the steps executed by the master and worker nodes. Let

$$\mathcal{L}_j(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^n \ell(y_{ji}, \langle \mathbf{x}_{ji}, \boldsymbol{\beta} \rangle), \quad j \in [m],$$

be the *empirical* loss at each machine. Our method starts by solving a local ℓ_1 regularized M-estimation program. At iteration $t = 0$, the master (first) machine obtains $\widehat{\boldsymbol{\beta}}_0$ as a minimizer of the following program

$$\min \mathcal{L}_1(\boldsymbol{\beta}) + \lambda_0 ||\boldsymbol{\beta}||_1. \tag{3}$$

The vector $\widehat{\boldsymbol{\beta}}_0$ is broadcasted to all other machines, which use it to compute a gradient of the local loss at $\widehat{\boldsymbol{\beta}}_0$. In particular, each worker computes $\nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_0)$ and communicates it back to the master. This constitutes one round of communication. At the iteration $t + 1$, the master solves the shifted ℓ_1 regularized problem

$$\widehat{\boldsymbol{\beta}}_{t+1} = \arg\min_{\boldsymbol{\beta}} \mathcal{L}_1(\boldsymbol{\beta}) + \left\langle \frac{1}{m} \sum_{j=1}^m \nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla \mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \boldsymbol{\beta} \right\rangle + \lambda_{t+1} ||\boldsymbol{\beta}||_1. \tag{4}$$

A minimizer $\widehat{\boldsymbol{\beta}}_{t+1}$ is communicated to other machines, which use it to compute the local gradient $\nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_{t+1})$ as before.

Formulation (4) is inspired by the proposal in Shamir et al. (2014), where the authors studied distributed optimization

for smooth and strongly convex empirical objectives. Compared to Shamir et al. (2014), we do not use any averaging scheme, which would require additional rounds of communication and, moreover, we add an ℓ_1 regularization term to ensure consistent estimation in high-dimensions. Different from the distributed first-order optimization approaches, the refined objective (4) leverages both global first-order information and local higher-order information. To see this, suppose we set $\lambda_{t+1} = 0$ and that $\mathcal{L}_j(\boldsymbol{\beta})$ is a quadratic objective with invertible Hessian. Then we have the following closed form solution for (4),

$$\widehat{\boldsymbol{\beta}}_{t+1} = \widehat{\boldsymbol{\beta}}_t - \left(\nabla^2 \mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t)\right)^{-1} \left(m^{-1} \sum_{j\in[m]} \nabla \mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)\right),$$

which is exactly a sub-sampled Newton updating rule. Unfortunately for high-dimensional problems, the Hessian is no longer invertible, and a ℓ_1 regularization is added to make the solution well behaved. The regularization parameter λ_t will be chosen in a way, so that it decreases with the iteration number t. As a result we will be able to show that the final estimator performs as well at the centralized solution. We discuss in details how to choose λ_t in the following section.

3. Main Result

We illustrate our main theoretical results in the context of sparse linear regression model

$$y_{ji} = \langle \mathbf{x}_{ji}, \boldsymbol{\beta}^* \rangle + \epsilon_{ji}, \qquad i \in [n], j \in [m], \tag{5}$$

where $\mathbf{x}_{ji}$ is a subgaussian p-dimensional vector of input variables and ϵ_{ji} is i.i.d. mean zero subgaussian noise. The loss function considered is the usual the squared loss $\ell(y, \widehat{y}) = \frac{1}{2}(y - \widehat{y})^2$. With this notation, the centralized approach leads to the lasso estimator (Tibshirani, 1996)

$$\widehat{\boldsymbol{\beta}}_{\text{centralize}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{m} \sum_{j=1}^{m} \mathcal{L}_j(\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1,$$

where the loss at worker j is

$$\mathcal{L}_j(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i\in[n]} (y_{ji} - \langle \boldsymbol{\beta}, \mathbf{x}_{ji} \rangle)^2.$$

Before stating the main result, we provide the definition of the subgaussian norm (Vershynin, 2012).

Definition 1 (Subgaussian norm)**.** *The subgaussian norm $||X||_{\psi_2}$ of a subgaussian p-dimensional random vector X, is defined as*

$$||X||_{\psi_2} = \sup_{x\in\mathbb{S}^{p-1}} \sup_{q>1} q^{-1/2} (\mathbb{E}|\langle X, x\rangle|^q)^{1/q},$$

where $\mathbb{S}^{p-1}$ is the p-dimensional unit sphere.

We also need an assumption on the restricted strong convexity constant (Negahban et al., 2012).

Assumption 2. *We assume that there exists a $\kappa > 0$, such that for any $\Delta \in \mathcal{C}(S, 3)$,*

$$\frac{1}{2n} ||\mathbf{X}_1 \Delta||_2^2 \geqslant \kappa ||\Delta||_2^2,$$

where

$$\mathcal{C}(S, 3) = \{\Delta \in \mathbb{R}^p \mid ||\Delta_{S^c}||_1 \leqslant 3||\Delta_S||_1\}$$

is a restricted cone in $\mathbb{R}^p$, and

$$\mathbf{X}_1 = [\mathbf{x}_{11}^T; \mathbf{x}_{12}^T; \ldots; \mathbf{x}_{1n}^T] \in \mathbb{R}^{n\times p}$$

is the data matrix on the master machine.

When $\mathbf{x}_{ji}$ are randomly drawn from a subgaussian distribution, Assumption (2) is satisfied with high probability as long as $n \gtrsim s \log p$ (Rudelson & Zhou, 2013).

We are now ready to state the estimation error bound for $\widehat{\boldsymbol{\beta}}_{t+1}$ obtained using Algorithm 1.

Theorem 3. *Assume that data are generated from a sparse linear regression model in (5) with $||\mathbf{x}_{ji}||_{\psi_2} \leqslant \sigma_X$ and $||\epsilon_{ji}||_{\psi_2} \leqslant \sigma$. Let*

$$\lambda_{t+1} = \frac{2}{mn} \left|\left| \sum_{j\in[m]} \sum_{i\in[n]} \mathbf{x}_{ji}\epsilon_{ji} \right|\right|_\infty + 2L \left(\max_{j,i} ||\mathbf{x}_{ji}||_\infty^2\right) \cdot ||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1 \cdot \sqrt{\frac{\log(2p/\delta)}{n}} \tag{6}$$

Then for $t \geqslant 0$ we have, with probability at least $1 - 2\delta$,

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \leqslant \frac{1 - a_n^{t+1}}{1 - a_n} \frac{48 s\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(p/\delta)}{mn}} + a_n^{t+1} \frac{s\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(np/\delta)}{n}}, \tag{7}$$

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2 \leqslant \frac{1 - a_n^{t+1}}{1 - a_n} \frac{12\sqrt{s}\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(p/\delta)}{mn}} + a_n^t b_n \frac{s\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(np/\delta)}{n}}, \tag{8}$$

where

$$a_n = \frac{96 s\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(2p/\delta)}{n}}$$

and

$$b_n = \frac{24\sqrt{s}\sigma\sigma_X}{\kappa} \sqrt{\frac{\log(np/\delta)}{n}}.$$

We can simplify the bound obtained in Theorem 3 by looking at the scaling with respect to n, m, s, and p, by treating κ, σ and σ_X as constants. Suppose $n \gtrsim s^2 \log p$ and set

$$\lambda_t \asymp \sqrt{\frac{\log p}{mn}} + \sqrt{\frac{\log p}{n}} \left(s\sqrt{\frac{\log p}{n}}\right)^t.$$

The following error bounds hold for Algorithm 1:

$$||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1 \lesssim_P s\sqrt{\frac{\log p}{mn}} + \left(s\sqrt{\frac{\log p}{n}}\right)^{t+1},$$

$$||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_2 \lesssim_P \sqrt{\frac{s\log p}{mn}} + \left(\sqrt{\frac{s\log p}{n}}\right)\left(s\sqrt{\frac{\log p}{n}}\right)^{t}.$$

We can compare the above bounds to the performance of the local and centralized lasso (Wainwright, 2009; Meinshausen & Yu, 2009; Bickel et al., 2009). For $\widehat{\boldsymbol{\beta}}_{\text{local}}$, we have

$$||\widehat{\boldsymbol{\beta}}_{\text{local}} - \boldsymbol{\beta}^*||_1 \lesssim_P s\sqrt{\frac{\log p}{n}}$$

and

$$||\widehat{\boldsymbol{\beta}}_{\text{local}} - \boldsymbol{\beta}^*||_2 \lesssim_P \sqrt{\frac{s\log p}{n}}.$$

For $\widehat{\boldsymbol{\beta}}_{\text{centralize}}$, we have

$$||\widehat{\boldsymbol{\beta}}_{\text{centralize}} - \boldsymbol{\beta}^*||_1 \lesssim_P s\sqrt{\frac{\log p}{mn}}$$

and

$$||\widehat{\boldsymbol{\beta}}_{\text{centralize}} - \boldsymbol{\beta}^*||_2 \lesssim_P \sqrt{\frac{s\log p}{mn}}.$$

We see that after one round of communication, we have

$$||\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}^*||_1 \lesssim_P s\sqrt{\frac{\log p}{mn}} + \frac{s^2\log p}{n}$$

and

$$||\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}^*||_2 \lesssim_P \sqrt{\frac{s\log p}{mn}} + \frac{s^{3/2}\log p}{n}.$$

These bounds match the results in Lee et al. (2015b) without expensive debiasing step. Furthermore, when $m \lesssim \frac{n}{s^2\log p}$, they match the performance of the centralized lasso. Finally, as long as $t \gtrsim \log m$ and $n \gtrsim s^2\log p$, it is easy to check that $\left(s\sqrt{\frac{\log p}{n}}\right)^{t+1} \lesssim s\sqrt{\frac{\log p}{mn}}$. Therefore,

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \lesssim_P s\sqrt{\frac{\log p}{mn}}$$

and

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2 \lesssim_P \sqrt{\frac{s\log p}{mn}},$$

which matches the centralized lasso performance without additional error terms. That is, as long as $n \gtrsim s^2\log p$, the rounds of communication to matches centralized procedure only increase logarithmically with the number of machines and independent of other parameters. Differently, for distributed learning methods studied in the literature for minimizing smooth objectives, the rounds of communication to match centralized procedure increase polynomially with m (see table 1 in (Zhang & Xiao, 2015)). This is because here we exploit the underlying restricted strong convexity from empirical loss functions, while prior work on distributed minimization of smooth objectives (Shamir et al., 2014; Zhang & Xiao, 2015) only consider strong convexity explicitly from regularization.

4. Generalized Theory and Proof Sketch

In order to establish Theorem 3, we prove an error bound on $\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ for a general loss $\ell(\cdot,\cdot)$ and $\widehat{\boldsymbol{\beta}}$ obtained using Algorithm 1. To simplify the presentation, we assume that the domain $\mathcal{X}$ is bounded and that the loss function $\ell(\cdot,\cdot)$ is smooth.

Assumption 4. *The loss $\ell(\cdot,\cdot)$ is L-smooth with respect to the second argument:*

$$\ell'(a,b) - \ell'(a,c) \leqslant L|b-c|, \qquad \forall a,b,c \in \mathbb{R}$$

Furthermore, $|\ell'''(a,b)| \leqslant M$ for all $a,b \in \mathbb{R}$.

Commonly used loss functions in statistical learning, including the squared loss for regression and logistic loss for classification, satisfy this assumption (Zhang et al., 2013b).

Next, we state the restricted strong convexity condition for a general loss function (Negahban et al., 2012).

Assumption 5. *There exists $\kappa > 0$ such that for any $\Delta \in \mathcal{C}(S,3)$*

$$\mathcal{L}_1(\boldsymbol{\beta}^* + \Delta) - \mathcal{L}_1(\boldsymbol{\beta}^*) - \langle \nabla\mathcal{L}_1(\boldsymbol{\beta}^*), \Delta\rangle \geqslant \kappa||\Delta||_2^2,$$

with $\mathcal{C}(S,3) = \{\Delta \in \mathbb{R}^p | ||\Delta_{S^c}||_1 \leqslant 3||\Delta_S||_1\}$.

The restricted strong convexity holds with high probability for a wide range of models and designs and it is commonly assumed for showing consistent estimation in high-dimensions (see, for example, van de Geer & Bühlmann, 2009; Negahban et al., 2012; Raskutti et al., 2010; Rudelson & Zhou, 2013, for details).

Our main theoretical result establishes a recursive estimation error bound, which relates the estimation error $||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||$ to that of the previous iteration $||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1$.

Theorem 6. *Suppose Assumption 4 and 5 holds. Let*

$$\begin{aligned}\lambda_{t+1} =& 2\left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty \\ &+ 2L\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)||\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_t||_1\sqrt{\frac{\log(2p/\delta)}{n}} \\ &+ 2M\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right)\left(||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1^2\right).\end{aligned} \tag{9}$$

Then with probability at least $1-\delta$, *we have*

$$||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_1 \leqslant \frac{48s}{\kappa}\left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty + \frac{48sL}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)||\boldsymbol{\beta}^*-\widehat{\boldsymbol{\beta}}_t||_1\sqrt{\frac{\log(2p/\delta)}{n}} + \frac{48sM}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right)\left(||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1^2\right),$$

and

$$||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_2 \leqslant \frac{12\sqrt{s}}{\kappa}\left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty + \frac{12\sqrt{s}L}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)||\boldsymbol{\beta}^*-\widehat{\boldsymbol{\beta}}_t||_1\sqrt{\frac{\log(2p/\delta)}{n}} + \frac{4\sqrt{s}M}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right)\left(||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1^2\right).$$

Theorem 6 upper bounds the estimation error $||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_1$ as a function of $||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1$. Applying Theorem 6 iteratively, we immediately obtain the following estimation error bound which depends on the quality of local ℓ_1 regularized estimation $||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1$.

Corollary 7. *Suppose the conditions of Theorem 6 are satisfied. Furthermore, suppose that for all* t, *we have*

$$M\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty\right)||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1 \leqslant L\sqrt{\frac{\log(2p/\delta)}{n}}. \quad (10)$$

Then with probability at least $1-\delta$, *we have*

$$||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_1 \leqslant a_n^{t+1}||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1 + (1-a_n)^{-1}(1-a_n^{t+1})\cdot\frac{48s}{\kappa}\cdot\left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty$$

and

$$||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_2 \leqslant a_n^t b_n\cdot||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1 + (1-a_n)^{-1}(1-a_n^{t+1})\cdot\frac{12\sqrt{s}}{\kappa}\cdot\left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty,$$

where

$$a_n = \frac{96sL}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)\sqrt{\frac{\log(2p/\delta)}{n}}$$

and

$$b_n = \frac{24\sqrt{s}L}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)\sqrt{\frac{\log(2p/\delta)}{n}}.$$

For the quadratic loss we have that $M=0$ and the condition in (10) holds. For other types of losses, condition in (10) will be true for t large enough when $m \gtrsim s^2$, leading to local exponential rate of convergence until reaching statistical optimal region.

4.1. Proof Sketch of Theorem 6

We first analyze how the estimation error bound decreases after one round of communication. In particular, we bound $||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||$ with $||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||$. Define

$$\widetilde{\mathcal{L}}_1(\boldsymbol{\beta},\widehat{\boldsymbol{\beta}}_t) = \mathcal{L}_1(\boldsymbol{\beta}) + \left\langle \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \boldsymbol{\beta}\right\rangle. \quad (11)$$

Then

$$\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta},\widehat{\boldsymbol{\beta}}_t) = \nabla\mathcal{L}_1(\boldsymbol{\beta}) + \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t).$$

The following lemma bounds the ℓ_∞ norm of $\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta},\widehat{\boldsymbol{\beta}}_t)$.

Lemma 8. *With probability at least* $1-\delta$, *we have*

$$\left|\left|\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)\right|\right|_\infty \leqslant \left|\left|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty + 2L\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)||\boldsymbol{\beta}^*-\widehat{\boldsymbol{\beta}}_t||_1\sqrt{\frac{\log(2p/\delta)}{n}} + M\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right)\left(||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1^2\right).$$

The lemma bounds the magnitude of the gradient of the loss at optimum point $\boldsymbol{\beta}^*$. This will be used to guide our choice of the ℓ_1 regularization parameter λ_{t+1} in (4). The following lemma shows that as long as λ_{t+1} is large enough, it is guaranteed that $\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*$ is in a restricted cone.

Lemma 9. *Suppose*

$$\lambda_{t+1}/2 \geqslant \left|\left|\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)\right|\right|_\infty.$$

Then with probability at least $1-\delta$, *we have* $\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^* \in \mathcal{C}(S,3)$.

Based on the conic condition and restricted strong convexity condition, we can obtain the recursive error bound stated in Theorem 6 following the proof strategy as in Negahban et al. (2012).

Applications Theorem 6 can be used to establish statistical guarantees for more general sparse learning problems, for example consider the logistic regression is a popular classification model where the binary label $y_{ji} \in \{-1,1\}$ is drawn according to a Bernoulli distribution:

$$\mathbb{P}(y_{ji}=\pm 1|\mathbf{x}_{ji}) = \frac{\exp(y_{ji}\langle\mathbf{x}_{ji},\boldsymbol{\beta}^*\rangle)}{\exp(y_{ji}\langle\mathbf{x}_{ji},\boldsymbol{\beta}^*\rangle)+1}, \quad (12)$$

we can establish local exponential convergence when applying Algorithm 1 to estimate $\boldsymbol{\beta}^*$ in the high-dimensional logistic model. Section A in Appendix provide formal guarantees and more illustrative examples.

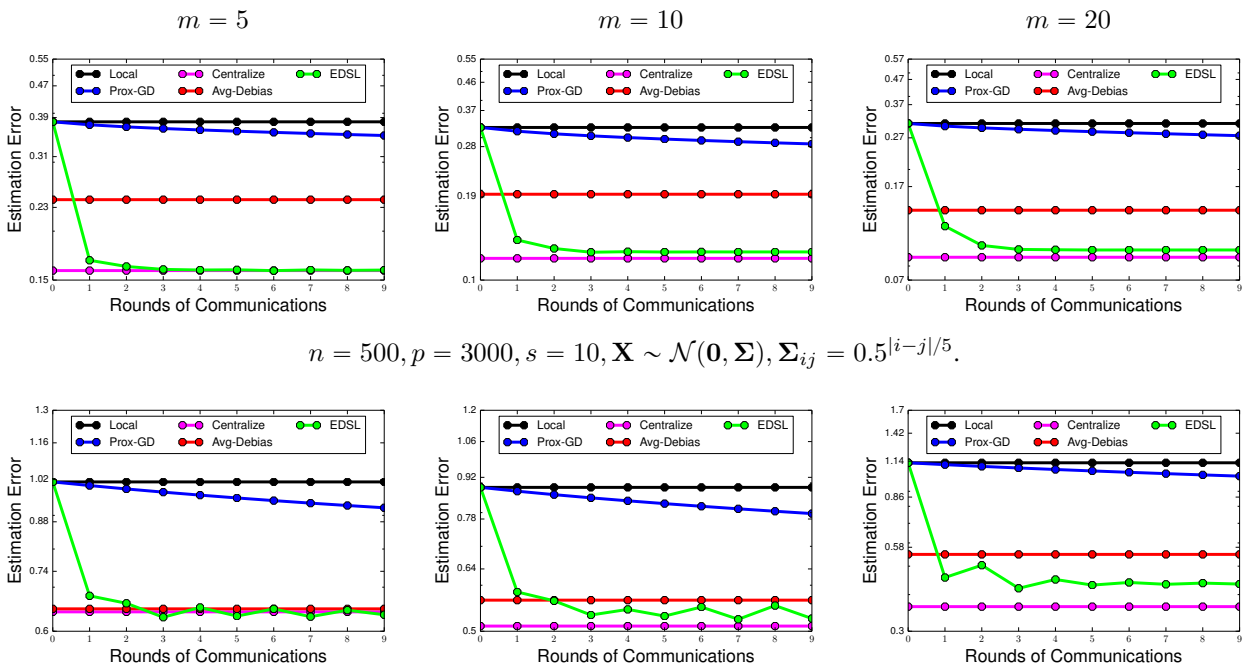


Figure 1. Comparison of various algorithms for distributed sparse learning on simulated data, first row: sparse linear regression, second row: sparse logistic regression.

5. Experiments

In this section we present empirical comparisons between various approaches on both simulated and real world datasets [3]. We run the algorithms for both distributed regression and classification problems, and compare with the following algorithms: i) Local; ii) Centralize; iii) Distributed proximal gradient descent (Prox GD); iv) Avg-Debias (Lee et al., 2015b) with hard thresholding, and v) the proposed EDSL approach.

5.1. Simulations

We first examine the algorithms on simulated data. We generate $\{\mathbf{x}_{ji}\}_{j\in[m],i\in[n]}$ from a multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma}$. The covariance $\mathbf{\Sigma}$ controls the condition number of the problem and we will varying it to see how the performance changes. We set $\Sigma_{ij} = 0.5^{|i-j|}$ for the well-conditioned setting and $\Sigma_{ij} = 0.5^{|i-j|/5}$ for the ill-conditioned setting. The response variable $\{y_{ji}\}_{j\in[m],i\in[n]}$ are drawn from (5) and (12) for regression and classification problems, respectively. For regression, the noise ϵ_{ji} is sampled from a standard normal distribution. The true model $\boldsymbol{\beta}^*$ is set to be s-sparse, where the first s-entries are sampled i.i.d. from a uniform distribution in $[0, 1]$, and the other entries are set to zero.

We run experiments with various (n, p, m, s) settings[4]. The estimation error $||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_2$ is shown versus rounds of communications for for Prox GD and the proposed EDSL algorithm. We also plot the estimation error of Local, Avg-Debias, and Centralize as horizontal lines, since the communication cost is fixed for for these algorithms[5][6]. Figure 1 summarize the results, averaged across 10 independent trials. We have the following observations:

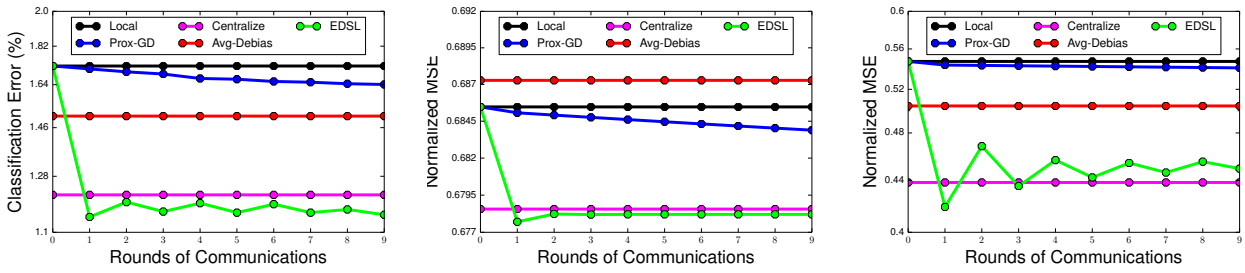
- The Avg-Debias approach obtained much better estimation error compared to Local after one round of communication and sometimes performed quite close to Centralize. However, in most cases, there is still a gap compared with Centralize, especially when the problem is not well-conditioned or m is large.
- ProxGD converges very slow when the condition number becomes bad ($\Sigma_{ij} = 0.5^{|i-j|/5}$ case).
- As theory suggests, EDSL obtained a solution that is

[3] Please refer to Section C in Appendix for full experimental results and more details

[4] n: sample size per machine, p: problem dimension, m: number of machines, s: true support size.

[5] these algorithms have zero, one-shot and full communications, respectively.

[6] To give some senses about computational cost, for a problem with $n = 200, p = 1000$, at each round EDSL takes about 0.048s, while Avg-Debias takes about 40.334s.



mnist 1 vs 2 connect4 dna

Figure 2. Comparison of various approaches for distributed sparse regression and classification on real world datasets.

competitive with Avg-Debias after one round of communication. The estimation error decreases to match performance of Centralize within few rounds of communications; typically less than 5, even though the theory suggests EDSL will match the performance of centralize within $\mathcal{O}(\log m)$ rounds of communication.

Above experiments illustrate our theoretical results in finite samples. As suggested by theory, when sample size per machine n is relatively small, one round of communication is not sufficient to make Avg-Debias matches the performance of centralized procedure. However, EDSL could match the performance of Avg-Debias with one round of communication and further improve the estimation quality by exponentially reducing the gap between centralized procedure with Avg-Debias, until matching the centralized performance. Thus, the proposed EDSL improves the Avg-Debias approach both computationally and statistically.

5.2. Real-world Data Evaluation

In this section, we compare the distributed sparse learning algorithms on several real world datasets. For all data sets, we use 60% of data for training, 20% as held-out validation set for tuning the parameters, and the remaining 20% for testing. We randomly partition data 10 times and report the average performance on the test set. For regression tasks, the evaluation metric is the normalized Mean Squared Error (normalized MSE), while for classification tasks we report the miss-classification error. We randomly partition the data on $m = 10$ machines. A subset of the results are plotted in Figure 2 where for some data sets the performance of Avg-Debias is significantly worse than others (mostly because the debiasing step fails), thus we omit these plots.

Since there is no well-specified model on these datasets, the curves behave quite differently on different data sets. However, a large gap between the local and centralized procedure is consistent as the later uses 10 times more data. Avg-Debias often fails on these real datasets and performs much worse than in the simulations, the main reason might be that the assumptions, such as well-specified model or generalized coherence condition, fail, then Avg-Debias can totally fail and produce solution even much worse than the local. Nevertheless, the proposed EDSL performs quite robust on real world data sets, and can often output a solution which is highly competitive with the centralized model within a few rounds of communications. We also observed a slight "zig-zag" behavior for EDSL approach on some data sets. For example, on the mushrooms data set, the predictive performance of EDSL is not stable. In sum, the experimental results on real world data sets verified that the proposed EDSL method is effective for distributed sparse learning problems.

6. Conclusion and Discussion

We proposed a novel approach for distributed learning with sparsity, which is efficient in both computation and communication. Our theoretical analysis showed that the proposed method works under weaker conditions than Avg-Debias estimator while matches its error bound with one-round communication. Furthermore, the estimation error can be improved with a logarithmic more rounds of communication until matching the centralized procedure. Experiments on both simulated and real-world data demonstrate that the proposed method significantly improves the performance over one shot averaging approaches, and matches the centralized procedure with few iterations.

There might be several ways to improve this work. As we see in real data experiments, the proposed approach can still perform slightly worse than the centralized approach on certain datasets. It is interesting to explore how to make EDSL provably work under even weaker assumptions. For example, EDSL requires $\mathcal{O}(s^2 \log p)$ samples per machine to match the centralized method in $\mathcal{O}(\log m)$ rounds of communications, however, it is not clear whether the sample size requirement can be improved, while still maintaining low-communication cost. Last but not the least, it is interesting to explore presented ideas to improve the computational cost of communication-efficient distributed multi-task learning with shared support (Wang et al., 2015).

References

Arjevani, Yossi and Shamir, Ohad. Communication complexity of distributed convex learning and optimization. *ArXiv e-prints, arXiv:1506.01900*, June 2015.

Balcan, Maria-Florina, Blum, Avrim, Fine, Shai, and Mansour, Yishay. Distributed learning, communication complexity and privacy. In Mannor, Shie, Srebro, Nathan, and Williamson, Robert C. (eds.), *JMLR W&CP 23: COLT 2012*, volume 23, pp. 26.1–26.22, 2012.

Battey, Heather, Fan, Jianqing, Liu, Han, Lu, Junwei, and Zhu, Ziwei. Distributed estimation and inference with statistical guarantees. *ArXiv e-prints, arXiv:1509.05457*, September 2015.

Bickel, Peter J., Ritov, Ya'acov, and Tsybakov, Alexandre B. Simultaneous analysis of lasso and Dantzig selector. *Ann. Stat.*, 37(4):1705–1732, 2009. doi: 10.1214/08-AOS620.

Boyd, Stephen P., Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

Braverman, Mark, Garg, Ankit, Ma, Tengyu, Nguyen, Huy L., and Woodruff, David P. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. *ArXiv e-prints, arXiv:1506.07216*, June 2015.

Cheng, Guang and Shang, Zuofeng. Computational limits of divide-and-conquer method. *ArXiv e-prints, arXiv:1512.09226*, December 2015.

Dekel, Ofer, Gilad-Bachrach, Ran, Shamir, Ohad, and Xiao, Lin. Optimal distributed online prediction using mini-batches. *J. Mach. Learn. Res.*, 13:165–202, 2012. ISSN 1532-4435.

Duchi, John C., Agarwal, Alekh, and Wainwright, Martin J. Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Trans. Automat. Control*, 57(3):592–606, 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2161027.

Duchi, John C., Jordan, Michael I., Wainwright, Martin J., and Zhang, Yuchen. Optimality guarantees for distributed statistical estimation. *ArXiv e-prints, arXiv:1405.0782*, May 2014.

Hoeffding, Wassily. Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.*, 58:13–30, 1963. ISSN 0162-1459.

Huang, Cheng and Huo, Xiaoming. A distributed one-step estimator. *ArXiv e-prints, arXiv:1511.01443*, November 2015.

Jaggi, Martin, Smith, Virginia, Takác, Martin, Terhorst, Jonathan, Krishnan, Sanjay, Hofmann, Thomas, and Jordan, Michael I. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pp. 3068–3076, 2014.

Javanmard, Adel. Inference and estimation in high-dimensional data analysis. *PhD dissertation, Stanford University*, 2014.

Javanmard, Adel and Montanari, Andrea. Confidence intervals and hypothesis testing for high-dimensional regression. *J. Mach. Learn. Res.*, 15(Oct):2869–2909, 2014.

Jordan, Michael I, Lee, Jason D, and Yang, Yun. Communication-efficient distributed statistical learning. *arXiv preprint arXiv:1605.07689*, 2016.

Lee, Jason D., Lin, Qihang, Ma, Tengyu, and Yang, Tianbao. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *ArXiv e-prints, arXiv:1507.07595*, July 2015a.

Lee, Jason D., Sun, Yuekai, Liu, Qiang, and Taylor, Jonathan E. Communication-efficient sparse regression: a one-shot approach. *ArXiv e-prints, arXiv:1503.04337*, 2015b.

Lu, Junwei, Cheng, Guang, and Liu, Han. Nonparametric heterogeneity testing for massive data. *ArXiv e-prints, arXiv:1601.06212*, January 2016.

Ma, Chenxin, Smith, Virginia, Jaggi, Martin, Jordan, Michael I., Richtrik, Peter, and Tak, Martin. Adding vs. averaging in distributed primal-dual optimization. *ArXiv e-prints, arXiv:1502.03508*, February 2015.

McCullagh, P. and Nelder, J. A. *Generalized linear models*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1989. ISBN 0-412-31760-5. doi: 10.1007/978-1-4899-3242-6. Second edition [of MR0727836].

Mcdonald, Ryan, Mohri, Mehryar, Silberman, Nathan, Walker, Dan, and Mann, Gideon S. Efficient large-scale distributed training of conditional maximum entropy models. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 1231–1239. Curran Associates, Inc., 2009.

Meinshausen, Nicolas and Bühlmann, Peter. High dimensional graphs and variable selection with the lasso. *Ann. Stat.*, 34(3):1436–1462, 2006.

Meinshausen, Nicolas and Yu, B. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Stat.*, 37(1):246–270, 2009.

Negahban, Sahand N, Ravikumar, Pradeep, Wainwright, Martin J., and Yu, Bin. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Stat. Sci.*, 27(4):538–557, 2012.

Nesterov, Yurii. A method of solving a convex program-

ming problem with convergence rate $\wr(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pp. 372–376, 1983.

Raskutti, Garvesh, Wainwright, Martin J, and Yu, Bin. Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11: 2241–2259, 2010.

Ravikumar, Pradeep, Wainwright, Martin J., and Lafferty, J. D. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Ann. Stat.*, 38(3):1287–1319, 2010.

Rosenblatt, Jonathan and Nadler, Boaz. On the optimality of averaging in distributed statistical learning. *ArXiv e-prints, arXiv:1407.2724*, July 2014.

Rudelson, Mark and Zhou, Shuheng. Reconstruction from anisotropic random measurements. *Information Theory, IEEE Transactions on*, 59(6):3434–3447, 2013.

Shamir, Ohad and Srebro, Nathan. Distributed stochastic optimization and learning. In *52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2014*, pp. 850–857. IEEE, 2014.

Shamir, Ohad, Srebro, Nathan, and Zhang, Tong. Communication efficient distributed optimization using an approximate newton-type method. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 1000–1008, 2014.

Smith, Virginia, Forte, Simone, Ma, Chenxin, Takac, Martin, Jordan, Michael I, and Jaggi, Martin. Cocoa: A general framework for communication-efficient distributed optimization. *arXiv preprint arXiv:1611.02189*, 2016.

Tibshirani, Robert J. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*, 58(1):267–288, 1996. ISSN 0035-9246.

van de Geer, Sara A. High-dimensional generalized linear models and the lasso. *Ann. Stat.*, 36(2):614–645, 2008.

van de Geer, Sara A. and Bühlmann, Peter. On the conditions used to prove oracle results for the lasso. *Electron. J. Stat.*, 3:1360–1392, 2009.

Vershynin, Roman. Introduction to the non-asymptotic analysis of random matrices. In Eldar, Y. C. and Kutyniok, G. (eds.), *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

Wainwright, Martin J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Trans. Inf. Theory*, 55(5):2183–2202, 2009. ISSN 0018-9448. doi: 10.1109/TIT.2009.2016018.

Wang, Jialei, Kolar, Mladen, and Srebro, Nathan. Distributed multitask learning. *ArXiv e-prints, arXiv:1510.00633*, October 2015.

Wu, Tong Tong, Chen, Yi Fang, Hastie, Trevor J., Sobel, Eric, and Lange, Kenneth L. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009. doi: 10.1093/bioinformatics/btp041.

Yang, Tianbao. Trading computation for communication: Distributed stochastic dual coordinate ascent. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 629–637. Curran Associates, Inc., 2013.

Yuan, M. and Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Zhang, Cun-Hui and Zhang, Stephanie S. Confidence intervals for low dimensional parameters in high dimensional linear models. *J. R. Stat. Soc. B*, 76(1):217–242, Jul 2013.

Zhang, Yuchen and Xiao, Lin. Communication-efficient distributed optimization of self-concordant empirical loss. *ArXiv e-prints, arXiv:1501.00263*, 2015.

Zhang, Yuchen, Wainwright, Martin J., and Duchi, John C. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pp. 1502–1510, 2012.

Zhang, Yuchen, Duchi, John C., Jordan, Michael I., and Wainwright, Martin J. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pp. 2328–2336, 2013a.

Zhang, Yuchen, Duchi, John C., and Wainwright, Martin J. Communication-efficient algorithms for statistical optimization. *J. Mach. Learn. Res.*, 14:3321–3363, 2013b. ISSN 1532-4435.

Zhang, Yuchen, Duchi, John C, and Wainwright, Martin J. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *arXiv preprint arXiv:1305.5029*, 2013c.

Zhao, Tianqi, Cheng, Guang, and Liu, Han. A partially linear framework for massive heterogeneous data. *ArXiv e-prints, arXiv:1410.8570*, October 2014a.

Zhao, Tianqi, Kolar, Mladen, and Liu, Han. A general framework for robust testing and confidence regions in high-dimensional quantile regression. *ArXiv e-prints, arXiv:1412.8724*, December 2014b.

Zhu, Ji and Hastie, Trevor J. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004. doi: 10.1093/biostatistics/kxg046.

Zinkevich, Martin, Weimer, Markus, Smola, Alexander J., and Li, Lihong. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing*, pp. 2595–2603. Curran Associates, Inc., 2010.

A. Illustrative Examples of General Sparse Learning Problems

In this section we discuss additional examples of high-dimensional statistical learning problems for which Theorem 6 is applicable.

A.1. Sparse Logistic Regression

For logistic model, performing maximum likelihood estimation (MLE) on (12) leads to the logistic loss function $\ell(y_{ji}, \langle \boldsymbol{\beta}, \mathbf{x}_{ji} \rangle) = \log(1 + \exp(-y_{ji}\langle \boldsymbol{\beta}, \mathbf{x}_{ji} \rangle))$. For high-dimensional problems, when we add a ℓ_1 regularization, we obtain the ℓ_1 regularized logistic regression model (Zhu & Hastie, 2004, Wu et al., 2009):

$$\widehat{\boldsymbol{\beta}}_{\text{centralize}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{mn} \sum_{j\in[m]} \sum_{i\in[n]} \log(1 + \exp(-y_{ji}\langle \boldsymbol{\beta}, \mathbf{x}_{ji} \rangle)) + \lambda ||\boldsymbol{\beta}||_1.$$

The logistic loss is $\frac{1}{4}$-smooth, and we also know $M = \frac{1}{4}$ because of self-concordance (Zhang & Xiao, 2015). Let $\mathcal{L}_j(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i\in[n]} \log(1 + \exp(-\mathbf{y}_{ji}\langle \boldsymbol{\beta}, \mathbf{x}_{ji} \rangle))$, (Negahban et al., 2012) showed that if $\mathbf{x}_{ji}$ are drawn from mean zero distribution with sub-Gaussian tails, then $\mathcal{L}_1(\boldsymbol{\beta})$ satisfies the restricted strong condition (5). Moreover, we have the following control on the quantity $\left|\left| \frac{1}{m}\sum_{j\in[m]} \nabla \mathcal{L}_j(\boldsymbol{\beta}^*) \right|\right|_\infty$.

Lemma 10. *Then we have the following upper bound holds in probability at least* $1 - \delta$*:*

$$\left|\left| \frac{1}{m} \sum_{j\in[m]} \nabla \mathcal{L}_j(\boldsymbol{\beta}^*) \right|\right|_\infty \lesssim ||\mathbf{x}_{ji}||_\infty \sqrt{\frac{2\log(p/\delta)}{mn}}.$$

The following ℓ_1 error bound states the estimation error for logistic regression with ℓ_1 regularization, which was established, for example, in (van de Geer, 2008, Negahban et al., 2012).

Lemma 11. *Under the model* (12)*, when* $n \geqslant (64/\kappa)s\log p$*, we have the following estimation error bound for* $\widehat{\boldsymbol{\beta}}_0$ *holds with probability at least* $1 - \delta$*:*

$$||\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*||_1 \lesssim \frac{s\sigma_X}{\kappa}\sqrt{\frac{2\log(np/\delta)}{n}}.$$

With above analysis for sparse logistic regression model with random design, we are ready to present the results for the estimation error bound which established local exponential convergence.

Corollary 12. *Under sparse logistic regression model with random design, and set* λ_{t+1} *as* (9)*. If the following condition holds for some* $T \geqslant 0$*:*

$$||\widehat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}^*||_1 \leqslant 4\sqrt{\frac{\log(2p/\delta)}{n}}. \tag{13}$$

Then with probability at least $1 - 2\delta$*, we have the following estimation error bound for all* $t \geqslant T$*:*

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \leqslant \frac{1 - a_n^{t-T+1}}{1 - a_n} \frac{96s\sigma\sigma_X}{\kappa}\sqrt{\frac{\log(p/\delta)}{mn}} + 4a_n^{t-T+1}\sqrt{\frac{\log(2p/\delta)}{n}}, \tag{14}$$

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2 \leqslant \frac{1 - a_n^{t-T+1}}{1 - a_n} \frac{4\sqrt{s}\sigma\sigma_X}{\kappa}\sqrt{\frac{\log(p/\delta)}{mn}} + 4a_n^{t-T}b_n\sqrt{\frac{\log(2p/\delta)}{n}}, \tag{15}$$

where

$$a_n = \frac{24s\sigma\sigma_X}{\kappa}\sqrt{\frac{\log(2p/\delta)}{n}} \quad \textit{and} \quad b_n = \frac{\sqrt{s}\sigma\sigma_X}{\kappa}\sqrt{\frac{\log(np/\delta)}{n}}.$$

A.2. High-dimensional Generalized Linear Models

The results are readily extendable to other high-dimensional generalized linear models (McCullagh & Nelder, 1989, van de Geer, 2008), where the response variable $y_{ji} \in \mathcal{Y}$ is drawn from the distribution

$$\mathbb{P}(y_{ji}|\mathbf{x}_{ji}) \propto \exp\left(\frac{y_{ji}\langle \mathbf{x}_{ji}, \boldsymbol{\beta}^* \rangle - \Phi(\langle \mathbf{x}_{ji}, \boldsymbol{\beta}^* \rangle)}{A(\sigma)}\right),$$

where $\Phi(\cdot)$ is a link function and $A(\sigma)$ is a scale parameter. Under the random subgaussian design, as long as the loss function has Lipschitz gradient, then the algorithm and corresponding estimation error bound and be applied.

A.3. High-dimensional Graphical Models

The results can also be used for the distributed unsupervised learning setting where the task is to learn a sparse graphical structure that represents the conditional independence between variables. Widely studied graphical models are Gaussian graphical models (Meinshausen & Bühlmann, 2006, Yuan & Lin, 2007) for continuous data and Ising graphical models (Ravikumar et al., 2010) for binary observations. As shown in (Meinshausen & Bühlmann, 2006, Ravikumar et al., 2010), these model selection problems can be reduced to solving parallel ℓ_1 regularized linear regression and logistic regression problems, respectively. Thus the approach presented in this paper can be readily applicable for these tasks.

B. Proofs

The section contains proofs of some theorems and lemmas stated in the main paper.

B.1. Proof of Lemma 8

Proof. Recall the definition of $\widetilde{\mathcal{L}}_1$ from (11). We have

$$\begin{aligned}\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t) &= \nabla\mathcal{L}_1(\boldsymbol{\beta}^*) + \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t)\\ &= \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*) + \nabla\mathcal{L}_1(\boldsymbol{\beta}^*) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t) - \left(\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*) - \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)\right).\end{aligned}$$

Using the triangle inequality

$$\begin{aligned}&\left\|\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)\right\|_\infty\\ &\leqslant \left\|\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right\|_\infty + \left\|\nabla\mathcal{L}_1(\boldsymbol{\beta}^*) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t) - \left(\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*) - \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)\right)\right\|_\infty.\end{aligned}$$

We focus on bounding the second term in the right-hand-side inequality above. Let $\tau_{ji} = \ell'(y_{ji}, \langle\boldsymbol{\beta}^*, \mathbf{x}_{ji}\rangle)$ and define $\mathbf{v}_{ji}(\widehat{\boldsymbol{\beta}}_t) \in \mathbb{R}^p$:

$$\begin{aligned}\mathbf{v}_{ji}(\widehat{\boldsymbol{\beta}}_t) &= \mathbf{x}_{ji}(\ell'(y_{ji},\langle\boldsymbol{\beta}^*,\mathbf{x}_{ji}\rangle) - \ell'(y_{ji},\langle\widehat{\boldsymbol{\beta}}_t,\mathbf{x}_{ji}\rangle))\\ &= \tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\left(\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*\right) + \mathbf{x}_{ji}\frac{\ell'''(y_{ji},\mathbf{u}_{ji})}{2}(\langle\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*, \mathbf{x}_{ji}\rangle)^2\end{aligned}$$

where $\mathbf{u}_{ji}$ is a number between $\langle\widehat{\boldsymbol{\beta}}_t, \mathbf{x}_{ji}\rangle$ and $\langle\boldsymbol{\beta}^*, \mathbf{x}_{ji}\rangle$. With this notation

$$\begin{aligned}&\left\|\nabla\mathcal{L}_1(\boldsymbol{\beta}^*) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t) - \left(\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*) - \frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t)\right)\right\|_\infty\\ &\leqslant \left\|\frac{1}{n}\sum_{i\in[n]}\mathbf{v}_{1i}(\widehat{\boldsymbol{\beta}}_t) - \frac{1}{mn}\sum_j\sum_i\mathbf{v}_{ji}(\widehat{\boldsymbol{\beta}}_t)\right\|_\infty\\ &\leqslant \left\|\frac{1}{n}\sum_i\tau_{1i}\mathbf{x}_{1i}\mathbf{x}_{1i}^T(\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*) - \frac{1}{mn}\sum_j\sum_i\tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T(\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*)\right\|_\infty + M\cdot\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right)\cdot||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1^2.\end{aligned}$$

The first term above can be further upper bounded by

$$\begin{aligned}
&\left\|\frac{1}{n}\sum_j \tau_{1i}\mathbf{x}_{1i}\mathbf{x}_{1i}^T(\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*)-\frac{1}{mn}\sum_j\sum_i \tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T(\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*)\right\|_\infty \\
&\quad\leqslant \left\|\frac{1}{n}\sum_j \tau_{1i}\mathbf{x}_{1i}\mathbf{x}_{1i}^T-\frac{1}{mn}\sum_j\sum_i \tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\right\|_\infty \cdot ||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1 \,. \\
&\quad\leqslant \left(\left\|\frac{1}{n}\sum_{i\in[n]} \tau_{1i}\mathbf{x}_{1i}\mathbf{x}_{1i}^T-\mathbb{E}\left[\tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\right]\right\|_\infty+\left\|\frac{1}{mn}\sum_j\sum_i \tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T-\mathbb{E}\left[\tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\right]\right\|_\infty\right)\cdot ||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1 \,.
\end{aligned}$$

Using Hoeffding's inequality together with a union bound, we have with probability at least $1-\delta$,

$$\left\|\frac{1}{n}\sum_{i\in[n]} \tau_{1i}\mathbf{x}_{1i}\mathbf{x}_{1i}^T-\mathbb{E}\left[\tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\right]\right\|_\infty \leqslant L\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)\sqrt{\frac{2\log(2p/\delta)}{n}},$$

and

$$\left\|\frac{1}{mn}\sum_j\sum_i \tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T-\mathbb{E}\left[\tau_{ji}\mathbf{x}_{ji}\mathbf{x}_{ji}^T\right]\right\|_\infty \leqslant L\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)\sqrt{\frac{2\log(2p/\delta)}{mn}}.$$

Combining the bounds, the proof of the lemma is complete. □

B.2. Proof of Lemma 9

Proof. The proof uses ideas presented in (Negahban et al., 2012). By triangle inequality we have

$$\begin{aligned}
||\widehat{\boldsymbol{\beta}}_{t+1}||_1-||\boldsymbol{\beta}^*||_1 =&||\boldsymbol{\beta}^*+(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}+(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1-||\boldsymbol{\beta}^*||_1 \\
\geqslant&||\boldsymbol{\beta}^*+(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1-||\boldsymbol{\beta}^*||_1 \\
=&||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1.
\end{aligned}$$

By the optimality of $\widehat{\boldsymbol{\beta}}_{t+1}$ for (4), we have

$$\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1},\widehat{\boldsymbol{\beta}}_t)+\lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1}||_1-\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)-\lambda_{t+1}||\boldsymbol{\beta}^*||_1\leqslant 0.$$

Thus

$$\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1},\widehat{\boldsymbol{\beta}}_t)-\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)+\lambda_{t+1}(||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1)\leqslant 0.$$

By the convexity of $\widetilde{\mathcal{L}}_1(\cdot,\widehat{\boldsymbol{\beta}}_t)$, we further have

$$\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1},\widehat{\boldsymbol{\beta}}_t)-\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)\geqslant\langle\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t),\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*\rangle.$$

Thus by Hölder's inequality

$$\begin{aligned}
0\geqslant&\langle\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t),\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*\rangle+\lambda_{t+1}(||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1) \\
\geqslant&-||\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*,\widehat{\boldsymbol{\beta}}_t)||_\infty||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_1+\lambda_{t+1}(||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1).
\end{aligned}$$

Under the assumption on λ_{t+1} we further have

$$\begin{aligned}
0\geqslant&-\frac{\lambda_{t+1}}{2}||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_1+\lambda_{t+1}(||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1) \\
=&\frac{\lambda_{t+1}}{2}||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_{S^c}||_1-\frac{3\lambda_{t+1}}{2}||(\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*)_S||_1,
\end{aligned}$$

which completes the proof. □

B.3. Proof of Theorem 6

Proof. For the term $\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1}, \widehat{\boldsymbol{\beta}}_t) - \widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t)$ we have

$$\begin{aligned}
\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1}, \widehat{\boldsymbol{\beta}}_t) - \widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t) =& \mathcal{L}_1(\widehat{\boldsymbol{\beta}}_{t+1}) + \left\langle \frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \widehat{\boldsymbol{\beta}}_{t+1} \right\rangle \\
& - \mathcal{L}_1(\boldsymbol{\beta}^*) - \left\langle \frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \boldsymbol{\beta}^* \right\rangle \\
\geqslant & \langle \nabla\mathcal{L}_1(\boldsymbol{\beta}^*), \widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^* \rangle + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2 \\
& + \left\langle \frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \widehat{\boldsymbol{\beta}}_{t+1} \right\rangle \\
& - \left\langle \frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \boldsymbol{\beta}^* \right\rangle \\
= & \left\langle \nabla\mathcal{L}_1(\boldsymbol{\beta}^*) + \frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\widehat{\boldsymbol{\beta}}_t) - \nabla\mathcal{L}_1(\widehat{\boldsymbol{\beta}}_t), \widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^* \right\rangle \\
& + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2 \\
= & \langle \nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t), \widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^* \rangle + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2,
\end{aligned}$$

where the first inequality we use the restricted strong convexity condition (5). Also by the optimality of $\widehat{\boldsymbol{\beta}}_{t+1}$ for (4), we have

$$\widetilde{\mathcal{L}}_1(\widehat{\boldsymbol{\beta}}_{t+1}, \widehat{\boldsymbol{\beta}}_t) - \widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t) + \lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1}||_1 - \lambda_{t+1}||\boldsymbol{\beta}^*||_1 \leqslant 0.$$

Combining above two inequalities we obtain with probability at least $1-\delta$:

$$\begin{aligned}
\lambda_{t+1}||\boldsymbol{\beta}^*||_1 - \lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1}||_1 \geqslant & \langle \nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t), \widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^* \rangle + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2 \\
\geqslant & -||\nabla\widetilde{\mathcal{L}}_1(\boldsymbol{\beta}^*, \widehat{\boldsymbol{\beta}}_t)||_\infty||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2 \\
\geqslant & -\frac{\lambda_{t+1}}{2}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 + \kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2.
\end{aligned}$$

By triangle inequality that $\lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \geqslant \lambda_{t+1}||\boldsymbol{\beta}^*||_1 - \lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1}||_1$, we have

$$\begin{aligned}
\kappa||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2^2 \leqslant & \frac{3\lambda_{t+1}}{2}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \\
= & \frac{3\lambda_{t+1}}{2}(||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1 + ||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_{S^c}||_1) \\
\leqslant & \frac{3\lambda_{t+1}}{2}(||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1 + 3||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1) \\
= & 6\lambda_{t+1}||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1 \\
\leqslant & 6\sqrt{s}\lambda_{t+1}||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_2 \\
\leqslant & 6\sqrt{s}\lambda_{t+1}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2.
\end{aligned}$$

We get

$$||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2 \leqslant \frac{6\sqrt{s}\lambda_{t+1}}{\kappa}.$$

Substitute λ_{t+1} in (9) concludes the proof for ℓ_2 estimation error bound. For $||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1$, we know

$$\begin{aligned}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \leqslant& ||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1 + ||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_{S^c}||_1 \\ \leqslant& 4||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_1 \leqslant 4\sqrt{s}||(\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*)_S||_2 \\ \leqslant& 4\sqrt{s}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2 \leqslant \frac{24s\lambda_{t+1}}{\kappa},\end{aligned}$$

which obtains the desired bound. □

B.4. Proof of Theorem 3

Proof. Theorem 3 follows from Theorem 6 after we verify some conditions. First, it is easy to see that the quadratic loss $L = 1, M = 0$. Under conditions of Theorem, with probability $1 - \delta$,

$$\left|\left|\frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty \lesssim \sigma\sigma_X\sqrt{\frac{\log(p/\delta)}{mn}}.$$

This follows from Corollary 5.17 of Vershynin (2012). Furthermore, with probability at least $1 - \delta$, we have

$$\max_{j\in[m],i\in[n]} ||\mathbf{x}_{ji}||_\infty \lesssim \sigma_X\sqrt{\log(mnp/\delta)}.$$

Finally,

$$||\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*||_1 \lesssim \frac{s\sigma\sigma_X}{\kappa}\sqrt{\frac{\log(np/\delta)}{n}},$$

with probability at least $1 - \delta$ (Wainwright, 2009, Meinshausen & Yu, 2009, Bickel et al., 2009). Plugging these bounds into Theorem 6 completes the proof.

□

B.5. Proof of Corollary 7

Proof. The proof proceeds by recursively applying Theorem 6 and sum a geometric sequence. For notation simplicity let

$$\begin{aligned}a =& \frac{48s}{\kappa}\left|\left|\frac{1}{m}\sum_{j\in[m]} \nabla\mathcal{L}_j(\boldsymbol{\beta}^*)\right|\right|_\infty, \\ b =& \left(\frac{48sL}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^2\right)\sqrt{\frac{4\log(2p/\delta)}{n}}\right), \\ c =& \frac{48sM}{\kappa}\left(\max_{j,i}||\mathbf{x}_{ji}||_\infty^3\right).\end{aligned}$$

By Theorem 6 we have

$$\begin{aligned}||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_1 \leqslant& a + b||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1 + c||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1^2 \\ \leqslant& a + 2b||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1 \\ \leqslant& a + 2b(a + 2b||\widehat{\boldsymbol{\beta}}_{t-1} - \boldsymbol{\beta}^*||_1) \leqslant \ldots \\ \leqslant& a\sum_{k=0}^{t}(2b)^k + (2b)^{t+1}||\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*||_1. \\ =& \frac{a(1-(2b)^{t+1})}{1-2b} + (2b)^{t+1}||\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*||_1, \end{aligned} \tag{16}$$

which completes the ℓ_1 estimation error bound. For $||\widehat{\boldsymbol{\beta}}_{t+1} - \boldsymbol{\beta}^*||_2$, we first use (16) to obtain

$$||\widehat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}^*||_1 \leqslant \frac{a(1-(2b)^t)}{1-(2b)} + (2b)^t||\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*||_1.$$

Then apply Theorem 6 to obtain that

$$\begin{aligned}||\widehat{\boldsymbol{\beta}}_{t+1}-\boldsymbol{\beta}^*||_2 \leqslant & \frac{a}{4\sqrt{s}}+\frac{(2b)}{4\sqrt{s}}||\widehat{\boldsymbol{\beta}}_t-\boldsymbol{\beta}^*||_1 \leqslant \frac{a}{4\sqrt{s}}+\frac{b}{4\sqrt{s}}\left(\frac{a(1-(2b)^t)}{1-(2b)}+(2b)^t||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1\right)\\ =&\frac{1}{4\sqrt{s}}\left(a+\frac{a((2b)-(2b)^{t+1})}{1-(2b)}\right)+\frac{(2b)^{t+1}||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1}{4\sqrt{s}}\\ =&\frac{a(1-(2b)^{t+1})}{4\sqrt{s}(1-(2b))}+\frac{(2b)^{t+1}||\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*||_1}{4\sqrt{s}},\end{aligned}$$

which concludes the proof. □

B.6. Proof of Lemma 10

Proof. By the definition of $\mathcal{L}_j(\boldsymbol{\beta})$, we have

$$\frac{1}{m}\sum_{j\in[m]}\nabla\mathcal{L}_j(\boldsymbol{\beta}^*)=\frac{1}{mn}\sum_{j\in[m]}\sum_{i\in[n]}\mathbf{x}_{ji}\left(y_{ji}-\frac{y_{ji}}{1+\exp(-y_{ji}\langle\boldsymbol{\beta},\mathbf{x}_{ji}\rangle)}\right).$$

It is easy to check that

$$\mathbb{E}\left[y_{ji}-\frac{y_{ji}}{1+\exp(-y_{ji}\langle\boldsymbol{\beta},\mathbf{x}_{ji}\rangle)}\right]=0,\quad\text{and}\quad\left|y_{ji}-\frac{y_{ji}}{1+\exp(--y_{ji}\langle\boldsymbol{\beta},\mathbf{x}_{ji}\rangle)}\right|\leqslant 1$$

and thus

$$\begin{aligned}&\mathbb{E}\left[\mathbf{x}_{ji}\left(y_{ji}-\frac{y_{ji}}{1+\exp(-y_{ji}\langle\boldsymbol{\beta},\mathbf{x}_{ji}\rangle)}\right)\right]=0,\\ &\left|\left|\mathbf{x}_{ji}\left(y_{ji}-\frac{y_{ji}}{1+\exp(--y_{ji}\langle\boldsymbol{\beta},\mathbf{x}_{ji}\rangle)}\right)\right|\right|_\infty\leqslant\max_{ji}\left(||\mathbf{x}_{ji}||_\infty\right).\end{aligned}$$

Appling Azuma-Hoeffding inequality (Hoeffding, 1963) and the union bound over $[p]$ leads to the desired bound. □

C. Full Experimental Results

We run the algorithms for both distributed regression and classification problems. The algorithms to be compared are:

- Local: the first machine just solves a related ℓ_1 regularized problem (lasso or ℓ_1 regularized logistic regression) with the optimal λ, and outputs the solution. Obviously this approach is communication free.
- Centralize: the master gathers all data from different machines together, and solves a centralized ℓ_1 regularized loss minimization problem with the optimal λ, and outputs the solution. This approach is communication expensive as all data needs to be communicated, but it usually gives us the best estimation and prediction performance.
- Prox GD: the distributed proximal gradient descent is ran on the ℓ_1 regularized objective, where we initialized the starting point with the first machine's solution.
- Avg-Debias: the method proposed in Lee et al. (2015b), with fine tuned regularization and hard thresholding parameters. This approach only requires one round of communication, where each machine sends a p-dimensional vector. However, Avg-Debias is computationally prohibitive because of the debiasing operation.
- EDSL: the proposed efficient distributed sparse learning approach, where the regularization level at each iteration is fine tuned on a held out test data set.

C.1. Simulations

The full experimental results plotted in Figure 3 and Figure 4, with various settings of (n, p, m, s), and condition numbers $1/\kappa$. We have the following observations:

Table 2. List of real-world datasets used in the experiments.

Name	#Instances	#Features	Task
a9a	48,842	123	Classification
connect-4	67,557	127	Regression
dna	2,000	181	Regression
mitface	6,977	362	Classification
mnist 1 vs 2	14,867	785	Classification
mnist	60,000	785	Regression
mushrooms	8,124	113	Classification
protein	17,766	358	Regression
spambase	4,601	57	Classification
usps	7,291	257	Regression
w8a	64,700	301	Classification
year	51,630	91	Regression

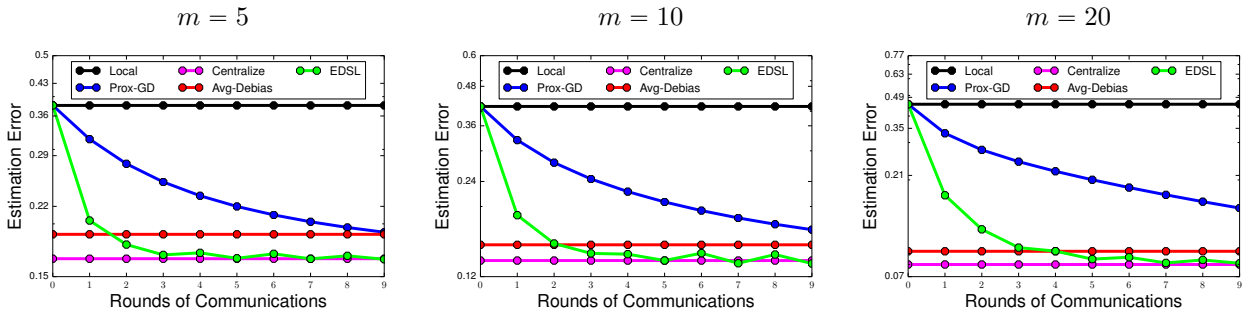
- The Avg-Debias approach obtained much better estimation error compared to Local after one round of communication and sometimes performed quite close to Centralize. However, in most cases, there is still a gap compared with Centralize, especially when the problem is not well-conditioned or the number of machines m is large.
- When the problem is well conditioned ($\Sigma_{ij} = 0.5^{|i-j|}$ case), Prox GD converges reasonably fast. However, it becomes very slow when the condition number becomes bad ($\Sigma_{ij} = 0.5^{|i-j|/5}$ case). We expect to observe a similar phenomenon for other first-order distributed optimization algorithms, such as accelerated proximal gradient or ADMM.
- As theory suggests, EDSL obtained a solution that is competitive with Avg-Debias after one round of communication. The estimation error decreases to match performance of Centralize within few rounds of communications; typically less than 5, even though the theory suggests EDSL will match the performance of centralize within $\mathcal{O}(\log m)$ rounds of communication.

C.2. Real-world Data Evaluation

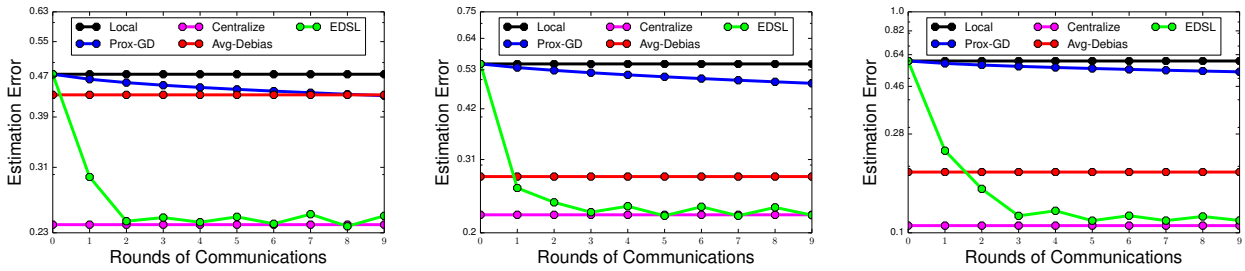
In real world data evaluation presented in Section 5.2, the datasets are publicly available from the LIBSVM website[7] and UCI Machine Learning Repository[8]. The statistics of these datasets are summarized in Table 2, where some of the multi-class classification datasets are adopted under the regression setting with squared losses. The results are plotted in Figure 5 where for some datasets the performance of Avg-Debias is significantly worse than others (mostly because the debiasing step fails), thus we omit these plots. The plots are shown in Figure 5 We have the following observations

- Since there is no well-specified model on these datasets, the curves behave quite differently on different data sets. However, a large gap between the local and centralized procedure is consistent as the later uses 10 times more data.
- Avg-Debias often fails on these real datasets and performs much worse than in simulations. The main reason might be that the assumptions, such as well-specified model or generalized coherence condition, fail, then Avg-Debias can totally fail and produce solution even much worse than the local.
- Prox GD approach still converges slowly in most of the cases.
- The proposed EDSL is quite robust on real world data sets, and can output a solution which is highly competitive with the centralized model within a few rounds of communications.
- There exits a slight "zig-zag" behavior for EDSL approach on some data sets. For example, on the mushrooms data set, the predictive performance of EDSL is not stable.

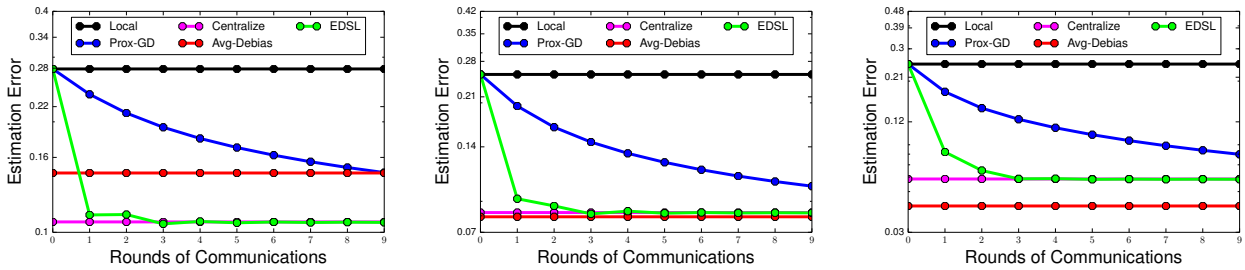
[7] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[8] http://archive.ics.uci.edu/ml/



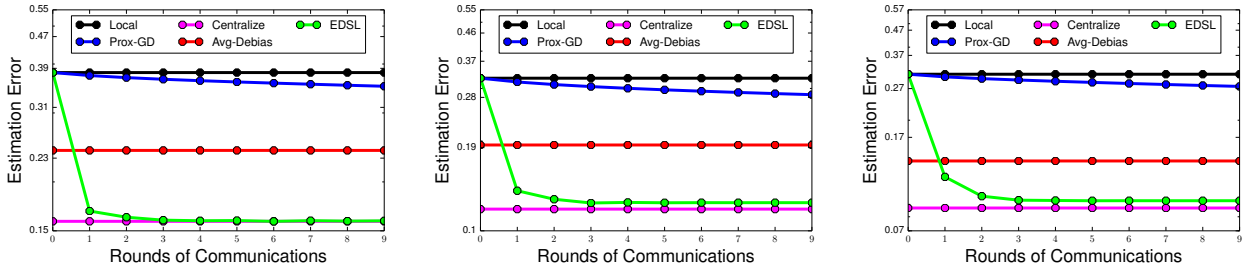
$n = 200, p = 1000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|}.$



$n = 200, p = 1000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|/5}.$

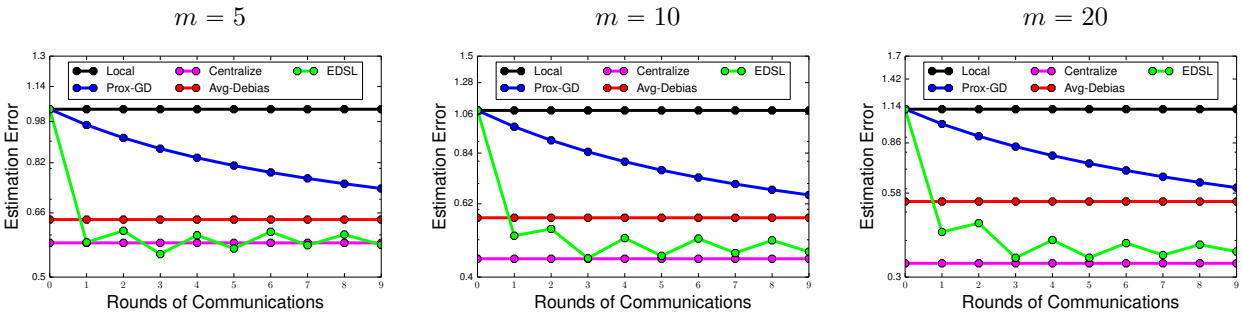


$n = 500, p = 3000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|}.$

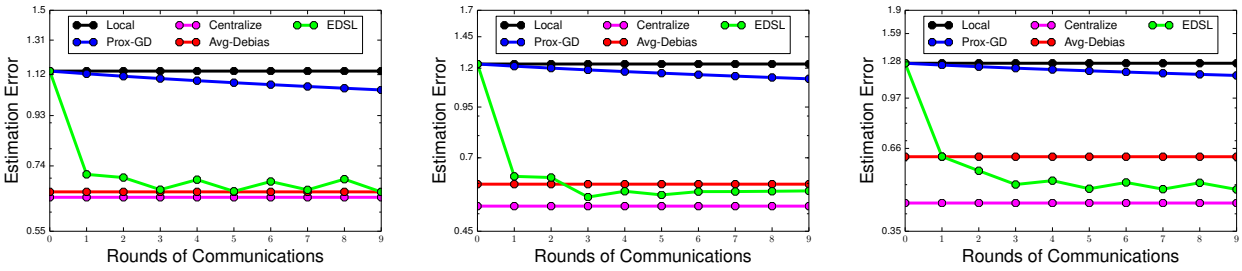


$n = 500, p = 3000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|/5}.$

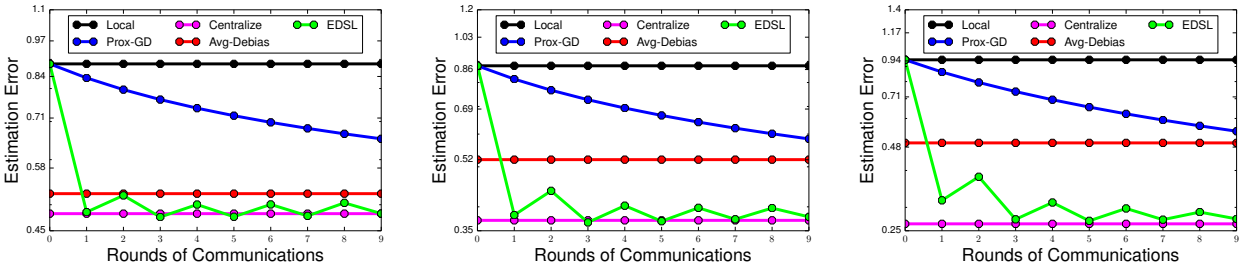
Figure 3. Comparison of various algorithms for distributed sparse regression, 1st and 3rd row: well-conditioned cases, 2nd and 4th row: ill-conditioned cases.



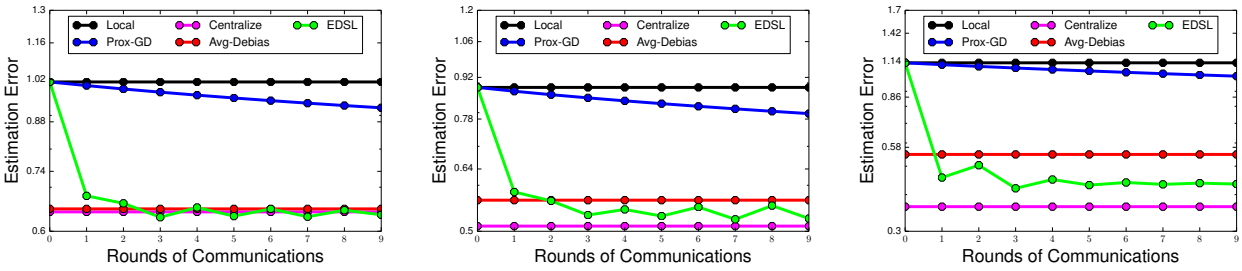
$n = 500, p = 1000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|}.$



$n = 500, p = 1000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|/5}.$



$n = 1000, p = 3000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|}.$



$n = 1000, p = 3000, s = 10, \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma}_{ij} = 0.5^{|i-j|/5}.$

Figure 4. Comparison of various algorithms for distributed sparse classification (logistic regression), 1st and 3rd row: well-conditioned cases, 2nd and 4th row: ill-conditioned cases.

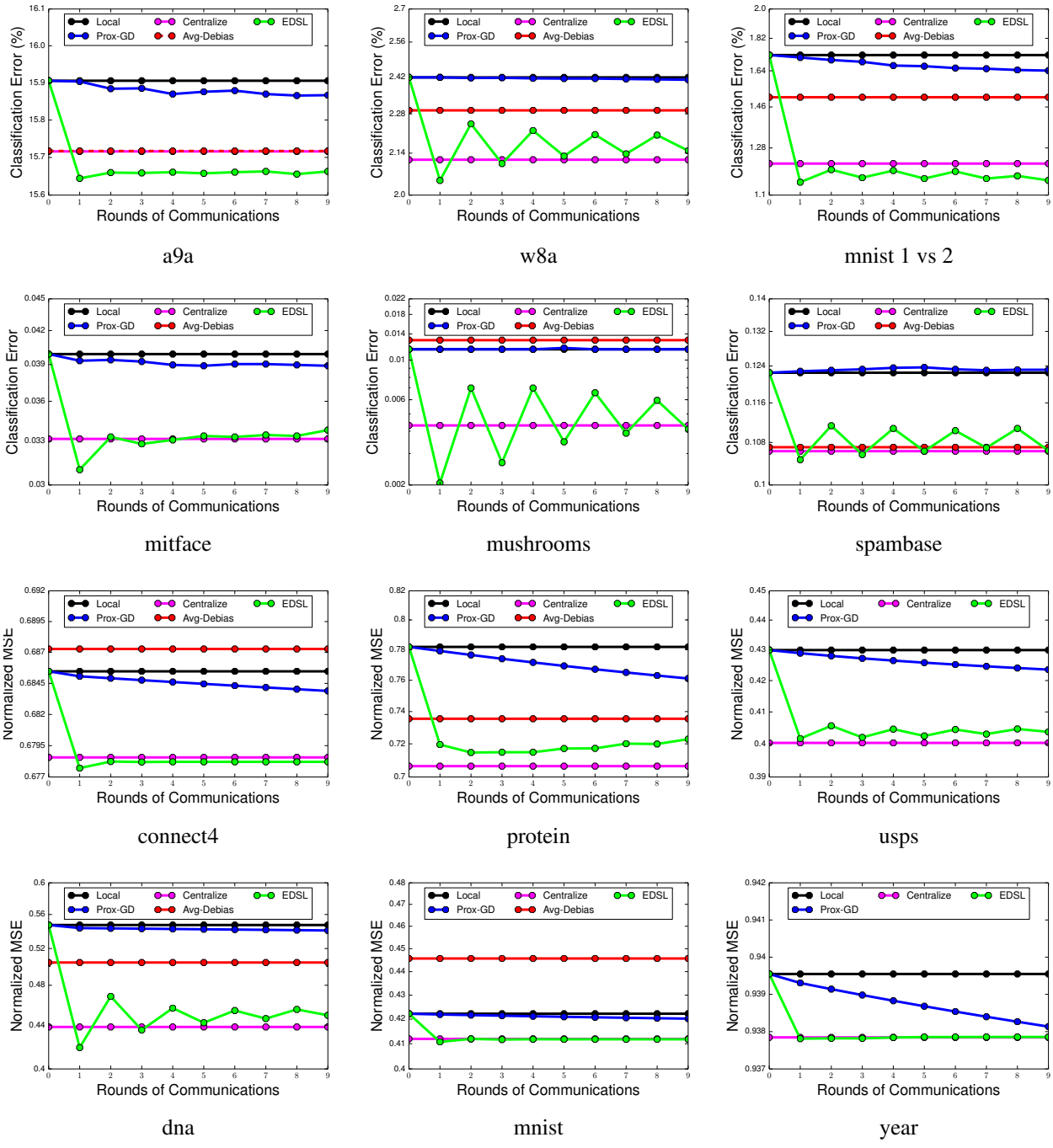


Figure 5. Comparison of various approaches for distributed sparse regression and classification on real world datasets. (Avg-Debias is omitted when it is significantly worse than others.)