Communication-Efficient Accurate Statistical Estimation

Jianqing Fan\textsuperscript{a}, Yongyi Guo\textsuperscript{a}, and Kaizheng Wang\textsuperscript{b}
\textsuperscript{a}Department of ORFE, Princeton University, Princeton, NJ; \textsuperscript{b}Department of IEOR, Columbia University, New York, NY

ABSTRACT

When the data are stored in a distributed manner, direct applications of traditional statistical inference procedures are often prohibitive due to communication costs and privacy concerns. This article develops and investigates two communication-efficient accurate statistical estimators (CEASE), implemented through iterative algorithms for distributed optimization. In each iteration, node machines carry out computation in parallel and communicate with the central processor, which then broadcasts aggregated information to node machines for new updates. The algorithms adapt to the similarity among loss functions on node machines, and converge rapidly when each node machine has large enough sample size. Moreover, they do not require good initialization and enjoy linear convergence guarantees under general conditions. The contraction rate of optimization errors is presented explicitly, with dependence on the local sample size unveiled. In addition, the improved statistical accuracy per iteration is derived. By regarding the proposed method as a multistep statistical estimator, we show that statistical efficiency can be achieved in finite steps in typical statistical applications. In addition, we give the conditions under which the one-step CEASE estimator is statistically efficient. Extensive numerical experiments on both synthetic and real data validate the theoretical results and demonstrate the superior performance of our algorithms.

1. Introduction

Statistical inference in modern era faces tremendous challenge on computation and storage. The exceedingly large size of data often makes it impossible to store all of them on a single machine. Moreover, many applications have individual agents (e.g., local governments, research labs, hospitals, smart phones) collecting data independently. Communication is prohibitively expensive due to the limited bandwidth, and direct data sharing raises concerns in privacy and loss of ownership. These constraints make it necessary to develop methodologies for distributed systems, solving statistical problems with divide-and-conquer procedures and communicating only certain summary statistics.

Distributed statistical inference has received considerable attention recently, covering a wide spectrum of topics including \(M\)-estimation (Zhang, Duchi, and Wainwright 2013; Chen and Xie 2014; Shamir, Srebro, and Zhang 2014; Rosenblatt and Nadler 2016; Wang, Kolar, et al. 2017; Lee et al. 2017b; Battey et al. 2018; Wang et al. 2018; Shi, Lu, and Song 2018; Banerjee, Durot, and Sen 2019), principal component analysis (Fan et al. 2019; Garber, Shamir, O., and Srebro 2017), nonparametric regression (Shang and Cheng 2017; Szabó and Van Zanten 2019; Han et al. 2018), quantile regression (Volgushev, Chao, and Cheng 2019; Chen, Liu, and Zhang 2021), bootstrap (Kleiner et al. 2014), confidence intervals (Jordon, Lee, and Yang 2019; Chen, Liu, and Zhang 2021), Bayesian methods (Wang and Dunson 2013; Jordan, Lee, and Yang 2019), etc. In the commonly used setting, the overall dataset is partitioned and stored on \(m\) node machines connected to a central processor. Most of the approaches only require one round of communication: the node machines work in parallel and send their results to the central processor, which then aggregates the information to get a final result. As typical examples, Zhang, Duchi, and Wainwright (2013) averaged the \(M\)-estimators on node machines; Battey et al. (2018) averaged debiased estimators; and Fan et al. (2019) averaged subspaces via eigen-decomposition. While these one-shot methods are communication-efficient, they only work with a small number of node machines (e.g., \(m = o(\sqrt{N})\), where \(N\) is the total sample size) and require large sample on each, as their theories heavily rely on asymptotic expansions of estimators. Such conditions are easily violated in practice.

Multi-round procedures come as a remedy, which alternate between local computations and global aggregations. It is possible to achieve optimal statistical precision after a few rounds of communications, under broader settings than those for one-shot procedures. Shamir, Srebro, and Zhang (2014) proposed a distributed approximate Newton (DANE) algorithm where, in each iteration, each node machine minimizes a modified loss function based on its own samples and the gradient information from all other machines obtained through communication. However, for non-quadratic losses, the analysis in Shamir, Srebro, and Zhang (2014) did not imply any advantage of DANE in terms of communication over distributed implementation of gradient descent. Other approximate Newton algorithms include Zhang and Xiao (2015), Wang

CONTACT Kaizheng Wang lw2934@columbia.edu Columbia University, Industrial Engineering and Operations Research, New York, NY 10027-6902.

© 2021 American Statistical Association

ARTICLE HISTORY

Received June 2019
Accepted August 2021

KEYWORDS

Communication efficiency; Distributed statistical estimation; Multi-round algorithms; Penalized likelihood

https://doi.org/10.1080/01621459.2021.1969238
et al. (2018), Chen, Liu, and Zhang (2021) and Crane and Roosta (2019). Jordan, Lee, and Yang (2019) developed a communication-efficient surrogate likelihood (CSL) framework for estimation and inference in random regular parametric models, penalized regression, and Bayesian statistics. A similar method also appears in Wang, Kolar, et al. (2017). These methods no longer have restrictions on the number of machines such as $m = o(\sqrt{N})$.

Due to the nature of Newton-type methods, existing theories for these algorithms heavily rely on good initialization or even self-concordance assumption on loss functions. They essentially focus on improving an initial estimator that is already consistent but not efficient, whose ideas coincide with the classical one-step estimator (Bickel 1975). Such initialization itself needs additional efforts and assumptions. Moreover, current results still require each machine to have sufficiently many samples so that loss functions on different machines are similar to each other. These all make the proposed methods unreliable in practice.

Aside from distributed statistical inference, there has also been a vast literature in distributed optimization. The ADMM (Boyd et al. 2011) is a celebrated example among the numerous algorithms that handle deterministic optimization problems with minimum structural assumption. Yet, the convergence can be quite slow and it cannot fully utilize the similarity among loss functions on node machines.

In this article, we develop and study two communication-efficient accurate statistical estimators (CEASE) based on multi-round algorithms for distributed statistical estimation. Our new algorithms extend the DANE algorithm (Shamir, Srebro, and Zhang 2014) to regularized empirical risk minimization. Moreover, we provide sharp convergence guarantees for general scenarios, even if the local loss functions are dissimilar and regularization is nonsmooth.

We assume that all the $m$ node machines have the same sample size $n$. Each has a regularized empirical risk function $f_k + g$ defined by the samples stored there, and the goal is to compute the minimizer of the overall regularized risk function $\frac{1}{m} \sum_{k=m}^{m} f_k + g$ to statistical precision. When $n$ is sufficiently large, their rates of convergence are better than or comparable to existing methods designed for this large-sample regime. For moderate or small $n$, they are still guaranteed to converge linearly even without good initialization, while other statistical methods fail. In addition, our algorithms take advantage of the similarity among $\{f_k\}_{k=1}^{m}$ and thus improve over general-purpose algorithms like ADMM. They interpolate between distributed algorithms for statistical estimation and general deterministic problems. Theoretical findings are verified by extensive numerical experiments. From a technical point of view, our algorithms use the proximal point algorithm (Rockafellar 1976) as the backbone and obtain inexact updates in a distributed manner. This turns out to be crucial for proving convergence under general conditions. Our techniques are potentially useful for studying other distributed algorithms.

The rest of this article is organized as follows. Section 2 introduces the algorithms. Section 3 presents deterministic convergence results. Section 4 provides guarantees in statistical problems. Section 5 shows numerical results on both synthetic and real data. Section 6 concludes the article and discusses possible future directions.

Here, we list the notations used throughout the article. We denote by $[n]$ the set $\{1, 2, \ldots, n\}$. We write $a_n = O(b_n)$ or $a_n \preceq b_n$ if there exists a constant $C > 0$ such that $a_n \leq Cb_n$ holds for sufficiently large $n$; and $a_n \asymp b_n$ if $a_n = O(b_n)$ and $b_n = O(a_n)$. Given $x, y \in \mathbb{R}^k$ and $r > 0$, we define $B(x, r) = \{z \in \mathbb{R}^k : ||z - x||_2 \leq r\}$ and $(x, y) = \sum_{j=1}^{k} x_j y_j$. For a convex function $h$ on $\mathbb{R}^k$, we let $\partial h(x)$ be its sub-differential set at $x \in \mathbb{R}^k$, and $\arg\min_{x \in \mathbb{R}^k} h(x)$ be the set of its minimizers if $\inf_{x \in \mathbb{R}^k} h(x) > -\infty$. We use $|| \cdot ||_2$ to denote the $\ell_2$ norm of a vector or operator norm of a matrix. For two sequences of random variables $\{X_n\}_{n=1}^{\infty}$ and $\{Y_n\}_{n=1}^{\infty}$, if $Y_n \geq 0$, we write $X_n = O_{\mathbb{P}}(Y_n)$ if for any $\varepsilon > 0$ there exists $C > 0$ such that $\mathbb{P}(|X_n| \geq CY_n) \leq \varepsilon$ for sufficiently large $n$. We use $||X||_{\psi_p} = \sup_{p \geq 1} \frac{1}{p} E[|X|^p]^{1/p}$ to refer to the sub-Gaussian norm of random variable $X$, and $||X||_{\psi_2} = \sup_{\|u\|_2 = 1} \|\langle u, X \rangle\|_{\psi_2}$ to denote the sub-Gaussian norm of random vector $X$.

2. The CEASE Algorithm

2.1. Problem Setup

Let $\mathcal{P}$ be an unknown probability distribution over some sample space $X$. For any parameter $\theta \in \mathbb{R}^p$, define its population risk $F(\theta) = \mathbb{E}_{X \sim \mathcal{P}} \ell(\theta; X)$ based on a loss function $\ell : \mathbb{R}^p \times X \rightarrow \mathbb{R}$. In parametric inference problems, $\ell$ is often chosen as the negative log-likelihood function of some parametric family. Under mild conditions, $F$ is well-defined and has a unique minimizer $\theta^*$. A ubiquitous problem in statistics and machine learning is to estimate $\theta^*$ given iid samples $\{X_i\}_{i=1}^{N}$ from $\mathcal{P}$, and the minimizer of the empirical risk $f(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; X_i)$ becomes a natural candidate. To achieve desirable precision in high-dimensional problems, it is often necessary to incorporate prior knowledge of $\theta^*$. A principled approach is the regularized empirical risk minimization

$$\min_{\theta \in \mathbb{R}^p} \left\{ f(\theta) + g(\theta) \right\},$$

(1)

denote by $g(\theta)$ is a deterministic penalty function. Common choices for $g(\theta)$ include the $\ell_2$ penalty $\lambda |\theta|^2_2$ (Hoeffl and Kennard 1970), the $\ell_1$ penalty $\lambda |\theta|_1$ (Tibshirani 1996), and a family of folded concave penalty functions $\|p_\lambda(\theta)\|_1$ such as SCAD (Fan and Li 2001) and MCP (Zhang 2010), where $\lambda > 0$ is a regularization parameter. Throughout the article, we assume that both $\ell$ and $g$ are convex in $\theta$, and $\ell$ is twice continuously differentiable in $\theta$. We allow $g$ to be non-smooth (e.g., the $\ell_1$ penalty).

Consider the distributed setting where the $N$ samples are stored on $m$ machines connected to a central processor. Denote by $\mathcal{I}_k$ the index set of samples on the $k$th machine and $f_k(\theta) = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \ell(\theta; X_i)$. For simplicity, we assume that $|\mathcal{I}_k|_{k=1}^{m}$ are disjoint, $N$ is a multiple of $m$, and $|\mathcal{I}_k| = n = N/m$ for all $k \in [m]$. Then (1) can be rewritten as

$$\min_{\theta \in \mathbb{R}^p} \left\{ f(\theta) + g(\theta) \right\}, \quad f(\theta) = \frac{1}{m} \sum_{k=1}^{m} f_k(\theta).$$

(2)

Each machine $k$ only has access to its local data and hence local loss function $f_k$ and the penalty $g$. We aim to solve Equation (2) in a distributed manner with both statistical efficiency and communication-efficiency.
2.2. Adaptive Gradient Enhancements and Distributed Algorithms in Large-Sample Regimes

In the large-sample regime, we drop the regularization term for now and consider the empirical risk minimization problem $\min_{\theta \in B} f(\theta)$ for estimating $\theta^* = \arg\min_{\theta \in B} F(\theta)$. In some problems, direct minimization of $f$ is costly, while it is easy to obtain some rough estimate $\hat{\theta}$ that is close to $\theta^*$ but not as accurate as the global minimizer $\hat{\theta} = \arg\min_{\theta \in B} f(\theta)$. Bickel (1975) proposed the one-step estimator based on the local quadratic approximation and shows that it is as efficient as the global minimizer $\hat{\theta}$ if the initial estimator $\hat{\theta}$ is accurate enough. Iterating this further results in multiple-step estimators that improve the optimization error in detail. Given samples $(\theta^*, y_i)$, one simply adds $\hat{\theta}$ to $f(\theta)$, which equals to

$$f(\theta) = f(\theta) + R(\theta), \quad \text{where} \quad R(\theta) = f(\theta) - f^{(1)}(\theta).$$

Since the linear function $f^{(1)}(\theta)$ can easily be communicated to each node machine whereas $R(\theta)$ can not, the latter is naturally replaced by its subsampled version at node $k$:

$$R_k(\theta) = f_k(\theta) - [f_k(\theta) + \langle \nabla f_k(\theta), \theta - \hat{\theta} \rangle],$$

where $f_k(\theta)$ is the loss function based on the data at node $k$. With this replacement, the target of optimization at node $k$ becomes $f^{(1)}(\theta) + R_k(\theta)$, which equals to

$$f_k(\theta) - \langle \nabla f_k(\hat{\theta}) - \nabla f(\hat{\theta}), \theta \rangle,$$

up to an additive constant. This function will be called gradient-enhanced loss function, in which the gradient at point $\hat{\theta}$ based on the local data is replaced by the global one. This function has one very nice fixed point at the global minimum $\hat{\theta}$: the minimizer of the adaptive gradient-enhanced function at $\theta = \hat{\theta}$ is still $\hat{\theta}$. This can easily be seen by verifying that the gradient at the point $\hat{\theta}$ is zero.

The idea of using such an adaptive gradient-enhanced function has been proposed in Shamir, Srebro, and Zhang (2014) and Jordan, Lee, and Yang (2019), though the motivations are different. Jordan, Lee, and Yang (2019) develop a Communication-efficient Surrogate Likelihood (CSL) method using the gradient-enhanced loss function $f_k(\theta) - \langle \nabla f_k(\theta) - \nabla f(\theta), \theta \rangle$ on the first machine, uses the minimizer on that machine as a new estimate, and iterates these steps until convergence. In the presence of a regularizer $g$ in Equation (1), one simply adds $g$ to the gradient-enhanced loss; see the Algorithm 1.

Note that in Algorithm 1, only the first machine solves optimization problems and others just evaluate gradients. These machines are idling while the first one is working hard. To fully utilize the computing power of machines and accelerate convergence, all the machines can optimize their corresponding gradient-enhanced loss functions in parallel and the central processor then aggregates the results. This is motivated by the distributed approximate Newton (DANE) algorithm (Shamir, Srebro, and Zhang 2014). Algorithm 2 describes the procedure in detail. Intuitively, the averaging step requires little computation but helps reduce the variance of estimators on node machines and enhance the accuracy.

We now illustrate Algorithm 2 in the context of linear regression. Given samples $\{(x_i, y_i)\}_{i \in [N]}$, the $k$th machine defines a quadratic loss function

$$f_k(\theta) = \frac{1}{2n} \sum_{i \in I_k} (y_i - x_i^\top \theta)^2 = \frac{1}{2} \theta^\top \Sigma_k \theta - \hat{w}_k^\top \theta + \frac{1}{2n} \sum_{i \in I_k} y_i^2,$$

Here, $\Sigma_k = \frac{1}{n} \sum_{i \in I_k} x_i x_i^\top$ and $\hat{w}_k = \frac{1}{n} \sum_{i \in I_k} x_i y_i$. The overall loss function is $f(\theta) = \frac{1}{m} \sum_{k=1}^m f_k(\theta) = \frac{1}{2m} \theta^\top \Sigma \theta - \hat{w}^\top \theta$, where $\Sigma = \frac{1}{m} \sum_{k=1}^m \Sigma_k$. Then the update of Algorithm 2 in one iteration is

$$\theta_{t+1,k} = (I - \Sigma_k^{-1} \hat{\Sigma}_k) \theta_t + \Sigma_k^{-1} \hat{w}_k,$$

$$\theta_{t+1} = (I - \frac{1}{m} \sum_{k=1}^m \Sigma_k^{-1} \hat{\Sigma}_k) \theta_t + \frac{1}{m} \sum_{k=1}^m \Sigma_k^{-1} \hat{w}_k. \quad (3)$$

Intuitively, this is a form of contraction toward the global minimizer $\theta^*$. As for the logistic regression, we can also write out the corresponding enhanced losses and minimize them using Newton’s method. Due to space limitations, we refer to Appendix C for details.
2.3. The CEASE Algorithm in General Regimes

Algorithms 1 and 2 are built upon large-sample regimes, with sufficiently strong convexity of \( f_k + g \) and small discrepancy between them. This requires the local sample size \( n \) to be large enough, which may not be the case in practice. Even worse, the required local sample size depends on structural parameters, making such a condition unverifiable. In fact, our numerical experiments confirm the instability of Algorithms 1 and 2 even for moderate \( n \). A naive method of remedy is to add strict convex quadratic regularization \( q(\theta) \). While this remedy can make the algorithm converge rapidly, the nonadaptive nature of \( q(\theta) \) will lead to a wrong target. Instead of using a fixed \( q \), we will adjust it according to current solutions. The idea stems from the proximal point algorithm (Rockafellar 1976).

**Definition 2.1.** For any convex function \( h : \mathbb{R}^p \to \mathbb{R} \), define the proximal mapping \( \text{prox}_h : \mathbb{R}^p \to \mathbb{R}^p, x \mapsto \argmin_{y \in \mathbb{R}^p} \{ h(y) + ||y - x||_2^2/2 \}. \)

For a given \( \alpha > 0 \), the proximal point algorithm for minimizing \( h \) iteratively computes

\[
x_{t+1} = \text{prox}_{\alpha^{-1}(f + g)}(x_t) = \argmin_{x \in \mathbb{R}^p} \{ h(x) + (\alpha/2)||x - x_t||_2^2 \},
\]

\( \forall t \geq 0, \)

starting from some initial value \( x_0 \). It is a strongly convex optimization, shrinking toward the current value \( x_t \). Under mild conditions, \( \{x_t\}_{t=0}^\infty \) converges linearly to some \( \tilde{x} \in \arg\min_{x \in \mathbb{R}^p} h(x) \) (Rockafellar 1976).

Now we take \( h = f + g \) and write the proximal point iteration for our problem (2)

\[
\theta_{t+1} = \text{prox}_{\alpha^{-1}(f + g)}(\theta_t) = \argmin_{\theta \in \mathbb{R}^p} \{ f(\theta) + g(\theta) + \alpha/2||\theta - \theta_t||_2^2 \}.
\]

Each iteration (5) is a distributed optimization problem, whose function object is not available to node machines. But it can be solved by Algorithms 1 and 2. Specifically, suppose we have already obtained \( \theta_t \) and aim for \( \theta_{t+1} \) in Equation (5). Letting \( \hat{g}(\theta) = g(\theta) + (\alpha/2)||\theta - \theta_t||_2^2 \), Algorithm 2 starting from \( \theta_0 = \theta_t \) produces iterations over \( s = 0, 1, \ldots \)

\[
\tilde{\theta}_{ts} = \argmin_{\theta \in \mathbb{R}^p} \{ f_k(\theta) + \hat{g}(\theta) + \langle \nabla f_k(\tilde{\theta}_s) - \nabla f_k(\theta), \theta \rangle \},
\]

\( k \in [m], \)

\[
\tilde{\theta}_{t+1} = \frac{1}{m} \sum_{k=1}^{m} \tilde{\theta}_{ts}.
\]

When \( \alpha + \rho_0 > \delta \), \( \{\tilde{\theta}_t\}_{t=0}^\infty \) converges Q-linearly\(^1\) to \( \theta_{t+1} \). On the other hand, there is no need to solve (5) exactly, as \( \text{prox}_{\alpha^{-1}(f + g)}(\theta_t) \) is merely an intermediate quantity for computing \( \theta \). We therefore only run one iteration of Algorithm 2 and use the resulting approximate solution as \( \theta_{t+1} \). This considerably simplifies the algorithm, reducing double loops to a single loop, and enhances statistical interpretability of the method as a multi-step estimator. However, it makes technical analysis more challenging. Similarly, we may also use one step of Algorithm 1 to compute the inexact proximal update.

The above discussions lead us to propose two CEASE in Algorithms 3 and 4, which use the proximal point algorithm as the backbone and obtain inexact updates in a distributed manner. They are regularized versions of Algorithms 1 and 2, with an additional proximal term in the objective functions. That term reduces relative differences of the local loss functions on individual machines, and is crucial for convergence when \( \{f_k\}_{k=1}^m \) are not similar enough. In Appendix A, we introduce a variant of Algorithm 4 which also stabilizes Algorithm 2.

In each iteration, Algorithm 3 has one round of communication and one optimization problem to solve. Although Algorithm 4 has two rounds of communication per iteration, only one round involves parallel optimization and the other is simply averaging. We will compare their theoretical guarantees as well as practical performances in the sequel.

---

\(^1\)According to Nocedal and Wright (2006), a sequence \( \{x_n\}_{n=1}^\infty \) in \( \mathbb{R}^p \) is said to converge Q-linearly to \( x^* \) in \( \mathbb{R}^p \) if there exists \( r \in (0, 1) \) such that \( ||x_{n+1} - x^*||_2 \leq r||x_n - x^*||_2 \) for \( n \) sufficiently large.
Algorithm 4 is an extension of the DANE algorithm in Shamir, Srebro, and Zhang (2014) to regularized empirical risk minimization. While DANE is originally motivated by mirror descent, we view it as a distributed implementation of the proximal point algorithm. The new perspective helps us obtain stronger convergence guarantees. Ideas from the proximal point algorithm have appeared in the literature of distributed stochastic optimization for different purposes such as accelerating first-order algorithms (Lee et al. 2017a) and regularizing sizes of updates (Wang, Wang, and Srebro 2017).

3. Deterministic Analysis

We first present in Section 3.1 the deterministic (almost sure) results for Algorithms 3 and 4 based on high-level structural assumptions. As special cases of these algorithms with $\alpha = 0$, Algorithms 1 and 2 will be analyzed in Section 3.2.

3.1. Deterministic Analysis of the CEASE Algorithm

Definition 3.1. Let $h : \mathbb{R}^p \rightarrow \mathbb{R}$ be a convex function, $\Omega \subseteq \mathbb{R}^p$ be a convex set, and $\rho \geq 0$. $h$ is $\rho$-strongly convex in $\Omega$ if $h(y) \geq h(x) + \langle g, y - x \rangle + \frac{\rho}{2}||y - x||^2_2$, $\forall x, y \in \Omega$ and $g \in \partial h(x)$.

Assumption 3.1 (Strong convexity). $f = \sum g$ has a unique minimizer $\hat{\theta} \in \mathbb{R}^p$, and is $\rho$-strongly convex in $B(\theta, R)$ for some $R > 0$ and $\rho > 0$.

Assumption 3.2 (Homogeneity). $||\nabla^2 f_k(\theta) - \nabla^2 f(\theta)||_2 \leq \delta, \forall k \in [m], \theta \in B(\hat{\theta}, R)$.

We will refer to $\delta$ as a homogeneity parameter. Based on both assumptions, we define

$$\rho_0 = \sup \left\{ c \in [0, \rho) : \{(f_k + g)^m \}_{k=1}^m \text{ are } c\text{-strongly convex in } B(\hat{\theta}, R) \right\}. \quad (6)$$

A simple but useful fact is $\max(\rho - \delta, 0) \leq \rho_0 \leq \rho$. In the most interesting problems, the population risk $F$ is smooth and strongly convex on any compact set. When $\{X_i\}_{i=1}^N$ are iid and the total sample size $N$ is large, the empirical risk $f$ concentrates around $F$ and inherits nice properties from the latter, making Assumption 3.1 hold easily.

On the other hand, since the empirical risk functions $\{f_k\}_{k=1}^m$ are iid stochastic approximations of the population risk $F$, they should not be too far away from their average $f$ provided that $n$ is not too small. Assumption 3.2 is a natural way of characterizing this similarity. It is a generalization of the concept "$\delta$-related functions" for quadratic losses in Arjevani and Shamir (2015). With high probability, it holds with reasonably small $\delta$ and large $R$ under general conditions. Large $n$ implies small homogeneity parameter $\delta$ and thus similar $\{f_k\}_{k=1}^m$. Assumption 3.2 always holds with $\delta = \max_{k \in [m]} \left\{ \sup_{\theta \in B(\hat{\theta}, R)} ||\nabla^2 f_k(\theta) - \nabla^2 f_k(\hat{\theta})||_2 + \sup_{\theta \in B(\hat{\theta}, R)} \sup_{\theta \in B(\hat{\theta}, R)} ||\nabla^2 f(\theta) - \nabla^2 f(\hat{\theta})||_2 \right\}$.

The following additional assumption on smoothness of the Hessian matrix of $f + g$ is not necessary for contraction, but it helps us obtain a much stronger result on the contraction rate of Algorithm 2, justifying the power of the simple averaging step.

Assumption 3.3 (Smoothness of Hessian). $g \in C^2(\mathbb{R}^p)$, and there exists $M \geq 0$ such that $||\nabla^2 f(\theta) + \nabla^2 g(\theta')||_2 - ||\nabla^2 f(\theta') + \nabla^2 g(\theta)||_2 \leq M||\theta - \theta'||_2, \forall \theta, \theta' \in B(\theta, R)$.

Theorem 3.1 gives contraction guarantees for Algorithms 3 and 4. It is deterministic and nonasymptotic by nature.

Theorem 3.1. Let Assumptions 3.1 and 3.2 hold. Consider the multistep estimators $\{\hat{\theta}_t\}_{t=1}^T$ generated by Algorithm 3 or 4. Suppose that $\theta_0 \in B(\theta, R)$ and $(\delta/(\rho_0 + \alpha))^2 < \rho/(\rho + 2\alpha)$.

(a) For both Algorithms 3 and 4, we have

$$||\hat{\theta}_{t+1} - \hat{\theta}||_2 \leq ||\hat{\theta}_t - \hat{\theta}||_2 \cdot \frac{\delta}{\rho_0 + \alpha} \cdot \sqrt{\frac{\rho^2 + 2\alpha \rho + \alpha}{\rho + \alpha}}, \quad 0 \leq t \leq T - 1; \quad (7)$$

(b) If Assumption 3.3 also holds, then for Algorithm 4, we have

$$||\hat{\theta}_{t+1} - \hat{\theta}||_2 \leq ||\hat{\theta}_t - \hat{\theta}||_2 \cdot \frac{\gamma_t \sqrt{\rho^2 + 2\alpha \rho + \alpha}}{\rho + \alpha}, \quad 0 \leq t \leq T - 1,$$

where we define $\gamma_t = \frac{\delta}{\rho_0 + \alpha} \cdot \min\{1, \frac{\delta}{\rho_0 + \alpha} (1 + \frac{M}{\rho_0 + \alpha}) \} \cdot \frac{\rho_0 + \alpha}{\rho_0 + \alpha}$.

(c) Both multiplicative factors in Equations (7) and (8) are strictly less than 1.

In the contraction factor in Equation (7), the two summands $\frac{\delta \sqrt{\rho^2 + 2\alpha \rho}}{\tau \rho + \alpha}$ and $\frac{\alpha}{\rho + \alpha}$ come from bounding the inexact proximal update $||\hat{\theta}_{t+1} - \text{prox}_{\alpha \gamma_t}(f + g)(\hat{\theta}_t)||_2$ and the residual $||\text{prox}_{\alpha \gamma_t}(f + g)(\hat{\theta}_t) - \hat{\theta}||_2$, respectively. Similar results hold for (8). The condition $(\delta/(\rho_0 + \alpha))^2 < \rho/(\rho + 2\alpha)$ ensures that both contraction factors are less than 1. Note that (8) requires Assumption 3.3, which forces $g$ to be smooth.

Theorem 3.1 shows the Q-linear convergence of the sequence $\{\hat{\theta}_t\}_{t=1}^\infty$ generated by both Algorithms 3 and 4 under quite general settings. The contraction rate depends explicitly on the structural parameters and the choice of $\alpha$. The local loss functions $\{f_k\}_{k=1}^m$ just need to be convex and smooth, and the convex penalty $g$ is allowed to be non-smooth, for example, the $\ell_1$ norm. On the contrary, most algorithms for distributed statistical estimation are only designed for smooth problems, and many of them are only rigorously studied when the loss functions are quadratic or self-concordant (Shamir, Srebro, and Zhang 2014; Zhang and Xiao 2015; Wang, Wang, and Srebro 2017). This is another important aspect of our contributions.

We immediately see from Theorem 3.1 that Algorithms 3 and 4 converge linearly as long as $[\delta/(\rho_0 + \alpha))^2 < \rho/(\rho + 2\alpha)$, which is guaranteed to hold by choosing sufficiently large $\alpha$. By contrast, however, we’ll show in Section 3.2 that Algorithms 1 and 2 (corresponding to $\alpha = 0$) hinges on the homogeneity assumption $\rho_0 > \delta$ in Theorem 3.2, that is, the functions $\{f_k\}_{k=1}^m$ must be similar enough. In the statistical setting, this requires the local sample size $n$ to be large. Therefore, proper regularization provides a safety net for the algorithms under general regimes with potentially insufficient local sample size. Corollary 3.1 below gives a guideline for choosing $\alpha$ to make Algorithms 3 and 4 converge in general.
Corollary 3.1. Let Assumptions 3.1 and 3.2 hold, \( \theta_0 \in B(\hat{\theta}, R/2) \), and \( \{\theta_t\}_{t=0}^\infty \) be the iterates of Algorithm 3 or 4. With any \( \alpha \geq 48^2 / \rho \), both algorithms converge with contraction factors in Equations (7) and (8) bounded by \( (1 - \frac{\rho}{10(\alpha + \rho)}) \).

On the other hand, consider the case where the local loss functions have small relative difference \( \delta/\rho \). In this case, Theorem 3.2 states that the contraction factors for unregularized versions \( (\alpha = 0) \) of Algorithms 3 and 4 are in the same order of \( \delta/\rho \) and \( (\delta/\rho)^2 \), respectively, which are smaller than the contracting factors with \( \alpha > 0 \). The following corollary characterizes the upper bound for \( \alpha \) so that the contraction factors remain at these small orders.

Corollary 3.2. Let Assumptions 3.1 and 3.2 hold, \( \theta_0 \in B(\hat{\theta}, R/2) \), and suppose \( \alpha \leq C\delta^2 / \rho \) for some constant \( C \). There exist constants \( C_1 \) and \( C_2 \) such that the followings hold when \( \delta/\rho \) is sufficiently small:

(a) Algorithms 3 and 4 have the contraction property
\[
||\theta_{t+1} - \hat{\theta}||_2 \leq C_1(\delta/\rho)||\theta_{t} - \hat{\theta}||_2, \quad 0 \leq t \leq T - 1;
\]
(b) If Assumption 3.3 also holds and \( ||\theta_{t} - \hat{\theta}||_2 \leq \rho/M \), then for Algorithm 4
\[
||\theta_{t+1} - \hat{\theta}||_2 \leq C_2(\delta/\rho)^2||\theta_{t} - \hat{\theta}||_2.
\]

Note that the second result above only holds given Assumption 3.3, which requires a smooth regularization \( g \). Corollary 3.2 reveals that by choosing \( \alpha \approx \delta^2 / \rho \), the contraction factors are essentially the same as those of the unregularized \( (\alpha = 0) \) algorithms when \( \delta/\rho \) is small. By combining Corollaries 3.1 and 3.2, we use \( \alpha \approx \delta^2 / \rho \) as a default choice for Algorithms 3 and 4 to become both fast and robust. They are reliable in general cases (Corollary 3.1) and efficient in nice cases (Corollary 3.2 and Theorem 3.1 with \( \alpha = 0 \)).

Algorithms 3 and 4 attain communication efficiency by utilizing similarity among local loss functions: The contraction factors in Corollary 3.2 go to zero if \( \delta/\rho \) does. In fact, both algorithms achieve \( \epsilon \)-accuracy within \( O(\max\{1, (\delta/\rho)^2\}) \) \( \log(||\theta_0 - \hat{\theta}||_2) \) rounds of communication. In contrast, the distributed accelerated gradient descent requires \( O(\sqrt{k_0}) \) \( \log(||\theta_0 - \hat{\theta}||_2) \) rounds of communication to achieve \( \epsilon \)-accuracy (Shamir, Srebro, and Zhang 2014), with \( k_0 \) being the condition number of \((f + g)\), which does not take advantage of sample size \( n \). As long as \( \delta/\rho \ll k_0^{1/4} \), Algorithms 3 and 4 communicate less than the distributed accelerated gradient descent. And again, our general results for Algorithms 3 and 4 also apply to the case with nonsmooth penalty functions while those for distributed accelerated gradient descent do not.

Moreover, if \( (f + g) \) is smooth and \( \theta_t \) is reasonably close to \( \hat{\theta} \), Corollary 3.2 shows that each iteration of Algorithm 4 is roughly equivalent to two iterations of Algorithm 3, although the former only has one round of optimization. The averaging step in Algorithm 4 reduces the error as much as the optimization step, while taking much less time. In this case, Algorithm 4 is preferable, and our numerical experiments also confirm this.

For unregularized empirical risk minimization, that is, \( g = 0 \) in Equation (2), Algorithm 4 reduces to an extension or a useful case of the DANE algorithm (Shamir, Srebro, and Zhang 2014). In this case, Theorem 3.1 and its corollaries refine the analysis of DANE (Shamir, Srebro, and Zhang 2014) in several aspects. On the one hand, our analysis handle both smooth and nonsmooth problems, while in Shamir, Srebro, and Zhang (2014), the theoretical analysis beyond quadratic loss requires extremal choice of tuning parameters and does not show any advantage over distributed implementation of the gradient descent. On the other hand, as mentioned in Section 2, we derive Algorithm 4 from the proximal point algorithm with a new prospective, which leads to sharp convergence analysis along with suggestions on choosing the tuning parameter \( \alpha \). As a by-product, we close a gap in the theory of DANE in non-quadratic settings. Our analysis techniques are potentially useful for other distributed optimization algorithms, especially when the loss is not quadratic.

3.2. Deterministic Analysis in Large-Sample Regimes

In this section, we restrict ourselves to large-sample regimes where the local sample size \( n \) is sufficiently large such that \( \rho_0 > \delta \geq 0 \), while in Shamir, Srebro, and Zhang (2014), the theory of DANE in non-quadratic settings. Our analysis techniques are potentially useful for other distributed optimization algorithms, especially when the loss is not quadratic.

Theorem 3.2. Let Assumptions 3.1 and 3.2 hold, and \( \rho_0 > \delta \geq 0 \). Consider the iterates \( \{\theta_t\}_{t=0}^\infty \) produced by Algorithm 1 or 2, with \( \theta_0 \in B(\hat{\theta}, R) \). Then
\[
||\theta_{t+1} - \hat{\theta}||_2 \leq (\delta/\rho_0)||\theta_{t} - \hat{\theta}||_2, \quad \forall t \geq 0.
\]

In addition, if Assumption 3.3 also holds, then for Algorithm 2, we have
\[
||\theta_{t+1} - \hat{\theta}||_2 \leq \frac{\delta}{\rho_0}||\theta_{t} - \hat{\theta}||_2 \cdot \min\left\{1, \frac{\delta}{\rho_0} \left(1 + \frac{M}{\rho_0} ||\theta_{t} - \hat{\theta}||_2, \quad \forall t \geq 0.
\right.
\]

Note that the last inequality requires Assumption 3.3 and thus a smooth regularization \( g \). The first part of Theorem 3.2 is a refinement of the analysis in Jordan, Lee, and Yang (2019), since we allow the initial estimator to be inaccurate and we have more explicit rates of contraction of optimization errors. This will be further demonstrated in Section 4.2.

The second part points out benefits of the averaging step, which is a novel result. Similar to the results on Algorithm 4, Theorem 3.2 shows that when \( \theta_t \) is close to \( \hat{\theta} \), with an additional standard assumption on Hessian smoothness, the averaging step alone in Algorithm 2 is almost as powerful as an optimization step in terms of contraction: The contracting constant will eventually be \( \frac{\delta}{\rho_0} \). With negligible computational cost, averaging significantly improves upon individual solutions \( \{\theta_{t,k}\}_{k=1}^\infty \) by doubling the speed of convergence.

4. Statistical Analysis

We further analyze the statistical properties of the above algorithms under a generalized linear model. Essentially, both the CSL methods and the CEASE algorithm are \( T \)-step estimators, starting from the initial estimator \( \theta_0 \). The question here is the effect of iterations in the multiple step estimators and the
role of the initial estimator. We start with statistical analysis of Algorithms 3 and 4 in Section 4.1, and then study Algorithms 1 and 2 in Section 4.2. In Section 4.3, we provide practical guidance when implementing the CEASE algorithm based on these analysis.

4.1. Multistep Estimators in General Regimes

The deterministic analysis in Section 3.2 applies to a wide range of statistical models. Here we consider the generalized linear model with canonical link, where our samples are iid pairs \((x_i, y_i)\) of covariates and responses and the conditional density of \(y_i\) given \(x_i\) is given by

\[
h(y_i; x_i, \theta^*) = c(x_i, y_i) \exp \left( y_i (x_i^\top \theta^*) - b(x_i^\top \theta^*) \right).
\]

For simplicity, we let the dispersion parameter to be 1 as we do not consider the issue of over-dispersion; \(b(. )\) is some known convex function, and \(c\) is a known function such that \(h\) is a valid probability density function. The negative log-likelihood of the whole data is an affine transformation of \(f(\theta) = \frac{1}{m} \sum_{k=1}^{m} f_k(\theta)\) with

\[
f_k(\theta) = \frac{1}{n} \sum_{i \in I_k} \left[ b(x_i^\top \theta) - y_i (x_i^\top \theta) \right].
\]

It is easy to verify that

\[
\nabla f_k(\theta) = \frac{1}{n} \sum_{i \in I_k} \left[ b'(x_i^\top \theta) x_i - y_i x_i \right]
\]

\[
\nabla^2 f_k(\theta) = \frac{1}{n} \sum_{i \in I_k} b''(x_i^\top \theta) x_i x_i^\top.
\]

Assume that \(x_i = (1, u_i^\top)^\top \in \mathbb{R}^p\), where \(\{u_i\}_{i=1}^N \subseteq \mathbb{R}^{p-1}\) are iid random covariate vectors with zero mean and covariance matrix \(\Sigma\). Suppose there exist universal positive constants \(A_1, A_2\) and \(A_3\) such that \(A_1 \leq \|\Sigma\|_2 \leq A_2\). Let \(\Sigma^* = \mathbb{E}(x_i x_i^\top) = \begin{pmatrix} 1 & 0 \\ 0 & \Sigma \end{pmatrix}\), \(g\) be a deterministic penalty function, and \(F(\theta) = \mathbb{E}f(\theta)\) be the population risk function. Below we impose some standard regularity assumptions.

**Assumption 4.1.**  
- \(\{S^{-1/2}u_i\}_{i=1}^N\) are iid Gaussian random vectors.
- For all \(x \in \mathbb{R}\), \(|b''(x)|\) and \(|b'''(x)|\) are bounded by some constant.
- \(\|\theta^*\|_2\) is bounded by some constant.

As in Assumptions 3.1 and 3.3, the following general assumptions are also needed for our analysis. Here, \(R\) is some positive quantity that satisfies \(R < A_4 p^{A_5}\) for some universal constants \(A_4\) and \(A_5\).

**Assumption 4.2.**  
There exists a universal constant \(\rho > 0\) such that \((F + g)\) is \(\rho\)-strongly convex in \(B(\theta^*, 2R)\).

The following smoothness assumption is only needed for a part of our theory; it is used to show that the averaging step in Algorithm 4 can significantly enhance the accuracy.

**Assumption 4.3.**  
\(g \in C^2(\mathbb{R}^p)\), and there exists a universal constant \(M \geq 0\) such that \(\|\nabla^2 F(\theta') + \nabla^2 g(\theta')\|_2 \leq M\|\theta' - \theta^*\|_2\), \(\forall \theta', \theta^* \in B(\theta^*, 2R)\).

Under the model assumptions above, we can explicitly determine the rate of \(\delta\) in Assumption 3.2. In particular, we will show in Lemma E.5 in Appendix E that

\[
\max_{k \in [m]} \max_{\theta \in B(\theta, \delta)} \|\nabla^2 f_k(\theta) - \nabla^2 f(\theta)\|_2 \\
= O_P \left( \frac{\|\Sigma\|_2}{n} \sqrt{\frac{\rho (\log p + \log N)}{n}} \right),
\]

provided that \(n \geq cp\) for an arbitrary positive constant \(c\). Therefore, with high probability, \(\delta \approx \|\Sigma\|_2 \sqrt{\rho (\log p + \log N)/n}\). Omitting the logarithmic terms, we see that the contraction factor is approximately \(\kappa \sqrt{p}/n\), where \(\kappa \triangleq \|\Sigma\|_2/\rho\) can be viewed as a condition number. This rate is more explicit on \(p\) and \(\kappa\) than that in Jordan, Lee, and Yang (2019), where finite \(p\) and \(\kappa\) are assumed. In addition, with a smooth regularization, Algorithm 4 benefits from the averaging step in that it improves the contraction rate to approximately \(\kappa^2 p/n\).

Let \(\tilde{\theta}_t\) be the \(t\)th iterate of one of the proposed algorithms. It is clear that the statistical error of the estimator \(\theta_t\) is bounded by its optimization error and the statistical error of \(\tilde{\theta}\):

\[
\|\theta_t - \theta^*\|_2 \leq \|\theta_t - \tilde{\theta}\|_2 + \|\tilde{\theta} - \theta^*\|_2.
\]

The second term is well-studied in statistics, which is of order \(O_P(\sqrt{p/N})\) under mild conditions. In the following theorem, we show that for the first term (i.e., the optimization error), with proper choice of \(\alpha\), each iteration of Algorithm 3 or 4 makes \(\theta_t\) closer to the global minimum \(\tilde{\theta}\) by some order depending on local sample size. Thus, through finite steps, the optimization errors are eventually negligible in comparison with statistical errors (assuming \(N\) is of order \((n/p)^a\) for a finite \(a\) in typical applications), and the distributed multistep estimator will work as well as the global minimum as if the data were aggregated in the central server.

**Theorem 4.1.** Suppose that Assumptions 4.1 and 4.2 hold, and with high probability the initial value \(\theta_0 \in B(\tilde{\theta}, R/2)\). Let \(\eta = \kappa^2 (\log N)/p\) and \(\kappa = \|\Sigma\|_2/\rho\). For any \(c_1, c_2 > 0\), there exists \(C > 0\) such that the followings hold with high probability:

(a) If \(n \geq c_1 p\) and \(\alpha \geq C \rho \eta\), then both Algorithms 3 and 4 have linear convergence

\[
\|\theta_t - \tilde{\theta}\|_2 \leq \left[ 1 - \frac{\rho}{10(\alpha + \rho)} \right]^t \|\theta_0 - \tilde{\theta}\|_2, \quad \forall t \geq 0;
\]

(b) If \(\eta\) is sufficiently small and \(\alpha \leq c_2 \rho \eta\), then for both algorithms

\[
\|\theta_t - \tilde{\theta}\|_2 = O_P(\eta^{t/2} \|\theta_0 - \tilde{\theta}\|_2), \quad \forall t \geq 0;
\]

in addition, if Assumption 4.3 also holds, then for Algorithm 4, we have

\[
\|\theta_t - \tilde{\theta}\|_2 = O_P(\eta^{t-t_0} \|\theta_{t_0} - \tilde{\theta}\|_2), \quad \forall t \geq t_0,
\]

where \(t_0 = \left[ \frac{2 \log(CMR/p\rho)}{\log(1/\eta)} \right]^{1/2}\).
In contrast to a fixed contraction derived by Shamir, Srebro, and Zhang (2014), Theorem 4.1 explains the significant benefits of large local sample size even in the presence of a non-smooth penalty: the optimization error shrinks by a factor that converges to zero explicitly in \( n \). As a brief illustration, let us consider the case with smooth loss functions, sufficient local sample size and no regularization. Let \( \theta_0 \) be the average of individual estimators on node machines. By Corollary 2 in Zhang, Duchi, and Wainwright (2013), this simple divide-and-conquer estimator has accuracy \( ||\theta_0 - \theta^*||_2 = O_p(\max\{\kappa \sqrt{\frac{p \log p}{n}}, \frac{p \log p}{n}\}) \). Using the explicit expression of \( \eta \), we can easily deduce from Theorem 4.1 (b) that the one-step estimator \( \theta_1 \) obtained by Algorithm 3 behaves the same as the global minimizer \( \hat{\theta} \) if the local sample size satisfies \( n^2 \gg N(\kappa^2 p \log N)(p + \kappa^2 \log p \). In this case, the local optimization in Algorithm 3 can further be replaced by using the explicit one-step estimator as in Bickel (1975) and Jordan, Lee, and Yang (2019), since the initial estimator is in a consistent neighborhood. More generally, the t-step estimator \( \theta_t \) has negligible optimization error under even weaker local sample size requirement: \( n^{t+2} \gg N(\kappa^2 p \log N)(p + \kappa^2 \log p \). A similar remark applies to Algorithm 4.

Similar to the deterministic results, the averaging step is about as effective as the optimization step when \( g \) is smooth and \( \theta_1 \) is sufficiently close to \( \hat{\theta} \) in that after a finite \( t_0 \) iterations. See a simplified example in Section 4.2 with \( \alpha = 0 \).

As for the initialization, the condition \( \theta_0 \in B(\hat{\theta}, R/2) \) is mild, since \( \hat{\theta} \) is usually a consistent estimate and \( ||\theta_0 - \theta^*||_2 \) is bounded (Assumption 4.1). In contrast with Jordan, Lee, and Yang (2019), we allow inaccurate initial value such as \( \theta_0 = 0 \) and give more explicit rates of contraction even when \( p \) and \( \kappa \) diverge. On the other hand, the accuracy of the initial estimator \( \theta_0 \) does help reduce the number of iterations.

Combining the results to be presented in Theorem 4.2, we will see that by choosing \( \alpha \propto \rho \eta \), Algorithms 3 and 4 inherit all the merits of Algorithms 1 and 2 in the large-\( n \) regime—fast linear contraction of rate \( \sqrt{n} \geq \kappa \sqrt{p (\log N)/n} \), and for Algorithm 4, a even faster rate of \( \eta \) for smooth functions and the loss and the penalty functions are smooth. These facts also guarantee that Algorithms 3 and 4 reach the statistical efficiency in \( O(\frac{\log \|\theta_0 - \theta^\|_2 + \log(N/p)}{\log(1/\eta)}) \) iterations. On the other hand, compared to Algorithms 1 and 2, Algorithms 3 and 4 overcome the difficulties with a small local sample size \( n \) in that as long as \( n/p \) is bounded away from some small constant (which is reasonable for many big-data problems of interest), shrinkage of optimization error is guaranteed. Moreover, while it is hard to check whether \( n \) is sufficiently large in practice, proper choice of \( \alpha \) always guarantees linear convergence, and the contraction rates adapt to the sample size \( n \). In this way, Algorithms 3 and 4 perfectly resolve the main issue of their vanilla versions.

We can get stronger results in the specific case of distributed linear regression, where the contraction rate has nearly no dependence on the conditional number \( \kappa \). Due to space constraints, we put all the details in Appendix B.

### 4.2. Multistep Estimators in Large-Sample Regimes

We now present the contraction of optimization error of Algorithms 1 and 2.

**Theorem 4.2.** Suppose that Assumptions 4.1 and 4.2 hold, and with probability tending to one, \( \theta_0 \in B(\hat{\theta}, R) \) for some \( R \gg ||\hat{\theta} - \theta^*||_2 \). For Algorithms 1 and 2, we have

\[
||\theta_t - \hat{\theta}||_2 = O_p(\eta^{t/2}||\theta_0 - \hat{\theta}||_2), \quad \forall t \geq 0,
\]

where \( \eta = \kappa^2 \rho \log(N)/n \). In addition, let Assumption 4.3 also hold. There exists some constant \( C \) such that for Algorithm 2, we have

\[
||\theta_t - \hat{\theta}||_2 = O_p(\eta^{t - t_0}||\theta_0 - \hat{\theta}||_2), \quad \forall t \geq t_0,
\]

where \( t_0 = \left\lfloor \frac{2 \log(CM(\rho))}{\log(1/\eta)} \right\rfloor \).

The strengthened result (9) requires Assumption 4.3 and thus smooth \( g \). Theorem 4.2 shows that when \( n \) is sufficiently large, Algorithms 1 and 2 behave similarly as Algorithms 3 and 4—faster convergence with the larger \( n \), mild restrictions on initialization, and averaging speeds up contraction given smooth loss. However, there is no convergence guarantee in general regimes. Section 5 further shows that with insufficient local sample size, the practical performance Algorithms 1 and 2 is less satisfactory.

Finally, to see how the averaging step reduces the statistical error (i.e., the distance between the estimator and \( \theta^* \)), we continue to look at the linear regression example mentioned at the end of Section 2.2. For simplicity, assume \( x_i \) are iid standard normal random vectors. Equations (3) and (4) can be expressed as

\[
\tilde{\Sigma}^{1/2}(\theta_{t+1,k} - \hat{\theta}) = \left( I - \tilde{\Sigma}^{1/2}_k \tilde{\Sigma}^{-1}_k \tilde{\Sigma}^{1/2}_k \right) \cdot \tilde{\Sigma}^{1/2}(\theta_t - \hat{\theta}),
\]

\[
\tilde{\Sigma}^{1/2}_k(\theta_{t+1} - \hat{\theta}) = \left[ I - \tilde{\Sigma}^{1/2}_k \left( \frac{1}{m} \sum_{k=1}^{m} \tilde{\Sigma}^{-1}_k \tilde{\Sigma}^{1/2}_k \right) \right] \cdot \tilde{\Sigma}^{1/2}(\theta_t - \hat{\theta}).
\]

If further \( n \gg p \), then the two contraction factors satisfy

\[
||I - \tilde{\Sigma}^{1/2}_k \tilde{\Sigma}^{-1}_k \tilde{\Sigma}^{1/2}_k||_2 = O_p(\sqrt{p/n}),
\]

\[
||I - \tilde{\Sigma}^{1/2}_k \left( \frac{1}{m} \sum_{k=1}^{m} \tilde{\Sigma}^{-1}_k \right) \tilde{\Sigma}^{1/2}_k||_2 = O_p(p/n).
\]

When the Algorithm 1 applies to this problem, the expression changes slightly to

\[
\tilde{\Sigma}^{1/2}(\theta_{t+1,1} - \hat{\theta}) = \left( I - \tilde{\Sigma}^{1/2}_k \tilde{\Sigma}^{-1}_k \tilde{\Sigma}^{1/2}_k \right) \cdot \tilde{\Sigma}^{1/2}(\theta_{t+1} - \hat{\theta}).
\]

The smaller magnitude in Equation (11) is due to the averaging.

Suppose that we initialize the algorithm using the one-shot average \( \theta_0 = \frac{1}{m} \sum_{k=1}^{m} \theta_k \), where \( \theta_k \) is the least-square solution on the kth machine. When \( n = O(\sqrt{Np}) \), we have \( \sqrt{p/N} = O(p/n) \). Zhang, Duchi, and Wainwright (2013) assert that \( ||\hat{\theta} - \theta^*||_2 = O_p(\sqrt{p/N}) \), \( ||\theta_0 - \hat{\theta}||_2 = O_p(p/n) \). Thus, the initial statistical error is \( ||\theta_0 - \theta^*||_2 = O_p(\max\{\sqrt{p/N}, p/n\}) = O_p(p/n) \).
By the contraction properties in Equations (10) and (11), the optimization errors are
\[
\|\theta_{1:k} - \bar{\theta}\|_2 = O_p(\sqrt{p/n}||\theta_0 - \bar{\theta}\|_2) = O_p(p^{3/2}/n^{3/2}),
\]
\[
||\theta_1 - \bar{\theta}||_2 = O_p(p/n)\|\theta_0 - \bar{\theta}\|_2 = O_p(p^2/n^2).
\]

We can see that when \( p^2/n^2 \ll \sqrt{p/N} \ll p^{3/2}/n^{3/2} \) (or equivalently, \( p^{3/4}N^{1/4} \ll n \ll p^{3/2}N^{1/2} \)), the optimization error is negligible for \( \theta_1 \), but is not negligible for \( \theta_{1:k} \). When \( n \) is smaller, even more iterations are needed. A refined analysis of distributed least squares is in Appendix B.

### 4.3. Guidance on Practice

We now provide some general guidance on how to implement the CEASE algorithm in practice. First, we recommend choosing an initialization depending on the magnitude of the local sample size \( n \). In particular, when \( n \) is not very large compared to the dimension \( p \), a zero initialization would be more robust. On the other hand, given a moderate or large \( n \), the one-shot average estimator will lead to extremely fast convergence.

Second, according to Theorem 4.1, it suffices to take \( \alpha \) of the order of \( p^2\alpha p\log N/n \). In practice, setting \( \alpha \) to be a small multiple of \( p/n \) seems suitable in many occasions.

Finally, as is already shown, both Algorithms 3 and 4 reach statistical efficiency in \( O(\log||\theta_0 - \bar{\theta}||_2 + \log(N/p)/\log(1/\eta)) \) iterations. In both our simulations and real data examples, the CEASE algorithms with a properly chosen \( \alpha \) converge to the centralized estimator within 10 iterations. With a moderate \( n \), a warm start further boosts the convergence speed.

### 5. Numerical Experiments

#### 5.1. Synthetic Data

We first conduct distributed logistic regression to illustrate the effect of local sample size and initialization on convergence. We keep the total sample size \( N = 10000 \) and the dimensionality \( p = 101 \) fixed, and generate the iid data \( \{(x_i, y_i)\}_{i=1}^N \) as follows: \( x_i = (1, u_i^T)^T \) with \( u_i \sim N(0_{p-1}, \Sigma) \) and \( \Sigma = \text{diag}(10, 5, 2, 1 \cdots 1) \in \mathbb{R}^{(p-1)\times(p-1)}; P(y_i = 1) = 1 - P(y_i = 0) = 1/(1 + e^{-x_i^T \theta^*}) \) where \( \theta^* \in \mathbb{R}^p \) is a random vector with norm 3 whose direction is chosen uniformly at random from the sphere. We use the natural logarithm of the estimation error \( ||\hat{\theta} - \theta^*||_2 \) to measure the performance of different algorithms, including multiple versions of the CEASE algorithms, GIANT (Wang et al. 2018), ADMM (Boyd et al. 2011) and accelerated gradient descent (Nesterov 1983).

Figure 1 shows how the estimation errors evolve with iterations. The curves show the average values over 100 independent runs; the error bands correspond to one standard deviation. The regimes “large \( n \),” “moderate \( n \)” and “small \( n \)” refer to \((n, m) = (2000, 5), (1000, 10), \) and \((250, 40)\); “zero initialization” and “good initialization” refer to \( \theta_0 = 0 \) (bottom panel) and \( \hat{\theta} \) (top panel), respectively. Here, \( \hat{\theta} \) is the one-shot distributed estimator (Zhang, Duchi, and Wainwright 2013) that averages the individual estimators on node machines. According to Figure 1, the standard deviation of each iterate is around 0.1. For the “large \( n \), good initialization” regime, all of the iterates are unsurprisingly very close to the optimal solution. Their error bands will cover up the curves. So we omit the bands in that case for the sake of clarity.

With proper regularization, the two CEASE algorithms are the only ones that converge rapidly in all scenarios. The purely

![Figure 1](image-url)  
**Figure 1.** Impacts of local sample size and initialization on convergence. The x-axis and y-axis are the number of iterations and \( \log||\theta_1 - \theta^*||_2 \). The dashed lines show the error of the minimizer of the overall loss function. The top and bottom panels use \( \theta \) and \( \hat{\theta} \) for initialization, respectively. CEASE(a) and CEASE(0) refer to Algorithm 4 with \( \alpha = 0.15p/n \) and 0; CEASE-single(a) and CEASE-single(0) refer to Algorithm 3 with \( \alpha = 0.15p/n \) and 0, respectively. In particular, CEASE-single(0) is equivalent to the CSL algorithm in Jordan, Lee, and Yang (2019).
and CEASE-single(0) refer to Algorithm 3 with \(\alpha = 0.15p/n\) and 0; CEASE-single(a) and CEASE-single(0) refer to Algorithm 3 with \(\alpha = 0.15p/n\) and 0, respectively. In particular, CEASE-single(0) is equivalent to the CSL algorithm in Jordan, Lee, and Yang (2019). GIANT and CEASE-single(0) do not converge to the optimal solution.

deterministic methods ADMM (Boyd et al. 2011) and accelerated gradient descent (Nesterov 1983) are also reliable but slow. Other distributed algorithms like unregularized CEASE and GIANT (Wang et al. 2018) easily fail when the local sample size is small or the initialization is uninformative. In addition, the CEASE with averaging (Algorithm 4) is superior to the one without averaging (Algorithm 3). For example, when \((n, m) = (1000, 10)\), the averaged CEASE with \(\alpha = 0\) converges while the one without averaging does not. Hence, the averaging step leads to better performance.

We also test the efficacy of our algorithms in the distributed \(\ell_1\)-regularized logistic regression, where the penalty \(g\) is nonsmooth (See Appendix D for details). To summarize, our simulations demonstrate several important properties of the CEASE Algorithms:

- In all scenarios, the CEASE Algorithms converge rapidly, usually within several steps, which is consistent with our theory;
- The CEASE Algorithms efficiently utilize statistical structures and similarities among local losses, and benefit from the averaging step with smooth loss functions;
- The CEASE Algorithms are also able to handle the most general situations (e.g., small local sample size, uninformative initialization) with convergence guarantees.

### 5.2. Real Data

As a real data example, we choose the Fashion-MNIST dataset (Xiao, Rasul, and Vollgraf 2017) as a testbed for comparison of algorithms. The whole dataset consists of 70000 grayscale images of fashion products in 10 classes, each of which has 6000 training samples and 1000 testing samples. We choose the 7th and 9th classes (Sneakers and Ankle boots) and the goal is to train a classifier that distinguishes them. Each image has \(28 \times 28 = 784\) pixels, represented by a feature vector in \([0, 1]^{784}\). The number of training (or testing) samples is \(6000 \times 2 = 12000\) (or \(1000 \times 2 = 2000\)). We randomly partition the training set and conduct logistic regression in a distributed manner. The performance metric is the classification error on the testing set. Figure 2 shows the average performance of the CEASE algorithms, ADMM, GIANT and AGD based on 100 independent runs, together with error bars showing one standard deviation. Here “large \(n\),” “moderate \(n\),” and “small \(n\)” refer to \((n, m) = (12000, 10), (480, 25)\) and \((240, 50)\), respectively. All of the iterations are initialized with the one-shot average (Zhang, Duchi, and Wainwright 2013). The experiments on this real data example also support our theoretical findings.

### 6. Discussions

We have developed two CEASE distributed estimators (Algorithms 3 and 4) for statistical estimation, with theoretical guarantees and superior performance on real data. Several new directions are worth exploring. First, while we assumed exact computation for simplicity, finer analysis should allow for inexact updates in practice. Second, we hope to extend the algorithms to decentralized and asynchronous settings. Third, distributed versions of confidence regions and hypothesis tests are of great importance, and our point estimation strategies may serve as a starting point. Finally, it will be interesting to explore non-convex statistical optimization problems such as mixture models and deep learning. We believe that the idea of gradient-enhanced loss function still plays an important role.

**Supplementary Material**

Supplementary material: The file “supplementary.pdf” contains more details and proofs of the results in this article.

**Acknowledgments**

We gratefully acknowledge NSF grants DMS-1662139, DMS-1712591, DMS-2053832, DMS-2052926, NIH grant 2R01-GM072611-15, and ONR grant N00014-19-1-2120. We acknowledge computing resources from Columbia University’s Shared Research Computing Facility project, which is supported by NIH Research Facility Improvement Grant 1G20-RR030893-01, and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract C090171, both awarded April 15, 2010.

**References**

Arjevani, Y., and Shamir, O. (2015), “Communication Complexity of Distributed Convex Learning and Optimization,” in Advances in Neural Information Processing Systems, 1756–1764. [1004]

Banerjee, M., Durot, C., and Sen, B. (2019), “Divide and Conquer in Nonstandard Problems and the Super-Efficiency Phenomenon,” The Annals of Statistics, 47, 720–757. [1000]
Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2018), “Distributed Testing and Estimation Under Sparse High Dimensional Models,” The Annals of Statistics, 46, 1352–1382. [1000]

Bickel, P. J. (1975), “One-Step Huber Estimates in the Linear Model,” Journal of the American Statistical Association, 70, 428–434. [1001,1002,1007]

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011), “Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers,” Foundations and Trends in Machine Learning, 3, 1–122. [1001,1008,1009]

Chen, X., and Xie, M.-g. (2014), “A Split-and-Conquer Approach for Distributed Statistical Inference,” The Annals of Statistics, 42, 1203–1233. [1000,1001]

Chen, X., and Xie, M.-g. (2014), “A Split-and-Conquer Approach for Analysis of Extraordinarily Large Data,” Statistica Sinica, 24, 1655–1684. [1000]

Craner, R., and Roosta, F. (2019), “Dingo: Distributed Newton-Type Method for Gradient-Norm Optimization,” in Advances in Neural Information Processing Systems, eds. H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox and R. Garnett, Vol. 32, Red Hook, New York: Curran Associates, Inc. [1001]

Fan, J., and Li, R. (2001), “First-Order Newton-Type Estimator for Distributed Estimation and Inference,” Journal of the American Statistical Association, 1–17, DOI: 10.1080/01621459.2021.1891925. [1001,1001]

Fan, J., and Xie, M.-g. (2014), “A Split-and-Conquer Approach for Analysis of Extraordinarily Large Data,” Statistica Sinica, 24, 1655–1684. [1000]

Crane, R., and Roosta, F. (2019), “Dingo: Distributed Newton-Type Method for Gradient-Norm Optimization,” in Advances in Neural Information Processing Systems, eds. H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox and R. Garnett, Vol. 32, Red Hook, New York: Curran Associates, Inc. [1001]

Fan, J., and Li, R. (2001), “Variable Selection Via Nonconcave Penalized Likelihood and Its Oracle Properties,” Journal of the American Statistical Association, 96, 1348–1360. [1001]

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2019), “Distributed Estimation of Principal Eigenspaces,” The Annals of Statistics, 47, 3009. [1000]

Garber, D., Shamir, O., and Srebro, N. (2017), “Communication-Efficient Algorithms for Distributed Stochastic Principal Component Analysis,” in International Conference on Machine Learning, Vol. 70, 1203–1212. [1000]

Han, Y., Mukherjee, P., Ozgur, A., and Weissman, T. (2018), “Distributed Statistical Estimation of High-Dimensional and Nonparametric Distributions,” in 2018 IEEE International Symposium on Information Theory (ISIT), 506–510, Piscataway, NJ: IEEE. [1000]

Hoerl, A. E., and Kennard, R. W. (1970), “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” Technometrics, 12, 55–67. [1000]

Jordan, M. I., Lee, J. D., and Yang, Y. (2019), “Communication-Efficient Distributed Statistical Inference,” Journal of the American Statistical Association, 114, 668–681. [1000,1001,1002,1005,1006,1007,1008,1009]

Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014), “A Scalable Bootstrap for Massive Data,” Journal of the Royal Statistical Society, Series B, 76, 795–816. [1000]

Lee, J. D., Lin, Q., Ma, T., and Yang, T. (2017a), “Distributed Stochastic Variance Reduced Gradient Methods by Sampling Extra Data With Replacement,” The Journal of Machine Learning Research, 18, 4404–4446. [1004]

Lee, J. D., Liu, Q., Sun, Y., and Taylor, J. E. (2017b), “Communication-Efficient Sparse Regression,” The Journal of Machine Learning Research, 18, 1–30. [1000]

Nesterov, Y. E. (1983), “A Method for Solving the Convex Programming Problem With Convergence Rate $O(1/k^2)$,” Dokl. Akad. Nauk SSSR, 269, 543–547. [1008,1009]

Nocedal, J., and Wright, S. J. (2006), Numerical Optimization (2nd ed.), New York, NY: Springer. [1003]

Robinson, P. M. (1988), “The Stochastic Difference Between Econometric Statistics,” Econometrica: Journal of the Econometric Society, 56, 531–548. [1002]

Rockafellar, R. T. (1976), “Monotone Operators and the Proximal Point Algorithm,” SIAM Journal on Control and Optimization, 14, 877–898. [1001,1003]

Rosenblatt, J. D. and Nadler, B. (2016), “On the Optimality of Averaging in Distributed Statistical Learning,” Information and Inference: A Journal of the IMA, 5, 379–404. [1000]

Shamir, O., Srebro, N., and Zhang, T. (2014), “Communication-Efficient Distributed Optimization Using an Approximate Newton-Type Method,” in International Conference on Machine Learning, 32, 1000–1008. [1000,1001,1002,1004,1005,1007]

Shang, Z., and Cheng, G. (2017), “Computational Limits of a Distributed Algorithm for Smoothing Splines,” The Journal of Machine Learning Research, 18, 3809–3845. [1000]

Shi, C., Lu, W., and Song, R. (2018), “A Massive Data Framework for M-Estimators With Cubic-Rate,” Journal of the American Statistical Association, 113, 1–12. [1000]

Szabo, B., and Van Zanten, H. (2019), “An Asymptotic Analysis of Distributed Nonparametric Methods,” The Journal of Machine Learning Research, 20, 87–1. [1000]

Tibshirani, R. (1996), “Regression Shrinkage and Selection Via The Lasso,” Journal of the Royal Statistical Society, Series B, 58, 267–288. [1001]

Volgushev, S., Chao, S.-K., and Cheng, G. (2019), “Distributed Inference for Quantile Regression Processes,” The Annals of Statistics, 47, 1634–1662. [1000]

Wang, J., Kolar, M., Srebro, N., and Zhang, T. (2017), “Efficient Distributed Learning With Sparsity,” in International Conference on Machine Learning, Sydney, Australia, 3636–3645. [1000,1001]

Wang, J., Wang, W., and Srebro, N. (2017), “Memory and Communication Efficient Distributed Stochastic Optimization With Minibatch Prox,” in Conference on Learning Theory, Amsterdam, Netherlands, 1882–1919. [1004]

Wang, S., Roosta-Khorasani, F., Xu, P., and Mahoney, M. W. (2018), “Giant: Globally Improved Approximate Newton Method for Distributed Optimization,” in Advances in Neural Information Processing Systems, 31, 2338–2348. [1000,1001,1008,1009]

Wang, X., and Dunson, D. B. (2013), “Parallelizing MCMC Via Weierstrass Sampler,” arXiv:1312.4605. [1000]

Xiao, H., Rasul, K., and Vollgraf, R. (2017). “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms,” arXiv:1708.07747. [1009]

Xiao, H., Rasul, K., and Vollgraf, R. (2017). “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms,” arXiv:1708.07747. [1009]

Zhang, C.-H. (2010), “Nearly Unbiased Variable Selection Under Minimax Concave Penalty,” The Annals of Statistics, 38, 894–942. [1001]

Zhang, Y., Duchi, J. C., and Wainwright, M. J. (2013), “Communication-Efficient Algorithms for Statistical Optimization,” Foundations and Trends® in Machine Learning, 7, 1–157, DOI:10.1561/2200000011. [1001,1002,1003,1004,1005,1006,1007,1008,1009]

Zhang, Y., and Xiao, L. (2015), “Disco: Distributed Optimization for Self-Concordant Empirical Loss,” in International Conference on Machine Learning, Lille, France, 362–370. [1000,1004]