Distributed Inexact Newton-type Pursuit for Non-Convex Sparse Learning

Bo Liu[†,‡] **Xiao-Tong Yuan**[§] **Lezi Wang**[†] **Qingshan Liu**[§] **Junzhou Huang**[♯] **Dimitris N. Metaxas**[†]
[†]Rutgers, The State University of New Jersey, Piscataway, NJ, USA
[‡]JD Digits, Mountain View, CA, USA
[§]B-DAT Lab, Nanjing University of Information Science and Technology, Nanjing, China
[♯] Tencent AI Lab, Shenzhen, China

Abstract

In this paper, we present a sample distributed greedy pursuit method for non-convex sparse learning under cardinality constraint. Given the training samples uniformly randomly partitioned across multiple machines, the proposed method alternates between local inexact sparse minimization of a Newton-type approximation and centralized global results aggregation. Theoretical analysis shows that for a general class of convex functions with Lipschitze continues Hessian, the method converges linearly with contraction factor scaling *inversely* to the local data size; whilst the communication complexity required to reach desirable statistical accuracy scales *logarithmically* with respect to the number of machines for some popular statistical learning models. For non-convex objective functions, up to a local estimation error, our method can be shown to converge to a local stationary sparse solution with sub-linear communication complexity. Numerical results demonstrate the efficiency and accuracy of our method when applied to large-scale sparse learning tasks including deep neural nets pruning.

1 Introduction

The following cardinality-constrained empirical risk minimization (ERM) problem is ubiquitous in high-dimensional sparse statistical learning:

$$\min_{w\in\mathbb{R}^p} F(w) := \frac{1}{N}\sum_{i=1}^{N} f(w; x_i, y_i), \quad \text{s.t. } \|w\|_0 \le k, \quad (1)$$

where $\{x_i, y_i\}_{i=1}^N$ are training samples, f is a general loss function, $\|w\|_0$ represents the number of non-zero entries in the parameter vector w, and k is an integer controlling the cardinality. Due to the presence of cardinality constraint, the problem is non-convex and NP-hard even when f is convex. In this paper, we are interested in distributed computing methods for solving such a non-convex ERM problem. In particular, we assume the training data $\mathcal{D} = \{D_1, ..., D_m\}$ with $N = mn$ samples is evenly and randomly distributed over m different machines; each machine j locally stores and accesses n training samples $D_j = \{x_{ji}, y_{ji}\}_{i=1}^n$. Let $F_j(w) := \frac{1}{n}\sum_{i=1}^n f(w; x_{ji}, y_{ji})$ be the local empirical risk evaluated on D_j. The global goal is to minimize the average of these local objectives under cardinality constraint:

$$\min_{w\in\mathbb{R}^p} F(w) = \frac{1}{m}\sum_{j=1}^{m} F_j(w), \quad \text{s.t.} \quad \|w\|_0 \le k. \quad (2)$$

We will refer to this model as ℓ_0-ERM throughout this paper.

1.1 Related work and motivation

Iterative hard thresholding. For the generic ℓ_0-minimization problem (1), the iterative hard thresholding (IHT) methods have demonstrated superior scalability in statistical learning models (Beck & Eldar, 2013; Yuan et al., 2014; Jain et al., 2014). The iteration procedure of IHT is as simple as a truncated version of gradient descent step: $w^{(t)} = \mathrm{H}_k\left(w^{(t-1)} - \eta\nabla F(w^{(t-1)})\right)$, where $\mathrm{H}_k(x)$ is a truncation operator which preserves the top k (in magnitude) entries of vector x and sets the remaining to be zero. Let $\bar{w}$ be a $\bar{k}$-sparse target solution. If $F(w)$ is L-smooth and μ_s-strongly-convex over any s-sparse vector space with $s = O(k)$, then it is known from (Jain et al., 2014) that with some sparsity level $k = O\left(\frac{L^2}{\mu_s^2}\bar{k}\right)$, IHT-style methods reach the estimation error level $\|w^{(t)} - \bar{w}\| = \mathcal{O}\left(\sqrt{k}\|\nabla F(\bar{w})\|_\infty/\mu_s\right)$ after

$$\mathcal{O}\left(\frac{L}{\mu_s}\log\left(\frac{\mu_s\|w^{(0)} - \bar{w}\|}{\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)\right) \quad (3)$$

rounds of iteration. A direct approach for distributed ℓ_0-ERM is a centralized map-reduce implementation of IHT: in the map step each machine calculates local gradient $\nabla F_j(w^{(t-1)})$ at $w^{(t-1)}$, and in the reduce step these

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

are averaged to obtain the full gradient $\nabla F(w^{(t-1)}) = \frac{1}{m}\sum_{j=1}^{m} F_j(w^{(t-1)})$ on a master machine, followed by the truncated gradient step. This distributed IHT approach was first introduced in (Patterson et al., 2014) for compressive sensing. As the iterates of distributed IHT are standard IHT iterates, the iteration complexity, and so the communication complexity, is identical to that of standard IHT. However, as suggested by (3), the linear dependence of the iteration complexity on the restricted condition number L/μ_s obviously makes the distributed IHT communication inefficient in ill-conditioned problems.

Distributed approximate Newton-type methods. For classical distributed ERM problems, the iteration complexity of first-order distributed approaches including gradient descent and ADMM (Boyd et al., 2011) also suffer from the unsatisfactory polynomial dependence on condition number. To alleviate this issue, Shamir et al. (2014) proposed a distributed approximate Newton-type (DANE) method that takes advantage of the stochastic nature of problem: the i.i.d. data samples $\{x_i, y_i\}$ are uniformly distributed and each local problem will become sufficiently similar to the global problem when data size increases. If $F(w)$ is quadratic with condition number L/μ, the communication complexity (in high probability) of DANE to reach ϵ-precision was shown to be $\mathcal{O}\left(\frac{L^2}{\mu^2 n}\log(mp)\log\left(\frac{1}{\epsilon}\right)\right)$, which has an improved dependence on the condition number L/μ which could scale as large as $\mathcal{O}(\sqrt{mn})$ in regularized learning problems. By applying Nesterov's acceleration technique, AIDE (Reddi et al., 2016b) further reduces the communication complexity of DANE to $\mathcal{O}\left(\sqrt{\frac{L}{\mu n^{1/2}}}\log(mp)\log\left(\frac{1}{\epsilon}\right)\right)$ in the quadratic case, which is nearly optimal for first-order distributed learning problems. For more general self-concordant empirical risk functions, Zhang & Lin (2015) proposed DiSCO as a distributed inexact damped Newton method with comparable communication complexity to AIDE. More recently, the EDSL (Wang et al., 2017) and TWT (Ren et al., 2017) methods extend DANE to solving ℓ_1-norm regularized ERM problems, obtaining similarly improved dependence of communication complexity on condition number. The common finding of this line of existing work is: when the local subproblems are well structured and sufficiently correlated to each other, the distributed Newton-type methods are able to approximate the global optimal solution in considerably fewer rounds of communication than the conventional first-order distributed learning methods.

Motivation. Despite the success of distributed approximate (inexact) Newton-type methods in regularized convex ERM learning, it is so far not clear if this class of methods generalizes equally well, both in theory and practice, to the non-convex ℓ_0-ERM model (2). This motivates us to explore the potential of DANE-type distributed ℓ_0-minimization methods in gaining improved communication efficiency over those first-order alternatives such as distributed IHT.

1.2 Overview of our approach and contribution

We propose the Distributed Inexact Newton-type PurSuit (D-INPS) method as a natural extension of DANE to distributed ℓ_0-ERM. The algorithm iterates between two main steps: 1) each worker machine (inexactly) solves a variance-reduced local ℓ_0-ERM which is constructed based on the local loss function and the current global gradient information; and 2) the master machine generates the next iterate via properly aggregating the local sparse solutions received from worker machines. In practice, the proposed method has been implemented on parameter server platform (Li et al., 2014) with actual performance evaluated on synthetic and real data high dimensional statistical learning tasks.

Although our method shares a similar algorithmic framework with DANE, its iteration complexity analysis turns out to be more challenging due to the presence of non-convex cardinality constraint $\|w\|_0 \le k$. Provided that n is sufficiently large and $F(w)$ is convex with restricted Lipschitz continuous Hessian (see Definition 2) and restricted condition number L/μ_s, we show in Theorem 2 that the estimation error $\|w^{(t)} - \bar{w}\| = \mathcal{O}\left(\sqrt{k}\|\nabla F(\bar{w})\|_\infty/\mu_s\right)$ can be guaranteed in overwhelming probability after

$$\mathcal{O}\left(\frac{1}{1-\frac{L}{\mu_s}\sqrt{\frac{\log(mp)}{n}}}\log\left(\frac{\mu_s\|w^{(0)}-\bar{w}\|}{\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)\right) \tag{4}$$

rounds of communication. To compare with (3), this above bound has clearly improved dependence on restricted condition number when data size is sufficiently large. In sharp contrast to the analysis of DANE (Shamir et al., 2014) and AIDE (Reddi et al., 2016b) which are restricted to quadratic problems, our bound in (4) is applicable to *a much wider problem spectrum* in machine learning. Provided that $n = \mathcal{O}\left(\frac{L^2\log(mp)}{\mu_s^2}\right)$ is sufficiently large and equipped with proper initialization, the bound can be shown to imply for some popular statistical learning models that the communication complexity scales logarithmically with respect to the number of machines. *In comparison, the sample complexity in (Wang et al., 2017; Ren et al., 2017) for ℓ_1-regularized ERM is $n = \mathcal{O}\left(\frac{s^2L^2\log p}{\mu_s^2}\right)$ which is inferior to ours. As another highlight of analysis, we have analyzed our method for non-convex functions, which to our knowledge has not been addressed in previous DANE-type sparse learning methods.*

1.3 Notation and organization

Notation. We denote $\mathrm{H}_k(x)$ as a truncation operator which preserves the top k (in magnitude) entries of vector x and forces the remaining to be zero. The notation $\mathrm{supp}(x)$ represents the index set of nonzero entries of x. We conventionally define $\|x\|_\infty = \max_i |[x]_i|$ and define $x_{\min} = \min_{i\in\mathrm{supp}(x)} |[x]_i|$. For an index set S, we define $[x]_S$ and $[A]_{SS}$ as the restriction of x to S and the restriction

of rows and columns of A to S, respectively. For an integer n, we abbreviate the index set $\{1, ..., n\}$ to $[n]$.

Organization. The rest of this paper is structured as follows: In Section 2 we introduce our distributed approximated Newton-type greedy pursuit method. The theoretical properties of the proposed method for convex and non-convex functions are then analyzed in Section 3 and Section 4, respectively. A brief survey on some other related work is given in Section 5. The numerical evaluation results are presented in Section 6. Finally, the concluding remarks are made in Section 7. Due to space limit, all the technical proofs of results are deferred to the supplement.

2 The DINPS Method

The high level algorithmic procedure of DINPS is outlined in Algorithm 1. Starting from an initial k-sparse approximation $w^{(0)}$, the procedure generates a sequence of intermediate k-sparse iterate $\{w^{(t)}\}_{t\geq 1}$ via distributed local sparse estimation and global synchronization among machines. More precisely, each iteration loop of DINPS can be decomposed into the following three consequent steps:

Map-reduce gradient computation. In this step, the global gradient $\nabla F(w^{(t-1)}) = \frac{1}{m}\sum_{j=1}^m \nabla F_j(w^{(t-1)})$ is evaluated at the current iterate via simple map-reduce averaging and distributed to all machines for local computation.

Local inexact sparse approximation. Based on the received gradient $\nabla F(w^{(t-1)})$, each machine j constructs at the current iterate a local objective function (5) and then inexactly estimate a local k-sparse solution $w_j^{(t)} \approx \arg\min_{\|w\|_0\leq k} P_j(w; w^{(t-1)}, \eta, \gamma)$ up to sparsity level $\bar{k} \leq k$ and ϵ-suboptimality. This inexact sparse optimization step can be implemented using IHT-style algorithms which have been witnessed to offer fast and accurate solutions for ℓ_0-estimation (Yuan et al., 2014; Jain et al., 2014).

Centralized results aggregation. We compute the truncated average $w^{(t)} = \mathrm{H}_k\left(\frac{1}{m}\sum_{j=1}^m w_j^{(t)}\right)$ as the next iterate generated from local sparse predictors. Here the truncation operation is conducted to maintain sparsity of output. As an alternative option, setting $w^{(t)} = w_1^{(t)}$ also works reasonably well in theory and practice. However, our numerical experience indicates that the truncated averaging strategy tends to make more balanced workload among machines and thus can produce slightly more accurate and stable solutions, especially for deep neural networks pruning.

The construction of the local objective (5) is inspired by the idea of leveraging the first-order gradient information and local higher-order information for local processing as originally introduced in DANE (Shamir et al., 2014). Compared to those first-order distributed methods (Boyd et al., 2011; Jaggi et al., 2014), such a way of local computation is known to be able to take advantage of inter-machine statistical correlation to dramatically reduce the frequency of communication. Similar local optimization strategy was also considered by (Wang et al., 2017; Ren et al., 2017) for l_1-regularized sparse learning. Different from these existing DANE-type approaches for convex optimization, our method is designed for the ℓ_0-ERM problem with non-convex cardinality constraint.

Algorithm 1: Distributed Inexact Newton-type PurSuit (D-INPS)

Input : Loss functions $\{F_j(w)\}_{j=1}^m$, sparsity level k, parameter $\gamma \geq 0$ and $\eta > 0$.

Initialization $w^{(0)} = 0$ *or* $w^{(0)} \approx \arg\min_{\|w\|_0\leq k} F_1(w)$.

for $t = 1, 2, ...$ **do**

Compute $\nabla F(w^{(t-1)}) = \frac{1}{m}\sum_{j=1}^m \nabla F_j(w^{(t-1)})$ and broadcast it to all workers;

for *all the workers* $j = 1, ..., m$ *in parallel* **do**

(i) Construct a local objective function:

$$P_j(w; w^{(t-1)} \mid \eta, \gamma) := F_j(w) + \langle \eta \nabla F(w^{(t-1)}) - \nabla F_j(w^{(t-1)}), w\rangle + \frac{\gamma}{2}\|w - w^{(t-1)}\|^2, \quad (5)$$

(ii) Estimate a k-sparse vector $w_j^{(t)}$ such that for any $\bar{k}$-sparse $\bar{w}$ with $\bar{k} \leq k$:

$$P_j(w_j^{(t)}; w^{(t-1)} \mid \eta, \gamma) \leq P_j(\bar{w}; w^{(t-1)} \mid \eta, \gamma) + \epsilon;$$

end

Compute $w^{(t)} = \mathrm{H}_k\left(\frac{1}{m}\sum_{j=1}^m w_j^{(t)}\right)$.

end

Output : $w^{(t)}$.

Concerning initialization, the simplest way is to set $w^{(0)} = 0$, i.e., starting the iteration from scratch. Since the data samples are assumed to be evenly and randomly distributed over machines, another reasonable option is to initialize with one of the local minimizers, say $w^{(0)} \approx \arg\min_{\|w\|_0\leq k} F_1(w)$, which is expected to be close to the global solution.

3 Analysis for Convex Functions

In this section, we analyze the rate-of-convergence performance of DINPS for convex objective functions.

3.1 Preliminaries

We start by introducing the concept of restricted strong convexity and smoothness which are conventionally used in analyzing greedy pursuit methods (Shalev-Shwartz et al., 2010; Yuan et al., 2014; Jain et al., 2014).

Definition 1 (Restricted Strong Convexity/Smoothness)**.** *For any integer $s > 0$, we say $f(w)$ is restricted μ_s-strongly-convex and L_s-smooth if $\frac{\mu_s}{2}\|w - w'\|^2 \leq f(w) - f(w') - \langle \nabla f(w'), w - w'\rangle \leq \frac{L_s}{2}\|w - w'\|^2$ holds for any $\forall w, w'$*

satisfying $\|w - w'\|_0 \leq s$.

We next introduce the concept of restricted Lipschitz continuous gradient and Hessian which characterizes the continuity of the gradient vector and Hessian matrix over sparse subspaces. To simplify the notation, we will use abbreviations $\nabla_S f := [\nabla f]_S$ and $\nabla^2_{SS} f := [\nabla^2 f]_{SS}$.

Definition 2 (Restricted Lipschitz Gradient/Hessian)**.** *We say $f(w)$ has Restricted Lipschitz Gradient with constant $\alpha_s \geq 0$ (or α_s-RLG) if $\|\nabla_S f(w) - \nabla_S f(w')\| \leq \alpha_s \|w - w'\|$ holds for all w, w' with $\|w - w'\|_0 \leq s$ and $S = supp(w) \cup supp(w')$. Moreover, suppose that $f(w)$ is twice continuously differentiable. We say $f(w)$ has Restricted Lipschitz Hessian with constant $\beta_s \geq 0$ (or β_s-RLH) if $\left\|\nabla^2_{SS} f(w) - \nabla^2_{SS} f(w')\right\| \leq \beta_s \|w - w'\|$.*

The RLH property of logistic loss function. Consider the logistic loss $f(w) = \frac{1}{n}\sum_{i=1}^n \log(1+\exp(-2y_i w^\top x_i))$ for some $y = (y_i) \in \{-1, +1\}^n$ and $X^n = (x_i) \in \mathbb{R}^{n\times p}$. We need to access the gradient and Hessian of the logistic loss $f(w)$. Let $\sigma(z) = 1/(1+\exp(-z))$ be the sigmoid function. It is easy to show that the gradient $\nabla f(w) = Xa(w)/n$ where $a(w) \in \mathbb{R}^n$ with $[a(w)]_i = -2y_i(1 - \sigma(2x_i w^\top u_i))$; and the Hessian $\nabla^2 f(w) = X\Lambda(w)X^\top/n$ where $\Lambda(w)$ is an $n \times n$ diagonal matrix whose diagonal entries $[\Lambda(w)]_{ii} = 4\sigma(2v_i w^\top u_i)(1-\sigma(2v_i w^\top u_i))$. The following proposition shows that the logistic loss has RLH. See Appendix B.1 for a proof of this result.

Proposition 1. *Given a cardinality number s. Assume that $\|[x_i]_s\| \leq r_s$ holds for all x_i. Let $\Sigma_n = \frac{1}{n}XX^\top$ be the sample covariance matrix. Then the logistic loss $f(w)$ has β_s-RLH with $\beta_s = 24 r_s \rho_s^{\max}(\Sigma_n)$.*

3.2 Results for quadratic objective functions

We first consider a special case where $F(w)$ is quadratic with RLH strength parameter $\beta_s \equiv 0$ for all s. The widely applied sparse least square regression model belongs to this case. We need in our analysis the concept of sparse largest/smallest eigenvalue of a square matrix.

Definition 3 (Sparse Largest/Smallest Eigenvalues)**.** *Let $H \in \mathbb{R}^{p\times p}$ be a square matrix. we define the largest s-sparse eigenvalue of H as $\rho_s^{\max}(H) = \max_{w\in\mathbb{R}^p}\left\{w^\top H w \mid \|w\|_0 \leq s, \|w\| = 1\right\}$, and the s-mallest s-sparse eigenvalue of H as $\rho_s^{\min}(H) = \min_{w\in\mathbb{R}^p}\left\{w^\top H w \mid \|w\|_0 \leq s, \|w\| = 1\right\}$.*

The following is a *deterministic* result on the sparse parameter estimation error of DINPS when the objective function $F(w)$ is quadratic.

Theorem 1. *Let $\bar{w}$ be a $\bar{k}$-sparse target vector with $\bar{k} \leq k$. Assume that each component $F_j(w)$ is quadratic with a Hessian matrix H_j and $\rho_{3k}^{\min}(H_j) \geq \mu_{3k} > 0$. Let $H = \frac{1}{m}\sum_{j=1}^m H_j$. Assume that $\max_j \|H_j - \eta H\| \leq \frac{\theta\mu_{3k}}{3.24}$ for some $\theta \in (0,1)$ and $\epsilon \leq \frac{k\eta^2\|\nabla F(\bar{w})\|_\infty^2}{5.29\mu_{3k}}$. Set $\gamma = 0$. Then Algorithm 1 will output solution $w^{(t)}$ satisfying*

$$\|w^{(t)} - \bar{w}\| \leq \frac{7.62\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}{(1-\theta)\mu_{3k}}$$

after $t \geq \frac{1}{1-\theta}\log\left(\frac{\mu_{3k}\|w^{(0)} - \bar{w}\|}{\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)$ rounds of iteration.

Proof. A proof of this result is given in Appendix B.2. □

The result established in Theorem 1 shows that under proper conditions: 1) the estimation error of DINPS is controlled by the multiplier of $\sqrt{k}\|\nabla F(\bar{w})\|_\infty$ which usually represents the optimal statistical error in high dimensional learning models; and 2) it enjoys a linear rate of convergence before moving into the error region.

We now turn to a *stochastic* setting where the samples are uniformly randomly distributed over m machines. The following lemma, which is based on a matrix concentration bound (Tropp, 2012), shows that the Hessian H_j is close to H when the sample size of each machine j is sufficiently large. The same result appears in (Shamir et al., 2014).

Lemma 1. *Assume that $\|\nabla^2 f(w^\top x_{ji}, y_{ji})\| \leq L$ holds for all $j \in [m]$ and $i \in [n]$. Let $H_j = \frac{1}{n}\sum_{i=1}^n \nabla^2 f(w^\top x_{ji}, y_{ji})$ and $H = \frac{1}{m}\sum_{j=1}^m H_j$. Then for each j, with probability at least $1 - \delta$ over the samples,*

$$\max_j \|H_j - H\| \leq \sqrt{\frac{32L^2\log(mp/\delta)}{n}}.$$

Equipped with Lemma 1, we are able to derive the following result as a specialization of Theorem 1 to the considered stochastic setting.

Corollary 1. *Let $\bar{w}$ be a $\bar{k}$-sparse target vector with $\bar{k} \leq k$. Assume that the samples are uniformly randomly distributed on m machines and the conditions in Theorem 1 hold. Assume $\|\nabla^2 f(w^\top x_{ji}, y_{ji})\| \leq L$ holds for all $j \in [m]$ and $i \in [n]$. Set $\gamma = 0$ and $\eta = 1$. For any $\delta \in (0,1)$, if $n > \frac{336L^2\log(mp/\delta)}{\mu_{3k}^2}$, then with probability at least $1 - \delta$, Algorithm 1 will output solution $w^{(t)}$ satisfying*

$$\|w^{(t)} - \bar{w}\| \leq \frac{7.62\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}{(1-\theta)\mu_{3k}}$$

after $t \geq \frac{1}{1-\theta}\log\left(\frac{\mu_{3k}\|w^{(0)} - \bar{w}\|}{\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)$ rounds of iteration with $\theta = \frac{L}{\mu_{3k}}\sqrt{\frac{336\log(mp/\delta)}{n}} < 1$.

Proof. See Appendix B.2 for a proof of this corollary. □

The main message conveyed by Corollary 1 is that for stochastic quadratic minimization, the contraction factor θ can be arbitrarily small given that the sample size $n =$

$\mathcal{O}\left(\frac{L^2\log(mp)}{\mu_{3k}^2}\right)$ is sufficiently large. This sample size complexity is superior to the corresponding $n = \mathcal{O}\left(\frac{k^2L^2\log p}{\mu_{3k}^2}\right)$ complexity established in (Wang et al., 2017; Ren et al., 2017) for ℓ_1-regularized sparse linear regression models.

3.3 Results for objective functions with RLH

Let us now consider the more general setting where the objective functions are twice differentiable with RLH. The following is a deterministic result on sparse parameter estimation error of DINPS in the considered setting.

Theorem 2. *Let $\bar{w}$ be a $\bar{k}$-sparse target vector with $\bar{k} \leq k$. Let $\bar{H}_j = \nabla^2 F_j(\bar{w})$ and $\bar{H} = \frac{1}{m}\sum_{j=1}^m \bar{H}_j$. Assume that: **(a)** $F_j(w)$ is μ_{3k}-strongly-convex and has β_{3k}-RLH; **(b)** $\max_j \|\bar{H}_j - \eta\bar{H}\| \leq \frac{\theta\mu_{3k}}{6.48}$ for some $\theta \in (0,1)$, $\|\nabla F(\bar{w})\|_\infty \leq \frac{\theta(1-\theta)\mu_{3k}^2}{21.45\eta(1+\eta)\beta_{3k}\sqrt{k}}$, and $\epsilon \leq \frac{k\eta^2\|\nabla F(\bar{w})\|_\infty^2}{5.29\mu_{3k}}$; **(c)** $\|w^{(0)} - \bar{w}\| \leq \frac{\theta\mu_{3k}}{3.24(1+\eta)\beta_{3k}}$. Set $\gamma = 0$. Then Algorithm 1 will output $w^{(t)}$ satisfying*

$$\|w^{(t)} - \bar{w}\| \leq \frac{7.62\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}{(1-\theta)\mu_{3k}}$$

after $t \geq \frac{1}{1-\theta}\log\left(\frac{\mu_{3k}\|w^{(0)}-\bar{w}\|}{\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)$ rounds of iteration.

Proof. A proof of this result is given in Appendix B.3. □

Given that $w^{(0)}$ is properly initialized and the gradient infinity norm $\|\nabla F(\bar{w})\|_\infty$ is sufficiently small, the estimation error of DINPS for RLH objectives is controlled by the multiplier of $\sqrt{k}\|\nabla F(\bar{w})\|_\infty$ which typically represents the optimal statistical error in sparse learning models; and the rate of convergence towards this error level is linear.

As a direct consequence of Theorem 2, if we further assume $\bar{w}_{\min} > \frac{7.62\eta\sqrt{k}\|\nabla F(\bar{w})\|_\infty}{(1-\theta)\mu_{3k}}$, then *support recovery* $\mathrm{supp}(w^{(t)}) \supseteq \mathrm{supp}(\bar{w})$ can be guaranteed at $w^{(t)}$.

Stochastic result. By plugging Lemma 1 to Theorem 2 we obtain the following stochastoc result of DINPS for objective functions with RLH.

Corollary 2. *Let $\bar{w}$ be a $\bar{k}$-sparse target vector with $\bar{k} \leq k$. Assume that the samples are uniformly randomly distributed on m machines and the conditions in Theorem 2 and Lemma 1 hold. Set $\gamma = 0$ and $\eta = 1$. For any $\delta \in (0,1)$, if $n > \frac{1344L^2\log(mp/\delta)}{\mu_{3k}^2}$, then with probability at least $1-\delta$, Algorithm 1 will output $w^{(t)}$ satisfying*

$$\|w^{(t)} - \bar{w}\| \leq \frac{7.62\sqrt{k}\|\nabla F(\bar{w})\|_\infty}{(1-\theta)\mu_{3k}}$$

after $t \geq \frac{1}{1-\theta}\log\left(\frac{\mu_{3k}\|w^{(0)}-\bar{w}\|}{\sqrt{k}\|\nabla F(\bar{w})\|_\infty}\right)$ rounds of iteration with $\theta = \frac{L}{\mu_{3k}}\sqrt{\frac{1344\log(mp/\delta)}{n}} < 1$.

Proof. See Appendix B.3 for a proof of this corollary. □

Corollary 2 shows that when objective functions have RLH, provided that sample size $n = \mathcal{O}\left(\frac{L^2\log(mp)}{\mu_{3k}^2}\right)$ is sufficiently large, the contraction factor θ can be well controlled to remove the dependency on condition number L/μ_{3k}. This sample complexity improves the corresponding $n = \mathcal{O}\left(\frac{k^2L^2\log p}{\mu_{3k}^2}\right)$ bound presented in (Wang et al., 2017; Ren et al., 2017) for distributed Lasso-type program.

On local initialization. The iteration complexity results established in Theorem 2 and Corollary 2 rely on the initialization error $\|w^{(0)} - \bar{w}\|$. Let us consider an ideal local initialization strategy of $w^{(0)} = \arg\min_{\|w\|_0 \leq k} F_1(w)$. If the component $F_1(w)$ is μ_{3k}-strongly convex then it can be verified that $\|w^{(0)} - \bar{w}\| \leq \frac{2.84\sqrt{k}\|\nabla F_1(\bar{w})\|_\infty}{\mu_{3k}}$. By plugging this error bound to Corollary 2, the iteration complexity of DINPS for RLH objectives can be bounded from above by

$$\mathcal{O}\left(\frac{1}{1-\frac{L}{\mu_s}\sqrt{\frac{\log(mp)}{n}}}\log\left(\frac{\|\nabla F_1(\bar{w})\|_\infty}{\|\nabla F(\bar{w})\|_\infty}\right)\right). \tag{6}$$

In the following example, we will show that the term $\log\left(\frac{\|\nabla F_1(\bar{w})\|_\infty}{\|\nabla F(\bar{w})\|_\infty}\right)$ scales as $\log(m)$ in logistic regression.

Implications for distributed sparse logistic regression. As an example, we briefly discuss the implications of our results for distributed sparse logistic regression models. The logistic loss over data D_j is defined as $F_j(w) = \frac{1}{n}\sum_{i=1}^n \log\left(1+\exp(-y_{ji}w^\top x_{ji})\right)$. Let $F(w) = \frac{1}{m}\sum_{j=1}^m F_j(w)$ be the average of local loss. From Proposition 1 we know that each local logistic loss has RLH. Suppose x_{ji} are sub-Gaussian with parameter σ. It is known that $\|\nabla F(\bar{w})\|_\infty = \mathcal{O}\left(\sigma\sqrt{\log p/(mn)}\right)$ and $\|\nabla F_j(\bar{w})\|_\infty = \mathcal{O}\left(\sigma\sqrt{\log p/n}\right)$ hold with high probability (Yuan et al., 2014). Then with the local initialization $w^{(0)} \approx \arg\min_{\|w\|_0 \leq k} F_1(w)$, the bound in (6) suggests that DINPS essentially needs $O(\log m)$ rounds of iteration/communication to reach the statistical error barrier $\mathcal{O}\left(\sigma\sqrt{k\log p/(mn)}\right)$.

4 Analysis for Non-Convex Functions

We now turn to study the case when $F(w)$ is non-convex which is of particular interest to deep learning. To analyze the global convergence behavior, we follow the convention to use the value $\|\nabla F(w)\|^2$ as a measurement of quality for approximate stationary solutions, keeping in mind that the estimation error criterion for convex problems is not applicable due to the hardness of non-convex problems (Reddi et al., 2016a). For our global analysis, we make two slight modifications of Algorithm 1 to adapt to non-convexity: i) estimate a k-sparse vector $w_j^{(t)}$ such

that $\|\nabla P_j(w_j^{(t)}; w^{(t-1)} \mid \eta^{(t)}, \gamma)\| \leq \epsilon$; and ii) update $w^{(t)} = w_1^{(t)}$, that is, we always set $w^{(t)}$ as the local solution of the first (or alternatively any other fixed) machine.

Theorem 3. *Assume that for all j, $F_j(w)$ is L_{2k}-smooth. Set $\gamma = (\eta+2)L_{2k}$. Then*

$$\min_{1\leq\tau\leq t} \|\nabla F(w^{(\tau)})\|^2 \leq \left(\frac{8(\eta+3)^2 L_{2k}(F(w^{(0)}) - F(w^*))}{\eta}\right)\frac{1}{t} + \frac{18(\eta+3)^2}{\eta^2}\epsilon^2,$$

where $F(w^*) = \min_{\|w\|_0\leq k} F(w)$.

Proof. A proof of this result is given in Appendix C. □

Remark 1. *To our knowledge, Theorem 3 is the first convergence result for IHT-style methods with non-convex objective functions. The precision barrier $\mathcal{O}(\epsilon^2)$ appeared in the bound is introduced by the local sparse solution whose gradient is generally non-vanishing. In the extreme case of dense learning where the cardinality constraint is inactive, the local solution precision ϵ can be arbitrarily small. This leads to a sub-linear convergence rate for the original DANE method with non-convex objective functions, which matches the bound established in (Reddi et al., 2016b).*

5 Other Related Work

ℓ_0-Minimization methods. Among numerous methods designed for the ℓ_0-ERM problem (1) (Bahmani et al., 2013; Liu et al., 2014), the IHT-style methods (Yuan et al., 2014; Jain et al., 2014, 2016) have gained significant interests and they have been witnessed to offer the fastest and most scalable solutions in many cases. The stochastic and variance reduction variants of IHT were developed to make the algorithm more efficient to handle large-scale data (Li et al., 2016; Nguyen et al., 2017; Zhou et al., 2018). More recently, a duality theory of ℓ_0-ERM along with a dual coordinate ascent based IHT algorithm was investigated in (Liu et al., 2017). In addition to these first-order sparsity recover methods, several Newton-type second-order greedy pursuit methods were proposed in (Yuan & Liu, 2014; Chen & Gu, 2017) to achieves faster rate of convergence.

Distributed optimization. A straight-forward single-iteration distributed estimation approach is averaging estimators locally optimized by different machines (Zinkevich et al., 2010; Huang & Huo, 2015). Although simple for implementation, it was shown in (Shamir et al., 2014) that such a one-shot estimator can be unsatisfactory in minimizing the population objective. Another popular class of distributed estimators are generated by distributed implementation of first-order multi-round approaches (Boyd et al., 2011; Shamir & Srebro, 2014). The iteration complexity, and so also the communication complexity, of these distributed estimators usually has strong dependence on the conditioning of problem. There is a recent trend to study the so called communication-efficient distributed learning methods (Jaggi et al., 2014; Jordan et al., 2018). A main theme of these methods is to reduce the number of inter-machine communication rounds through designing more aggressive and balanced local computation schemes. Recently, a family of distributed approximate/inexact Newton-type methods (Shamir et al., 2014; Zhang & Lin, 2015; Reddi et al., 2016b; Wang et al., 2018) has become popular for communication-efficient learning, due to their milder dependence on the condition number when the local problems are sufficiently correlated to the global one. The distributed algorithm for network, such as gossip algorithms are proposed in (Boyd et al., 2006; Colin et al., 2016).

6 Experiments

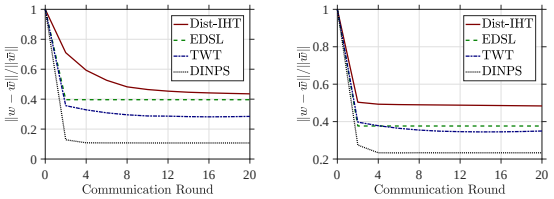
In this section, we present empirical results of DINPS on a number of synthetic and real-world sparse learning problems, including sparse linear/logistic regression, sparse bilinear regression and deep neural nets pruning. The considered algorithms are implemented with C++ and tested on multiple machines with 3.0GHz CPU interconnected by Ethernet. The machine communication interface is implemented by parameter server (Li et al., 2014).

6.1 Sparse linear regression

We first compare DINPS with distributed IHT (Dist-IHT) (Patterson et al., 2014), efficient distributed sparse learning (EDSL) (Wang et al., 2017) and two-way trauncation (TWT) (Ren et al., 2017) on simulated sparse linear regression tasks. Recollect that EDSL and TWT are DANE-type distributed computing methods for solving the Lasso-type estimation problem. A synthetic $N \times p$ design matrix is generated with each data sample x_i drawn from Gaussian distribution $\mathcal{N}(0, \Sigma)$ with $\Sigma_{j,k} = 1$ if $j = k$, and $1.1^{-\frac{|j-k|}{\gamma}}$ otherwise. A $\bar{k}$-sparse model parameter $\bar{w} \in \mathbb{R}^p$ is generated with the top $\bar{k}$ entries uniformly randomly valued in interval $(0, 1)$ and all the other entries set to be zero. The response variables $\{y_i\}_{i=1}^N$ are generated by $y_i = \bar{w}^\top x_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, 1)$.

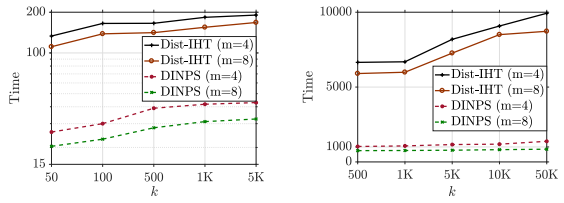
We fix the training sample size to be $N = 5 \times 10^3$, $p = 10^4$, $\bar{k} = 100$, the number of machines to be $m = 8$ and vary the value of γ to be 2 and 8. The convergence is measured by relative estimation error $\|w - \bar{w}\|/\|\bar{w}\|$. The algorithm hyper-parameters are tuned by grid search for optimal performance.

The convergence curves of the considered algorithms with respect to round of communication are shown in Figure 1. From these curves we can observe: 1) DINPS, EDSL and TWT converge quickly after a few rounds of master-worker communication, while Dist-IHT method needs substantially more rounds of communication to reach the comparable ac-



(a) $\gamma = 2$. (b) $\gamma = 8$.

Figure 1: Simulation study on sparse linear regression: communication efficiency comparison with varying γ values.



(a) rcv1. (b) kdd2010-algebra.

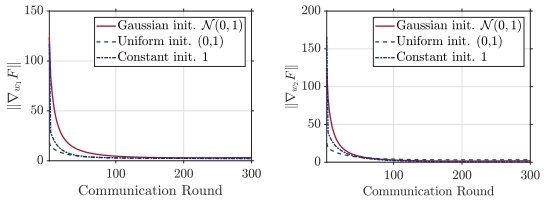
Figure 2: Real-data experiments on sparse logistic regression: computation time (in second) comparison.

curacy of DINPS; 2) When convergence is attained, DINPS outputs more accurate sparse solution than EDSL and TWT, mainly because DINPS directly works on the cardinality-constrained formulation while the EDSL and TWT are for a ℓ_1-norm relaxed formulation which tends to introduce bias in sparse learning. In conclusion, DINPS simultaneously achieves higher communication efficiency and model estimation accuracy than the two state-of-the-art methods.

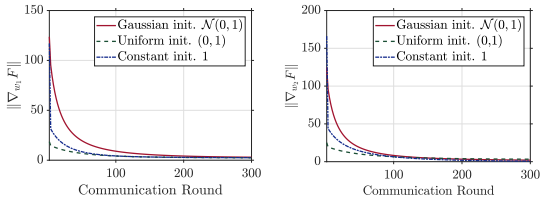
6.2 Sparse ℓ_2-regularized logistic regression

Next we evaluate the performance of DINPS in sparse ℓ_2-regularized binary logistic regression tasks. We compare the training time of DINPS with Dist-IHT on two real-world datasets: rcv1 ($N = 6 \times 10^5, p \approx 4.7 \times 10^5$) and kdd2010-algebra ($N \approx 8 \times 10^6, p \approx 2 \times 10^7$). For both datasets, the training samples are evenly distributed onto $m = 4$ and 8 machines, and the ℓ_2-regularization strength is set as 10^{-5}.

Figure 2 shows the computation time of algorithms under varying sparsity level $k \in \{0.05, 0.1, 0.5, 1, 5\} \times 10^3$ for rcv1 and $k \in \{0.05, 0.1, 5, 1, 5\} \times 10^4$ for kdd2010-algebra, with number of machines $m = 4$ and 8. For any sparsity level, we first run Dist-IHT until it reaches a sub-optimality $|F(w^{(t)}) - F(w^{(t-1)})|/|F(w^{(t)})| \leq 10^{-4}$ or maximum number of iteration, and then record the running time of DINPS with different machine number m to reach the same level of sub-optimality. Each model training is repeated 5 times to calculate the average computation time. It can be clearly seen that DINPS is consistently more efficient than Dist-IHT in a wide range of sparsity level and number of machines.



(a) $m = 4$.



(b) $m = 8$.

Figure 3: Simulation study on sparse bilinear regression: convergence curves of $\|\nabla_{w_j} F\|$, $j = 1, 2$, with respect to communication round for $m = 4, 8$ under different initialization schemes.

We further compare the training loss values of DINPS, EDSL and TWT evaluated on the considered data sets with $k = \{100, 1000\}$ and $m = \{2, 4, 8\}$. For all these algorithms, we set the termination condition as $|F(w^{(t)}) - F(w^{(t-1)})|/|F(w^{(t)})| \leq 10^{-4}$ and average the training loss over 5 data splits. Table 1 and Table 2 respectively show the results of the considered algorithms on rcv1 and kdd2010-algebra. It is observable that DINPS slightly outperforms EDSL and TWT in training accuracy.

6.3 Sparse bilinear regression

This is a simulated experiment to verify our convergence analysis of DINPS for non-convex functions. Here we consider a non-convex regression problem in which the training samples $\{X_i, y_i\}_{i=1}^N, X_i \in \mathbb{R}^{p_1 \times p_2}, y_i \in \mathbb{R}$ are generated according to a bilinear model $y_i = \bar{w}_1^\top X_i \bar{w}_2 + \varepsilon_i$, where $\bar{w}_1 \in \mathbb{R}^{p_1}$ and $\bar{w}_2 \in \mathbb{R}^{p_2}$ are two sparse vectors whose non-zero entries are uniformly randomly drawn from interval (0,1), $X_i \sim \mathcal{N}(0, I)$ and $\varepsilon_i \sim \mathcal{N}(0, 0.5)$. The objective is to minimize $F(w_1, w_2) := \frac{1}{2N} \sum_{i=1}^N \|y_i - w_1^\top X_i w_2\|^2$ with constraint $\|w_1\|_0 \leq k_1, \|w_2\|_0 \leq k_2$. We test with $p_1 = 40$, $\|\bar{w}_1\|_0 = 20$, $p_2 = 20$, $\|\bar{w}_1\|_0 = 10$, $k_1 = \|\bar{w}_1\|_0$, $k_2 = \|\bar{w}_2\|_0$ and $N = 10^4$.

We study the global convergence of DINPS under three different schemes for initializing the entries of $w_1^{(0)}$ and $w_2^{(0)}$: (1) Gaussian random initialization $\mathcal{N}(0, 1)$, (2) uniform random initialization $(0, 1)$, and (3) constant initialization 1. For solving the local ℓ_0-minimization problem (5), we alternately optimize w_1 and w_2 using IHT. The convergence curves of $\|\nabla_{w_1} F\|$ and $\|\nabla_{w_2} F\|$ with respect to round of communication are plot in Figure 3 for machine number $m = 4, 8$ under different initialization schemes. From this

	$k = 100$			$k = 1K$		
	EDSL	TWT	DINPS	EDSL	TWT	DINPS
$m = 2$	0.3237	0.2820	**0.2709**	0.2201	0.1823	**0.1551**
$m = 4$	0.3255	0.2828	**0.2717**	0.2225	0.1842	**0.1554**
$m = 8$	0.3298	0.2830	**0.2723**	0.2236	0.1861	**0.1555**

Table 1: Sparse ℓ_2-regularized logistic regression: model training loss comparison on rcv1.

	$k = 100$			$k = 1K$		
	EDSL	TWT	DINPS	EDSL	TWT	DINPS
$m = 2$	0.3959	0.3832	**0.3709**	0.3503	0.3422	**0.3314**
$m = 4$	0.4049	0.3874	**0.3712**	0.3521	0.3460	**0.3347**
$m = 8$	0.4060	0.3902	**0.3723**	0.3526	0.3463	**0.3356**

Table 2: Sparse ℓ_2-regularized logistic regression: model training loss comparison on kdd2010-algebra.

group of curves we can see that the ℓ_2-norm of parameter gradient converges quickly to a stable state after sufficient communication among machines, which is consistent with the theoretical results established in Theorem 3.

6.4 Sparse deep neural networks

Finally, we apply DINPS to distributed training of convolutional neural networks under layer-wise sparsity constraint over neuron connections[1]. Such sparse neural networks have recently been shown to be able to efficiently compress model size without sacrificing accuracy such as in image classification problems (Han et al., 2015; Jin et al., 2016; Wen et al., 2016). In our experiment, we test with LeNet3 (LeCun et al., 1998) on mnist digit dataset and VGG16 (Simonyan & Zisserman, 2014) on cifar10 dataset[2]. For both networks, we prune 50% of the parameters in convolutional layers and 80% of the parameters in fully connected layers. To initialize DINPS, we train a dense network by applying the Federated-Averaging (FedAvg) method (McMahan et al., 2017) designed for distributed neural work training on the given data partition. For local processing, the IHT-style algorithm from (Jin et al., 2016) is adopted to prune the deep nets based on local data. We compare the sparse network output by DINPS against the dense network by FedAvg in prediction accuracy and model size. The experiment is replicated 5 times with average results reported. The considered algorithms are implemented on Apache MXNet platform and tested on a cluster of Nvidia K80 GPUs.

Table 3 lists the experimental results on $m = 2, 4, 8$ machines. It can be observed from these results that the sparse networks trained by the DINPS have quite competitive or even superior prediction accuracy to the dense nets obtained by FedAvg, while the former has much fewer model parameters than the latter. This set of empirical results confirm that

[1]Code is available at https://github.com/wanglezi/DINPS.

[2]We follow the network structure definition in https://github.com/chengyangfu/pytorch-vgg-cifar10

	mnist: LeNet3		cifar10: VGG16	
	FedAvg	DINPS	FedAvg	DINPS
$m = 2$	1.49	**1.43**	7.03	**6.96**
$m = 4$	1.51	**1.44**	7.52	**7.48**
$m = 8$	1.55	**1.46**	7.83	**7.67**
model size	266K	**53K**	15.24M	**7.46M**

Table 3: Distributed sparse neural nets training: validation set classification error (in $\%$) and model size comparison.

DINPS is an accurate and communication-efficient distributed optimization method for pruning deep neural networks.

7 Conclusion

We proposed DINPS as a Newton-type communication-efficient distributed computing method for non-convex sparse minimization under cardinality constraint. At each iteration, each worker machine inexactly solves an ℓ_0-constrained minimization problem constructed based on the local data and global gradient, followed by sparse parameter aggregation and map-reduce gradient computation on the master machine. For generic convex loss functions, the communication complexity of DINPS has been shown to scale logarithmically with respect to the number of machines and its required per-machine sample complexity is lower than the prior DANE-type sparse learning methods. For non-convex loss functions, we have established sub-linear rate of convergence for DINPS. Extensive empirical results confirmed our theoretical predictions and demonstrated the advantages of DINPS over the state-of-the-art methods.

Acknowledgements

Qingshan Liu is supported in part by Natural Science Foundation of China under Grant 61532009. Xiao-Tong Yuan is supported by Natural Science Foundation of China under Grant 61522308 and Grant 61876090, and by Tencent AI Lab Rhino-Bird Joint Research Program No.JR201801.

References

Bahmani, Sohail, Raj, Bhiksha, and Boufounos, Petros T. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(Mar):807–841, 2013. 6

Beck, Amir and Eldar, Yonina C. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, 2013. 1

Boyd, Stephen, Ghosh, Arpita, Prabhakar, Balaji, and Shah, Devavrat. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006. 6

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 2, 3, 6

Chen, Jinghui and Gu, Quanquan. Fast newton hard thresholding pursuit for sparsity constrained nonconvex optimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017. 6

Colin, Igor, Bellet, Aurélien, Salmon, Joseph, and Clémençon, Stéphan. Gossip dual averaging for decentralized optimization of pairwise functions. In *International Conference on Machine Learning*, 2016. 6

Han, Song, Pool, Jeff, Tran, John, and Dally, William. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 2015. 8

Huang, Cheng and Huo, Xiaoming. A distributed one-step estimator. *arXiv preprint arXiv:1511.01443*, 2015. 6

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*, 2014. 3, 6

Jain, Prateek, Tewari, Ambuj, and Kar, Purushottam. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, 2014. 1, 3, 6

Jain, Prateek, Rao, Nikhil, and Dhillon, Inderjit. Structured sparse regression via greedy hard-thresholding. *Neural Information Processing Systems*, 2016. 6

Jin, Xiaojie, Yuan, Xiao-Tong, Feng, Jiashi, and Yan, Shuicheng. Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423*, 2016. 8

Jordan, Michael I, Lee, Jason D, and Yang, Yun. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 2018. 6

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

Li, Mu, Andersen, David G, Smola, Alex J, and Yu, Kai. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, 2014. 2, 6

Li, Xingguo, Zhao, Tuo, Arora, Raman, Liu, Han, and Haupt, Jarvis. Stochastic variance reduced optimization for nonconvex sparse learning. In *International Conference on Machine Learning*, 2016. 6, 11

Liu, Bo, Yuan, Xiao-Tong, Wang, Lezi, Liu, Qingshan, and Metaxas, Dimitris N. Dual iterative hard thresholding: From non-convex sparse minimization to non-smooth concave maximization. In *International Conference on Machine Learning*, 2017. 6

Liu, Ji, Ye, Jieping, and Fujimaki, Ryohei. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. In *International Conference on Machine Learning*, pp. 503–511, 2014. 6

McMahan, H Brendan, Moore, Eider, Ramage, Daniel, Hampson, Seth, et al. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017. 8

Nguyen, N., Needell, D., and Woolf, T. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. 63(11):6869–6895, 2017. 6

Patterson, Stacy, Eldar, Yonina C, and Keidar, Idit. Distributed compressed sensing for static and time-varying networks. *IEEE Transactions on Signal Processing*, 62 (19):4931–4946, 2014. 2, 6

Reddi, Sashank J, Hefny, Ahmed, Sra, Suvrit, Poczos, Barnabas, and Smola, Alex. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, 2016a. 5

Reddi, Sashank J, Konečnỳ, Jakub, Richtárik, Peter, Póczós, Barnabás, and Smola, Alex. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016b. 2, 6

Ren, Jineng, Li, Xingguo, and Haupt, Jarvis. Communication-efficient algorithm for distributed sparse learning via two-way truncation. *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2017. 2, 3, 5, 6

Shalev-Shwartz, Shai, Srebro, Nathan, and Zhang, Tong. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20:2807–2832, 2010. 3

Shamir, Ohad and Srebro, Nathan. Distributed stochastic optimization and learning. In *52nd Annual Allerton*

Conference on Communication, Control, and Computing, 2014. 6

Shamir, Ohad, Srebro, Nati, and Zhang, Tong. Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, 2014. 2, 3, 4, 6

Shen, Jie and Li, Ping. A tight bound of hard thresholding. *The Journal of Machine Learning Research*, 18(1):7650–7691, 2017. 11

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

Tropp, Joel A. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. 4

Wang, Jialei, Kolar, Mladen, Srebro, Nathan, and Zhang, Tong. Efficient distributed learning with sparsity. In *International Conference on Machine Learning*, 2017. 2, 3, 5, 6

Wang, Shusen, Roosta-Khorasani, Farbod, Xu, Peng, and Mahoney, Michael W. Giant: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 2338–2348, 2018. 6

Wen, Wei, Wu, Chunpeng, Wang, Yandan, Chen, Yiran, and Li, Hai. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, 2016. 8

Yuan, Xiao-Tong and Liu, Qingshan. Newton greedy pursuit: A quadratic approximation method for sparsity-constrained optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 6

Yuan, Xiao-Tong, Li, Ping, and Zhang, Tong. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning*, 2014. 1, 3, 5, 6

Zhang, Yuchen and Lin, Xiao. Disco: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning*, 2015. 2, 6

Zhou, Pan, Yuan, Xiaotong, and Feng, Jiashi. Efficient stochastic gradient hard thresholding. In *Advances in Neural Information Processing Systems*, pp. 1988–1997, 2018. 6

Zinkevich, Martin, Weimer, Markus, Li, Lihong, and Smola, Alex J. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2010. 6