Optimal Distributed Learning of Linear Models with Two Rounds of Communication

Anonymous Authors[1]

Abstract

We present the OWA algorithm for distributed empirical risk minimization. OWA requires only two rounds of communication and has communication complexity comparable to the naive averaging distributed estimator. Given m machines, each with n data points, the error of OWA shrinks at the rate of $O(1/\sqrt{mn})$, which matches the optimal error rate of the single machine oracle. It is the first communication-efficient algorithm to achieve this error rate. Furthermore, this error rate holds under mild conditions. For example, OWA does not require the loss to be convex.

1. INTRODUCTION

Many datasets are too large to fit in the memory of a single machine. To analyze them, we must partition the data onto many machines and use distributed algorithms. Existing algorithms fall into one of two categories:

Interactive algorithms require many rounds of communication between machines. These algorithms often resemble standard iterative algorithms where each iteration is followed by a communication step. The appeal of interactive algorithms is that they enjoy the same statistical regret bounds as standard sequential algorithms. That is, given m machines and n data points per machine, interactive algorithms typically have error that decays as $O(1/\sqrt{mn})$. But, there are two downsides. First, these algorithms can be slow in practice because communication is the main bottleneck in modern distributed architectures. Second, these algorithms require special implementations and do not work with off-the-shelf statistics libraries provided by (for example) Python, R, and Matlab.

Non-interactive algorithms require only a single round of communication. They are significantly faster than interactive algorithms and easily implemented with standard libraries. The downside is worse regret bounds. All existing algorithms have worst case behavior where the error decays as $O(1/\sqrt{n})$, completely independent of the number of machines m. Recent work has shown that no non-interactive algorithm can achieve regret bounds comparable to an interactive one.

In this paper, we propose a *semi-interactive* distributed algorithm called *optimal weighted averaging* (OWA). OWA performs two rounds of communication, so it is not subject to the regret bounds of non-interactive algorithms and achieves the optimal error rate of $O(1/\sqrt{mn})$. In the second round of communication, the machines send only a small amount of data. So OWA retains the speed advantages of non-interactive estimators. Furthermore, OWA is easily implemented in a MapReduce architecture with standard packages. The implementation used in our experiments requires only a few dozen lines of Python.

The next section introduces notation and formally describes our problem setting. Section 3 describes the OWA algorithm. We take special care to show how the algorithm can be implemented with off-the-shelf optimizers. Section 4 compares OWA to existing distributed algorithms. We highlight how the analysis of existing algorithms requires more limiting assumptions than OWA's, and show in detail why existing non-interactive regret bounds do not apply to OWA. Section 5 provides a simple proof that OWA achieves the optimal $O(1/\sqrt{mn})$ regret under mild conditions. Section 6 shows experimentally that our algorithm performs well on synthetic and real world advertising data. We emphasize that our algorithm is robust to the strength of regularization, which is one of the reasons it performs well in practice.

2. PROBLEM SETTING

Let $\mathcal{Y} \subseteq \mathbb{R}$ be the space of response variables, $X \subseteq \mathbb{R}^d$ be the space of covariates, and $\mathcal{W} \subseteq \mathbb{R}^d$ be the parameter space. We assume a linear model where the loss of data point $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ given the parameter $\mathbf{w} \in \mathcal{W}$ is denoted by $\ell(y, \mathbf{x}^\mathsf{T}\mathbf{w})$. The optimal parameter vector is denoted by $\mathbf{w}^*$. We do not require that the model be correctly specified, nor do we require that ℓ be convex with respect to $\mathbf{w}$. Let $Z \subset \mathcal{X} \times \mathcal{Y}$ be a dataset of mn i.i.d. observations. Finally, let $r : \mathcal{W} \to \mathbb{R}$ be a regularization function (typically the L1 or L2 norm) and $\lambda \in \mathbb{R}$ be the regularization strength. Then the regularized empirical risk minimizer (ERM) is

$$\hat{\mathbf{w}}^{erm} = \operatorname*{arg\,max}_{\mathbf{w}\in\mathcal{W}} \sum_{(\mathbf{x},y)\in Z} \ell(y, \mathbf{x}^\mathsf{T}\mathbf{w}) + \lambda r(\mathbf{w}). \tag{1}$$

In the remainder of this paper, it should be understood that all ERMs are regularized.

Assume that the dataset Z has been partitioned onto m machines so that each machine i has dataset Z_i of size n, and all the Z_i are disjoint. Then each machine calculates the local ERM

$$\hat{\mathbf{w}}_i^{erm} = \operatorname*{arg\,max}_{\mathbf{w}\in\mathcal{W}} \sum_{(\mathbf{x},y)\in Z_i} \ell(y, \mathbf{x}^\mathsf{T}\mathbf{w}) + \lambda r(\mathbf{w}). \tag{2}$$

Solving for $\hat{\mathbf{w}}_i^{erm}$ requires no communication with other machines. Our goal is to merge the $\hat{\mathbf{w}}_i^{erm}$s into a single improved estimate. A baseline merging procedure is the averaging estimator:

$$\hat{\mathbf{w}}^{ave} = \frac{1}{m}\sum_{i=1}^{m} \hat{\mathbf{w}}_i^{erm}. \tag{3}$$

This estimator is well studied, and in Section 4 we compare this previous work to our own. Here we briefly recall that the quality of an estimator $\hat{\mathbf{w}}$ can be measured by the error $\|\hat{\mathbf{w}} - \mathbf{w}^*\|$. We can use the triangle inequality to decompose this error as

$$\|\hat{\mathbf{w}} - \mathbf{w}^*\| \le \|\hat{\mathbf{w}} - \mathbb{E}\hat{\mathbf{w}}\| + \|\mathbb{E}\hat{\mathbf{w}} - \mathbf{w}^*\|. \tag{4}$$

We refer to $\|\hat{\mathbf{w}} - \mathbb{E}\hat{\mathbf{w}}\|$ as the variance of an estimator and $\|\mathbb{E}\hat{\mathbf{w}} - \hat{\mathbf{w}}\|$ as the bias. The $\hat{\mathbf{w}}^{ave}$ estimator is known to have lower variance than the estimator $\hat{\mathbf{w}}_i^{erm}$ trained on a single machine, but the same bias. Our goal is to design an estimator that reduces both variance and bias.

3. THE OWA ESTIMATOR

We propose a modification to the averaging estimator called the *optimal weighted average* (OWA). OWA uses a second round of optimization to calculate the optimal linear combination of the $\hat{\mathbf{w}}_i^{erm}$s. This second optimization occurs over a small fraction of the dataset, so its computational and communication cost is negligible.

3.1. Warmup: the Full OWA

To motivate our estimator, we first present a less efficient estimator that uses the entire dataset for the second round of optimization. Define the matrix $\hat{W} : \mathbb{R}^{d\times m}$ to have ith column equal to $\hat{\mathbf{w}}_i^{erm}$. Now consider the estimator

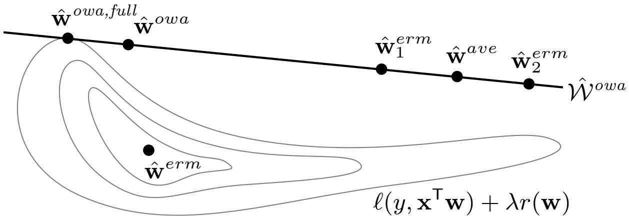
$\hat{\mathbf{w}}^{owa,full}$
$\hat{\mathbf{w}}^{owa}$
$\hat{\mathbf{w}}_1^{erm}$
$\hat{\mathbf{w}}^{ave}$
$\hat{\mathbf{w}}_2^{erm}$
$\hat{\mathcal{W}}^{owa}$
$\hat{\mathbf{w}}^{erm}$
$\ell(y, \mathbf{x}^\mathsf{T}\mathbf{w}) + \lambda r(\mathbf{w})$

Figure 1. OWA performs a second optimization to find the best parameter vector in $\hat{\mathcal{W}}^{owa}$. Since $\hat{\mathcal{W}}^{owa}$ has low dimension, this optimization needs relatively little data to ensure that with high probability $\hat{\mathbf{w}}^{owa}$ has lower empirical loss than $\hat{\mathbf{w}}^{ave}$.

$$\hat{\mathbf{w}}^{owa,full} = \hat{W}\hat{\mathbf{v}}^{owa,full}, \tag{5}$$

where

$$\hat{\mathbf{v}}^{owa,full} = \operatorname*{arg\,max}_{\mathbf{v}\in\mathbb{R}^m} \sum_{(\mathbf{x},y)\in Z} \ell\left(y, \mathbf{x}^\mathsf{T}\hat{W}\mathbf{v}\right) + \lambda r(\hat{W}\mathbf{v}). \tag{6}$$

Notice that $\hat{\mathbf{w}}^{owa,full}$ is just the empirical risk minimizer when the parameter space $\mathcal{W}$ is restricted to the subspace $\hat{\mathcal{W}}^{owa} = \operatorname{span}\{\hat{\mathbf{w}}_i^{erm}\}_{i=1}^m$. In other words, the $\hat{\mathbf{v}}^{owa,full}$ vector contains the optimal weights to apply to each $\hat{\mathbf{w}}_i^{erm}$ when averaging. Figure 1 shows graphically that no other estimator in $\hat{\mathcal{W}}^{owa}$ can have lower regularized empirical loss than $\hat{\mathbf{w}}^{owa,full}$.

3.2. The OWA Estimator

Calculating the weights $\hat{\mathbf{v}}^{owa,full}$ directly is infeasible because it requires access to the full dataset. Fortunately, we do not need to consider all the data points for an accurate estimator. The parameter space $\hat{\mathcal{W}}^{owa}$ is m-dimensional. So intuitively, we only need $O(m)$ data points to solve the second optimization to our desired accuracy. This intuition motivates the OWA estimator. Let $Z_i^{owa} \subset Z_i$ be a set of n^{owa} data points uniformly sampled from Z_i without replacement, and let Z^{owa} be the union of the Z_i^{owa}s. Then the OWA estimator is defined as

$$\hat{\mathbf{w}}^{owa} = \hat{W}\hat{\mathbf{v}}^{owa}, \tag{7}$$

where

$$\hat{\mathbf{v}}^{owa} = \operatorname*{arg\,max}_{\mathbf{v}\in\mathbb{R}^m} \sum_{(\mathbf{x},y)\in Z^{owa}} \ell\left(y, \mathbf{x}^\mathsf{T}\hat{W}\mathbf{v}\right) + \lambda r(\hat{W}\mathbf{v}). \tag{8}$$

We present two algorithms for calculating $\hat{\mathbf{w}}^{owa}$. Algorithm 1 uses only a single round of communication,

Algorithm 1 Calculating $\hat{\mathbf{w}}^{owa}$ in one round

```
Preconditions:
    each machine $i$ already has dataset $Z_i$
    the master machine additionally has $Z^{owa}$
Round 1, each machine $i$ independently:
    calculates $\hat{\mathbf{w}}_i^{erm}$ using (2)
    transmits $\hat{\mathbf{w}}_i^{erm}$ to the master
The master calculates $\hat{\mathbf{w}}^{owa}$ using (7) and (8)
```

Algorithm 2 Calculating $\hat{\mathbf{w}}^{owa}$ in two rounds

```
Preconditions:
    each machine $i$ already has dataset $Z_i$
Round 1, each machine $i$ independently:
    calculates $\hat{\mathbf{w}}_i^{erm}$ using (2)
    broadcasts $\hat{\mathbf{w}}_i^{erm}$ to all other machines
Round 2, each machine $i$ independently:
    constructs $\hat{W} = (\hat{\mathbf{w}}_1^{erm}, ..., \hat{\mathbf{w}}_m^{erm})$
    samples a dataset $Z_i^{owa} \subset Z_i$ of size $n^{owa}$
    calculates $Z_i^{proj} = \{(\mathbf{x}^\mathsf{T}\hat{W}, y) : (\mathbf{x}, y) \in Z^{owa}\}$
    sends $Z_i^{proj}$ to a master machine
The master calculates $\hat{\mathbf{w}}^{owa}$ using (7) and (8)
```

but requires that a predesignated master machine already have a copy of the Z^{owa} dataset. In the only round, each machine calculates $\hat{\mathbf{w}}_i^{erm}$ independently, then transfers the result to the master. A total of $O(dm)$ bits are transfered to the server. (The parameter vector has d dimensions and there are m machines.) The averaging estimator transfers the same information, and so has the same $O(dm)$ communication complexity. The only difference between the two algorithms is the way the master machine merges the local estimates.

If the master machine does not already have a copy of Z^{owa}, then we must transfer a copy to the master. Algorithm 2 exploits the structure of the $\hat{\mathbf{w}}^{owa}$ estimator to transmit this data efficiently. In the first round, each machine calculates $\hat{\mathbf{w}}_i^{erm}$ independently. The result is then broadcast to every other machine, instead of just the master. A total of $O(dm^2)$ bits are transmitted in this round. (The parameter vector has d dimensions, there are m machines, and each machine transmits to each other machine.) In the second round, each machine projects its local dataset Z_i^{owa} onto the space $\hat{\mathcal{W}}^{owa}$. These projected data points are then transmitted to the master. A total of $O(m^2 n^{owa})$ bits are transmitted. (The projected data points each have dimension m, there are m machines, and there are n^{owa} data points per machine.) Theorem 2 suggests that n^{owa} should be set to $O(mn/d)$. So the total data transmitted in both rounds is $O(dm^2 + m^3 n/d)$.

3.3. Implementing with Existing Optimizers

Equations 6 and 8 cannot be solved directly using off the shelf optimizers because existing optimizers do not support the non-standard regularization term $r(\hat{W}\mathbf{v})$. In practice, it is sufficient to approximate this term by L2 regularization directly on the $\mathbf{v}$ vector:

$$\lambda r(\hat{W}\mathbf{v}) \approx \lambda_2 \|\mathbf{v}\|^2, \tag{9}$$

where λ_2 is a new hyperparameter. We provide two justifications for this approximation:

1. When we want the parameter vector $\mathbf{w}$ to be sparse (and so the regularizer r is the L1 norm), we have no reason to believe that the $\mathbf{v}$ vector should be sparse. The desired sparsity is induced by the regularization when solving for $\hat{\mathbf{w}}_i^{erm}$s and maintained in any linear combination of the $\hat{\mathbf{w}}_i^{erm}$s.
2. As the size of the dataset increases, the strength of the regularizer decreases. In this second optimization, the dimensionality of the problem is small, so it is easy to add more data to make the influence of the regularization negligible.

The new λ_2 regularization parameter should be set by cross validation. This will be a fast procedure, however, because there are only $mn^{owa} \ll mn$ data points to optimize over, they have dimensionality $m \ll d$, and the L2 regularized problem is much easier to solve than the L1 problem. Furthermore, this cross validation can be computed locally on the master machine without any communication. We demonstrate empirically in Section 6 that the overhead of determining λ_2 is negligible.

4. RELATED WORK

There are two main categories of related work: alternative estimators, and bounds on the communication complexity of distributed learning.

4.1. Alternative Estimators

The simplest and most popular non-interactive estimator is the averaging estimator:

$$\hat{\mathbf{w}}^{ave} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{w}}_i^{erm}. \tag{10}$$

Previous analysis of $\hat{\mathbf{w}}^{ave}$ makes a number of limiting assumptions. McDonald et al. (2009) analyze $\hat{\mathbf{w}}^{ave}$ in the special case of L2 regularized maximum entropy models. They provide tail bounds on $\|\mathbf{w}^* - \hat{\mathbf{w}}^{ave}\|$, showing that the deviation $\|\mathbb{E}\hat{\mathbf{w}}^{ave} - \hat{\mathbf{w}}^{ave}\|$ reduces as $O((mn)^{-1/2})$, but that the bias $\|\mathbf{w}^* - \mathbb{E}\hat{\mathbf{w}}^{ave}\|$ reduces only as $O(n^{-1/2})$. Their analysis uses a martingale technique that requires the radius of the dataset be independent of the size of the dataset. This is a particularly limiting assumption as even the simple case of normally-distributed data does not satisfy it. Zhang et al. (2012) provide a more general analysis showing that the mean squared error (MSE) $\mathbb{E}\|\mathbf{w}^* - \hat{\mathbf{w}}^{ave}\|^2$ decays as $O((mn)^{-1} + n^{-2})$. This matches the optimal MSE of $\hat{\mathbf{w}}^{erm}$ whenever $m < n$. Their analysis also requires a number of technical assumptions. For example, they assume the parameter space $\mathcal{W}$ is bounded. This assumption does not hold under the standard Bayesian interpretation of L2 regularization as a Gaussian prior of the parameter space. They further make strong convexity and $8th$ order smoothness assumptions which guarantee that $\hat{\mathbf{w}}_i^{erm}$ is a "nearly unbiased estimator" of $\mathbf{w}^*$. Most recently, Rosenblatt and Nadler (2016) analyze $\hat{\mathbf{w}}^{ave}$ in the asymptotic regime as the number of data points $n \to \infty$. This analysis is more general than previous analyses, but it does not hold in the finite sample regime. Our analysis of OWA in Section 5 requires no assumptions of boundedness or convexity and holds in the finite sample regime.

Other research has focused on modifications to the $\hat{\mathbf{w}}^{ave}$ estimator to reduce bias. Zinkevich et al. (2010) show that if the training sets partially overlap each other (instead of being disjoint), then the resulting estimator will have lower bias. Lee et al. (2015) and Battey et al. (2015) independently develop techniques for debiasing L1-regularized problems. Zhang et al. (2012) provide a debiasing technique that works for any estimator. It works as follows. Let $r \in (0, 1)$, and Z_i^r be a bootstrap sample of Z_i of size rn. Then the bootstrap average estimator is

$$\hat{\mathbf{w}}^{boot} = \frac{\hat{\mathbf{w}}^{ave} - r\hat{\mathbf{w}}^{ave,r}}{1-r}, \tag{11}$$

where

$$\begin{aligned} \hat{\mathbf{w}}^{ave,r} &= \frac{1}{m}\sum_{i=1}^{m} \hat{\mathbf{w}}_i^{erm,r}, \\ \hat{\mathbf{w}}_i^{erm,r} &= \operatorname*{arg\,max}_{\mathbf{w}} \sum_{(\mathbf{x},y)\in Z_i^r} \ell(y, \mathbf{x}^\mathsf{T}\mathbf{w}) + \lambda r(\mathbf{w}). \end{aligned} \tag{12}$$

The intuition behind this estimator is to use the bootstrap sample to directly estimate and correct for the bias. This estimator enjoys a MSE that decays as $O((mn)^{-1} + n^{-3})$ under similar assumptions as their analysis of $\hat{\mathbf{w}}^{ave}$. There are two main limitations to $\hat{\mathbf{w}}^{boot}$. First, the optimal value of r is not obvious and setting the parameter requires cross validation on the entire data set. Our proposed $\hat{\mathbf{w}}^{owa}$ estimator has a similar parameter λ_2 that needs tuning, but this tuning happens on a small fraction of the data and always with the L2 regularizer. So properly tuning λ_2 is more efficient than r. Second, performing a bootstrap on an unbiased estimator increases the variance. This means that $\hat{\mathbf{w}}^{boot}$ could perform worse than $\hat{\mathbf{w}}^{ave}$ on unbiased estimators. Our $\hat{\mathbf{w}}^{owa}$ estimator, in contrast, will perform at least as well as $\hat{\mathbf{w}}^{ave}$ with high probability, as seen in Figure 1. In Section 6, we show that our estimator has better empirical performance.

Liu and Ihler (2014) propose a more Bayesian approach inspired by Merugu and Ghosh (2003). Instead of averaging the model's parameters, they directly "average the models" with the following KL-average estimator:

$$\hat{\mathbf{w}}^{kl} = \operatorname*{arg\,min}_{\mathbf{w}\in\mathcal{W}} \sum_{i=1}^{m} \mathrm{KL}\left(p(\cdot;\hat{\mathbf{w}}_i^{erm}) \,\middle\|\, p(\cdot;\mathbf{w})\right). \tag{13}$$

The minimization is performed via a bootstrap sample from the smaller models. This method has three main advantages. First, it is robust to reparameterizations of the model. Second, it is statistically optimal for the class of non-interactive optimization methods. (We show in the next section that this optimality bound does not apply to our $\hat{\mathbf{w}}^{owa}$ estimator due to our semi-interactive setting.) Third, this method is general enough to work for any model, whereas our proposed OWA method works only for linear models. The main downside of the KL-average is that the minimization has a prohibitively high computational cost. Let n^{kl} be the size of the bootstrap sample. Then the original implementation's MSE shrinks as $O((mn)^{-1} + (nn^{kl})^{-1})$. This implies that the bootstrap procedure requires as many samples as the original problem to get a MSE that shrinks at the same rate as the averaging estimator. Han and Liu (2016) provide a method to reduce this rate to $O((mn)^{-1} + (n^2 n^{kl})^{-1})$ using control variates, but the procedure remains prohibitively expensive. Their experiments show the procedure scaling only to datasets of size $mn \approx 10^4$, whereas our experiments involve a dataset of size $mn \approx 10^8$.

Surprisingly, Zhang et al. (2013b) show that in the special case of kernel ridge regression, a reduction in bias is not needed to have the MSE of $\hat{\mathbf{w}}^{ave}$ decay at the optimal sequential rate. By a careful choice of regularization parameter λ, they cause $\hat{\mathbf{w}}_i^{erm}$ to have lower bias but higher variance, so that the final estimate of $\hat{\mathbf{w}}^{ave}$ has both reduced bias and variance. This suggests that a merging procedure that reduces bias is not crucial to good performance if we set the regularization parameter correctly. Typically there is a narrow range of good regularization parameters, and finding a λ in this range is expensive computationally. We show experimentally in Section 6 that our method has significantly reduced sensitivity to λ. Therefore, it is computationally cheaper to find a good λ for our method than for the other methods discussed in this section.

4.2. Performance Bounds

Performance bounds come in two flavors: statistical and information theoretic. On the statistical side, Liu and Ihler (2014) show that for any non-interactive distributed estimator $\hat{\mathbf{w}}$, the quantity $\|\hat{\mathbf{w}} - \hat{\mathbf{w}}^{erm}\|^2$ decays as $\Omega(\gamma^2\mathcal{I}^{-1}/n^2)$. Here γ is the statistical curvature of the model and $\mathcal{I}$ is the Fisher information. Furthermore, they show that their KL-averaging estimator $\hat{\mathbf{w}}^{kl}$ matches this bound. One consequence of this bound is that no non-interactive learner can achieve the optimal $O(1/\sqrt{mn})$ error rate on models with nonzero statistical curvature. The vast majority of models used in practice have nonzero curvature. The only models with zero curvature are full exponential families. OWA is able to "break" this bound and achieve optimal error rate because of its semi-interactive setting. A crucial assumption of Liu and Ihler's analysis is that the merge function not depend on the data.

Shamir (2014), Zhang et al. (2013a), and Garg et al. (2014) all provide information theoretic lower bounds on the sample complexity of non-interactive learning problems. As above, however, their results are not applicable in our semi-interactive setting. Braverman et al. (2016) provide a bound that applies in all distributed settings. In particular, they show that the minimax optimal error rate for least squares regression requires $\Omega(m \cdot \min\{n, d\})$ bits of communication. This bound essentially matches the communication complexity of Algorithm 1.

5. ANALYSIS

In this section, we show that the statistical error $\|\mathbf{w}^* - \hat{\mathbf{w}}^{owa}\|$ decreases at the rate $O(1/\sqrt{mn})$. The proof is broken into two steps. First we show that $\hat{\mathcal{W}}^{owa}$ is a good subspace to optimize over in the sense that the distance between $\hat{\mathcal{W}}^{owa}$ and $\mathbf{w}^*$ is small. Then we show that $\hat{\mathbf{w}}^{owa}$ is a good parameter to choose within $\hat{\mathcal{W}}^{owa}$. Our analysis depends on estimators obeying the following mild condition.

The Sub-Gaussian Tail (SGT) Condition. *Let $\hat{\mathbf{w}}$ be a linear estimator trained on n data points of dimension d. Let $t > 0$. Then, with probability at least $1 - \exp(-t)$,*

$$\|\mathbf{w}^* - \hat{\mathbf{w}}\| \leq O\left(\sqrt{dt/n}\right). \tag{14}$$

The SGT condition is known to hold in many situations of interest. In the asymptotic regime when $n \to \infty$, very strong results of this form have been known since the 1960s. Chapter 7 of Lehmann (1999) provides an elementary introduction to this work. Lehman proves an asymptotic version of the SGT requiring only that ℓ be three times differentiable and that the data points be i.i.d.

Similar results hold in the non-asymptotic case $n < \infty$. The simplest results place distributional assumptions on the data. For example, Negahban et al. (2009) considers the case when the data points are sub-Gaussian, the likelihood satisfies a "restricted strong convexity condition," and the regularizer is decomposable. Their resulting theorems are actually much stronger than the SGT condition: They prove that the dependence on the dimension d in Equation 14 can be replaced by the number of non-zero elements in the optimal parameter vector $\mathbf{w}^*$. For sparse models, this is a major improvement. The strongest non-asymptotic results known to the authors are due to Spokoiny (2012). Spokoiny does not place any distributional assumption on the data, captures sparsity information through dependence on the Fisher information, and does not even require the data to be i.i.d. Spokoiny's only assumption is that the empirical loss admit a local approximation via the "bracketing device," which is a generalization of the Taylor expansion.

The following lemma is an easy consequence of the SGT condition. It formalizes the key idea that $\hat{\mathcal{W}}^{owa}$ is a good subspace to optimize over.

Lemma 1. *Assume the $\hat{\mathbf{w}}_i^{erm}$s satisfy the SGT condition. Let $t > 0$. Then with probability at least $1 - \exp(-t)$,*

$$\min_{\mathbf{w} \in \hat{\mathcal{W}}^{owa}} \|\mathbf{w} - \mathbf{w}^*\| \leq O(\sqrt{dt/mn}). \tag{15}$$

Proof. Using independence of the $\hat{\mathbf{w}}_i^{erm}$s and the SGT condition, we have that

$$\Pr\left[\min_{\mathbf{w}\in\hat{\mathcal{W}}^{owa}} \|\mathbf{w}-\mathbf{w}^*\| \le O(\sqrt{dt/mn})\right] \tag{16}$$

$$\ge \Pr\left[\min_{i=1\dots m} \|\hat{\mathbf{w}}_i^{erm}-\mathbf{w}^*\| \le O(\sqrt{dt/mn})\right] \tag{17}$$

$$=1-\Pr\left[\min_{i=1\dots m} \|\hat{\mathbf{w}}_i^{erm}-\mathbf{w}^*\| > O(\sqrt{dt/mn})\right] \tag{18}$$

$$=1-\left(\Pr\left[\|\hat{\mathbf{w}}_1^{erm}-\mathbf{w}^*\| > O(\sqrt{dt/mn})\right]\right)^m \tag{19}$$

$$=1-\left(1-\Pr\left[\|\hat{\mathbf{w}}_1^{erm}-\mathbf{w}^*\| \le O(\sqrt{dt/mn})\right]\right)^m \tag{20}$$

$$\ge 1-\left(1-(1-\exp(-t/m))\right)^m \tag{21}$$

$$=1-\exp(-t). \tag{22}$$

□

Next, we introduce a smoothness condition that will connect the $\mathbf{w}$ minimizing (15) to $\hat{\mathbf{w}}^{owa,full}$.

The Hessian Condition. *For any vector $\mathbf{w}$ satisfying $\|\mathbf{w}\| \le \|\mathbf{w}^* - \hat{\mathbf{w}}^{owa,full}\|$, we have that*

$$q_{lo}\|\mathbf{w}-\mathbf{w}^*\|^2 \le \mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{w}^*) \le q_{hi}\|\mathbf{w}-\mathbf{w}^*\|^2 \tag{23}$$

where $\mathcal{L}(\mathbf{w}) = \sum_{(\mathbf{x},y)\in Z} \ell(y;\mathbf{x}^\mathsf{T}\mathbf{w}) + \lambda r(\mathbf{w})$.

This condition is somewhat easier to understand in the asymptotic regime as $n \to \infty$. In this regime, we have that $\|\mathbf{w}^* - \hat{\mathbf{w}}^{owa,full}\| \to 0$, so (23) is equivalent to requiring that the condition number of the Hessian $\nabla^2\mathcal{L}(\mathbf{w}^*)$ be $q_{\text{hi}}/q_{\text{lo}}$.

We are now ready to bound the error of $\hat{\mathbf{w}}^{owa,full}$.

Theorem 1. *Assume the Hessian Condition and that the $\hat{\mathbf{w}}_i^{erm}$s satisfy the SGT condition. Let $t > 0$. Then with probability at least $1-\exp(-t)$,*

$$\|\hat{\mathbf{w}}^{owa,full} - \mathbf{w}^*\| \le O\left(\sqrt{\left(\frac{q_{hi}}{q_{lo}}\right)\left(\frac{dt}{mn}\right)}\right). \tag{24}$$

Proof. Let $\pi_{\hat{\mathcal{W}}^{owa}}\mathbf{w}^*$ denote the vector in $\hat{\mathcal{W}}^{owa}$ with minimum distance to $\mathbf{w}^*$. That is, $\pi_{\hat{\mathcal{W}}^{owa}}\mathbf{w}^*$ is the vector minimizing (15). Then by the Hessian Condition, we have that

$$q_{\text{lo}}\|\hat{\mathbf{w}}^{owa,full} - \mathbf{w}^*\|^2 \le \mathcal{L}(\hat{\mathbf{w}}^{owa,full}) - \mathcal{L}(\mathbf{w}^*) \tag{25}$$

$$\le \mathcal{L}(\pi_{\hat{\mathcal{W}}^{owa}}\mathbf{w}^*) - \mathcal{L}(\mathbf{w}^*) \tag{26}$$

$$\le q_{\text{hi}}\|\pi_{\hat{\mathcal{W}}^{owa}}\mathbf{w}^*\|^2. \tag{27}$$

And so

$$\|\hat{\mathbf{w}}^{owa,full} - \mathbf{w}^*\| \le \sqrt{\frac{q_{\text{hi}}}{q_{\text{lo}}}}\|\pi_{\hat{\mathcal{W}}^{owa}}\mathbf{w}^*\|. \tag{28}$$

The result follows by Lemma 1. □

Next, we show that if n^{owa} is set properly, then $\hat{\mathbf{w}}^{owa}$ and $\hat{\mathbf{w}}^{owa,full}$ have the same error bounds.

Theorem 2. *Assume the Hessian Condition and that both the $\hat{\mathbf{w}}_i^{erm}$s and $\hat{\mathbf{v}}^{owa}$ satisfy the SGT condition. Let $n^{owa} = mn/d$ and $t > 0$. Then we have with probability at least $1-\exp(-t)$,*

$$\|\hat{\mathbf{w}}^{owa} - \mathbf{w}^*\| \le O\left(\sqrt{\left(\frac{q_{hi}}{q_{lo}}\right)\left(\frac{dt}{mn}\right)}\right) \tag{29}$$

Proof. We have by the triangle inequality that

$$\|\hat{\mathbf{w}}^{owa} - \mathbf{w}^*\| \le \|\hat{\mathbf{w}}^{owa} - \hat{\mathbf{w}}^{owa,full}\| + \|\hat{\mathbf{w}}^{owa,full} - \mathbf{w}^*\| \tag{30}$$

Theorem 1 bounds the rightmost term above. Recall that the $\hat{\mathbf{v}}^{owa}$ estimator is trained on mn^{owa} data points of dimension m (see Equation 8). If we consider the true data distribution of this data to be the empirical distribution of sampling from Z, then the SGT condition for $\hat{\mathbf{v}}^{owa}$ states that

$$\|\hat{\mathbf{w}}^{owa} - \hat{\mathbf{w}}^{owa,full}\| \le O\left(\sqrt{mt/mn^{owa}}\right). \tag{31}$$

$$= O\left(\sqrt{t/n^{owa}}\right). \tag{32}$$

Substituting for n^{owa} gives the desired result. □

6. EXPERIMENTS

We evaluate OWA on two logistic regression tasks. The first task uses synthetic data. The second task uses real world ad-click data from the Tencent search engine. In each experiment, we compare $\hat{\mathbf{w}}^{owa}$ with four baseline estimators: the naive estimator using the data from only a single machine $\hat{\mathbf{w}}_i^{erm}$; the averaging estimator $\hat{\mathbf{w}}^{ave}$; the bootstrap estimator $\hat{\mathbf{w}}^{boot}$; and the oracle estimator of all data trained on a single machine $\hat{\mathbf{w}}^{erm}$. The $\hat{\mathbf{w}}^{boot}$ estimator has a parameter r that needs to be tuned. In all experiments we evaluate $\hat{\mathbf{w}}^{boot}$ with $r \in \{0.005, 0.01, 0.02, 0.04, 0.1, 0.2\}$, which is a set recommended in the original paper (Zhang et al., 2012), and then report only the value of r with highest true likelihood. Thus we are reporting an overly optimistic estimate of the performance of $\hat{\mathbf{w}}^{boot}$, and as we shall see $\hat{\mathbf{w}}^{owa}$ still tends to perform better.

6.1. Synthetic Data

We generate the data according to the following sparse logistic regression model. Each component of the true parameter vector $\mathbf{w}^*$ is sampled i.i.d. from a spike and slab distribution. With probability 0.9, the component is 0; with probability 0.1, the component is sampled from a standard normal distribution. The data points

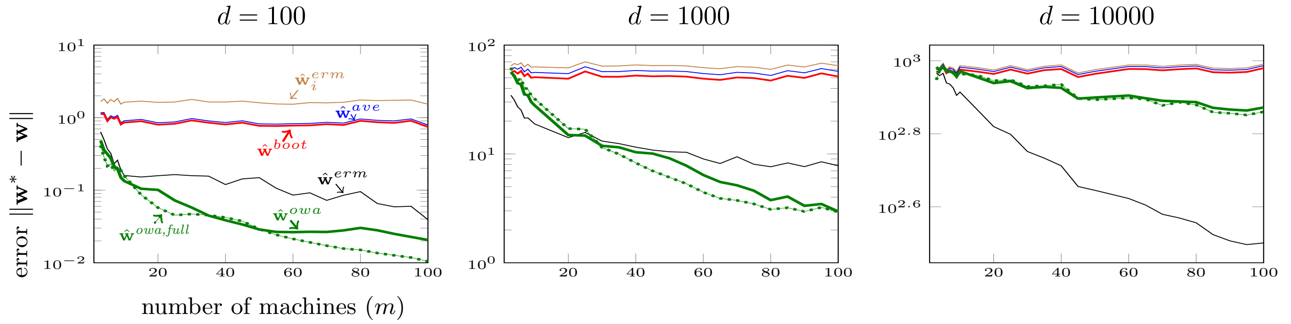

Figure 2. In all three figures, the number of data points per machine is $m = 1000$. Therefore, the left figure shows scalability in the low dimension regime, the middle figure in a medium dimension regime, and the right figure in a high dimension regime. $\hat{\mathbf{w}}^{owa}$ scales well with the number of machines in all cases. In particular, it scales with m at the same rate as $\hat{\mathbf{w}}^{erm}$, whereas $\hat{\mathbf{w}}^{ave}$ and $\hat{\mathbf{w}}^{boot}$ do not scale well with m on this synthetic data. Surprisingly, $\hat{\mathbf{w}}^{owa}$ outperforms the oracle estimator trained on all of the data $\hat{\mathbf{w}}^{erm}$ in some situations.

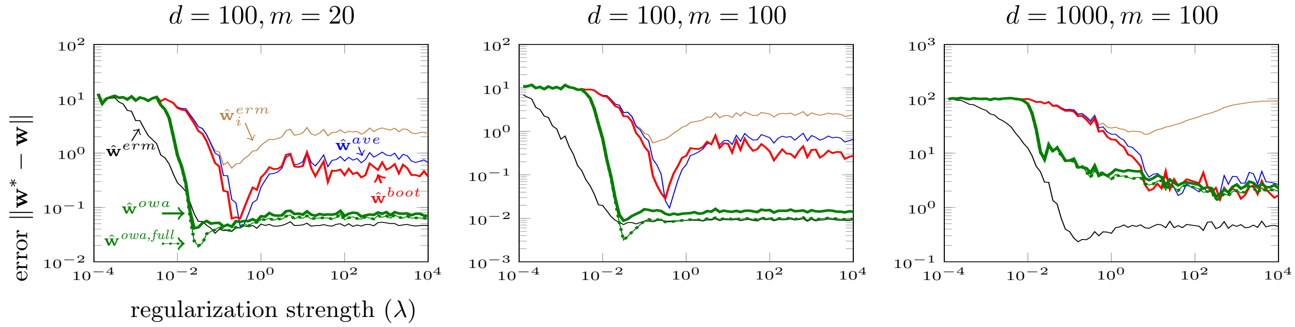

Figure 3. OWA is robust to the regularization strength. Surprisingly, additional regularization introduced by OWA lets it outperform the oracle estimator $\hat{\mathbf{w}}^{erm}$ in some cases. Our theory states that as $m \to d$, we have that $\hat{\mathcal{W}}^{owa} \to \mathcal{W}$, and so $\hat{\mathbf{w}}^{owa} \to \hat{\mathbf{w}}^{erm}$. This is confirmed in the middle experiment. In the left experiment, $m < d$, but $\hat{\mathbf{w}}^{owa}$ still behaves similarly to $\hat{\mathbf{w}}^{erm}$. In the right experiment, $\hat{\mathbf{w}}^{owa}$ has similar performance as $\hat{\mathbf{w}}^{ave}$ and $\hat{\mathbf{w}}^{boot}$ but is more robust to λ.

are then sampled as

$$\mathbf{x}_i \sim \mathcal{N}(0, I), \quad y_i = \left(1 + \exp(-\mathbf{x}_i^\mathsf{T}\mathbf{w}^*)\right)^{-1}. \tag{33}$$

The primary advantage of synthetic data is that we know the model's true parameter vector. So for each estimator $\hat{\mathbf{w}}$ that we evaluate, we can directly calculate the error $\|\hat{\mathbf{w}} - \mathbf{w}^*\|$. We run two experiments on the synthetic data. In both experiments, we use the L1 regularizer to induce sparsity in our estimates of $\mathbf{w}^*$. Results are qualitatively similar when using a Laplace, Gaussian, or uniform prior on $\mathbf{w}^*$, and with L2 regularization.

Our first experiment shows how the estimators scale as the number of machines m increases. We fix $n = 1000$ data points per machine, so the size of the dataset mn grows as we add more machines. This simulates the typical "big data" regime where data is abundant, but processing resources are scarce. For each value of m, we generate 50 datasets and report the average of the results. Our $\hat{\mathbf{w}}^{owa}$ estimator was trained with $n^{owa} = 128$. The results are shown in Figure 2. As the analysis predicted, the performance of $\hat{\mathbf{w}}^{owa}$ scales much better than $\hat{\mathbf{w}}^{ave}$ and $\hat{\mathbf{w}}^{boot}$. Surprisingly, in the low dimensional regimes, $\hat{\mathbf{w}}^{owa}$ outperforms the single machine oracle $\hat{\mathbf{w}}^{erm}$.

One issue that has been overlooked in the literature on non-interactive distributed estimation is how to best set λ. There are two natural ways to use cross validation and grid search. The first is: for each λ in the grid, perform the full training procedure including all communications and merges. Then select the λ with lowest cross validation error. Unfortunately, this requires many rounds of communication (one for each λ we are testing). This extra communication largely negates the main advantage of non-interactive learners. The second is: each machine independently uses cross validation to select the λ that best fits the data locally when calculating $\hat{\mathbf{w}}_i^{erm}$. In our experiments we use this second method due to three advantages. First, there is no additional communication because model selection is a completely local task. Second, existing optimizers have built-in model selection routines which make the process easy to implement. We used the default model selection procedure from Python's SciKit-Learn (Pedregosa et al., 2011). Third, the data may be best fit using different regularization strengths for each machine. It is unclear which method previous literature used to select λ. As mentioned in Section 3.3, OWA requires an additional round of cross validation during the master's merge procedure to set λ_2. This cross validation is particularly fast and requires no communication with other machines.

Our second experiment shows the importance of proper λ selection. We evaluate the performance of the estimators with λ varying from 10^{-4} to 10^4 on a grid of 80 points. Figure 3 shows the results. The $\hat{\mathbf{w}}^{owa}$ estimator is more robust to the choice of λ than the other distributed estimators.

6.2. Real World Advertising Data

We evaluate the estimators on real world data from the KDD 2012 Cup (Niu et al., 2012). The goal is to predict whether a user will click on an ad from the Tencent internet search engine. This dataset was previously used to evaluate the performance of $\hat{\mathbf{w}}^{boot}$ (Zhang et al., 2012). This dataset is too large to fit on a single machine, so we must use distributed estimators, and we do not provide results of the oracle estimator $\hat{\mathbf{w}}^{erm}$ in our figures. There are 235,582,879 distinct data points, each of dimension 741,725. The data points are sparse, so we use the L1 norm to encourage sparsity in our final solution. The regularization strength was set using cross validation in the same manner as for the synthetic data. For each test, we split the data into 80 percent training data and 20 percent test data. The training data is further subdivided into 128 partitions, one for each of the machines used. It took about 1 day to train the local model on each machine in our cluster.

Our first experiment tests the sensitivity of the n^{owa} parameter on large datasets. We fix $m = 128$, and allow n^{owa} to vary from 2^0 to 2^{20}. Recall that the number of data points used in the second optimization is mn^{owa}, so when $n^{owa} = 2^{20}$ nearly the entire data set is used. We repeated the experiment 50 times, each time using a different randomly selected set Z^{owa} for the second optimization. Figure 4 shows the results. Our $\hat{\mathbf{w}}^{owa}$ estimator has lower loss than $\hat{\mathbf{w}}^{ave}$ using only 16 data points per machine (approximately 4×10^{-8} percent of the full training set) and $\hat{\mathbf{w}}^{owa}$ has converged to its final loss value with only 1024 data points per machine (approximately 2.7×10^{-6} percent of the full training set). This justifies our claim that only a small number of data points are needed for the second round of optimization, and so the communication complexity of $\hat{\mathbf{w}}^{owa}$ is essentially the same as $\hat{\mathbf{w}}^{ave}$. The computation is also very fast due to the lower dimensionality and L2 regularization in the second round of optimization. When $n^{owa} = 2^{10}$, computing the merged model took only minutes (including the cross validation time to select λ_2). This time is negligible compared to the approximately 1 day it took to train the models on the individual machines.

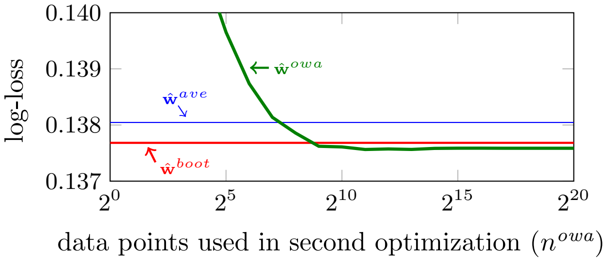

Figure 4. Relatively few data points are needed in the second round of optimization for $\hat{\mathbf{w}}^{owa}$ to converge.

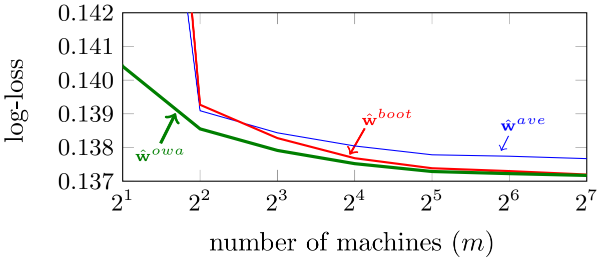

Figure 5. Performance of the parallel estimators on advertising data as the number of machines m increases.

Our last experiment shows the performance as we scale the number of machines m. The results are shown in Figure 5. Here, $\hat{\mathbf{w}}^{owa}$ performs especially well in the low m setting. For large m, $\hat{\mathbf{w}}^{owa}$ continues to slightly outperform $\hat{\mathbf{w}}^{boot}$ without the need for an expensive model selection procedure to determine the r parameter.

7. DISCUSSION

We introduced a new distributed estimation algorithm called OWA. OWA is the first algorithm with communication complexity comparable to the averaging estimator that achieves the optimal $O(1/\sqrt{mn})$ error rate. Although OWA's analysis is more general than the analysis of similar distributed estimators, it is limited to linear models. In order to make OWA work for non-linear models, we need a method for projecting data points into the parameter space to perform the second round of optimization.

References

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

Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. *Symposium on the Theory of Computing*, 2016.

Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of distributed statistical estimation and dimensionality. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2014.

Jun Han and Qiang Liu. Bootstrap model aggregation for distributed statistical learning. *Advances in Neural Information Processing Systems*, 2016.

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

Erich Leo Lehmann. *Elements of large-sample theory*. Springer Science & Business Media, 1999.

Qiang Liu and Alexander T Ihler. Distributed estimation, information loss and exponential families. In *Advances in Neural Information Processing Systems*, pages 1098–1106, 2014.

Ryan McDonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon S Mann. Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems*, pages 1231–1239, 2009.

Srujana Merugu and Joydeep Ghosh. Privacy-preserving distributed clustering using generative models. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 211–218. IEEE, 2003.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

Yanzhi Niu, Yi Wang, Gordon Sun, Aden Yue, Brian Dalessandro, Claudia Perlich, and Ben Hamner. The tencent dataset and KDD-cup'12. 2012.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Jonathan D Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference*, 5(4):379–404, 2016.

Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems 27*, pages 163–171, 2014.

Vladimir Spokoiny. Parametric estimation. finite sample theory. *The Annals of Statistics*, 40(6):2877–2909, 2012.

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

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

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Divide and conquer kernel ridge regression. In *COLT*, 2013b.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.