Communication-Efficient Distributed Optimization using an Approximate Newton-type Method

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago and the Department of Computer Science, Technion, Haifa, Israel

Tong Zhang TZHANG@STAT.RUTGERS.EDU
Department of Statistics, Rutgers University, Piscataway NJ, USA, and Baidu Inc., Beijing, China

Abstract

We present a novel Newton-type method for distributed optimization, which is particularly well suited for stochastic optimization and learning problems. For quadratic objectives, the method enjoys a linear rate of convergence which provably *improves* with the data size, requiring an essentially constant number of iterations under reasonable assumptions. We provide theoretical and empirical evidence of the advantages of our method compared to other approaches, such as one-shot parameter averaging and ADMM.

1. Introduction

We consider the problem of distributed optimization, where each of m machines has access to a function $\phi_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \ldots, m$, and we would like to minimize their average

$$\phi(w) = \frac{1}{m} \sum_{i=1}^{m} \phi_i(w). \tag{1}$$

We are particularly interested in a stochastic optimization (learning) setting, where the ultimate goal is to minimize some stochastic (population) objective (e.g. expected loss or generalization error)

$$F(w) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[f(w, z)] \tag{2}$$

and each of the m machines has access to n i.i.d. samples $z_i^1, \ldots, z_i^n$ from the source distribution $\mathcal{D}$, for a total of $N = nm$ independent samples evenly and randomly distributed among the machines. Each machine i can construct a local empirical (sample) estimate of $F(w)$:

$$\phi_i(w) = \hat{F}_i(w) = \frac{1}{n} \sum_{j=1}^{n} f(w, z_i^j) \tag{3}$$

and the overall empirical objective is then:

$$\phi(w) = \hat{F}(w) = \frac{1}{m} \sum_{i=1}^{m} \hat{F}_i(w) = \frac{1}{nm} \sum_{i,j} f(w, z_i^j). \tag{4}$$

We can then use the *empirical risk minimizer* (ERM)

$$\hat{w} = \arg\min \hat{F}(w) \tag{5}$$

as an approximate minimizer of $F(w)$. Since our interest lies mostly with this stochastic optimization setting, we will denote $\hat{w} = \arg\min \phi(w)$ even when the optimization objective $\phi(w)$ is not an empirical approximation to a stochastic objective.

When considering distributed optimization, two resources are at play: the amount of processing on each machine, and the communication between machines. In this paper, we focus on algorithms which alternate between a local optimization procedure at each machine, and a communication round involving simple map-reduce operations such as distributed averaging of vectors in $\mathbb{R}^d$. Since the cost of communication is very high in practice (Bekkerman et al., 2011), our goal is to develop methods which quickly optimize the empirical objective $\hat{F}(\cdot)$, using a minimal number of such iterations.

One-Shot Averaging A straight-forward single-iteration approach is for each machine to optimize its own local objective, obtaining

$$\hat{w}_i = \arg\min \phi_i(w), \tag{6}$$

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

and then to compute their average:

$$\bar{w} = \frac{1}{m}\sum_{i=1}^{m}\hat{w}_i. \tag{7}$$

This approach, which we refer to as "one-shot parameter averaging", was recently considered in Zinkevich et al. (2010) and further analyzed by Zhang et al. (2013). The latter also proposed a bias-corrected improvement which perturbs each $\hat{w}_i$ using the optimum on a bootstrap sample. This approach gives only an approximate minimizer of $\phi(w)$ with some finite suboptimality, rather then allowing us converge to $\hat{w}$ (i.e. to obtain solutions with any desired suboptimality ϵ). Although approximate solutions are often sufficient for stochastic optimization, we prove in Section 2 that the one-shot solution $\bar{w}$ can be much worse in terms of minimizing the population objective $F(w)$, compared to the actual empirical minimizer $\hat{w}$. It does not seem possible to address this suboptimality by more clever averaging, and instead additional rounds of communications appear necessary.

Gradient Descent One possible multi-round approach to distributed optimization is a distributed implementation of gradient descent: at each iteration each machine calculates $\nabla\phi_i(w^{(t)})$ at the current iterate $w^{(t)}$, and then these are averaged to obtain the overall gradient $\nabla\phi(w^{(t)})$, and a gradient step is taken. As the iterates are then standard gradient descent iterates, the number of iterations, and so also number of communication rounds, is linear in the conditioning of the problem – or, if accelerated gradient descent is used, proportional to the square root of the condition number: If $\phi(w)$ is L-smooth and λ-strongly convex, then

$$\mathcal{O}\left(\sqrt{\frac{L}{\lambda}}\log\left(\frac{1}{\epsilon}\right)\right) \tag{8}$$

iterations are needed to attain an ϵ-suboptimal solution. The polynomial dependence on the condition number may be disappointing, as in many problems the parameter of strong convexity λ might be very small. E.g., when strong convexity arises from regularization, as in many stochastic optimization problems, λ decreases with the overall sample size $N = nm$, and is typically at most $1/\sqrt{nm}$ (Sridharan et al. 2008; Shalev-Shwartz et al. 2009; and see also Section 4.3 below). The number of iterations / communication rounds needed for distributed accelerated gradient descent then scales as $\sqrt[4]{nm}$, i.e. increases polynomially with the sample size.

Instead of gradient descent, one may also consider more sophisticated methods which utilize gradient information, such as quasi-Newton methods. For example, a distributed implementation using L-BFGS has been proposed in (Agarwal et al., 2011). However, no guarantee better then (8) can be ensured for gradient-based methods (Nemirovsky & Yudin, 1983), and we thus may still get a polynomial dependence on the sample size.

ADMM and other approaches Another popular approach is distributed alternating direction method of multipliers (ADMM, e.g. Boyd et al. 2011), where the machines alternate between computing shared dual variables in a distributed manner, and solving augmented Lagrangian problems with respect to their local data. However, the convergence of ADMM can be slow. Although recent works proved a linear convergence rate under favorable assumptions (Deng & Yin, 2012; Hong & Luo, 2012), we are not aware of any analysis where the number of iterations / communication rounds doesn't scale strongly with the condition number, and hence the sample size, for learning applications. A similar dependence occurs with other recently-proposed algorithms for distributed optimization (e.g. Yang, 2013; Mahajan et al., 2013; Dekel et al., 2012; Cotter et al., 2011; Duchi et al., 2012). We also mention that our framework is orthogonal to much recent work on distributed coordinate descent methods (e.g. Recht et al., 2011; Richtárik & Takác, 2013), which assume the data is split feature-wise rather than instance-wise.

Our Method The method we propose can be viewed as an approximate Newton-like method, where at each iteration, instead of a gradient step, we take a step appropriate for the geometry of the problem, *as estimated on each machine separately*. In particular, for quadratic objectives, the method can be seen as taking approximate Newton steps, where each machine i implicitly uses its local Hessian $\nabla^2\phi_i(w)$ (although no Hessians are explicitly computed!). Unlike ADMM, our method can take advantage of the fact that for machine learning applications, the sub-problems are usually similar: $\phi_i \approx \phi$. We refer to our method as DANE—Distributed Approximate NEwton.

DANE is applicable to any smooth and strongly convex problem. However, as is typical of Newton and Newton-like methods, its generic analysis is not immediately apparent. For general functions, we can show convergence, but cannot rigorously prove improvement over gradient descent. Instead, in order to demonstrate DANE's advantages and give a sense of its benefits, we focus our theoretical analysis on quadratic objectives. For stochastic quadratic objectives, where $f(w,z)$ is L-smooth and λ-strongly convex in $w \in \mathbb{R}^d$, we show that

$$\mathcal{O}\left(\frac{(L/\lambda)^2}{n}\log(dm)\log(\frac{1}{\epsilon})\right) \tag{9}$$

iterations are sufficient for DANE to find $\tilde{w}$ such that with high probability $\hat{F}(\tilde{w}) \leq \hat{F}(\hat{w}) + \epsilon$. When L/λ is fixed and the number of examples n per machine is large (the

regime considered by Zhang et al. 2013), (9) establishes convergence after a *constant* number of iterations / communication rounds. When λ scales as $1/\sqrt{nm}$, as discussed above, (9) yields convergence to the empirical minimizer in a number of iterations that scales roughly linearly with the number of machines m, but *not* with the sample size $N = nm$. To the best of our knowledge, this is the first algorithm which provably has such a behavior. We also provide evidence for similar behavior on non-quadratic objectives.

Notation and Definitions For vectors, $\|v\|$ is always the Euclidean norm, and for matrices $\|A\|_2$ is the spectral norm. We use $\lambda \preccurlyeq A \preccurlyeq L$ to indicate that the eigenvalues of A are bounded between λ and L. We say that a twice differentiable[1] function $f(w)$ is λ-strongly convex or L-smooth, iff for all w, its Hessian is bounded from below by λ (i.e. $\lambda \preccurlyeq \nabla^2 f(w)$), or above by L (i.e. $\nabla^2 f(w) \preccurlyeq L$) respectively.

2. Stochastic Optimization and One-shot Parameter Averaging

In a stochastic optimization setting, where the true objective is the population objective $F(w)$, there is a limit to the accuracy with which we can minimize $F(w)$ given only $N = nm$ samples, even using the exact empirical minimizer $\hat{w}$. It is thus reasonable to compare the suboptimality of $F(w)$ when using the exact $\hat{w}$ to what can be attained using distributed optimization with limited communication.

When $f(w,z)$ has gradients with bounded second moments, namely when $\forall_w \mathbb{E}_z \left[\|\nabla_w f(w,z)\|^2\right] \leq G^2$, and $F(w)$ is λ-strongly convex, then (Shalev-Shwartz et al., 2009)[2]

$$\mathbb{E}[F(\hat{w})] \leq F(w^*) + \mathcal{O}\left(\frac{G^2}{\lambda N}\right) = \inf_w F(w) + \mathcal{O}\left(\frac{G^2}{\lambda nm}\right) \tag{10}$$

where $w^* = \arg\min F(w)$ is the population minimizer and the expectation is with respect to the random sample of size $N = nm$. One might then ask whether a suboptimality of $\epsilon = \mathcal{O}\left(\frac{G^2}{\lambda nm}\right)$ can be also be achieved using a few, perhaps only one, round of communication. This is different from seeking a distributed optimization method that achieves any arbitrarily small empirical suboptimality, and thus converges to $\hat{w}$, but might be sufficient in terms of stochastic optimization.

For one-shot parameter averaging, Zhang et al. (2013, Corollary 2) recently showed that for λ-strongly convex objectives, and when moments of the first, second and third derivatives of $f(w,z)$ are bounded by G, L, and M respectively[3], then

$$\mathbb{E}\,\|\bar{w} - w^*\|^2 \leq \tilde{\mathcal{O}}\left(\frac{G^2}{\lambda^2 nm} + \frac{G^4 M^2}{\lambda^6 n^2} + \frac{L^2 G^2 \log d}{\lambda^4 n^2}\right), \tag{11}$$

where $\bar{w}$ is the one-shot average estimator defined in (7). This implies that the population suboptimality $\mathbb{E}[F(\bar{w})] - F(w^*)$ is bounded by

$$\tilde{\mathcal{O}}\left(\frac{LG^2}{\lambda^2 nm} + \frac{LG^4 M^2}{\lambda^6 n^2} + \frac{L^3 G^2 \log d}{\lambda^4 n^2}\right). \tag{12}$$

Zhang et al. (2013) argued that the dependence on the sample size mn above is essentially optimal: the dominant term (as $n \to \infty$, and in particular when $n \gg m$) scales as $1/(nm)$, which is the same as for the empirical minimizer $\hat{w}$ (as in eq. 10), and so one-shot parameter averaging achieves the same population suboptimality rate, using only a single round of communication, as the best rate we can hope for using unlimited communication, or if all $N = nm$ samples were on the same machine. Moreover, the $\mathcal{O}(n^{-2})$ terms can be replaced by a $\mathcal{O}(n^{-3})$ term using an appropriate bias-correction procedure.

However, this view ignores the dependence on the other parameters, and in particular the strong convexity parameter λ, which is much worse in (12) relative to (10). The strong convexity parameter often arises from an explicit regularization, and decays as the sample size increases. E.g., in regularized loss minimization and SVM-type problems (Sridharan et al., 2008), as well as more generally for stochastic convex optimization (Shalev-Shwartz et al., 2009), the regularization parameter, and hence the strong convexity parameter, decreases as $\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{nm}}$. In practice, λ is often chosen even smaller, possibly as small as $\frac{1}{N}$. Unfortunately, substituting $\lambda = \mathcal{O}(1/\sqrt{nm})$ in (12) results in a useless bound, where even the first term does not decrease with the sample size.

Of course, this strong dependence on λ might be an artifact of the analysis of Zhang et al.. However, in Theorem 1 below, we show that even in a simple one-dimensional example, when $\lambda \leq \mathcal{O}(1/\sqrt{n})$, the population sub-optimality of the one-shot estimator (using m machines and a total of nm samples), can be no better then the population sub-optimality using just n samples, and much worse than what

[1] All our results hold also for weaker definitions of smoothness and strong convexity which do not require twice differentiability.

[2] More precisely, (Shalev-Shwartz et al., 2009) shows this assuming $\|\nabla_w f(w,z)\|^2 \leq G^2$ for all w, z, but the proof easily carries over to this case.

[3] The exact conditions in Zhang et al. (2013) refer to various high order moments, but are in any case satisfied when $\|\nabla_w f(w,z)\| \leq G$, $\|\nabla_w^2 f(w,z)\|_2 \leq L$ and $\nabla^2 f(w,z)$ is M-Lipschitz in the spectral norm. For learning problems, all derivatives of the objective can be bounded in terms of a bound on the data and bounds on the derivative of a scalar loss function, and are less of a concern to us.

can be attained using nm samples. In other words, one-shot averaging does not give any benefit over using only the data on a single machine, and ignoring all other $(m-1)n$ data points.

Theorem 1. *For any per-machine sample size $n \geq 9$, and any $\lambda \in \left(0, \frac{1}{9\sqrt{n}}\right)$, there exists a distribution $\mathcal{D}$ over examples and a stochastic optimization problem on a convex set[4] $\mathcal{W} \subset \mathbb{R}$, such that:*

- *$f(w;z)$ is λ-strongly convex, infinitely differentiable, and $\forall_{w \in \mathcal{W}}\, \mathbb{E}_z[\|\nabla f(w;z)\|^2] \leq 9$.*
- *For any number of machines m, if we run one-shot parameter averaging to compute $\bar{w}$, it holds for some universal constants C_1, C_2, C_3, C_4 that*

$$\mathbb{E}[\|\bar{w} - w^*\|^2] \geq \frac{C_1}{\lambda^2 n}\,,\ \mathbb{E}[\|\hat{w} - w^*\|^2] \leq \frac{C_2}{\lambda^2 nm}$$

$$\mathbb{E}[F(\bar{w})] - F(w^*) \geq \frac{C_3}{\lambda n}\,,\ \mathbb{E}[F(\hat{w})] - F(w^*) \leq \frac{C_4}{\lambda nm}$$

The intuition behind the construction of Theorem 1 is that when λ is small, the deviation of each machine output $\hat{w}_i$ from w^* is large, and its expectation is biased away from w^*. The exact bias amount is highly problem-dependent, and cannot be eliminated by any fixed averaging scheme. Since bias is not reduced by averaging, the optimization error does not scale down with the number of machines m. The full construction and proof appear in appendix A. In the appendix we also show that the bias correction proposed by Zhang et al. to reduce the lower-order terms in equation (11) does not remedy this problem.

3. Distributed Approximate Newton-type Method

We now describe a new iterative method for distributed optimization. The method performs two distributed averaging computations per iteration, and outputs a predictor $w^{(t)}$ which, under suitable parameter choices, converges to the optimum $\hat{w}$. The method, which we refer to as DANE (**D**istributed **A**pproximate **NE**wton-type Method) is described in Figure 1.

DANE maintains an agreed-upon iterate $w^{(t)}$, which is synchronized among all machines at the end of each iteration. In each iteration, we first compute the gradient $\nabla\phi(w^{(t-1)})$ at the current iterate, by averaging the local gradients $\nabla\phi_i(w^{(t-1)})$. Each machine then performs a separate local optimization, based on its own local objective $\phi_i(w)$ and the computed global gradient $\nabla\phi(w^{(t)})$, to obtain a local iterate $w_i^{(t)}$. These local iterates are averaged to obtain the centralized iterate $w^{(t)}$.

The crux of the method is the local optimization performed on each machine at each iteration:

$$w_i^{(t)} = \arg\min_w [\phi_i(w) - (\nabla\phi_i(w^{(t-1)}) - \eta\nabla\phi(w^{(t-1)}))^\top w + \frac{\mu}{2}\|w - w^{(t-1)}\|_2^2] \tag{13}$$

To understand this local optimization, recall the definition of the Bregman divergence corresponding to a strongly convex function ψ:

$$D_\psi(w'; w) = \psi(w') - \psi(w) - \langle\nabla\psi(w), w' - w\rangle.$$

Now, for each local objective ϕ_i, consider the regularized local objective, defined as

$$h_i(w) = \phi_i(w) + \frac{\mu}{2}\|w\|^2$$

and its corresponding Bregman divergence:

$$D_i(w'; w) = D_{h_i}(w'; w) = D_{\phi_i}(w'; w) + \frac{\mu}{2}\|w' - w\|^2.$$

It is not difficult to check that the local optimization problem (13) can be written as

$$w_i^{(t)} = \arg\min_w \phi(w^{(t-1)}) + \langle\nabla\phi(w^{(t-1)}), w - w^{(t-1)}\rangle + \frac{1}{\eta}D_i(w; w^{(t-1)}), \tag{14}$$

where we also added the terms $\phi(w^{(t-1)}) + \langle\nabla\phi(w^{(t-1)}), w^{(t-1)}\rangle$ which do not depend on w and do not affect the optimization. The first two terms in (14) are thus a linear approximation of the overall objective $\phi(w)$ about the current iterate $w^{(t-1)}$, and do not depend on the machine i. What varies from machine to machine is the potential function used to localize the linear approximation. The update (14) is in-fact a mirror descent update (Nemirovski & Yudin, 1978; Beck & Teboulle, 2003) using the potential function h_i, and step size η.

Let us examine this form of update. When $\mu \to \infty$ the potential function essentially becomes a squared Euclidean norm, as in gradient descent updates. In fact, when $\eta, \mu \to \infty$ as $\tilde{\eta} \doteq \frac{\eta}{\mu}$ remains constant, the update (14) becomes a standard gradient descent update on $\phi(w)$ with stepsize $\tilde{\eta}$. In this extreme, the update does not use the local objective $\phi_i(w)$, beyond the centralized calculation of $\nabla\phi(w)$, the updates (14) are the same on all machines, and the second round of communication is not needed. DANE reduces to distributed gradient descent, with its iteration complexity of $\mathcal{O}\left(\frac{L}{\lambda}\log(1/\epsilon)\right)$.

[4] Following the framework of Zhang et al., we present an example where the optimization is performed on a bounded set, which ensures that the gradient moments are bounded. However, this is not essential and the same result can be shown when the domain of optimization is $\mathbb{R}$.

Procedure DANE

Parameter: learning rate $\eta > 0$ and regularizer $\mu > 0$
Initialize: Start at some $w^{(0)}$, e.g. $w^{(0)} = 0$
Iterate: for $t = 1, 2, \ldots$
Compute $\nabla\phi(w^{(t-1)}) = \frac{1}{m}\sum_{i=1}^{m} \nabla\phi_i(w^{(t-1)})$ and distribute to all machines
For each machine i, solve $w_i^{(t)} = \arg\min_w \left[\phi_i(w) - (\nabla\phi_i(w^{(t-1)}) - \eta\nabla\phi(w^{(t-1)}))^\top w + \frac{\mu}{2}\|w - w^{(t-1)}\|_2^2\right]$
Compute $w^{(t)} = \frac{1}{m}\sum_{i=1}^{m} w_i^{(t)}$ and distribute to all machines $(*)$
end

Figure 1. Distributed Approximate NEwton-type method (DANE)

At the other extreme, consider the case where $\mu = 0$ and all local objectives are equal, i.e. $h_i(w) = \phi_i(w) = \phi(w)$. Substituting the definition of the Bregman divergence into (14), or simply investigating (13), we can see that $w_i^{(t)} = \arg\min \phi_i(w) = \arg\min \phi(w) = \hat{w}$. That is, DANE converges in a single iteration to the overall empirical optimum. This is an ideal Newton-type iteration, where the potential function is perfectly aligned with the objective.

Of course, if $\phi_i(w) = \phi(w)$ for all machines i, we would not need to perform distributed optimization in the first place. Nevertheless, as $n \to \infty$, we can hope that $\phi_i(w)$ are similar enough to each other, such that (14) *approximates* such an ideal Newton-type iteration, gets us very close to the optimum, and very few such iterations are sufficient.

In particular, consider the case where $\phi_i(w)$, and hence also $\phi(w)$ are quadratic. In this case, the Bregman divergence $D_i(w; w^{(t-1)})$ takes the form:

$$\frac{1}{2}(w - w^{(t-1)})^\top(\nabla^2\phi_i(w^{(t-1)}) + \mu I)(w - w^{(t-1)}), \quad (15)$$

and the update (14) can be solved in closed form:

$$\begin{aligned} w_i^{(t)} &= w^{(t-1)} - \eta(\nabla^2\phi_i(w^{(t-1)}) + \mu I)^{-1}\nabla\phi(w^{(t-1)}) \\ w^{(t)} &= w^{(t-1)} \\ &\quad - \eta\left(\frac{1}{m}\sum_i (\nabla^2\phi_i(w^{(t-1)}) + \mu I)^{-1}\right)\nabla\phi(w^{(t-1)}). \end{aligned} \quad (16)$$

Contrast this with the true Newton update:

$$w^{(t)} = w^{(t-1)} - \eta\left(\frac{1}{m}\sum_i \nabla^2\phi_i(w^{(t-1)})\right)^{-1}\nabla\phi(w^{(t-1)}). \quad (17)$$

The difference here is that in (16) we approximate the inverse of the average of the local Hessians with the average of the inverse of the Hessians (plus a possible regularizer). Again we see that the DANE update (16) approximates the true Newton update (17), which can be performed in a distributed fashion *without communicating the Hessians*.

For a quadratic objective, a single Newton update is enough to find the exact optimum. In Section 4 we rigorously analyze the effects of the distributed approximation, and quantify the number of DANE iterations (and thus rounds of communication) required.

For a general convex, but non-quadratic, objective, the standard Newton approach is to use a quadratic approximation to the ideal Bregman divergence D_ϕ. This leads to the familiar quadratic Newton update in terms of the Hessian. DANE uses a different sort of approximation to D_ϕ: we use a *non-quadratic* approximation, based on the entire objective and not just a local quadratic approximation, but approximate the potential on each node separately. In the stochastic setting, this approximation becomes better and better, and thus the required number of iterations decrease, as $n \to \infty$.

Since it is notoriously difficult to provide good global analysis for Newton-type methods, we will investigate the global convergence behavior of DANE carefully in the next Section but only for quadratic objective functions. This analysis can also be viewed as indicative for non-quadratic objectives, as locally they can be approximated by quadratics and so should enjoy the same behavior, at least asymptotically. For non-quadratics, we provide a rigorous convergence guarantee when the stepsize η is sufficiently small or the regularization parameter μ is sufficiently large (in Section 5). However, this analysis does not show a benefit over distributed gradient descent for non-quadratics. We partially bridge this gap by showing that even in the non-quadratic case, the convergence rate improves as the local problems ϕ_i become more similar.

4. DANE for Quadratic Objectives

In this Section, we analyze the performance of DANE on quadratic objectives. We begin in Section 4.1 with an analysis of DANE for arbitrary quadratic objectives $\phi_i(w)$, without stochastic assumptions, deriving a guarantee in terms of the approximation error of the true Hessian. Then in Section 4.2 we consider the stochastic setting where the

instantaneous objective $f(w,z)$ is quadratic in w, utilizing a bound on the approximation error of the Hessian to obtain a performance guarantee for DANE in terms of the smoothness and strong convexity of $f(w,z)$. In Section 4.3 we also consider the behavior for stochastic optimization problems, where λ is set as a function of the sample size $N = nm$.

4.1. Quadratic $\phi_i(w)$

We begin by considering the case where each local objective $\phi_i(w)$ is quadratic, i.e. has a fixed Hessian. The overall objective $\phi(w)$ is then of course also quadratic.

Theorem 2. *After t iterations of DANE on quadratic objectives with Hessians $H_i = \nabla^2\phi_i(w)$, we have:*

$$\|w^{(t)} - \hat{w}\| \leq \|I - \eta\tilde{H}^{-1}H\|_2^t \|w^{(0)} - \hat{w}\|,$$

where $H = \nabla^2\phi(w) = \frac{1}{m}\sum_{i=1}^m H_i$ and $\tilde{H}^{-1} = \frac{1}{m}\sum_{i=1}^n (H_i + \mu I)^{-1}$.

The proof appears in Appendix B. The theorem implies that if $\|I - \eta\tilde{H}^{-1}H\|_2$ is smaller than 1, we get a linear convergence rate. Indeed, we would expect $\|I - \eta\tilde{H}^{-1}H\|_2 \ll 1$ as long as η is close to 1 and $\tilde{H}$ is a good approximation for the true Hessian H, hence $\tilde{H}^{-1}H \approx I$. In particular, if H is not too ill-conditioned, and all H_i are sufficiently close to their average H, we can indeed ensure $\tilde{H} \approx H$. This is captured by the following lemma (whose proof appears in Appendix C):

Lemma 1. *If $0 < \lambda \preccurlyeq H \preccurlyeq L$ and for all i, $\|H_i - H\|_2 \leq \beta$, then setting $\eta = 1$ and $\mu = \max\{0, \frac{8\beta^2}{\lambda} - \lambda\}$, we have:*

$$\|I - \tilde{H}^{-1}H\|_2 \leq \begin{cases} \frac{4\beta^2}{\lambda^2} & \text{if } \frac{4\beta^2}{\lambda^2} \leq \frac{1}{2} \\ 1 - \frac{\lambda^2}{16\beta^2} & \text{otherwise.} \end{cases}$$

In the next Section, we consider the stochastic setting, where we can obtain bounds for $\|H_i - H\|_2$ that improve with the sample size, and plug these into Lemma 1 and Theorem 2 to obtain a performance guarantee for DANE.

4.2. Stochastic Quadratic Problems

We now turn to a stochastic quadratic setting, where $\phi_i(w) = \hat{F}_i(w)$ as in (3), and the instantaneous losses are smooth and strongly convex quadratics. That is, for all z, $f(w,z)$ is quadratic in w and $\lambda \preccurlyeq \nabla_w^2 f(w,z) \preccurlyeq L$.

We first use a matrix concentration bound to establish that all Hessians $H_i = \nabla^2\hat{F}_i(w)$ are close to each other, and hence also to their average:

Lemma 2. *If $0 \preccurlyeq \nabla_w^2 f(w,z) \preccurlyeq L$ for all z, then with probability at least $1-\delta$ over the samples, $\max_i \|H_i - H\|_2 \leq \sqrt{\frac{32L^2\log(dm/\delta)}{n}}$, where $H_i = \nabla^2\hat{F}_i(w)$ and $H = \nabla^2\hat{F}(w)$.*

The proof appears in appendix D. Combining Lemma 2, Lemma 1 and Theorem 2, we can conclude:

Theorem 3. *In the stochastic setting, and when the instantaneous losses are quadratic with $\lambda \preccurlyeq \nabla f(w,z) \preccurlyeq L$, then after*

$$t = \mathcal{O}\left(\frac{(L/\lambda)^2}{n}\log\left(\frac{dm}{\delta}\right)\log\left(\frac{L\|w^0 - \hat{w}\|^2}{\epsilon}\right)\right)$$

iterations of DANE, we have, with probability at least $1-\delta$, that $\hat{F}(w^{(t)}) \leq \hat{F}(\hat{w}) + \epsilon$.

The proof appears in Appendix E. From the theorem, we see that if the condition number L/λ is fixed, then as $n \to \infty$ the number of required iterations decreases. In fact, for any target sub-optimality ϵ, as long as the sample size is at least logarithmically large, namely $n = \Omega\left((L/\lambda)^2\log(dm)\log(\frac{1}{\epsilon})\right)$, we can obtain the desired accuracy after a constant or even a single DANE iteration! This is a mild requirement on the sample size, since N generally increases at least linearly with $1/\epsilon$.

We next turn to discuss the more challenging case where the condition number decays with the sample size.

4.3. Analysis for Regularized Objectives

Consider a stochastic convex optimization scenario where the instantaneous objectives $f(w,z)$ are not strongly convex. For example, this is the case in linear prediction (including linear and kernel classification and regression, support vector machines, etc.), and more generally for generalized linear objectives of the form $f(w,z) = \ell_z(\langle w, \Psi(z)\rangle)$. For such generalized linear objectives, the Hessian $\nabla_w^2 f(w,z)$ is rank-1, and so certainly not strongly convex, even if $\ell_z(\cdot)$ is strongly convex.

Confronted with such non-strongly-convex objectives, a standard approach is to perform empirical minimization on a *regularized* objective (Shalev-Shwartz et al., 2009). That is, to define the regularized instantaneous objective

$$f_\lambda(w,z) = f(w,z) + \frac{\lambda}{2}\|w\|^2 \tag{18}$$

and minimize the corresponding empirical objective $\hat{F}_\lambda$. The instantaneous objective $f_\lambda(w,z)$ of the modified stochastic optimization problem is now λ-strongly convex. If $f(w,z)$ are G-Lipschitz in w, then we have (Shalev-Shwartz et al., 2009):

$$\begin{aligned} F(\hat{w}_\lambda) \leq F_\lambda(\hat{w}_\lambda) &\leq F_\lambda(w_\lambda^*) + \mathcal{O}\left(\frac{G^2}{\lambda N}\right) \\ &= \inf_w \left(F(w) + \frac{\lambda}{2}\|w\|^2\right) + \mathcal{O}\left(\frac{G^2}{\lambda N}\right) \\ &\leq \inf_{\|w\|\leq B} F(w) + \mathcal{O}\left(\lambda B^2 + \frac{G^2}{\lambda N}\right), \end{aligned}$$

where $\hat{w}_\lambda = \arg\min \hat{F}_\lambda(w)$ and $w^*_\lambda = \arg\min F_\lambda(w)$. The optimal choice of λ in the above is $\lambda = \sqrt{\frac{G^2}{B^2 N}}$, where B is a bound on the predictors we would like to compete with, and with this λ we get the optimal rate:

$$F(\hat{w}_\lambda) \leq \inf_{\|w\| \leq B} F(w) + \mathcal{O}\left(\sqrt{\frac{B^2 G^2}{N}}\right). \qquad (19)$$

It is thus instructive to consider the behavior of DANE when $\lambda = \Theta\left(\sqrt{\frac{G^2}{B^2 N}}\right) = \Theta\left(\sqrt{\frac{G^2}{B^2 nm}}\right)$. Plugging this choice of λ into Theorem 3, we get that the number of DANE iterations behaves as:

$$\mathcal{O}\left(\frac{L^2 B^2}{G^2} \cdot m \cdot \log(dm) \log(1/\epsilon)\right). \qquad (20)$$

That is, unlike distributed gradient descent, or any other relevant method we are aware of, the number of required iterations / communication rounds does *not* increase with the sample size, and only scales linearly with the number of machines.

5. Convergence Analysis for Non-Quadratic Objectives

As discussed above, it is notoriously difficult to obtain generic global analysis of Newton-type methods. Our main theoretical result in this paper is the analysis for quadratic objectives, which we believe is also instructive for non-quadratics. Nevertheless, we complement this with a convergence analysis for generic objectives.

We therefore return to considering generic convex objectives $\phi_i(w)$. We also do not make any stochastic assumptions. We only assume that each $\phi_i(w)$ is L_i-smooth and λ_i strongly convex, and that the combined objective $\phi(w)$ is L-smooth and λ-strongly convex.

Theorem 4. *Assume that for all i, w, z, $\lambda_i \preccurlyeq \nabla^2 \phi_i(w) \preccurlyeq L_i$ and $\lambda \preccurlyeq \nabla^2 \phi(w) \preccurlyeq L$. Let*

$$\rho = \frac{1}{m} \sum_{i=1}^{m} \left[\frac{1}{\mu + L_i} - \frac{\eta L}{2(\mu + \lambda_i)^2}\right] \eta\lambda.$$

If $\rho > 0$, then the DANE iterates satisfy $\phi(w^{(t)}) - \phi(\hat{w}) \leq (1-\rho)^t [\phi(w^{(0)}) - \phi(\hat{w})]$.

The proof appears in Appendix F. The theorem establishes that with any $\mu > 0$ and small enough step-size η, DANE converges to $\hat{w}$. If each $\phi_i(w)$ is strongly convex, we can also take $\mu = 0$ and sufficiently small η and ensure convergence to $\hat{w}$. However, the optimal setting of η and μ above is to take $\mu \to \infty$ and set $\eta = \mu/L$, in which case $\rho \to \lambda/L$, and we recover distributed gradient descent, with the familiar gradient descent guarantee.

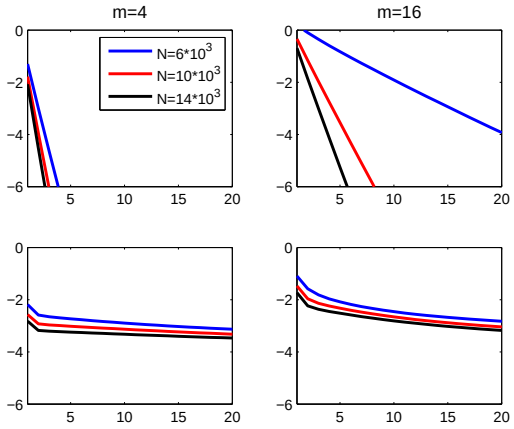

Figure 2. Synthetic dataset: Convergence rate for different number of machines m and sample sizes N. The top row presents results for DANE, and the bottom row for ADMM. The x-axis is the iteration number, and the y-axis is the logarithm (in base 10) of the suboptimality.

We again emphasize that the analysis above is weak and does not take into account the relationship between the local objectives $\phi_i(w)$. We believe that the quadratic analysis of Section 4 better captures the true behavior of DANE. Moreover, we can partially bridge this gap by the following result, which shows that a variant of DANE enjoys a linear convergence rate which improves as the local objectives ϕ_i become more similar to ϕ (the proof is in Appendix G):

Theorem 5. *Assume that in the DANE procedure, we replace step $(*)$ by $w^{(t)} = w_1^{(t)}$, and define $h(\cdot) = h_1(\cdot)$. If there exists $\gamma > 0$ such that $\forall w, w'$, we have $\gamma D_h(w; w') \leq D_\phi(w; w') \leq \eta^{-1} D_h(w; w')$, then*

$$D_h(\hat{w}; w^{(t)}) \leq (1 - \eta\gamma)^t D_h(\hat{w}; w^{(0)}).$$

If μ is small and $\phi_i \approx \phi$, then we expect $\gamma \approx 1$ and $\eta \approx 1$. In this case, $\eta\gamma \approx 1$, leading to fast convergence.

6. Experiments

In this section, we present preliminary experimental results on our proposed method. In terms of tuning η, μ, we discovered that simply picking $\eta = 1, \mu = 0$ (which makes DANE closest to a Newton-type iteration, as discussed in Section 3) often results in the fastest convergence. However, in unfavorable situations (such as when the data size per machine is very small), this can also lead to non-convergence. In those cases, convergence can be recovered by slightly increasing μ to a small positive number. In the experiments, we considered $\mu = 0, 3\lambda$. These are considerably smaller than what our theory indicates, and we leave the question of the best parameter choice to future research.

We begin by considering a simple quadratic problem us-

	COV1						ASTRO						MNIST-47					
m	2	4	8	16	32	64	2	4	8	16	32	64	2	4	8	16	32	64
$\mu = 0$	2	2	2	2	2	3	6	6	6	6	12	*	5	5	5	5	6	*
$\mu = 3\lambda$	9	9	9	9	9	9	14	14	14	14	14	14	10	10	10	10	10	10
ADMM	3	3	5	9	16	31	24	20	16	16	14	20	23	23	27	21	31	28

Figure 3. Number of iterations required to reach $< 10^{-6}$ accuracy on 3 datasets, for varying number of machines m. Results are for DANE using $\eta = 1$ and $\mu = 0, \lambda, 3\lambda$, and for ADMM. * Indicates non-convergence after 100 iterations.

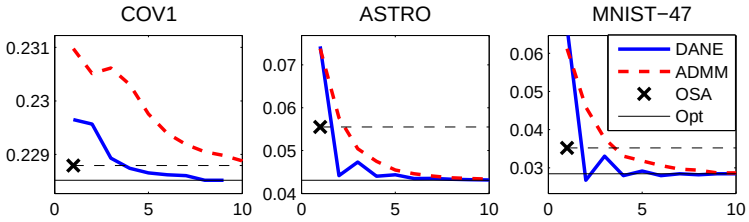

Figure 4. Average regularized smooth-hinge loss on the test set as a function of the iteration number. OSA represents bias-corrected one-shot parameter averaging, which requires a single iteration. 'Opt' is the average loss of the exact regularized loss minimizer.

ing a synthetic dataset, where all parameters can be explicitly controlled. We generated N i.i.d. training examples (x, y) according to the model $y = \langle x, w^* \rangle + \xi$, $x \sim \mathcal{N}(0, \Sigma), \xi \sim \mathcal{N}(0, 1)$, where $x \in \mathbb{R}^{500}$, the covariance matrix Σ is diagonal with $\Sigma_{i,i} = i^{-1.2}$, and w^* is the all-ones vector. Given a set of examples $\{x, y\}$ which is assumed to be randomly split to different machines, we then solved a standard ridge regression problem of the form $\min_w \frac{1}{N} \sum_{i=1}^{N} (\langle x, w \rangle - y)^2 + 0.005 w^2$, using DANE (with $\eta = 1, \mu = 0$). Figure 2 shows the convergence behavior of the algorithm for different number of machines m as the total number of examples N (and hence also the data size per machine) increases. For comparison, we also implemented distributed ADMM (Boyd et al., 2011), which is a standard method for distributed optimization but does not take advantage of the statistical similarity between problems at different machines. The results for DANE clearly indicate a linear convergence rate, and moreover, that the rate of convergence *improves* with the data size, as predicted by our analysis. In contrast, while more data improves the ADMM accuracy after a fixed number of iterations, the convergence rate is slower and does not improve with the data size[5].

We now turn to present results for solving a smooth non-quadratic problem, this time using non-synthetic datasets. Specifically, we solved a regularized loss minimization problem of the form $\min_w \frac{1}{N} \sum_{i=1}^{N} \ell(y_i \langle x_i, w \rangle) + \frac{\lambda}{2} \|w\|^2$, where ℓ is the smooth hinge loss (as in (Shalev-Shwartz & Zhang, 2013)) and the training examples $\{(x_i, y_i)\}$ are randomly split among different machines. We experimented on 3 datasets: COV1 and ASTRO-PH (as used in e.g. (Shalev-Shwartz & Zhang, 2013; Rakhlin et al., 2012)), as well as a subset of the MNIST digit recognition dataset which focuses on discriminating the 4 from the 7 digits[6]. In figure 3, we present the number of iterations required for DANE to reach accuracy $< 10^{-6}$ for $\eta = 1$ and $\mu = 0, 3\lambda$, and for different number of machines. We also report results for ADMM on the same datasets. As in the synthetic case, DANE explicitly takes advantage of the similarity between problems on different machines, and we indeed observe that it tends to converge in less iterations than ADMM. Finally, note that for $\mu = 0$ and many machines (i.e. few data points per machine), DANE may not converge, and increasing μ fixes this at the cost of slowing down the average convergence rate.

Finally, we examine the convergence on these datasets in terms of the average loss on the test set. In figure 4, we present the results for $m = 64$ machines on the three datasets, using DANE (with $\mu = 3\lambda$) and ADMM. We also present for comparison the objective value obtained using one-shot parameter averaging (OSA), using bias correction as proposed in (Zhang et al., 2013). The figure highlights the practical importance of multi-round communication algorithms: while DANE and ADMM converge to the value achieved by the regularized loss minimizer, the single-round OSA algorithm may return a significantly suboptimal result.

Acknowledgements: Ohad Shamir and Nathan Srebro are supported by the Intel ICRI-CI Institute. Ohad Shamir is further supported by an Israel Science Foundation grant 425/13 and an FP7 Marie Curie CIG grant.

[5] To be fair, ADMM performs a single distributed averaging computation per iteration, while DANE performs two. However, counting iterations is a more realistic measure of performance, since both methods also perform a full-scale local optimization at each iteration.

[6] We used $\lambda = 10^{-5}$ for COV1, $\lambda = 0.0005$ for ASTRO and $\lambda = 0.001$ for MNIST-47. For MNIST-47, we randomly chose 10,000 examples as the training set, and the rest of the examples as a test set.

References

Agarwal, A., Chapelle, O., Dudík, M., and Langford, J. A reliable effective terascale linear learning system. *CoRR*, abs/1110.4198, 2011.

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.

Bekkerman, R., Bilenko, M., and Langford, J. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

Boyd, S.P., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Cotter, A., Shamir, O., Srebro, N., and Sridharan, K. Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, 2011.

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13: 165–202, 2012.

Deng, W. and Yin, W. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, Rice University Technical Report TR12-14, 2012.

Duchi, J., Agarwal, A., and Wainwright, M. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Automat. Contr.*, 57(3): 592–606, 2012.

Hong, M. and Luo, Z.-Q. On the linear convergence of the alternating direction method of multipliers. *CoRR*, abs/1208.3922, 2012.

Mahajan, D., Keerthy, S., Sundararajan, S., and Bottou, L. A parallel sgd method with strong convergence. *CoRR*, abs/1311.0636, 2013.

Nemirovski, A. and Yudin, D. On cesaro's convergence of the gradient descent method for finding saddle points of convex-concave functions. *Doklady Akademii Nauk SSSR*, 239(4), 1978.

Nemirovsky, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

Recht, B., Re, C., Wright, S., and Niu, F. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

Richtárik, P. and Takác, M. Distributed coordinate descent method for learning with big data. *CoRR*, abs/1310.2059, 2013.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *COLT*, 2009.

Sridharan, K., Shalev-Shwartz, S., and Srebro, N. Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems*, pp. 1545–1552, 2008.

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

Yang, T. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *NIPS*, 2013.

Zhang, Y., Duchi, J., and Wainwright, M. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

Zinkevich, M., Weimer, M., Smola, A., and Li, L. Parallelized stochastic gradient descent. In *NIPS*, pp. 2595–2603, 2010.