Large Scale Empirical Risk Minimization via Truncated Adaptive Newton Method

Mark Eisen
University of Pennsylvania

Aryan Mokhtari
Massachusetts Institute of Technology

Alejandro Ribeiro
University of Pennsylvania

Abstract

Most second order methods are inapplicable to large scale empirical risk minimization (ERM) problems because both, the number of samples N and number of parameters p are large. Large N makes it costly to evaluate Hessians and large p makes it costly to invert Hessians. This paper propose a novel adaptive sample size second-order method, which reduces the cost of computing the Hessian by solving a sequence of ERM problems corresponding to a subset of samples and lowers the cost of computing the Hessian inverse using a truncated eigenvalue decomposition. Although the sample size is grown at a geometric rate, it is shown that it is sufficient to run a single iteration in each growth stage to track the optimal classifier to within its statistical accuracy. This results in convergence to the optimal classifier associated with the whole set in a number of iterations that scales with $\log(N)$. The use of a truncated eigenvalue decomposition result in the cost of each iteration being of order p^2. Theoretical performance gains manifest in practical implementations.

1 Introduction

A fundamental tension in learning is between the problem we would like to solve, which is the minimization (M) of a statistical risk (SR), and the problem we actually can solve, which is the minimization (M) of an empirical risk (ER). Indeed, it is customary to define classifiers or regressors as the optimal argument of a stochastic program formulated with respect to the probability distribution of the data. This distribution is unknown but it is possible to acquire N independent training samples and formulate an ERM problem to approximate the statistical program. The solutions of SRM and ERM are not equivalent but their distance – the statistical accuracy – vanishes as N grows [1, 4, 34].

When it comes to finding solutions of ERM problems, most existing methods do not exploit the connection between statistical and empirical risk minimization. Workhorse stochastic gradient descent methods [3, 27] as well as the more recent variance reduction [7, 12, 24, 30] and quasi-Newton [6, 18, 19, 31] variants work just as well for any finite sum minimization problem. This is not necessarily a drawback but it is nonetheless true that not exploiting the connection between SRM and ERM leaves some performance gains on the table. Adaptive sample size methods attempt to collect these gains [16, 20].

To explain adaptive sample size methods, suppose the statistical accuracy of an ERM formulation with n samples is $1/n$. If we consider two ERM formulations with n and $2n$ samples, the difference between the corresponding minimizers is of order $3/2n$ from the triangle inequality, because each is within its respective statistical accuracy. Thus, when doubling the size of the training set, the respective solutions are not far from each other when measured relative to the statistical accuracy; a fact that holds *irrespectively* of n. We can then use the solution of the ERM problem for n samples as a warm start for solving the problem with $2n$ samples. Since the optima are close, we can expect to solve the ERM problem with $2n$ samples in a few iterations and keep progressing *geometrically* growing the sample size. This conceptual argument can be formalized to show that adaptive sample size methods yield meaningful reductions in the number of operations that are needed for convergence relative to those of conventional stochastic optimization [20].

Given that adaptive sample size methods solve what is effectively a sequence of deterministic problems they open up the opportunity to utilize Newton's method in ERM [16] – which is hampered in stochastic optimization by the difficulty of computing unbiased estimates of Newton steps. For a problem of dimension p, each Newton iteration would have a cost of order Np^2 to compute a Hessian and of order p^3 to invert it. However, since subsequent optima are close, we need to run just a few iterations at each growth stage and since we are growing the sample size geometrically we

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

only need in the order of $\log(N)$ iterations. In this paper we propose a novel second-order adaptive sample size method that uses a truncated eigenvalue decomposition to reduce the cost of each Newton iteration to order p^2 and derive conditions to permit geometric growth of the sample set while running a *single* iteration at each growth stage. This conditions roughly imply the ability to double the sample size so that the total number of iterations needed to find an optimal classifier is of order $\log_2 N$. We reemphasize that each of these iterations incurs a cost of order np^2 to compute a Hessian and of order p^2 to invert it.

The truncated eigenvalue decomposition that we use here keeps the eigenvectors associated with the k largest eigenvalues and ignores the remaining $p-k$. The advantages of the resulting k-TAN method rely on the ability to use small k. This is possible because when we approximate the statistical loss with the empirical loss we induce an error on the order of statistical accuracy. Any further error induced by an eigenvalue truncation is negligible. Many eigenvalues in a high dimensional problems are small and thus contribute little to the Newton direction. We can therefore safely discard all small eigenvalues that induce additional error of the same order as the statistical accuracy, providing potentially significant reduction in computational cost.

1.1 Related Work

Adaptive sample size methods grow the sample size geometrically. This is a departure from conventional stochastic descent methods that process one sample per iteration [3, 27]. The same holds true for Nesterov-based methods [2, 23], variance reduction [12, 24], stochastic average gradient [7, 30], stochastic majorization-minimization [8, 15], hybrid [13], and dual coordinate methods [32, 33]. The philosophy also differs from stochastic second order methods such as subsampled Newton [9, 25, 28, 29], incremental Hessian [10], stochastic dual Newton ascent [26], and stochastic quasi-Newton methods [14, 17–19, 21, 31]. These methods utilize second order information but fail to achieve and exploit quadratic convergence rates because they are eventually dominated by noise.

As is the case of the truncated adaptive Newton (k-TAN) method that we propose here, the adaptive (Ada) sample size Newton method is such that a single iteration suffices each time the sample size grows – this also follows as a particular case of the results presented here with $k = p$ and is proven in [16]. First order adaptive sample size methods are analyzed in [20]. First order methods are applicable when $p \gg N$, Ada Newton is applicable when $p < N$, and k-TAN fills the intermediate niche of $p > N$ but $p \not\gg N$.

2 Problem Formulation

We consider in this paper the empirical risk minimization (ERM) problem for a convex function $f(\mathbf{x}, z)$, where z is a realization of a random variable Z. More specifically, we seek the optimal variable $\mathbf{x} \in \mathbb{R}^p$ that minimizes the expected loss $L(\mathbf{x}) := \mathbf{E}_Z[f(\mathbf{x}, z)]$. Define $\mathbf{x}^*$ as the variable that minimizes the expected loss, i.e.

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} \mathbf{E}_Z[f(\mathbf{x}, z)]. \tag{1}$$

In general, the problem in (1) cannot be solved without knowing the distribution of Z. As an alternative, we traditionally consider the case that we have access to N samples of Z, labelled $z_1, z_2, \ldots, z_n$. Define then the functions $f_i(\mathbf{x}) = f(\mathbf{x}, z_i)$ for $i = 1, 2, \ldots, N$ and an associated empirical risk function $L_n := (1/n)\sum_{i=1}^n f_i(\mathbf{x})$ as the statistical mean over the first $n \leq N$ samples. We say that function $L_n(\mathbf{x})$ approximates the original expected loss $L(\mathbf{x})$ with statistical accuracy V_n if the difference between the empirical risk function $L_n(\mathbf{x})$ and the expected loss $L(\mathbf{x})$ is upper bounded by V_n for all $\mathbf{x}$ with high probability (w.h.p.). The statistical accuracy V_n is typically bounded by $V_n = \mathcal{O}(1/\sqrt{n})$ [34] or the stronger $V_n = \mathcal{O}(1/n)$ for a set of common problems [1, 4].

Observe that the sampled loss function L_n is of an order V_n difference from the true loss function L and, consequently, any additional change of the same order has negligible effect. It is therefore common to regularize non-strongly convex loss functions L_n by a term of order V_n. We then seek the minimum argument of the regularized risk function R_n,

$$\mathbf{x}_n^* := \operatorname*{argmin}_{\mathbf{x}} R_n(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}} \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}) + \frac{cV_n}{2}\|\mathbf{x}\|^2, \tag{2}$$

where c is a scalar constant. The solution $\mathbf{x}_n^*$ minimizes the regularized risk function using the first n samples, which is of order V_n from the expected loss function L. It follows then that by setting $n = N$ we find a solution $\mathbf{x}_N^*$ in (2) that solves the original problem in (1) up to the statistical accuracy V_N of using all N samples.

The problem in (2) is strongly convex and can be solved using any descent method. In particular, Newton's method uses a curvature-corrected gradient to iteratively update a variable $\mathbf{x}$, and is known to converge to the optimal argument $\mathbf{x}_n^*$ at a very fast quadratic rate. To implement Newton's method, it is necessary to compute the gradient $\nabla R_n(\mathbf{x})$ and Hessian $\nabla^2 R_n(\mathbf{x})$ as

$$\nabla R_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}) + cV_n\mathbf{x}, \tag{3}$$

$$\nabla^2 R_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(\mathbf{x}) + cV_n\mathbf{I}. \tag{4}$$

The variable $\mathbf{x}$ is updated in Newton's method as

$$\mathbf{x}^+ = \mathbf{x} - \nabla^{-2} R_n(\mathbf{x}) \nabla R_n(\mathbf{x}). \tag{5}$$

Solving (1) to the full statistical accuracy V_N (i.e. solving (2) for $n = N$) using Newton's method would then require the computation of individual gradients and Hessians for N functions f_i for computational cost of $\mathcal{O}(Np^2)$ at each iteration. Furthermore, the computation of the Hessian inverse in (5) requires a cost of $\mathcal{O}(p^3)$, bringing at total of $\mathcal{O}(Np^2 + p^3)$ for an iteration of Newton's method using the whole dataset. In addition, the initial iterate may be far from the optimal solution and, in this case, a line search method is needed, which requires computation of the function value multiple times. Finally, the algorithm suffers from a slow sublinear convergence rate outside the local quadratic convergence region. These issues make Newton's method computationally infeasible when both N and p are large. In this paper we show how this complexity can be reduced by gradually increasing the sample size n and approximating the inverse of the respective Hessian $\nabla^2 R_n(\mathbf{x})$.

3 k-Truncated Adaptive Newton (k-TAN) Method

We propose the k-Truncated Adaptive Newton (k-TAN) as a low cost alternative to solving (1) to its statistical accuracy. In the k-TAN method, at each iteration we start from a point $\mathbf{x}_m$ within the statistical accuracy of R_m, i.e. $R_m(\mathbf{x}_m) - R_m(\mathbf{x}_m^*) \leq V_m$. We geometrically increase the sample size to $n = \alpha m$, where $\alpha > 1$, and compute $\mathbf{x}_n$ using an approximated Newton method on the increased sample size risk function R_n. More specifically, we update a decision variable $\mathbf{x}_m$ associated with R_m to a new decision variable $\mathbf{x}_n$ associated with R_n with the Newton-type update

$$\mathbf{x}_n = \mathbf{x}_m - \hat{\mathbf{H}}_{n,k}^{-1} \nabla R_n(\mathbf{x}_m), \tag{6}$$

where $\hat{\mathbf{H}}_{n,k}$ is a matrix approximating the Hessian $\nabla^2 R_n(\mathbf{x}_m)$ and parametrized by $k \in \{1, 2, \ldots, p\}$. In particular, we are interested in an approximation matrix $\hat{\mathbf{H}}_{n,k}$ whose inverse $\hat{\mathbf{H}}_{n,k}^{-1}$ can be computed with complexity less than $\mathcal{O}(p^3)$ and is a good approximation for the true Hessian inverse $\nabla^2 R_n(\mathbf{x}_m)^{-1}$. To define such a matrix, consider $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_p$ to be the eigenvalues of the Hessian of empirical risk $\nabla^2 L_n(\mathbf{x}_m)$, with associated eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$. We perform an eigenvalue decomposition of $\nabla^2 L_n(\mathbf{x}_m) = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, where $\mathbf{U} := [\mathbf{v}_1, \ldots, \mathbf{v}_p] \in \mathbb{R}^{p \times p}$ and $\mathbf{\Sigma} := \text{diag}(\mu_1, \ldots, \mu_p) \in \mathbb{R}^{p \times p}$. We can then define the truncated eigenvalue decomposition with rank k as $\hat{\nabla}^2 L_n(\mathbf{x}_m) := \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{U}_k^T$, where $\mathbf{U}_k := [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{p \times k}$ and $\mathbf{\Sigma}_k := \text{diag}(\mu_1, \ldots, \mu_k) \in \mathbb{R}^{k \times k}$. The full approximated Hessian $\hat{\mathbf{H}}_{n,k}$ is subsequently defined as the rank k approximation of $\nabla^2 L_n(\mathbf{x}_m)$ regularized by $cV_n\mathbf{I}$, i.e.,

$$\hat{\mathbf{H}}_{n,k} := \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{U}_k^T + cV_n \mathbf{I}. \tag{7}$$

The inverse of the approximated Hessian $\hat{\mathbf{H}}_{n,k}$ can then be computed directly using $\mathbf{U}_k$ and $\mathbf{\Sigma}_k$ as

$$\hat{\mathbf{H}}_{n,k}^{-1} := \mathbf{U}_k [(\mathbf{\Sigma}_k + cV_n \mathbf{I})^{-1} - (cV_n \mathbf{I})^{-1}] \mathbf{U}_k^T + (cV_n)^{-1} \mathbf{I}. \tag{8}$$

Observe that setting $k = p$ leads to exact Hessian inverse computation, i.e., $\hat{\mathbf{H}}_{n,k}^{-1} = \nabla^2 R_n(\mathbf{x}_m)^{-1}$, and recovers the AdaNewton method in [16]. To understand how we may determine k, consider that the full Hessian computed in (3) is $\nabla^2 L_n(\mathbf{x}_m)$ regularized by $cV_n\mathbf{I}$. Therefore, the eigenvalues of $\nabla^2 L_n(\mathbf{x}_m)$ less than cV_n are made negligible by the regularization, and can be left out of the approximation. We thus select the k largest eigenvalues of the Hessian which are larger than $\rho c V_n$ for some truncation parameter $0 < \rho < 1$.

To analyze the computational complexity of (8), observe that the inverse computation in (8) requires only the inversion of diagonal matrices, and thus the primary cost in computing the k largest eigenvalues $\mathbf{\Sigma}_k$ and associated eigenvectors $\mathbf{U}_k$. Indeed, the truncated eigenvalue decomposition $\{\mathbf{U}_k, \mathbf{\Sigma}_k\}$ can in general be computed with at most complexity $\mathcal{O}(kp^2)$, with recent randomized algorithms even finding $\{\mathbf{U}_k, \mathbf{\Sigma}_k\}$ with complexity $\mathcal{O}(p^2 \log k)$ [11]. This results in a total cost of, at worst, $\mathcal{O}((\log k + n)p^2)$ to perform the update in (6), thus removing a $\mathcal{O}(p^3)$ cost.

In this paper we aim to show that while we geometrically increase the size of the training set, a single iteration of the truncated Newton method in (6) is sufficient to solve the new risk function within its statistical accuracy. To state this result we first assume the following assumptions hold.

Assumption 1 *The loss functions $f(\mathbf{x}, \mathbf{z})$ are convex with respect to $\mathbf{x}$ for all values of $\mathbf{z}$. Moreover, their gradients $\nabla f(\mathbf{x}, \mathbf{z})$ are Lipschitz continuous with constant M.*

Assumption 2 *The loss functions $f(\mathbf{x}, \mathbf{z})$ are self-concordant with respect to $\mathbf{x}$ for all $\mathbf{z}$.*

Assumption 3 *Define the local region of L_n as $\mathcal{L}_n := \{\mathbf{x} \mid L_n(\mathbf{x}) - \min_{\mathbf{x}} L_n(\mathbf{x}) \leq V_n\}$ The difference between the gradients of the empirical loss L_n and the statistical average loss L is bounded by $V_n^{1/2}$ for all $\mathbf{x} \in \mathcal{L}_n$ with high probability,*

$$\sup_{\mathbf{x} \in \mathcal{L}_n} \|\nabla L(\mathbf{x}) - \nabla L_n(\mathbf{x})\| \leq V_n^{1/2}, \qquad \textit{w.h.p.} \tag{9}$$

Based on Assumption 1, the regularized empirical risk gradients ∇R_n are Lipschitz continuous with constant $M + cV_n$. Assumption 2 states the loss functions are additionally self concordant which is a customary assumption in the analysis of second-order methods. It also follows that the functions R_n are therefore self concordant. Assumption 3 bounds the difference between gradients of the expected loss and the empirical loss with n samples by $V_n^{1/2}$ for points local to the optimum. This bound is reasonable for the convergence

of gradients to their statistical averages due to the law of large numbers.

We are interested in establishing the result that, as we increase n at each step, the k-TAN method stays in the quadratic region of the the associated risk function R_n. While we are after all more interested in being within the statistical accuracy of expected risk L, indeed being within the statistical accuracy of R_n further implies being within the statistical accuracy of L. More explicitly, we wish to show the sample size can be increased from m to $n = \alpha m$ such that $\mathbf{x}_m$ is in the quadratic region of R_n. Moreover, if $\mathbf{x}_m$ is indeed in the quadratic region of R_n, then we demonstrate that a single step of k-TAN as in (6) produces a point $\mathbf{x}_n$ that is within the statistical accuracy V_n of the risk R_n. We formalize these condition in the following theorem.

Theorem 1 *Consider the k-TAN method defined in* (6)-(8) *and suppose that the constant k for low rank factorization is defined as $k = \min\{k \mid \mu_{k+1} \leq \rho c V_n\}$ where ρ is a free parameter chosen from the interval $(0, 1]$. Further consider the variable $\mathbf{x}_m$ as a V_m-optimal solution of the risk R_m, i.e., a solution such that $R_m(\mathbf{x}_m) - R_m(\mathbf{x}_m^*) \leq V_m$. Let $n = \alpha m > m$ and suppose Assumptions* (1)-(3) *hold. If the sample size n is chosen such that*

$$\left(\frac{2(M + cV_m)V_m}{cV_n}\right)^{1/2} + \frac{2(n-m)}{nc^{1/2}} + \frac{\left((2+\sqrt{2})c^{1/2} + c\|\mathbf{x}^*\|\right)(V_m - V_n)}{(cV_n)^{1/2}} \leq \frac{1}{4} \qquad (10)$$

and

$$\frac{16}{(3-\rho)^4}\Big[36K^2(1+\rho)^2V_m^2 + 30K^{3/2}\rho(1+\rho)V_m^{3/2} + 6K\rho^2 V_m\Big] \leq V_n \qquad (11)$$

are satisfied, where $K = 3 + c/2\|\mathbf{x}^*\|^2(1 - 1/\sqrt{\alpha})$., then the variable $\mathbf{x}_n$ computed from (6) *has the suboptimality of V_n with high probability, i.e.,*

$$R_n(\mathbf{x}_n) - R_n(\mathbf{x}_n^*) \leq V_n, \qquad \textit{w.h.p.} \qquad (12)$$

The result in Theorem 1 establishes the required conditions to guarantee that the iterates $\mathbf{x}_n$ always stay within the statistical accuracy of the risk R_n. The expression in (10) provides a condition on growth rate $\alpha = n/m$ to ensures that iterate $\mathbf{x}_m$, which is a V_m-suboptimal solution for R_m, is within the quadratic convergence neighborhood of Newton's method for R_n. The second condition in (11) ensures that a single iteration of k-TAN is sufficient for the updated variable $\mathbf{x}_n$ to be within the statistical accuracy of R_n. Note that the first term in the left hand side of (11) is quadratic with respect to V_m and comes from the quadratic convergence of Newton's method, while the second and third terms of respective orders $V_m^{3/2}$ and V_m are the outcome of Hessian approximation. Indeed, these terms depend on ρ, which is the upper bound on ratio of the discarded eigenvalues $\mu_{k+1}, \ldots, \mu_p$ to the regularization cV_n. The truncation must be enough such that ρ is sufficiently small to make (11) hold. It is worth mentioning, as a sanity check, if we set $\rho = 0$ then we will keep all the eigenvalues and recover the update of Newton's method which makes the non-quadratic terms in (11) zero.

The conditions in Theorem 1 are cumbersome but can be simplified if we focus on large m and assume that the inequality $V_m \leq \alpha V_n$ holds for $n = \alpha m$. Then, (10) and (11) can be simplified to

$$\left(\frac{2\alpha M}{c}\right)^{1/2} + \frac{2(\alpha - 1)}{\alpha c^{1/2}} \leq \frac{1}{4}, \qquad (13)$$

$$\frac{96[3 + c/2\|\mathbf{x}^*\|^2(1 - 1/\sqrt{\alpha})]\rho^2}{(3-\rho)^2} \leq \frac{1}{\alpha}, \qquad (14)$$

respectively. Observe that first condition is dependent of α and the second condition depends on α and ρ. Thus, a pair (α, ρ) must be chosen that satisfies (13) for the result in Theorem 1 to hold. We point out if c is chosen such that $c > 16(2\sqrt{M} + 1)^2$, then one such pair is $\alpha = 2$ and $\rho \leq 9/(21\sqrt{c\|\mathbf{x}^*\|^2 + 16} + 3)$. Consequently, when m is large, by choosing small enough ρ, we may double the sample size with each update in until $n = N$, after which we will have obtained a point $\mathbf{x}_N$ such that $R_N(\mathbf{x}_N) - R_N(\mathbf{x}_N^*) \leq V_N$. After $\log_2 N$ iterations (roughly $2N$ samples processed), we solve the full risk function R_N to within the statistical accuracy V_N. At each iteration, the truncated inverse step requires cost $\mathcal{O}(p^2 \log k)$. Computing Hessians over $2N$ samples requires cost $\mathcal{O}(2Np^2)$, resulting in a total complexity of $\mathcal{O}(p^2(2N + \log_2 N \log k))$.

In practice, these may be chosen in a backtracking manner, in which the iterate $\mathbf{x}_m$ is updated using an estimate (α, ρ) pair. If the resulting iterate $\mathbf{x}_n$ is not in statistical accuracy $R_n(\mathbf{x}_n) - R_n(\mathbf{x}_n^*) \leq V_n$, the increase factor α is decreased by factor $\beta < 1$ and ρ is decreased by factor $\delta < 1$. Since $R_n(\mathbf{x}_n^*)$ is not known in practice, the suboptimality can be upper bounded using strong convexity as $R_n(\mathbf{x}_n) - R_n(\mathbf{x}_n^*) \leq \|\nabla R_n(\mathbf{x}_n)\|^2/(2cV_n)$.

The resulting method is presented in Algorithm 1. After preliminaries and initializations in Steps 1-4, the backtracking loop starts in Step 6 with the sample size increase by rate α. The gradient is computed in Step 7 and the Hessian $\nabla^2 L_n$ and its low rank approximation $\hat{\nabla}^2 L_n$ are evaluated in Step 8. Then, the Hessian inverse approximation is computed according to (8) in Step 9 to perform the update of k-TAN in Step 10, based on (6). The factors α and ρ are then decreased using the backtracking parameters and the statistical accuracy condition is checked. We stress that, while $\nabla R_n(\mathbf{x}_n)$ must be computed to check the exit condition

Algorithm 1 k-TAN

1: **Parameters:** $\alpha_0 > 1$, $\rho_0 < 1$, and $0 < \beta, \delta < 1$
2: **Input:** Initial sample size $n = m_0$ and argument $\mathbf{x}_n = \mathbf{x}_{m_0}$ with $\|\nabla R_n(\mathbf{x}_n)\| < (\sqrt{2c})V_n$
3: **while** $n \leq N$ **do** {main loop}
4: Update argument and index: $\mathbf{x}_m = \mathbf{x}_n$ and $m = n$. Reset factor $\alpha = \alpha_0$, $\rho = \rho_0$.
5: **repeat** {sample size backtracking loop}
6: Increase sample size: $n = \min\{\alpha m, N\}$.
7: Compute gradient [cf. (3)]: $\nabla R_n(\mathbf{x}_m) = \frac{1}{n}\sum_{i=1}^n \nabla f(\mathbf{x}_m, z_i) + cV_n\mathbf{x}_m$
8: Compute $\nabla^2 L_n = \frac{1}{n}\sum_{i=1}^n \nabla^2 f(\mathbf{x}_m, z_i)$ and find the low rank decomposition $\hat{\nabla}^2 L_n$ [11]: $\hat{\nabla}^2 L_n = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{U}_k^T, \;\; k = \min\{k|\mu_{k+1} \leq \rho c V_n\}$
9: Compute regularized Hessian inverse [cf. (8)]: $\hat{\mathbf{H}}_{n,k}^{-1} = \mathbf{U}_k[(\mathbf{\Sigma}_k + cV_n\mathbf{I})^{-1} - \frac{1}{cV_n}\mathbf{I}]\mathbf{U}_k^T + \frac{1}{cV_n}\mathbf{I}$
10: Compute Newton update [cf. (6)]: $\mathbf{x}_n = \mathbf{x}_m - \hat{\mathbf{H}}_{n,k}^{-1}\nabla R_n(\mathbf{x}_m)$
11: Backtrack sample rate $\alpha = \beta\alpha$, truncation $\rho = \delta\rho$.
12: **until** $\|\nabla R_n(\mathbf{x}_n)\| < (\sqrt{2c})V_n$
13: **end while**

in Step 12, the gradient for these samples must be computed in any case in the following iteration, so no additional computation is added by this step.

4 Convergence Analysis

In this section, we study the convergence properties of the k-TAN method and in particular prove the result in Theorem 1.

4.1 Preliminaries

Before proceeding with the analysis of k-TAN, we first present two propositions that relate current iterate $\mathbf{x}_m$ to the suboptimality and quadratic convergence region to the increased sample size risk R_n. Define $S_n(\mathbf{x})$ to be the n-suboptimality of point $\mathbf{x}$ with respect to R_n, i.e.

$$S_n(\mathbf{x}) := R_n(\mathbf{x}) - R_n(\mathbf{x}_n^*), \tag{15}$$

where $\mathbf{x}_n^*$ is the point that minimizes R_n. In the following proposition, we establish a bound on $S_n(\mathbf{x}_m)$ in terms of the statistical accuracy of R_m, i.e., V_m.

Proposition 2 *Consider a point $\mathbf{x}_m$ that minimizes the risk function R_m to within its statistical accuracy V_m, i.e. $S_m(\mathbf{x}_m) \leq V_m$. If the sample size is increased from m to $n = \alpha m$ and $(1/\sqrt{\alpha})V_m \leq V_n \leq (1/\alpha)V_m$, w.h.p the sub-optimality $S_n(\mathbf{x}_m)$ is upper bounded by*

$$S_n(\mathbf{x}_m) \leq \left[3 + \frac{c}{2}\|\mathbf{x}^*\|^2\left(1 - \frac{1}{\sqrt{\alpha}}\right)\right]V_m. \tag{16}$$

Proposition 2 demonstrates a bound on the n-suboptimality $S_n(\mathbf{x}_m)$ of a point $\mathbf{x}_m$ whose m-suboptimality $S_m(\mathbf{x}_m)$ is within statistical accuracy V_m. We stress that the condition $(1/\sqrt{\alpha})V_m \leq V_n \leq (1/\alpha)V_m$ holds for both the standard cases of $V_n = \mathcal{O}(1/n)$ and $V_n = \mathcal{O}(1/\sqrt{n})$. It is also necessary to establish conditions on increase rate α such the $\mathbf{x}_m$ is also in the quadratic convergence region of R_n. Traditional analysis of Newton's method characterizes quadratic convergence in terms of the Newton decrement $\lambda_n(\mathbf{x}) := \|\nabla^2 R_n(\mathbf{x})^{-1/2}\nabla R_n(\mathbf{x})\|$. The iterate $\mathbf{x}$ is said to be in the quadratic convergence region of R_n when $\lambda_n(\mathbf{x}) < 1/4$—see [5, Chapter 9.6.4]. The conditions for current iterate $\mathbf{x}_m$ to be within this region are presented in the following proposition. The proof can be found in [16, Proposition 3].

Proposition 3 *Define $\mathbf{x}_m$ as an V_m optimal solution of the risk R_m, i.e., $R_m(\mathbf{x}_m) - R_m(\mathbf{x}_m^*) \leq V_m$. In addition, define $\lambda_n(\mathbf{x}) := \left(\nabla R_n(\mathbf{x})^T\nabla^2 R_n(\mathbf{x})^{-1}\nabla R_n(\mathbf{x})\right)^{1/2}$ as the Newton decrement of variable $\mathbf{x}$ associated with the risk R_n. If Assumption 1-3 hold, then Newton's method at point $\mathbf{x}_m$ is in the quadratic convergence phase for the objective function R_n, i.e., $\lambda_n(\mathbf{x}_m) < 1/4$, if we have*

$$\left[\frac{2(M + cV_m)V_m}{cV_n}\right]^{1/2} + \frac{2(n-m)}{nc^{1/2}} + \frac{(\sqrt{2c} + 2\sqrt{c} + c\|\mathbf{x}^*\|)(V_m - V_n)}{(cV_n)^{1/2}} \leq \frac{1}{4}, \quad \textit{w.h.p.} \tag{17}$$

4.2 Analysis of k-TAN

To analyze the k-TAN method, it is necessary to study the error incurred from approximating the Hessian inverse in (7) with rank k. Since we are only interested in solving each risk function R_n to within its statistical accuracy V_n, some approximation error can be afforded. In the following lemma, we characterize the error between an approximate and exact Newton steps using the chosen rank k of the approximation and the associated eigenvalues of the Hessian.

Lemma 4 *Consider the k-TAN update in* (6)-(8) *for some $k = \{0, 1, \ldots, p\}$. Define $\epsilon_n := \mu_{k+1}/(cV_n)$. Considering the notation $\mathbf{H}_n := \hat{\mathbf{H}}_{n,p} = \nabla^2 R_n(\mathbf{x}_m)$, it holds*

$$\|\hat{\mathbf{H}}_{n,k}^{-1}\nabla R_n(\mathbf{x}_m) - \mathbf{H}_n^{-1}\nabla R_n(\mathbf{x}_m)\| \leq \epsilon_n\|\mathbf{H}_n^{-1}\nabla R_n(\mathbf{x}_m)\|. \tag{18}$$

Proof: Check the supplementary material. ■

The result in Lemma 4 gives us an upper bound on the error incurred in single iteration of a rank k approximation of the

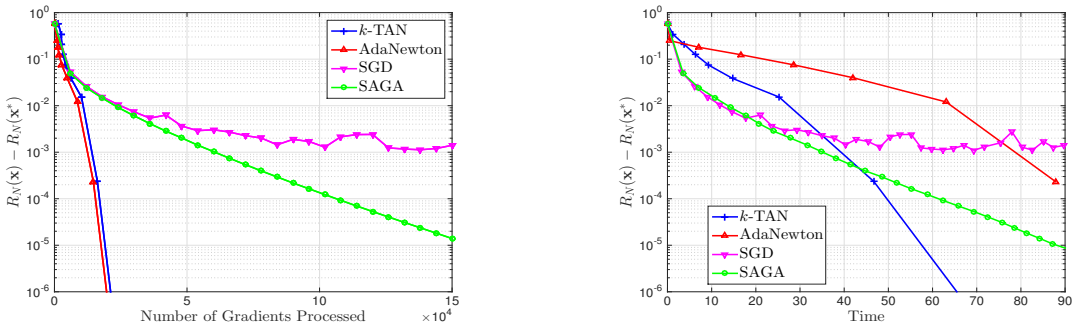

Figure 1: Convergence of k-TAN, AdaNewton, SGD, and SAGA in terms of number of processed gradients (left) and runtime (right) for the GISETTE handwritten digit classification problem.

Newton step versus an exact Newton step. To make ϵ_n small, a sufficiently large k must be chosen such that μ_{k+1} is in the order of V_n. The size of k will therefore depend on the distribution of the eigenvalues of particular empirical risk function. However, in practical datasets of high dimension, it is often the case that most eigenvalues of the Hessian will be close to 0, in which case k can be made very small. This trend is supported by our numerical experiments on real world data sets in Section 5 and the Appendix of this paper.

With the results of Proposition 2 and Lemma 4 in mind, we can characterize the n-suboptimality of the updated variable $\mathbf{x}_n$ from (6). This is stated formally in the following Lemma.

Lemma 5 *Consider the k-TAN update in* (6)-(8). *If $\mathbf{x}_m$ is in the quadratic neighborhood of $\mathbf{R}_n$, i.e. $\lambda_n(\mathbf{x}_m) < 1/4$, then the n-suboptimality $S_n(\mathbf{x}_n) = R_n(\mathbf{x}_n) - R(\mathbf{x}_n^*)$ can be upper bounded by*

$$S_n(\mathbf{x}_n) \leq \frac{16}{(3-\epsilon_n)^4}[36(1+\epsilon_n)^2 S_n(\mathbf{x}_m)^2 + 30\epsilon_n(1+\epsilon_n)S_n(\mathbf{x}_m)^{3/2} + 6\epsilon_n^2 S_n(\mathbf{x}_m)]. \tag{19}$$

With Lemma 5 we establish a bound on n-suboptimality $S_n(\mathbf{x}_n)$ of the $\mathbf{x}_n$ obtained from the k-TAN update in (6). Note that $S_n(\mathbf{x}_n)$ is bounded by terms proportional to the n-suboptimality of the previous point, $S_n(\mathbf{x}_m)$. We can then establish that $S_n(\mathbf{x}_n)$ is indeed upper bounded by the statistical accuracy V_n if we combine the results in (16) and (19) to obtain Theorem 1.

To be more precise, from Proposition 3 the condition in (10) ensures that $\mathbf{x}_m$ will be in the quadratic region of R_n, i.e., $\lambda_n(\mathbf{x}_m) < 1/4$. Now according to the result in Lemma 5, the conditions required for (19) are satisfied and this result holds. From Proposition 2 we can bound the n-suboptimality of the previous iterate $S_n(\mathbf{x}_m)$ by a constant multiply by KV_m, where K is defined as $K := 3 + \frac{c}{2}\|\mathbf{x}^*\|^2(1 - 1/\sqrt{\alpha})$.

For notational simplicity, we focus on the case in which the statistical accuracy is $V_m = \mathcal{O}(1/m)$, as in (16). Furthermore, based on the definition in Lemma 4, we can replace the truncation error ϵ_n by $\mu_{k+1}/(cV_n)$. Also, if ρ is chosen such that $\mu_{k+1} \leq \rho c V_n$, we can conclude that ϵ_n is bounded above by ρ. Substituting these bounds into (19) yields

$$S_n(\mathbf{x}_n) \leq \frac{16}{(3-\rho)^4}[36K^2(1+\rho)^2 V_m^2 + 30K^{3/2}\rho(1+\rho)V_m^{3/2} + 6K\rho^2 V_m]. \tag{20}$$

Therefore, if the condition in (11) is satisfied, the result in (20) leads to $S_n(\mathbf{x}_n) \leq V_n$ and the claim in (12) follows.

5 Experiments

We compare the performance of the k-TAN method to existing optimization methods on large scale machine learning problems of practical interest. In particular, we consider a regularized logistic loss function, with regularization parameters $V_n = 1/n$ and $c = 1$. The k-TAN method is compared against the second order method AdaNewton [16] and two first order methods—SGD and SAGA [7]. Here, we study the performance of these methods on the logistic regression problem for multiple datasets. First, the GISETTE handwritten digit classification from the NIPS 2003 feature selection challenge and, second, the RCV1 dataset for classifying news stories from the Reuters database. Further experiments on the KDD Cup 2009 and KDD Cup 2004 competition datasets (the latter is included in the supplementary material). In all experiments, the optimal value $R_N(\mathbf{x}^*)$ was found a priori using full batch gradient descent for the purposes of evaluating suboptimality of the methods.

The GISETTE dataset includes $N = 6000$ samples of dimension $p = 5000$. We use a constant step size of 0.08 for SAGA and a diminishing step size for SGD. In both k-TAN and AdaNewton, the sample size is increased by a factor of $\alpha = 2$ at each iteration (the condition $\|\nabla R_n(\mathbf{x}_n)\| < (\sqrt{2c})V_n$ is always satisfied) starting with an initial size of $m_0 = 124$. For both of these methods, we initially run gradient descent on R_{m_0} for 100 iterations so that we may begin in the statistical accuracy V_{m_0}. For k-TAN, the truncation k is observed to be able to afford a cutoff of around $0.01p$ in all of our simulations.

In Figure 1, the convergence results of the four methods

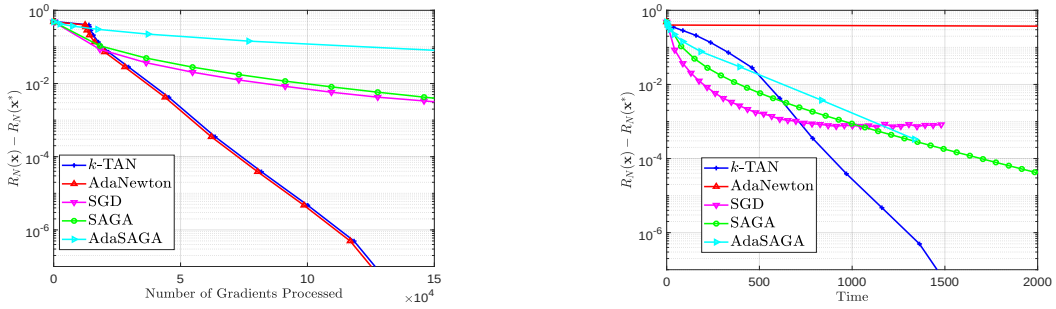

Figure 2: Convergence of k-TAN, AdaNewton, SGD, and SAGA, and AdaSAGA in terms of number of processed gradients (left) and runtime (right) for the RCV1 text classification problem.

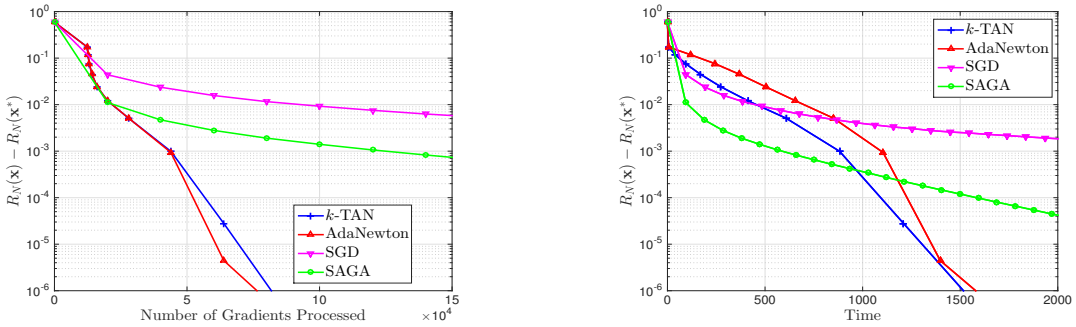

Figure 3: Convergence of k-TAN, AdaNewton, SGD, and SAGA in terms of number of processed gradients (left) and runtime (right) for the ORANGE text classification problem.

for GISETTE data is shown. The left plot demonstrates the sub-optimality with respect to the number of gradients, or samples, processed. In particular, k-TAN and AdaNewton compute m gradients per iterations, while SGD and SAGA compute 1 gradient per iteration. Observe that the second order methods all converge with a smaller number of total processed gradients than the first order methods, reaching after 2.5×10^4 samples a sub-optimality of 10^{-7}. We point out that, while k-TAN only approximates the Hessian inverse, its convergence path follows that of AdaNewton exactly. Indeed, both algorithms reach the statistical accuracy of $1/N = 1.6 \times 10^{-4}$ after 15000 samples, or just over two passes over the dataset. To see the gain in terms of computation time of k-TAN over AdaNewton and other methods, we present in the right image of Figure 1 the convergence in terms of runtime. In this case, k-TAN outperforms all methods, reaching a sub-optimality of 4×10^{-6} after 60 seconds, while AdaNewton reaches only a sub-optimality of 10^{-3} after 80 seconds. Note that first order methods have lower cost per iteration than all second order methods. Thus, SAGA is able to converge to 2×10^{-5} after 80 seconds.

For a high dimensional problem, we consider the RCV1 dataset with $N = 18242$ and $p = 47236$. We use a constant step size of 0.1 for SAGA and SGD and truncate sizes of around $0.001p$ for k-TAN, while keeping the parameters for the other methods the same. For this dataset we additionally include the adaptive sample size first order method, AdaSAGA [20]. The results of these simulations are shown in Figure 2. In the left image, observe that, in terms of processed gradients, the second order methods again outperform the first order, as expected, with k-TAN again following the path of AdaNewton. Given the high dimension p, the cost of computing the inverse in AdaNewton provides a large bottleneck. The gain in terms of computation time can then be best seen in the right image of Figure 2. Observe that AdaNewton becomes entirely ineffective in this case. The k-TAN method, alternatively, continues to descend at a fast rate because of the inverse truncation step. For this set k-TAN outperforms all the other methods, reaching an error of 10^{-7} after 1500 seconds. Since both n and p are large, SAGA performs well on this dataset due to small cost per iteration.

We perform additional numerical experiments on the ORANGE dataset used for customer relationship prediction in KDD Cup 2009. We use $N = 20000$ samples with dimension $p = 14472$. The convergence results are shown in Figure 3. Observe in the right hand plot that all second order methods, perform similarly well on this dataset. The first order methods, including SAGA, do not converge after 2000 seconds. Also, note that, in this experiment, we were able to reduce the truncation size k to around 0.1% of p.

5.1 Comparison of truncation lengths

While Theorem 1 provides a theoretical and principled way of selecting the truncation parameter k, we also provide a

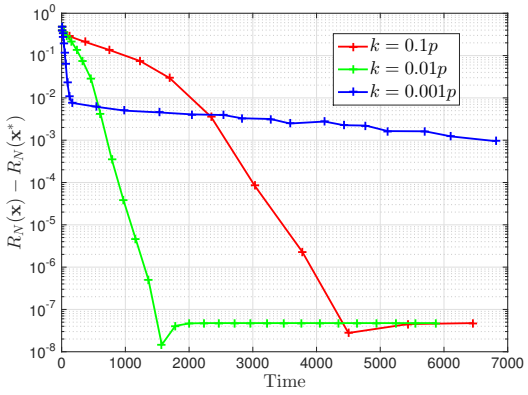

Figure 4: Convergence of k-TAN in terms of runtime for RCV1 dataset for various selections of truncation length k.

brief empirical analysis of various choices of this parameter. Naturally, the choice of k will tradeoff the time required to compute the Hessian inverse with the accuracy of the inversion, and by extension the ability of the k-TAN update to remain in the quadratic convergence region of the increased sample size function. Figure 4 shows the performance of k-TAN in terms of run time when various truncation lengths k are selected on the RCV1 dataset. We test truncation lengths of three different orders, $0.1p$, $0.01p$, and $0.001p$. Observe that the smallest truncation, $k = 0.001p$, descends much faster than the other two, but is unable to achieve a low accuracy because the truncation is too strong for the k-TAN method to progressively achieve accurate solutions. On the other hand, both $k = 0.01p$ and $k = 0.1p$ retain enough eigenvalues to achieve a strong accuracy, but the former does so with considerably less runtime. These results confirm that there is indeed a truncation length that both minimizes computation time while still achieving the desired accuracy. Note that this length can be selected via the backtracking step presented in Algorithm 1.

6 Discussion

We demonstrated the success of the proposed k-TAN method on solving large scale ERM problems both theoretically and empirically. The k-TAN method reduces the total cost in solving (1) to its statistical accuracy in two ways: (i) progressively increasing the sample size to reduce the costs of computing gradients and Hessians, and (ii) using a low rank approximation of the Hessian to reduce the cost of inversion. The gain provided by k-TAN relative to existing methods is therefore most significant in ERM problems with large sample size N and dimension p. To see this, consider the alternatives previously considered

- Stochastic first order methods, such as SAGA [7] and SVRG [12] have the overall complexity of $\mathcal{O}(N \log(N)p)$ to achieve statistical accuracy of the full training set if $V_N = \mathcal{O}(1/N)$.
- Newton's method computes gradients and Hessians over the entire dataset and inverts a matrix of size p at each step, requiring a total cost of $\mathcal{O}(M(Np^2 + p^3))$, where M is number of iterations required to converge.
- AdaNewton [16] computes gradients and Hessians for a subset of samples and inverts a matrix of size p at each step, while increases the size of the subset geometrically. By doubling the sample size, the statistical accuracy can be reached in $\log_2 N$ steps after a total of $2N$ passes over the dataset, for a total cost of $\mathcal{O}(2Np^2 + \log_2(N)p^3)$. While Hessian computation cost is reduced, for high dimensional problems the inversion step leads to a costly algorithm.

The k-TAN method computes gradients and Hessians on a increasing subset of data in the same manner as AdaNewton, put reduces the inversion cost at each iteration to $\mathcal{O}(p^2 \log k)$, resulting in a total cost of $\mathcal{O}(2Np^2 + p^2 \log_2 N \log k)$, or an effective cost of $\mathcal{O}(Np^2)$, if the size of the initial training set is large enough.

7 Acknowledgements

This work is supported by ARL DCIST CRA W911NF-17-2-0181 and Intel Science and Technology Center for Wireless Autonomous Systems (ISTC-WAS).

References

[1] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[4] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20, Vancouver, British Columbia, Canada*, pages 161–168, 2007.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[6] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[7] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada*, pages 1646–1654, 2014.

[8] Aaron Defazio, Justin Domke, and Tiberio Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the 31st international conference on machine learning (ICML-14)*, pages 1125–1133, 2014.

[9] Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, Quebec, Canada*, pages 3052–3060, 2015.

[10] Mert Gürbüzbalaban, Asuman Ozdaglar, and Pablo Parrilo. A globally convergent incremental Newton method. *Mathematical Programming*, 151(1):283–313, 2015.

[11] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26. Lake Tahoe, Nevada, United States.*, pages 315–323, 2013.

[13] Jakub Konecnỳ and Peter Richtárik. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2(2.1):3, 2013.

[14] Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic Newton method. *arXiv*, 2015.

[15] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, pages 2283–2291, 2013.

[16] Aryan Mokhtari, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, and Alejandro Ribeiro. Adaptive Newton method for empirical risk minimization to statistical accuracy. In *Advances in Neural Information Processing Systems*, pages 4062–4070, 2016.

[17] Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. IQN: An incremental quasi-Newton method with local superlinear convergence rate. *arXiv preprint arXiv:1702.00709*, 2017.

[18] Aryan Mokhtari and Alejandro Ribeiro. RES: Regularized stochastic BFGS algorithm. *IEEE Transactions on Signal Processing*, 62(23):6089–6104, 2014.

[19] Aryan Mokhtari and Alejandro Ribeiro. Global convergence of online limited memory BFGS. *Journal of Machine Learning Research*, 16:3151–3181, 2015.

[20] Aryan Mokhtari and Alejandro Ribeiro. First-order adaptive sample size methods to reduce complexity of empirical risk minimization. In *Advances in Neural Information Processing Systems*, page (to appear), 2017.

[21] Philipp Moritz, Robert Nishihara, and Michael I. Jordan. A linearly-convergent stochastic L-BFGS algorithm. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain*, pages 249–258, 2016.

[22] Yu Nesterov. *Introductory Lectures on Convex Programming Volume I: Basic course*. Citeseer, 1998.

[23] Yurii Nesterov et al. *Gradient methods for minimizing composite objective function*. 2007.

[24] Lam Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. *arXiv preprint arXiv:1703.00102*, 2017.

[25] Mert Pilanci and Martin J Wainwright. Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *arXiv preprint arXiv:1505.02250*, 2015.

[26] Zheng Qu, Peter Richtárik, Martin Takác, and Olivier Fercoq. SDNA: stochastic dual Newton ascent for empirical risk minimization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1823–1832, 2016.

[27] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[28] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods I: globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016.

[29] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods II: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016.

[30] Nicolas Le Roux, Mark W. Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25. Lake Tahoe, Nevada, United States.*, pages 2672–2680, 2012.

[31] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico*, pages 436–443, 2007.

[32] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14:567–599, 2013.

[33] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1-2):105–145, 2016.

[34] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.