Leverage Score Sampling for Faster Accelerated Regression and ERM

Naman Agarwal NAMANAGARWAL@GOOGLE.COM
Google AI Princeton, Princeton, USA

Sham Kakade SHAM@CS.WASHINGTON.EDU
University of Washington, Seattle, USA

Rahul Kidambi RKIDAMBI@CORNELL.EDU
Cornell University, Ithaca, USA

Yin Tat Lee YINTAT@UW.EDU
University of Washington and Microsoft Research, Seattle, USA

Praneeth Netrapalli PRANEETH@MICROSOFT.COM
Microsoft Research, Bengaluru, India

Aaron Sidford SIDFORD@STANFORD.EDU
Stanford University, Palo Alto, USA

Editors: Aryeh Kontorovich and Gergely Neu

Abstract

Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and a vector $b \in \mathbb{R}^d$, we show how to compute an ϵ-approximate solution to the regression problem $\min_{x\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{A}x - b\|_2^2$ in time $\widetilde{O}((n + \sqrt{d \cdot \kappa_{\text{sum}}})s \log \epsilon^{-1})$ where $\kappa_{\text{sum}} = \operatorname{tr}\left(\mathbf{A}^\top \mathbf{A}\right)/\lambda_{\min}(\mathbf{A}^\top \mathbf{A})$ and s is the maximum number of non-zero entries in a row of $\mathbf{A}$. This improves upon the previous best running time of $\widetilde{O}((n + \sqrt{n \cdot \kappa_{\text{sum}}})s \log \epsilon^{-1})$.

We achieve our result through an interesting combination of leverage score sampling, proximal point methods, and accelerated coordinate descent methods. Further, we show that our method not only matches the performance of previous methods up to polylogarithmic factors, but further improves whenever leverage scores of rows are small. We also provide a non-linear generalization of these results that improves the running time for solving a broader class of ERM problems and expands the set of ERM problems provably solvable in nearly linear time.

Keywords: Convex Optimization, Empirical Risk Minimization, Randomized Algorithms

1. Introduction

Given $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, the regression problem $\min_{x\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{A}x - b\|_2^2$ is one of the most fundamental problems in optimization and a prominent tool in machine learning. It is one of the simplest empirical risk minimization (ERM) problems and a prominent proving ground for developing new provably efficient algoirthms for solving large scale learning problems.

Regression is long known to be solve-able directly by fast matrix multiplication in $O(nd^{\omega-1})$ time where $\omega < 2.373$ (Williams, 2012) is the matrix multiplication constant and recent work has improved the running time to $\widetilde{O}((\operatorname{nnz}(\mathbf{A}) + d^\omega)\log(1/\epsilon))$,[1] i.e. linear time plus the time needed

1. Throughout we use $\widetilde{O}$ to hide factors polylogarithmic in n,d,$\kappa \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{A}^\top\mathbf{A})/\lambda_{\min}(\mathbf{A}^\top\mathbf{A})$, and M (see Definition 3 and Definition 6).

© 2020 N. Agarwal, S. Kakade, R. Kidambi, Y.T. Lee, P. Netrapalli & A. Sidford.

to solve a nearly square linear system (Clarkson and Woodruff, 2013; Li et al., 2013; Nelson and Nguyen, 2013; Cohen et al., 2015; Cohen, 2016). However, for large $\mathbf{A}$ even a super-quadratic running time of $\Omega(d^\omega)$ can be prohibitively expensive. Consequently, over the past decade improving this running time of regression under mild regularity assumptions on $\mathbf{A}$ has been an active area of research.

In this paper we improve the best known running time for solving regression under standard regularity assumptions. In particular we consider the following regression problem.

Definition 1 (The Regression Problem) *Given $\mathbf{A} \in \mathbb{R}^{n \times d}$ with rows $a_1, ..., a_n$ and given $b \in \mathbb{R}^n$, we consider the* regression problem $\min_{x \in \mathbb{R}^d} f_{\mathbf{A},b}(x)$ *where*

$$f_{\mathbf{A},b}(x) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{A}x - b\|_2^2 = \sum_{i \in [n]} \frac{1}{2}\left(a_i^\top x - b_i\right)^2 .$$

The central problem of this paper is to get faster regression algorithms defined as follows.

Definition 2 (Regression Algorithm) *We call an algorithm a $\mathcal{T}(\mathbf{A})$-time* regression algorithm *if for any $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^d$, and $\epsilon \in (0, \frac{1}{2})$ with high probability (w.h.p) in n in time $O(\mathcal{T}(\mathbf{A}) \log \epsilon^{-1})$ the algorithm outputs a vector y such that*

$$f_{\mathbf{A},b}(y) - \min_x f_{\mathbf{A},b}(x) \leq \epsilon \cdot \left(f_{\mathbf{A},b}(x_0) - \min_x f_{\mathbf{A},b}(x)\right) . \tag{1}$$

Note that if x_* is a minimizer of $f_{\mathbf{A},b}(x)$ then the guarantee (1) is equivalent to the following

$$\|y - x_*\|_{\mathbf{A}^\top \mathbf{A}}^2 \leq \epsilon \|x_0 - x_*\|_{\mathbf{A}^\top \mathbf{A}}^2 \tag{2}$$

where $\|x\|_{\mathbf{M}}^2 \stackrel{\text{def}}{=} x^\top \mathbf{M} x$ for $\mathbf{M} \succeq 0$. The goal of this paper is to provide regression algorithms with improved running times depending on n, d, and the following regularity parameters.

Definition 3 *(Regularity Parameters) We let $\lambda_{\min}(\mathbf{A}^\top \mathbf{A})$ and $\lambda_{\max}(\mathbf{A}^\top \mathbf{A})$ denote the smallest and largest eigenvalues of $\mathbf{A}^\top \mathbf{A}$, $\kappa(\mathbf{A}^\top \mathbf{A}) \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{A}^\top \mathbf{A})/\lambda_{\min}(\mathbf{A}^T \mathbf{A})$ denote the* condition number *of $\mathbf{A}^\top \mathbf{A}$, $\kappa_{\text{sum}}(\mathbf{A}^\top \mathbf{A}) \stackrel{\text{def}}{=} \text{tr}\left(\mathbf{A}^\top \mathbf{A}\right)/\lambda_{\min}(\mathbf{A}^T \mathbf{A})$ denote the* total condition number *of $\mathbf{A}^\top \mathbf{A}$, and $s(\mathbf{A})$ denote the maximum number of non-zero entries in a row of $\mathbf{A}$. Occasionally, we drop the terms in parenthesis when they are clear from context.*

In this paper we provide an $\widetilde{O}((n + \sqrt{d \cdot \kappa_{\text{sum}}})s \log(1/\epsilon)$ time algorithm for solving regression, improving upon the previous best running time of $\widetilde{O}((n + \sqrt{n \cdot \kappa_{\text{sum}}})s \log(1/\epsilon)$. (See Table 1 for the history of running time improvements to this problem.)

1.1. Previous Results

Classic iterative methods such as gradient descent and accelerated gradient descent (Nesterov, 1983) solve the regression problem with running times of $O(n \cdot s(\mathbf{A}) \cdot \kappa(\mathbf{A}^\top \mathbf{A}))$ and $O(n \cdot s(\mathbf{A}) \cdot \sqrt{\kappa(\mathbf{A}^\top \mathbf{A})})$ respectively. While these running times are super-linear whenever $\kappa(\mathbf{A}^\top \mathbf{A})$ is super constant there has been a flurry of recent papers showing that using sampling techniques faster running times can be achieved. These often yield nearly linear running times when n is sufficiently

Regression Running Times	
Algorithms	Running time
Naive Matrix Multiplication	$O(nd^2)$
Gradient Descent	$\widetilde{O}((\text{nnz}(\mathbf{A})\kappa)\log(1/\epsilon))$
Fast Matrix Multiplication (Williams, 2012)	$O(nd^{\omega-1})$ where $\omega < 2.373$
Accelerated Gradient Descent (Nesterov, 1983)	$\widetilde{O}((\text{nnz}(\mathbf{A})\sqrt{\kappa})\log(1/\epsilon))$
Row Sampling (Li et al., 2013; Cohen et al., 2015) Subspace Embeddings (Nelson and Nguyen, 2013)	$\widetilde{O}((\text{nnz}(\mathbf{A}) + d^{\omega})\log(1/\epsilon))$
SAG (Roux et al., 2012; Defazio et al., 2014) SVRG (Johnson and Zhang, 2013a)	$\widetilde{O}((\text{nnz}(\mathbf{A}) + \kappa_{\text{sum}} s)\log(1/\epsilon))$
APP (Frostig et al., 2015a) Catalyst (Lin et al., 2015) Katyusha (Allen Zhu, 2016)	$\widetilde{O}((\text{nnz}(\mathbf{A}) + s\sqrt{n\kappa_{\text{sum}}})\log(1/\epsilon))$
This Paper	$\widetilde{O}((\text{nnz}(\mathbf{A}) + s\sqrt{d\kappa_{\text{sum}}})\log(1/\epsilon))$

Table 1: History of improvements to the running time for regression in terms of the problem parameters $\text{nnz}(\mathbf{A})$, n, d, $\kappa \stackrel{\text{def}}{=} \kappa(\mathbf{A}^\top\mathbf{A})$, $\kappa_{\text{sum}} \stackrel{\text{def}}{=} \kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})$, and $s \stackrel{\text{def}}{=} s(\mathbf{A})$.

larger than d (Shalev-Shwartz and Zhang, 2012; Johnson and Zhang, 2013a; Shalev-Shwartz and Zhang, 2016; Allen Zhu, 2016).

Using recent advances in accelerated coordinate descent (Allen Zhu et al., 2016; Nesterov and Stich, 2017) coupled with proximal point methods (Frostig et al., 2015a; Lin et al., 2015) the previous fastest iterative algorithm is as follows:

Theorem 4 (Previous Best Regression Running Times) *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$, there is a $\mathcal{T}(\mathbf{A})$-time regression algorithm with*

$$\mathcal{T}(\mathbf{A}) = \widetilde{O}\left(\left(n + \frac{\sum_{i\in[n]}\|a_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}}\right)\cdot s(\mathbf{A})\right) = \widetilde{O}((n + \sqrt{n\cdot\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})})\cdot s(\mathbf{A})).$$

The equality in this theorem follows directly from Cauchy Schwartz, as

$$\sum_{i\in[n]}\|a_i\|_2 \leq \sqrt{n\cdot\text{tr}(\mathbf{A}^\top\mathbf{A})} = \sqrt{n\cdot\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})\cdot\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}\,.$$

We provide a generalization of the above theorem as Theorem 8 which is more useful to our analysis and provide a proof in the Appendix (Section C).

1.2. Our Results

The work in this paper is motivated by the natural question, *can this running time of Theorem 4 be further improved*? Despite the running time lower bound of $\sqrt{n\cdot\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})}$ shown in Woodworth and Srebro (2016),[2] in this paper we give an affirmative answer improving the $\sqrt{n\cdot\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})}$ term in Theorem 4 to $\sqrt{d\cdot\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})}$. The main result of this paper is the following:

2. Their lower bound involves a function with $d \gg n$. However, $d \ll n$ is more common as we explain.

Theorem 5 (Improved Regression Running Time) *Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, Algorithm 1 is a $\mathcal{T}(\mathbf{A})$-time regression algorithm that succeeds w.h.p in n where*

$$\mathcal{T}(\mathbf{A}) = \widetilde{O}\left(\text{nnz}(\mathbf{A}) + \left(d + \frac{\sum_{i\in[n]} \|a_i\|_2 \cdot \sqrt{\sigma_i(\mathbf{A})}}{\sqrt{\lambda_{\min}(\mathbf{A}^T\mathbf{A})}}\right) \cdot s(\mathbf{A})\right)$$

and $\sigma_i(\mathbf{A}) = \|a_i\|^2_{(\mathbf{A}^\top\mathbf{A})^{-1}}$ for all $i \in [n]$.

Up to polylogarithmic factors Theorem 5 is an improvement over Theorem 4 as $\sigma_i(\mathbf{A}) \in [0, 1]$. This improvement can be substantial as $\sigma_i(\mathbf{A})$ can be as small as $O(d/n)$, e.g. if $\mathbf{A}$ is an entry-wise random Gaussian matrix. Compared to Theorem 4 whose second term in running time grows as n, our second term is always independent of n due to the following:

$$\sum_{i\in[n]} \|a_i\|_2 \sqrt{\sigma_i(\mathbf{A})} \leq \sqrt{\sum_{i\in[n]} \|a_i\|_2^2 \sum_{i\in[n]} \|a_i\|^2_{(\mathbf{A}^\top\mathbf{A})^{-1}}} = \sqrt{\text{tr}\left(\mathbf{A}^\top\mathbf{A}\right) \text{tr}(\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top)} \leq \sqrt{d\kappa_{\text{sum}}}. \tag{3}$$

Therefore in Theorem 5 we have $\mathcal{T}(\mathbf{A}) = \widetilde{O}((n + \sqrt{d \cdot \kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})}) \cdot s(\mathbf{A}))$.

This improvement from n to d can be significant as n (the number of samples) is in some cases orders of magnitude larger than d (the number of features). For example, in the LIBSVM dataset[3], in 87 out of 106 non-text problems, we have $n \geq d$, 50 of them have $n \geq d^2$ and in the UCI dataset,[4] in 279 out of 301 non-text problems, we have $n \geq d$, 195 out of them have $n \geq d^2$.

Furthermore, in Section 5 we show how to extend our results to ERM problems more general than regression. In particular we consider the following ERM problem

Definition 6 (ERM) *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rows $a_1, ..., a_n$ and functions $\{\psi_1 \dots \psi_n\} \in \mathbb{R} \to \mathbb{R}$ such that each $\psi_i : \mathbb{R} \to \mathbb{R}$ is twice differentiable and satisfies*

$$\forall\, x \in \mathbb{R}^d \quad \frac{1}{M} \leq \psi''(x) \leq M \tag{4}$$

we wish to minimize $F(x) : \mathbb{R}^d \to \mathbb{R}$ over $x \in \mathbb{R}^d$ where

$$F(x) \stackrel{\text{def}}{=} \sum_{i\in[n]} f_i(x) = \sum_{i\in[n]} \psi_i(a_i^\top x)$$

This assumption (4) is satisfied by many ERM problems where the $f_i(x)$ are regularized directly. For example, the δ-regularized logistic function $f(x) = \log(1 + \exp(-x)) + \frac{\delta}{2}x^2$ satisfies $\delta \leq f''(x) \leq 1/4 + \delta$ for all x and therefore under appropriate rescaling yields $M = 1/\sqrt{\delta}$ which may be small in many instances. Showing how to adapt the above result to ERM problems with arbitrary regularization, like logistic regression, we leave to future work. The following is our main theorem regarding ERM:

3. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
4. http://archive.ics.uci.edu/ml/datasets.html

Theorem 7 *Given an ERM problem(Definition 6) and an initial point x_0, there exists an algorithm that produces a point x' such that $F(x') - \min_{x\in\mathbb{R}^d} F(x) \leq \epsilon\left(F(x_0) - \min_{x\in\mathbb{R}^d} F(x)\right)$ which succeeds w.h.p in n in total time*

$$\tilde{O}\left(\left(\text{nnz}(\mathbf{A}) + \left(dM^5 + \sum_{i=1}^{n} \frac{\|a_i\|_2\sqrt{\sigma_i(\mathbf{A})}M^3}{\sqrt{\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}}\right)s(\mathbf{A})\right)\log\left(\frac{1}{\epsilon}\right)\right)$$

where $\sigma_i(\mathbf{A}) \stackrel{\text{def}}{=} \|a_i\|^2_{[\mathbf{A}^\top\mathbf{A}]^{-1}}$ are the leverage scores with respect to a_i.

Note that Theorem 7 interpolates our regression results, i.e. it recovers our results for regression in the special case of $M = 1$. To better understand the bound in Theorem 7, note that following the derivation in Equation (3) we have that the running time in Theorem 7 is bounded by

$$\tilde{O}\left(\left(\text{nnz}(\mathbf{A}) + \left(dM^5 + \sum_{i=1}^{n} M^3\sqrt{d\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A})}\right)s(\mathbf{A})\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

The best known bound for the ERM problem as defined in Definition 6 given by (Frostig et al., 2015a; Lin et al., 2015; Allen Zhu, 2016) is

$$\tilde{O}\left(\left(\text{nnz}(\mathbf{A}) + \sum_{i=1}^{n} M\sqrt{n\kappa_{\text{sum}}(\mathbf{A}^\top\mathbf{A}))}\right)s(\mathbf{A})\log\left(\frac{1}{\epsilon}\right)\right)$$

In this case Theorem 7 should be seen as implying that under Assumption (4) the effective dependence on the number of examples on the running time for ERM can be reduced to at most dM^5.

Again, we remark that the running time bound of Theorem 7 should be viewed as a proof of concept that our regression machinery can be used to improve the running time of ERM. We leave it as future work to both improve Theorem 7's dependence on M and have it extend to a broader set of problems. For example, we believe the the running time can be immediately improved to

$$\tilde{O}\left(\left(\text{nnz}(\mathbf{A}) + \left(dM^4 + \sum_{i=1}^{n} \frac{\|a_i\|_2\sqrt{\sigma_i}M^3}{\sqrt{\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}}\right)s(\mathbf{A})\right)\log\left(\frac{1}{\epsilon}\right)\right)$$

by using a proximal version of Theorem 15, which is alluded to in the work of Allen Zhu et al. (2016). Note that this improvement leads to the effective number of examples being bounded by dM^4.

1.3. Our Approach

Our algorithm follows from a careful combination and analysis of a recent suite of advances in numerical linear algebra. First, we use the previous fastest regression algorithm, Theorem 4, which is the combination of recent advances in accelerated coordinate descent (Allen Zhu et al., 2016; Nesterov and Stich, 2017) and proximal point methods (Frostig et al., 2015a; Lin et al., 2015). Then, we show that if we have estimates of the leverage scores of the rows of $\mathbf{A}$, a natural recently popularized measure of importance, (Spielman and Srivastava, 2008; Li et al., 2013; Cohen et al., 2015) we can use concentration results on leverage score sampling and preconditioning to obtain a faster regression algorithm. (See Section 3.)

A powerful well known fact is that given a regression algorithm leverage scores can be estimated in nearly linear time plus the time needed to solve $\widetilde{O}(1)$ regression problems (Spielman and Srivastava, 2008). Consequently, to achieve the improved running time when we do not have leverage scores we are left with a chicken and egg problem. Fortunately, recent work (Li et al., 2013; Cohen et al., 2015) has shown that such a problem can be solved in several ways. We show that the technique in Li et al. (2013) carefully applied can be used to obtain our improved running time for both estimating leverage scores and solving regression problems with little overhead (See Section 4).

For application to a broader class of ERM problems most parts of the regression procedure generalize naturally. The key ingredient is the generalization of the preconditioning step to the case when we are sampling non-quadratic functions. For this, we prove concentration results on sampling from ERM inspired from Frostig et al. (2015b) to show that it suffices to solve ERM on a sub-sampling of the components that may be of intrinsic interest. (See Section 5)

In summary our algorithms are essentially a careful blend of accelerated coordinate descent and concentration results coupled with the iterative procedure in Li et al. (2013) and the Johnson Lindenstrauss machinery of Spielman and Srivastava (2008) to compute leverage scores. Ultimately the algorithms we provide are fairly straightforward, but it provides a substantial running time improvement that we think is of intrinsic interest.

Finally, we remark that there is another way to achieve the $\sqrt{d \cdot \kappa_{\text{sum}}(\mathbf{A})}$ improvement over $\sqrt{n \cdot \kappa_{\text{sum}}(\mathbf{A})}$. One could use subspace embeddings (Clarkson and Woodruff, 2013; Nelson and Nguyen, 2013; Cohen, 2016) and preconditioning to reduce the regression problem to a regression problem on a $\widetilde{O}(d) \times d$ matrix and then apply Theorem 4 to solve the $\widetilde{O}(d) \times d$ regression problem. While this works, it has three shortcomings relevant to our approach. Firstly note that even if the original rows of the matrix are s sparse, the the rows of the sketched matrix might become $\Omega(d)$ sparse, and the final running time would have an additional $\widetilde{O}(d^2)$ term our method does not. Second, it is unclear if this approach yields our more fine-grained running time dependence on leverage scores that appears in Theorem 5 which we believe to be significant. Thirdly it is unclear how to extend the approach to the ERM setting.

1.4. Paper Organization

After providing requisite notation in Section 2 we prove Theorem 5 in Sections 3 and 4. We first provide the algorithm for regression given leverage score estimates in Section 3 and further provide the algorithm to compute the estimates in Section 4. Note that the algorithm for computing leverage scores makes use of the algorithm for regression given leverage scores as a sub-routine. In Section 5 we state and prove Theorem 14 which forms the bulk of the proof of Theorem 7. The proof of Theorem 7 is provided in the Appendix. Finally we collect the proofs of all the lemmas/theorems deferred from the main paper due to space constraints in the Appendix.

2. Notation

For symmetric matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ and $x \in \mathbb{R}^d$ we let $\|x\|_{\mathbf{M}}^2 = x^\top \mathbf{M} x$. For symmetric matrix $\mathbf{N} \in \mathbb{R}^{d \times d}$ we use $\mathbf{M} \preceq \mathbf{N}$ to denote the condition that $x^\top \mathbf{M} x \leq x^\top \mathbf{N} x$ for all $x \in \mathbb{R}^d$ and we define $\prec$, $\succeq$, and $\succ$ analogously. We use $\text{nnz}(\mathbf{A})$ to denote the number of non-zero entries in $\mathbf{A}$ and for $b \in \mathbb{R}^n$, we let $\text{nnz}(b)$ denote the number of nonzero entries in b.

3. Regression Algorithm Given Leverage Score Estimates

The regression algorithm we provide in this paper involves two steps. First we find which rows of $\mathbf{A}$ are important in terms of *leverage score*. Second, we use these leverage scores to sample the matrix and solve the regression problem on the sampled matrix using the following theorem which is a generalization of Theorem 4 that is useful for our analysis.

Theorem 8 (Previous Best Regression Running Time) *Let $\mathbf{A}$ and $\mathbf{B}$ be matrices with the same number of columns. Suppose that $\mathbf{B}$ has n rows and $\left(\frac{5}{6}\right)\mathbf{B}^\top\mathbf{B} \preceq \mathbf{A}^\top\mathbf{A} \preceq \left(\frac{6}{5}\right)\mathbf{B}^\top\mathbf{B}$, then there is a $\mathcal{T}(\mathbf{A})$-time regression algorithm with*

$$\mathcal{T}(\mathbf{A}) = \widetilde{O}\left(\mathrm{nnz}(\mathbf{A}) + \left(n + \frac{\sum_{i\in[n]}\|b_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{B}^\top\mathbf{B})}}\right)\cdot s(\mathbf{B})\right).$$

Theorem 8 is a consequence of results on accelerated coordinate descent (Allen Zhu et al., 2016; Nesterov and Stich, 2017) and approximate proximal point (Frostig et al., 2015a; Lin et al., 2015). We defer the proof to the Appendix (Section C). In the rest of the section we define leverage scores and analyze the second step of our algorithm.

Definition 9 (Leverage Score) *For $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rows $a_1, ..., a_n \in \mathbb{R}^d$ we denote the* leverage score of row $i \in [n]$ *by $\sigma_i(\mathbf{A}) \stackrel{\text{def}}{=} a_i^\top \left(\mathbf{A}^\top\mathbf{A}\right)^+ a_i$.*

Note that $\sigma_i(\mathbf{A}) \in (0, 1]$ for all $i \in [n]$ and $\sum_{i\in[n]}\sigma_i(\mathbf{A}) = \mathrm{rank}(\mathbf{A})$. The following lemma shows that sampling rows of $\mathbf{A}$ according to overestimates of leverage scores yields a good approximation to $\mathbf{A}$ after appropriate re-scaling (Cohen et al., 2015; Spielman and Srivastava, 2008):

Lemma 10 (Leverage Score Sampling (Cohen et al. (2015))) *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $\delta \in (0, \frac{1}{2})$, and let $u \in \mathbb{R}^n$ be overestimates of leverage scores of $\mathbf{A}$; i.e. $u_i \geq \sigma_i(\mathbf{A})$ for all $i \in [n]$. Define $p_i \stackrel{\text{def}}{=} \min\left\{1, k\delta^{-2}u_i\log n\right\}$ for a sufficiently large constant $k > 0$ and let $\mathbf{H} \in \mathbb{R}^{n\times n}$ be a random diagonal matrix where $\mathbf{H}_{ii} = \frac{1}{p_i}$ independently with probability p_i and $\mathbf{H}_{ii} = 0$ otherwise. With high probability in n, $\mathrm{nnz}(\mathbf{H}) = O(d\cdot\delta^{-2}\cdot\log n)$ and $(1-\delta)\cdot\mathbf{A}^\top\mathbf{A} \preceq \mathbf{A}^\top\mathbf{H}\mathbf{A} \preceq (1+\delta)\cdot\mathbf{A}^\top\mathbf{A}$.*

Algorithm 1 outlines the procedure to solve regression given overestimates of leverage score.

Algorithm 1: $\texttt{SolveUsingLS}_{\mathbf{A},u}(x_0, b, \epsilon, u)$

Let $p_i = \min\{1, k'\cdot u_i\log n\}$ where k' is a sufficiently large absolute constant.
repeat
  Let $\mathbf{H} \in \mathbb{R}^{n\times n}$ be a diagonal matrix where independently for all $i \in [n]$ we let $\mathbf{H}_{ii} = \frac{1}{p_i}$ with probability p_i and 0 otherwise.
  Let $\mathbf{B} = \sqrt{\mathbf{H}}\mathbf{A}$.
until $\sum_{i\in[n]}\|b_i\|_2 \leq 2\cdot\sum_{i\in[n]}\sqrt{k'\cdot u_i\log n}\cdot\|a_i\|_2$;
Invoke Theorem 8 on $\mathbf{A}$ and $\mathbf{B}$ to find y such that

$$f_{\mathbf{A},b}(y) - \min_x f_{\mathbf{A},b}(x) \leq \epsilon\cdot\left(f_{\mathbf{A},b}(x_0) - \min_x f_{\mathbf{A},b}(x)\right).$$

Output: y.

Theorem 11 *If $u \in \mathbb{R}^n$ satisfies $\sigma_i(\mathbf{A}) \leq u_i \leq 4 \cdot \sigma_i(\mathbf{A}) + [n \cdot \kappa(\mathbf{A}^\top \mathbf{A})]^{-1}$ for all $i \in [n]$ then $\texttt{SolveUsingLS}_{\mathbf{A},u}$ is a $\mathcal{T}(\mathbf{A})$-time regression algorithm where*

$$\mathcal{T}(\mathbf{A}) = \widetilde{O}\left(\text{nnz}(\mathbf{A}) + \left(d + \frac{\sum_{i\in[n]} \sqrt{\sigma_i(\mathbf{A})} \cdot \|a_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}}\right) \cdot s(\mathbf{A})\right) .$$

Proof Let $k' = \delta^{-2} \cdot k$ for $\delta = \frac{1}{10}$ and k defined in Lemma 10. Applying Lemma 10 yields that for every iteration of the inner loop, we have with high probability in n that

$$\left(\frac{5}{6}\right) \mathbf{A}^\top \mathbf{A} \preceq \mathbf{A}^\top \mathbf{H} \mathbf{A} \preceq \left(\frac{6}{5}\right) \mathbf{A}^\top \mathbf{A} \tag{5}$$

where $\mathbf{A}^\top \mathbf{H} \mathbf{A} = \sum_{i\in[n]\,:\,\mathbf{H}_{ii}\neq 0} b_i b_i^\top$, $b_i \stackrel{\text{def}}{=} \frac{1}{\sqrt{p_i}} a_i$ and $p_i \stackrel{\text{def}}{=} \min\{1, k' \cdot u_i \log n\}$. Note that

$$\mathbb{E}\left[\sum_{i\in[n]} \|b_i\|_2\right] = \sum_{i\in[n]} \frac{p_i}{\sqrt{p_i}} \|a_i\|_2 \leq \sum_{i\in[n]} \sqrt{k' u_i \log n} \|a_i\|_2 .$$

Consequently, by Markov's inequality, with probability at least $1/2$

$$\sum_{i\in[n]} \|b_i\|_2 \leq 2 \cdot \sum_{i\in[n]} \sqrt{k' \cdot u_i \log n} \cdot \|a_i\|_2$$

Therefore the loop in the algorithm terminates with high probability in n in $O(\log n)$ iterations. Consequently, the loop takes only $O(\text{nnz}(\mathbf{A}) + n \log n)$-time and since we only sampled $O(\log n)$ many independent copies of $\mathbf{A}^\top \mathbf{H} \mathbf{A}$, the guarantee (5) again holds with high probability in n.

Using the guarantee (5) and Theorem 8 on $\mathbf{A}$ and $\mathbf{B} \stackrel{\text{def}}{=} \sqrt{\mathbf{H}}\mathbf{A}$, we can produce a y we need in time $O(\log(\epsilon^{-1}))$ times

$$\widetilde{O}\left(\left(\text{nnz}(\mathbf{A}) + \left(d \log n + \frac{1}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}} \sum_{i\in[n]} \|b_i\|_2\right) \cdot s(\mathbf{A})\right)\right)$$

where we used that $\mathbf{B}$ has at most $O(d \log n)$ rows with high probability in n. Since we know

$$\sum_{i\in[n]} \|b_i\|_2 \leq 2 \sum_{i\in[n]} \sqrt{k' \cdot u_i \log n} \cdot \|a_i\|_2,$$

all that remains is to bound $\sum_{i\in[n]} \sqrt{u_i} \|a_i\|_2$. However, $\mathbf{A}^\top \mathbf{A} \preceq \lambda_{\max}(\mathbf{A}^\top \mathbf{A}) \mathbf{I}$ and therefore

$$\mathbf{I} \preceq \lambda_{\max}(\mathbf{A}^\top \mathbf{A})(\mathbf{A}^\top \mathbf{A})^{-1} \text{ and } \|a_i\|_2 \leq \sqrt{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})} \cdot \sigma_i(\mathbf{A}) .$$

Consequently, Cauchy Schwartz and $\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \leq \text{tr}(\mathbf{A}^\top \mathbf{A})$ yields

$$\frac{1}{\sqrt{n}} \sum_{i\in[n]} \|a_i\|_2 \leq \sqrt{\sum_{i\in[n]} \|a_i\|_2^2} \leq \frac{1}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}} \sum_{i\in[n]} \|a_i\|_2^2 \leq \sqrt{\kappa(\mathbf{A}^\top \mathbf{A})} \sum_{i\in[n]} \sqrt{\sigma_i(\mathbf{A})} \cdot \|a_i\|_2.$$

Since $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ this yields

$$\sum_{i\in[n]} \sqrt{u_i} \cdot \|a_i\|_2 \leq 2 \sum_{i\in[n]} \sqrt{\sigma_i(\mathbf{A})} \cdot \|a_i\|_2 + \frac{1}{\sqrt{n \cdot \kappa(\mathbf{A}^\top \mathbf{A})}} \sum_{i\in[n]} \|a_i\|_2 \leq 3 \sum_{i\in[n]} \sqrt{\sigma_i(\mathbf{A})} \cdot \|a_i\|_2$$

which in turn yields the result as $\widetilde{O}$ hides factors poly-logarithmic in n and d. ■

4. Regression Algorithm Without Leverage Score Estimates

In the previous section we showed that we can solve regression in our desired running time provided we have a constant factor upper approximation to leverage scores. Here we show how to apply this procedure repeatedly to estimate leverage scores as well. We do this by first adding a large multiple of the identity to our matrix and then gradually decreasing this multiple while maintaining estimates for leverage scores along the way. This is a technique introduced in Li et al. (2013) and we leverage it tailored to our setting.

A key technical ingredient for this algorithm is the following well-known result on the reduction from leverage score computation to regression with little overhead. Formally, Lemma 12 states that you can compute constant multiplicative approximations to all leverage scores of a matrix in nearly linear time plus the time needed to solve $\widetilde{O}(1)$ regression problems. Algorithm 2 details the procedure for computing leverage scores.

Algorithm 2: $\texttt{ComputeLS}(\mathbf{A}, \delta, \mathcal{A})$

Let $k = c\log(n)$ and $\epsilon = \frac{\delta^2}{(18nd\log n \cdot \kappa(\mathbf{A}^\top\mathbf{A}))^2}$ where c is some large enough constant.

for $j = 1, \cdots, k$ **do**

Let $v_j \in \mathbb{R}^n$ be a random Gaussian vector, i.e. each entry follows $N(0, I)$.

Use algorithm $\mathcal{A}$ to find a vector y_j such that

$$f_{\mathbf{A},v_j}(y_j) - \min_x f_{\mathbf{A},v_j}(x) \leq \epsilon(f_{\mathbf{A},v_j}(0) - \min_x f_{\mathbf{A},v_j}(x)) .$$

end

Let $\tau_i = \frac{1}{k}\sum_{j=1}^k (e_i^\top \mathbf{A}^\top y_j)^2$ for all $i = 1, \cdots, n$.

Output: $\frac{\tau}{1-\delta/3} + \frac{\delta}{2n\cdot\kappa(\mathbf{A}^\top\mathbf{A})}$.

Lemma 12 (Computing Leverage Scores) *For $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $\mathcal{A}$ be a $\mathcal{T}(\mathbf{A})$-time algorithm for regression on $\mathbf{A}$. For $\delta \in (\frac{1}{n}, \frac{1}{2})$, in time $O((\mathrm{nnz}(\mathbf{A}) + \mathcal{T}(\mathbf{A})\log\epsilon^{-1})\delta^{-2}\log n)$ where we set $\epsilon = \delta^2(18n\cdot d\cdot\log n\cdot\kappa(\mathbf{A}^\top\mathbf{A}))^{-2}$, with high probability in n, the algorithm $\texttt{ComputeLS}(\mathbf{A}, \delta, \mathcal{A})$ outputs $\tau \in \mathbb{R}^n$ such that $\sigma_i(\mathbf{A}) \leq \tau_i \leq (1+\delta)\sigma_i(\mathbf{A}) + \delta\cdot[n\cdot\kappa(\mathbf{A}^\top\mathbf{A})]^{-1}$. for all $i \in [n]$.*

We defer the proof of Lemma 12 to the Appendix (Section A). Combining the algorithm for estimating leverage scores `ComputeLS` (Algorithm 2) with our regression algorithm given leverage scores `SolveUsingLS` (Theorem 11) yields our solver (Algorithm 3). We first provide a technical lemma regarding invariants maintained by the algorithm (Lemma 13). The proof of Lemma 13 is deferred to the Appendix(Section B) due to space constraints.

Lemma 13 *In the algorithm $\texttt{Solve}_{\mathbf{A},\epsilon}$ (See Algorithm 3) the following invariant is satisfied*

$$\sigma_i(\mathbf{A}_\eta) \leq u_i \leq 4\cdot\sigma_i(\mathbf{A}_\eta) + [n\cdot\kappa(\mathbf{A}_\eta^\top\mathbf{A}_\eta)]^{-1}. \tag{6}$$

We now prove Theorem 5 using Lemma 13 and Algorithm 3.

Proof [Proof of Theorem 5] Lemma 13 shows that u is always a good enough estimate of $\sigma_i(\mathbf{A}_\eta)$ throughout the algorithm to invoke `SolveUsingLS` with Theorem 11. Note that `SolveUsingLS` is

Algorithm 3: $\texttt{Solve}_{\mathbf{A}}(x_0, b, \epsilon)$

Let $\mathbf{A}_\eta = \begin{pmatrix} \mathbf{A} \\ \sqrt{\eta}\mathbf{I} \end{pmatrix}$, $\eta = \lambda_{\max}(\mathbf{A}^\top \mathbf{A})$ and $u_i = \begin{cases} \frac{1}{\eta}\|a_i\|_2^2 & \text{if } 1 \le i \le n \\ 1 & \text{if } n+1 \le i \le n+d \end{cases}$ [5]

repeat

  $u \leftarrow 2 \cdot \texttt{ComputeLS}(\mathbf{A}_\eta, \frac{1}{4}, \mathcal{A})$ for algorithm $\mathcal{A}$ given by $\texttt{SolveUsingLS}_{\mathbf{A}_\eta, u}$.

  $\eta \leftarrow \frac{3}{4} \cdot \eta$.

until $\eta > \frac{1}{10}\lambda_{\min}(\mathbf{A}^\top \mathbf{A})$;

Set $\eta \leftarrow 0$. Let $\bar{b} = \begin{pmatrix} b \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{n+d}$.

Apply algorithm $\texttt{SolveUsingLS}_{\mathbf{A}_0, u}$ to find y such that

$$f_{\mathbf{A}_0, \bar{b}}(y) - \min_x f_{\mathbf{A}_0, \bar{b}}(x) \le \epsilon(f_{\mathbf{A}_0, \bar{b}}(x_0) - \min_x f_{\mathbf{A}_0, \bar{b}}(x)) \,.$$

Output: y

called from within the invocation of ComputeLS and then in particular at the last step when η is set to 0 and the invariant ensures that the output of the algorithm is as desired by Theorem 5.

During the whole algorithm, $\texttt{ComputeLS}(\mathbf{A}_\eta, \frac{1}{4}, \mathcal{A})$ is called $\Theta(\log(\kappa(\mathbf{A}^\top \mathbf{A})))$ times. Each time ComputeLS is called, SolveUsingLS is called $\Theta(\log(n))$ many times. Therefore upto log factors all that remains is to bound the running time of the individual invocation of SolveUsingLS. We use Theorem 11 for this purpose. Note that, for $\lambda \ge 0$ and $i \in [n]$ we have $\sigma_i(\mathbf{A}_\lambda) \le \sigma_i(\mathbf{A}_0)$ and since $\mathbf{A}_\lambda^\top \mathbf{A}_\lambda \ge \lambda \mathbf{I}$ we have that $\lambda \le \lambda_{\min}(\mathbf{A}_\lambda^\top \mathbf{A}_\lambda)$. Furthermore, since $\lambda_{\min}(\mathbf{A}_\lambda^\top \mathbf{A}_\lambda) \ge \lambda_{\min}(\mathbf{A}^\top \mathbf{A})$ we have that the running time follows from the following and Theorem 11:

$$\frac{\sum_{i \in [n+d]} \sqrt{\sigma_i(\mathbf{A}_\lambda)} \cdot \|a_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{A}_\lambda^\top \mathbf{A}_\lambda)}} \le \sum_{i \in [n]} \frac{\sqrt{\sigma_i(\mathbf{A})} \cdot \|a_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}} + \sum_{i \in [d]} \frac{\sqrt{\lambda}}{\sqrt{\lambda}} \,.$$

Finally note that all the statements hold with high probability in n but they are invoked logarithmically many times and hence by a union bound, we see that the procedure succeeds with high probability in n. ■

5. Extension for ERM Problems

In this section we consider the ERM problem (cf. Definition 6). We propose Algorithm 4 as the main sub-routine to solve the ERM problem. Theorem 14 provides the error guarantee and bounds the running time of Algorithm 4. Note that Theorem 14 provides a constant factor decrease in the error which can be repeated via a standard reduction to provide ϵ error as required by Theorem 7. We formally provide the reduction and the proof of Theorem 7 in the Appendix (Section D).

Algorithm 4 takes as input, estimates of leverage scores of the matrix $\mathbf{A}^\top \mathbf{A}$ and creates an estimator of the true function by sampling component functions according to the probability distribution given by the leverage scores and appropriate re-scaling. Further, it reformulates the estimator as a sum of variance reduced components akin to Johnson and Zhang (2013b). The algorithm then

approximately minimizes the estimator using an off-the-shelf ERM minimizer $\mathcal{A}$ (Theorem 15). This step can be seen as analogous to the preconditioned iteration in the case of linear regression.

Algorithm 4: $\texttt{ERMSolve}(x_0, \{\tau_i\}_{i=1}^n, F(x) = \sum_{i=1}^n f_i(x)), m)$

Define for $k = 1 \to n$, $p_k \stackrel{\text{def}}{=} \frac{\tau_k}{\sum_j \tau_j}$.
Let $\mathcal{D}(j)$ be the distribution over $[1, \ldots n]$ such that $\forall\ k \quad \Pr_{j \sim \mathcal{D}}(j = k) = p_k$
Define for $k = 1 \to n$, $\tilde{f}_k(x) \stackrel{\text{def}}{=} \frac{1}{p_k}\left[f_k(x) - \nabla f_k(x_0)^\top x\right] + \nabla F(x_0)^\top x$
Sample m integers $i_1 \ldots i_m \in [n]$ independently from $\mathcal{D}$.
if $\sum_{t=1}^m \frac{\|a_{i_t}\|_2}{\sqrt{p_{i_t}}} \leq 10m \sum_{k=1}^n \|a_k\|_2 \sqrt{p_k}$ **then**
  Set $F_m(x) = \frac{1}{m}\sum_{t=1}^m \tilde{f}_{i_t}(x)$.
  Use Theorem 15 to find x' such that

$$F_m(x') - \min F_m(x) \leq \frac{1}{512M^4}\left(F_m(x_0) - \min F_m(x)\right)$$

end
Output: x'

Theorem 14 *Given an ERM problem (Definition 6) and numbers u_i which are over estimates of leverage scores i.e. $u_i \geq \sigma_i$, set parameters such that $\tau_i = \min\{1, 20u_i \log(d)\}$, $m = 160\left((\sum_j \tau_j) \cdot M^4\right)$ then we have that Algorithm 4 produces a point x' such that*

$$F(x') - \min_{x \in \mathbb{R}^d} F(x) \leq \frac{1}{2}\left(F(x_0) - \min_{x \in \mathbb{R}^d} F(x)\right)$$

with probability at least $1/2$. Further Algorithm 4 can be implemented in total time

$$\tilde{O}\left(\left(mM + \sum_{i=1}^n \frac{\|a_i\|_2 \sqrt{\tau_i} M^3}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}}\right) s(\mathbf{A})\right).$$

5.1. Proof of Theorem 14

We first provide a generalization of Theorem 4 for the ERM setting and then prove Theorem 14.

Theorem 15 (Acc. Coordinate Descent for ERM) *Consider the ERM problem (cf. Definition 6) with ψ_i such that $\forall\ x \quad \psi_i''(x) \in [\mu_i, L_i]$ and λ such that $\forall\ x\ \nabla^2 F(x) \succeq \lambda I$. Given a point x_0, there exists an algorithm $\mathcal{A}$ which produces a point x' w.h.p in n such that*

$$F(x') - \min_{x \in \mathbb{R}^d} F(x^*) \leq \epsilon(F(x_0) - \min_{x \in \mathbb{R}^d} F(x^*))$$

in total time proportional to

$$\tilde{O}\left(\left(\sum_{i=1}^n \sqrt{\frac{L_i}{\mu_i}} + \sum_{i=1}^n \|a_i\|_2 \sqrt{\frac{L_i}{\lambda}}\right) s(\mathbf{A}) \log(\epsilon^{-1})\right)$$

The proof of Theorem 15 is a direct consequence of Allen Zhu et al. (2016) and is deferred to the Appendix (Section F). We will use Algorithm $\mathcal{A}$ guaranteed by Theorem 8 as a subroutine in the Algorithm 4.

Proof For convenience we restate the definitions provided in Algorithm 4. Given parameters $\{\tau_1 \dots \tau_n\}$ we define a probability distribution $\mathcal{D}$ over $\{1, \dots n\}$ such that

$$\forall\, k \in [n] \;\; p_k \stackrel{\text{def}}{=} Pr_{j \sim \mathcal{D}}(j = k) \stackrel{\text{def}}{=} \frac{\tau_k}{\sum \tau_k} \quad . \tag{7}$$

We define approximations to f_k for $k \in [n]$ as

$$\tilde{f}_k(x) \stackrel{\text{def}}{=} \frac{1}{p_k}\left[f_k(x) - \nabla f_k(x_i)^T x\right] + \nabla F(x_0)^T x \quad . \tag{8}$$

Further we sample m integers $\{i_1, \dots i_m\}$ independently from $\mathcal{D}$ and we define the approximation

$$F_m(x) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{t=1}^{m} \tilde{f}_{i_t}(x) \quad . \tag{9}$$

Define $x_* \stackrel{\text{def}}{=} \text{argmin}_x F(x)$. To prove the theorem we will prove two key properties. Firstly the choice of the sample size $m = \Omega(\sum_{k=1}^n \tau_k M^4)$ is sufficient to ensure that approximately minimizing $F_m(x)$ makes constant multiplicative factor progress on F. Secondly we will bound the running time of the coordinate descent procedure ($\mathcal{A}$ from Theorem 8). Consider the random matrix

$$\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \stackrel{\text{def}}{=} \frac{1}{m}\sum_{t=1}^{m} \frac{a_{i_t} a_{i_t}^\top}{p_{i_t}} \quad . \tag{10}$$

and define the event $\mathcal{E}_1$ to be the following event.

$$\mathcal{E}_1 \stackrel{\text{def}}{=} \{0.5\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq 2\mathbf{A}^\top\mathbf{A}\} \quad . \tag{11}$$

We use a concentration inequality Lemma 24 (stated and proved in Appendix Section G) to ensure

$$Pr(\mathcal{E}_1) \geq 1 - 1/d$$

The following lemma bounds the number of samples required for exact minimization of $F_m(x)$ to lead to constant decrease in error under the event $\mathcal{E}_1$. Due to space constraints we provide the proof of the Lemma in the Appendix(Section E)

Lemma 16 *Consider an ERM problem $F(x) = \sum f_i(x)$ as defined in Definition 6. Let F_m be as defined in (9) and $\tilde{\mathbf{A}}$ be as defined in (10). Let*

$$x_m \stackrel{\text{def}}{=} \text{argmin}_{x \in \mathbb{R}^d} F_m(x).$$

Let $\mathcal{E}_1 \stackrel{\text{def}}{=} \{0.5\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq 2\mathbf{A}^\top\mathbf{A}\}$ and let $Pr(\mathcal{E}_1) \geq p$. Then if we set $m \geq 160(\sum_j \tau_j)\cdot M^4$, we have that

$$\Pr\left(F(x_m) - F(x_*) \leq O\left(\frac{1}{4}\left(F(x_0) - F(x_*)\right)\right)\right) \geq p - \frac{1}{10} \tag{12}$$

For the rest of the proof we will assume that the event $\mathcal{E}_1$ and the property (12) holds. Lemma 16 implies that this happens with probability at least $7/10$. An application of Markov's inequality gives us that the condition in the if statement in Algorithm 4 i.e.

$$\sum_{t=1}^{m} \frac{\|a_{i_t}\|_2}{\sqrt{p_{i_t}}} \leq 10m \sum_{k=1}^{n} \|a_k\|_2 \sqrt{p_k} = 10\mathbb{E}\left[\sum_{t=1}^{m} \frac{\|a_{i_t}\|_2}{\sqrt{p_{i_t}}}\right] \tag{13}$$

happens with probability at least $9/10$. Putting the above together via a union bound gives us that with probability at least $6/10$ all three of the following happen: $\mathcal{E}_1$, Condition (12) and the execution of the if loop (i.e. Condition 13 is met). We now show that under the above conditions we get sufficient decrease in error. Firstly note that by definition we have that

$$F_m(x') - F_m(x_m) \leq \frac{1}{512M^4}\left(F_m(x_0) - F_m(x_m)\right) \quad . \tag{14}$$

Note that if event $\mathcal{E}_1$ happens then

$$\forall\, x \quad \frac{1}{2M}\mathbf{A}^\top\mathbf{A} \leq \nabla^2 F_m(x) \leq 2M\mathbf{A}^\top\mathbf{A} \quad . \tag{15}$$

Now consider the RHS of (14)

$$\begin{aligned} F_m(x_0) - F_m(x_m) &\leq M\|x_0 - x_m\|^2_{\mathbf{A}^\top\mathbf{A}} \leq 2M(\|x_0 - x_*\|^2_{\mathbf{A}^\top\mathbf{A}} + \|x_m - x_*\|^2_{\mathbf{A}^\top\mathbf{A}}) \\ &\leq 4M^2(F(x_0) - F(x_*) + F(x_m) - F(x_*)) \\ &\leq 5M^2(F(x_0) - F(x_*)) \end{aligned} \tag{16}$$

The first inequality follows from (15), second from triangle inequality, third by noting that F is $\frac{1}{M}$ strongly convex in $\mathbf{A}^\top\mathbf{A}$ norm (Assumption (4)) and the fourth from Lemma 16. Further,

$$\begin{aligned} F(x') - F(x_m) &\leq \nabla F(x_m)^\top(x' - x_m) + \frac{M}{2}\|x' - x_m\|^2_{\mathbf{A}^\top\mathbf{A}} \\ &\leq \|\nabla F(x_m)\|_{[\mathbf{A}^\top\mathbf{A}]^{-1}}\|(x' - x_m)\|_{\mathbf{A}^\top\mathbf{A}} + \frac{M}{2}\|x' - x_m\|^2_{\mathbf{A}^\top\mathbf{A}} \\ &\leq \sqrt{2M(F(x_m) - F(x_*))}\|(x' - x_m)\|_{\mathbf{A}^\top\mathbf{A}} + \frac{M}{2}\|x' - x_m\|^2_{\mathbf{A}^\top\mathbf{A}} \\ &\leq \sqrt{2M(F(x_m) - F(x_*))}\sqrt{4M(F_m(x') - F_m(x_m))} + 2M^2(F_m(x') - F_m(x_m)) \\ &\leq \frac{1}{8M}\sqrt{(F(x_m) - F(x_*))}\sqrt{(F_m(x_0) - F_m(x_m))} + \frac{1}{256M^2}(F_m(x_0) - F_m(x_m)) \\ &\leq \frac{1}{3}(F(x_0) - F(x_*)) \end{aligned} \tag{17}$$

The first and third inequality follow by noting that F is M smooth and $1/M$ strongly convex in $\mathbf{A}^\top\mathbf{A}$ norm. Fourth inequality follows by noting that if event $\mathcal{E}_1$ holds, F_m is $1/2M$ strongly convex in $\mathbf{A}^\top\mathbf{A}$ norm. Fifth inequality follows from (14) and sixth inequality from (16) and Lemma 16. (17) together with (12) implies that with probability at least $6/10$, we have that

$$F(x') - F(x_*) \leq \frac{1}{2}(F(x_0) - F(x_*))$$

We will now bound the running time of the procedure via Theorem 15. Define L_{i_t} and μ_{i_t} to be respectively the smoothness and strong convexity parameters of the components $\tilde{f}_{i_t}/m$. Note that $L_{i_t} \leq \frac{M}{mp_{i_t}}$ and $\mu_{i_t} \geq \frac{1}{Mmp_{i_t}}$. Note that event $\mathcal{E}_1$ gives us that $\forall\ x\ \nabla^2 F_m(x) \succeq \frac{1}{2M}\lambda_{min}(\mathbf{A}^\top \mathbf{A})$. A direct application of Theorem 15 using the bounds on L_{i_t} and μ_{i_t} gives us that the total running time is bounded by

$$\tilde{O}\left(\left(\sum_{t=1}^{m} M + \sum_{t=1}^{m} \|a_{i_t}\|_2 \sqrt{\frac{M}{mp_{i_t}} \cdot \frac{M}{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}}\right) s(\mathbf{A}) \log(\epsilon^{-1})\right) \leq \tilde{O}\left(mM + \sum_{i=1}^{n} \frac{\|a_i\|_2 \sqrt{\tau_i} M^3}{\sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})}}\right)$$

The inequality follows from Condition (13) and the definitions of p_k, m.

■

Acknowledgments

Sham Kakade acknowledges funding from the Washington Research Foundation for Innovation in Data-intensive Discovery, the National Science Foundation Grant under award CCF1637360 (Algorithms in the Field) and award CCF-1703574. Rahul Kidambi acknowledges support from the NSF Award 1740822.Yin Tat Lee acknowledges support from NSF awards CCF-1749609, CCF-1740551, DMS-1839116, and Microsoft Research Faculty Fellowship. Aaron Sidford acknowledges support from the NSF CAREER Award CCF-1844855

References

Zeyuan Allen Zhu. Katyusha: Accelerated variance reduction for faster SGD. *CoRR*, abs/1603.05953, 2016.

Zeyuan Allen Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1110–1119, 2016.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 81–90, 2013. doi: 10.1145/2488608.2488620.

Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 278–287, 2016. doi: 10.1137/1.9781611974331.ch21.

Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 181–190, 2015. doi: 10.1145/2688073.2688113.

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: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1646–1654, 2014.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2540–2548, 2015a.

Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *COLT*, pages 728–763, 2015b.

Nick Harvey. Matrix concentration, 2012.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 315–323, 2013a.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013b.

S Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. 2009.

Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 127–136, 2013. doi: 10.1109/FOCS.2013.22.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3384–3392, 2015.

Jelani Nelson and Huy L. Nguyen. OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 117–126, 2013. doi: 10.1109/FOCS.2013.21.

Yu. Nesterov. A method for solving a convex programming problem with convergence rate 1/kˆ2. *Doklady AN SSSR*, 269:543–547, 1983.

Yurii Nesterov and Sebastian U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017. doi: 10.1137/16M1060182.

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: 26th Annual Conference on Neural Information Processing Systems 2012.*

Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., pages 2672–2680, 2012.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *CoRR*, abs/1209.1873, 2012.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Math. Program.*, 155(1-2):105–145, 2016. doi: 10.1007/s10107-014-0839-0.

Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 563–568, 2008. doi: 10.1145/1374376.1374456.

Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 887–898, 2012. doi: 10.1145/2213977.2214056.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems*, pages 3639–3647, 2016.

Appendix A. Computing Leverage Scores given a Regression Algorithm (Lemma 12)

In this section we give a proof of Lemma 12 which bounds the running time of computing leverage scores assuming access to a regression algorithm. The main algorithm is given as Algorithm 2.

Proof [Proof of Lemma 12] Let $y_j^* = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top v_j$ be the minimizer of $f_{\mathbf{A},v_j}(x)$. (2) shows that

$$\|\mathbf{A}y_j - \mathbf{A}y_j^*\|_2^2 \leq \epsilon \cdot v_j^\top \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top v_j.$$

Using $v_j \sim N(0, I)$, we have that

$$v_j^\top \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top v_j \leq 2d \cdot \log(n)$$

with probability $1 - n^{-\Theta(1)}$. Hence, we have that

$$\left|e_i^\top \mathbf{A}y_j - e_i^\top \mathbf{A}y_j^*\right| \leq \|\mathbf{A}y_j - \mathbf{A}y_j^*\|_2 \leq \sqrt{2\epsilon d \cdot \log(n)}.$$

Using this and

$$\left|e_i^\top \mathbf{A}y_j^*\right| \leq \sqrt{e_i^\top \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top e_i}\sqrt{v_j^\top \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top v_j} \leq \sqrt{2d \cdot \log(n)}\,,$$

we have that

$$\left|\left(e_i^\top \mathbf{A}y_j\right)^2 - \left(e_i^\top \mathbf{A}y_j^*\right)^2\right| \leq 6\sqrt{\epsilon} d \cdot \log(n).$$

Using the definition of ϵ, we have that

$$\left|\frac{1}{k}\sum_{j=1}^{k}(e_i^\top \mathbf{A}y_j)^2 - \frac{1}{k}\sum_{j=1}^{k}(e_i^\top \mathbf{A}y_j^*)^2\right| \leq 6\sqrt{\epsilon} d \cdot \log(n) \leq \frac{\delta}{3n \cdot \kappa(\mathbf{A}^\top \mathbf{A})} \tag{18}$$

Also, we note that

$$\frac{1}{k}\sum_{j=1}^{k}(e_i^\top \mathbf{A}y_j^*)^2 = \frac{1}{k}\sum_{j=1}^{k}(e_i^\top \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top v_j)^2.$$

Since $v_j \sim N(0, I)$ and $k = c\log(n)/\delta^2$ where c is some large enough constant, Johnson-Lindenstrauss lemma shows that, with high probability in n for all $i \in [n]$

$$\left(1-\frac{\delta}{3}\right)\sigma_i(\mathbf{A}) \le \frac{1}{k}\sum_{j=1}^{k}(e_i^\top \mathbf{A}y_j^*)^2 \le \left(1+\frac{\delta}{3}\right)\sigma_i(\mathbf{A})$$

Combining this with (18) gives the result.

Finally, to check the success probability of this algorithm, we note that we solved $O(\delta^{-2}\log n)$ many regression problems and each one has success probability $1 - n^{-\Theta(1)}$. Also, the Johnson–Lindenstrauss lemma succeed with probability $1 - n^{-\Theta(1)}$. This gives the result. ■

Appendix B. Proof of Invariants in Algorithm 3(Lemma 13)

Proof Note that $\mathbf{A}_\eta^\top\mathbf{A}_\eta = \mathbf{A}^\top\mathbf{A} + \eta\mathbf{I}$. Consequently, since initially $\eta = \lambda_{\max}(\mathbf{A}^\top\mathbf{A})$ we have that initially $\eta\mathbf{I} \preceq \mathbf{A}_\eta^\top\mathbf{A}_\eta \preceq 2\eta\mathbf{I}$. Consequently, we have that initially $\sigma_i(\mathbf{A}_\eta) \le u_i \le 2\sigma_i(\mathbf{A}_\eta)$ and therefore satisfies the invariant (6).

Now, suppose at the start of the repeat loop, u satisfies the invariant (6). In this case the the assumptions needed to invoke `SolveUsingLS` by Theorem 11 are satisfied. Hence, after the line $u \leftarrow 2\cdot$ `ComputeLS`$(\mathbf{A}_\eta, \frac{1}{4}, \mathcal{A})$, by Lemma 12 we have that for all $i \in [n]$

$$2\sigma_i(\mathbf{A}_\eta) \le u_i \le 2\left(1+\frac{1}{4}\right)\sigma_i(\mathbf{A}_\eta) + \frac{2}{4n\cdot\kappa(\mathbf{A}_\eta^\top\mathbf{A}_\eta)}.$$

Now, letting $\eta' = \frac{3}{4}\eta$ we see that $(3/4)\sigma_i(\mathbf{A}_\eta) \le \sigma_i(\mathbf{A}_{\eta'}) \le (4/3)\sigma_i(\mathbf{A}_\eta)$ and direct calculation shows that invariant (6) is still satisfied after changing η to η'.

All the remains is to consider the last step when we set $\eta = 0$. When this happens $\eta < \frac{1}{10}\lambda_{\min}(\mathbf{A}^\top\mathbf{A})$. and therefore $\sigma_i(\mathbf{A}_\eta)$ is close enough to $\sigma_i(\mathbf{A})$ and the invariant (6) is satisfied. ■

Appendix C. Previous Best Regression Runtime - Proof of Theorem 8

First we give the theorems encapsulating the results we use and then use them to prove Theorem 4 in the case when $\mathbf{A} = \mathbf{B}$. We then prove the case when $\mathbf{A} \neq \mathbf{B}$. Theorem 17 describes the fastest coordinate descent algorithm known by Allen Zhu et al. (2016). Theorem 18 describes the reduction Frostig et al. (2015a) to from regression to coordinate decent via proximal point.

Theorem 17 (Corollary of Thm 5.1 of Allen Zhu et al. (2016)) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable σ-strongly convex function for $\mu > 0$. Further suppose that for all $x \in \mathbb{R}^n$ and $i \in [n]$ it is the case that $\frac{\partial^2}{\partial x_i^2}f(x) \le L_i$ for $i \in [n]$ and the partial derivative $\frac{\partial}{\partial x_i}f(x)$ can be computed in $O(s)$ time. Then there exists an algorithm which given any $\epsilon > 0$ finds a $y \in \mathbb{R}^n$ such that*

$$f(y) - \min_x f(x) \le \epsilon\left(f(x_0) - \min_x f(x)\right).$$

in expected running time $O(s\sum_i \sqrt{L_i/\mu})$.

Theorem 18 (Corollary of Thm 4.3 of Frostig et al. (2015a)) *Given* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *with rows* $a_1, ..., a_n$ *and* $c \in \mathbb{R}^n$. *Consider the function* $p(x) = \sum_{i=1}^n \phi_i(a_i^\top x)$ *where* ϕ_i *are convex functions. Suppose that* $\lambda\mathbf{I} \preceq \nabla^2 p(x) \preceq L\mathbf{I}$ *for all* $x \in \mathbb{R}^d$. *Let* $\kappa = L/\lambda$. *Let dual problem* $g_s(y) = \sum_{i=1}^n \phi_i^*(y_i) + \frac{1}{2\lambda}\|\mathbf{A}^\top y\|_2^2 - s^\top \mathbf{A}^\top y$.

Suppose that for any $s \in \mathbb{R}^d$, *any* $y_0 \in \mathbb{R}^n$ *and any* $0 \le \epsilon \le \frac{1}{2}$, *we can compute* y *in expected running time* $\mathcal{T}_\epsilon$ *such that*

$$g_s(y) - \min_y g_s(y) \le \epsilon(g_s(y_0) - \min_y g_s(y)). \tag{19}$$

Then, for any x_0 *and any* $\epsilon \in (0, \frac{1}{2})$ *we can find* x *such that*

$$p(x) - \min_x p(x) \le \epsilon\left(p(x_0) - \min_x p(x)\right)$$

in time $\tilde{O}(\mathcal{T}_\delta \log(1/\epsilon))$ *w.h.p. in* n *where* $\delta = \Theta(n^{-2}\kappa^{-4})$ *and* $\tilde{O}$ *includes logarithmic factors in* n, κ.

We note that although the guarantees of Thm 5.1 of Allen Zhu et al. (2016) and Thm 4.3 of Frostig et al. (2015a) are not stated in the form of Theorems 17 and 18. They can be easily converted to the form above by noticing that the expected running time of the procedure in Thm 4.3 of Frostig et al. (2015a) using Theorem 17 is $\tilde{O}(T_\delta \log(1/\epsilon))$ which can then be boosted to high probability in n using Lemma 19. We now give the proof of Theorem 8.

Proof [Proof of Theorem 8 when $\mathbf{A} = \mathbf{B}$] Let

$$p(x) = \sum_{i=1}^n \phi_i(a_i^\top x) \text{ where } \phi_i(x) = \frac{1}{2}(x - b_i)^2 .$$

Then, we have that $\phi_i^*(y) = \frac{1}{2}y^2 + b_i y$ and hence

$$g_s(y) = \sum_{i=1}^n \phi_i^*(y_i) + \frac{1}{2\lambda}\|\mathbf{A}^\top y\|_2^2 - s^\top \mathbf{A}^\top y = \frac{1}{2}\|y\|_2^2 + b^\top y + \frac{1}{2\lambda}\|\mathbf{A}^\top y\|_2^2 - s^T \mathbf{A}^T y.$$

Note that $g_s(y)$ is 1 strongly convex and

$$\frac{d^2}{dy_i^2} g_s(y) = 1 + \frac{1}{\lambda}\|a_i\|_2^2 \stackrel{\text{def}}{=} L_i .$$

Hence, Theorem 17 finds y satisfying (19) in time

$$O\left(s(\mathbf{A}) \cdot \sum_{i\in[n]} \sqrt{1 + \frac{1}{\lambda}\|a_i\|_2^2} \log(\epsilon^{-1})\right) = O\left(\left(n + \frac{1}{\sqrt{\lambda}}\sum_{i\in[n]} \|a_i\|_2\right) \cdot s(\mathbf{A}) \cdot \log(\epsilon^{-1})\right).$$

Hence, this shows that the primal can be solved in time

$$O\left(\left(n + \frac{1}{\sqrt{\lambda}}\sum_{i\in[n]} \|b_i\|_2\right) \cdot s \cdot \log(n \cdot \kappa) \cdot \log(\kappa\epsilon^{-1})\right)$$

where we used $\mathbf{A} = \mathbf{B}$ at the end. ■

Proof [Proof of Theorem 8 for the case $\mathbf{A} \neq \mathbf{B}$] The proof involves two steps. First, we show that given any point x_0, we can find a new point x that is closer to the minimizer. Then, we bound how many steps it takes. To find x, we consider the function

$$f_{x_0}(x) = \frac{1}{2}\|\mathbf{B}x - \mathbf{B}x_0\|_2^2 + \langle \mathbf{A}x_0 - c, \mathbf{A}x - \mathbf{A}x_0 \rangle .$$

Let z be the minimizer of f_{x_0} and x^* be the minimizer of $\frac{1}{2}\|\mathbf{A}x - b\|_2^2$. Note that

$$z = x_0 - (\mathbf{B}^\top \mathbf{B})^{-1}\mathbf{A}^\top \eta \text{ with } \eta = \mathbf{A}x_0 - b, \text{ and } x^* = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top b.$$

Hence, we have that

$$\begin{aligned}\frac{1}{2}\|\mathbf{A}z - \mathbf{A}x^*\|_2^2 &= \frac{1}{2}\|\mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \eta - \mathbf{A}(\mathbf{B}^\top \mathbf{B})^{-1}\mathbf{A}^\top \eta\|_2^2 \\ &= \frac{1}{2}\eta \mathbf{A}^\top (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \eta - \eta \mathbf{A}^\top (\mathbf{B}^\top \mathbf{B})^{-1}\mathbf{A}^\top \eta + \frac{1}{2}\|\mathbf{A}(\mathbf{B}^\top \mathbf{B})^{-1}\mathbf{A}^\top \eta\|_2^2.\end{aligned}$$

Using that $\frac{5}{6}\mathbf{B}^\top \mathbf{B} \preceq \mathbf{A}^\top \mathbf{A} \preceq \frac{6}{5}\mathbf{B}^\top \mathbf{B}$, we have

$$\frac{1}{2}\|\mathbf{A}z - \mathbf{A}x^*\|_2^2 \leq \frac{4}{10}\eta \mathbf{A}^\top (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \eta = \frac{4}{10}\|\mathbf{A}x_0 - \mathbf{A}x^*\|_2^2. \tag{20}$$

However, it is difficult to reduce to the case when $\mathbf{A} = \mathbf{B}$ to minimize the function f_{x_0} due to the extra linear term. To address this issue, we assume $\mathbf{B} = [\overline{\mathbf{B}}; \sqrt{\frac{\lambda}{100}}\mathbf{I}]$ by appending an extra identity term. Note that this only adds a small matrix $\frac{\lambda}{100}\mathbf{I}$ and hence we still have $\frac{5}{6}\mathbf{B}^\top \mathbf{B} \preceq \mathbf{A}^\top \mathbf{A} \preceq \frac{6}{5}\mathbf{B}^\top \mathbf{B}$ but with a slightly different constant which will not affect the proof for (20). Due to the extra identity term, $f_{x_0}(x)$ reduces to an expression of the form $\frac{1}{2}\|\mathbf{B}x - d\|_2^2 + C$ for some vector d and constant C. We can now apply Theorem 8 for the case $\mathbf{A} = \mathbf{B}$ and get an x such that

$$f_{x_0}(x) - f_{x_0}(z) \leq \frac{1}{200}\left(f_{x_0}(x_0) - f_{x_0}(z)\right). \tag{21}$$

in time

$$O\left(\left(n + \frac{\sum_{i\in[n]}\|b_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{B}^\top \mathbf{B})}}\right) \cdot s(\mathbf{B}) \cdot \log(n\kappa) \cdot \log(\kappa)\right) .$$

Note that the extra terms in $\mathbf{B}$ does not affect the minimum eigenvalue and it increases $\frac{1}{\sqrt{\lambda}}\sum_{i\in[n]}\|b_i\|_2$ by atmost n.

Now, using the formula of z, the guarantee (21) can be written as

$$\|\mathbf{B}x - \mathbf{B}z\|_2^2 \leq \frac{1}{200}\|\mathbf{A}x_0 - \mathbf{A}x^*\|_2^2 .$$

Using that $\frac{5}{6}\mathbf{B}^\top \mathbf{B} \preceq \mathbf{A}^\top \mathbf{A} \preceq \frac{6}{5}\mathbf{B}^\top \mathbf{B}$, we have

$$\|\mathbf{A}x - \mathbf{A}z\|_2 \leq \frac{1}{10}\|\mathbf{A}x_0 - \mathbf{A}x^*\|_2 .$$

Combining this with (20), we have that

$$\|\mathbf{A}x - \mathbf{A}x^*\|_2 \leq 0.9\|\mathbf{A}x_0 - \mathbf{A}x^*\|_2.$$

Hence, we get closer to x^* by constant factor. Therefore, to achieve (2), we only need to repeat this process $\log(1/\epsilon)$ times. Hence, the total running time is

$$O\left(\left(n + \frac{\sum_{i\in[n]}\|b_i\|_2}{\sqrt{\lambda_{\min}(\mathbf{B}^\top\mathbf{B})}}\right)\cdot s(\mathbf{B})\log^2(n\kappa)\log(\epsilon^{-1})\right)$$

■

Appendix D. Reduction from High Probability Solvers to Expected Running Times - Proof of Theorem 7

In this section we provide Lemma 19 which reduces the problem of achieving ϵ accuracy with high probability to the problem of achieving an accuracy c with probability at least δ for some constants c, δ. Note that a naive reduction suffers an additional $\log\log(1/\epsilon)$ term which we avoid. The reduction helps provide a concise proof of Theorem 7 based on Theorem 14.

Lemma 19 *Consider being given a function $F : \mathbb{R}^d \to \mathbb{R}$ and define $x^* \stackrel{\text{def}}{=} \operatorname{argmin}_x F(x)$. Let $\mathcal{A}$ be an algorithm such that given any point x_0 the algorithm runs in time $\mathcal{T}$ and produces a point x' such that*

$$F(x') - F(x^*) \leq c\,(F(x_0) - F(x^*))$$

with probability at least $1-\delta$ for given universal constants $c, \delta \in [0,1]$. Further suppose there exists a procedure $\mathcal{P}$ which given a point x can produce an estimate m in time $\mathcal{T}'$ such that $F(x) - F(x^*) \in [m/r, rm]$ for some given $r \geq 1$. Then there exists a procedure that given a point x_0 outputs a point x' such that

$$F(x') - F(x^*) \leq \epsilon\,(F(x_0) - F(x^*))$$

and the expected running time of the procedure is bounded by $O\left((\mathcal{T}+\mathcal{T}')\log(r)\log(\epsilon^{-1})\right)$ where $O()$ hides constant factors in c, δ. Moreover for any γ we have a procedure that produces a point x' such that

$$F(x') - F(x^*) \leq \epsilon\,(F(x_0) - F(x^*))$$

with probability at least $1-\gamma$ with a total running time of $O\left((\mathcal{T}+\mathcal{T}')\log(r)\log(\epsilon^{-1})\log(\gamma^{-1})\right)$

We first use Lemma 19 to prove Theorem 7 and then provide a proof of Lemma 19.

Proof [Proof of Theorem 7] We make use of Lemma 19 plugging in Algorithm 4 as the procedure $\mathcal{A}$. Note that c, δ are both $1/2$ as guaranteed by Theorem 14. Moreover since $F(x)$ is such that that $\forall \quad x \quad M\lambda_{min}(\mathbf{A}^\top\mathbf{A}) \preceq \nabla^2 F(x) \preceq M\lambda_{max}(\mathbf{A}^\top\mathbf{A})$, we can use $\|\nabla F(x)\|_2^2$ as an estimator for $F(x) - F(x^*)$. The corresponding r for it is bounded by $M^2\kappa(\mathbf{A}^\top\mathbf{A})$.

Finally note that the running time guaranteed in Theorem 14 depends on the quality of the estimates of the leverage scores input to it. We invoke Lemma 12 for computing accurate estimates of leverage scores. Putting together the above arguments finishes the proof for Theorem 7. ■

Proof [Proof of Lemma 19] To show the lemma we will show the existence of a procedure (described in Algorithm 5) which produces a point x' such that

$$F(x') - F(x^*) \leq 1/2\,(F(x_0) - F(x^*)) \tag{22}$$

with expected running time bounded by $O\left((\mathcal{T} + \mathcal{T}')\log(r)\right)$. Applying this procedure $O(\log(\epsilon^{-1})$ and using linearity of expectation gives us the Lemma 19. Consider the following procedure to prove Lemma 19.

Algorithm 5: `Reduction`$(x_0, F(x), \mathcal{P}, \mathcal{A}, c, \delta, r)$

Set $T = \log_{c^{-1}}(2r^2)$
repeat
  for $i = 0 \to T$ **do**
    for $j = 0 \to \log_{\delta^{-1}}(2\log_{c^{-1}}(2r^2)$ **do**
      Set $x_{ij} = \mathcal{A}(x_i, F)$
    end
    Set $x_{i+1} = \min_j x_{ij}$
  end
  Compute error estimates $E_1 = \mathcal{P}(x_0), E_2 = \mathcal{P}(x_T)$
  Set $E = \frac{E_2}{E_1}$.
until $E \leq 0.5$;
Output: x_T

Note that since for every x_{ij} we have that

$$F(x_{ij}) - F(x^*) \leq c\,(F(x_i) - F(x^*))$$

with probability at least δ, therefore we have that

$$F(x_{i+1}) - F(x^*) \leq c\,(F(x_i) - F(x^*))$$

with probability at least $1 - \delta^{\log_{\delta^{-1}}(2\log_{c^{-1}}(r^2)} = 1 - \frac{1}{2\log_{c^{-1}}(r^2)}$. Taking a union bound over the outer loop gives us that with probability at least $1/2$ we have that

$$F(x_T) - F(x^*) \leq \frac{1}{2r^2}\,(F(x_i) - F(x^*))$$

Moreover by the property of the estimates given by $\mathcal{P}$ we know that in this case we have that $E \leq 0.5$. Therefore we have that with probability at least $1/2$ the repeat loop computes an x_T that reduces error by at least a factor of $1/2$ and we can verify it. Therefore in expectation the loop runs a total of 2 times. The total runtime of the above procedure can easily seen to be $O\left((\mathcal{T} + \mathcal{T}')\log(r)\log(\epsilon^{-1}\right)$.

Further suppose we are given a procedure with the guarantee that for any ϵ in expected running time $\mathcal{T}_\epsilon$ it produces a point x' such that

$$F(x') - \min F(x) \leq \epsilon(F(x_0) - \min F(x))$$

We now run this procedure for time $\mathcal{T}_{\epsilon/2}$. By Markov's inequality with probability at least $1/2$ we have a point that satisfies

$$F(x') - \min F(x) \leq \epsilon(F(x_0) - \min F(x))$$

It is now easy to see that if we repeat the above procedure $\log(\gamma^{-1})$ many times and take the x with the minimum value we have a point x' such that

$$F(x') - \min F(x) \leq \epsilon(F(x_0) - \min F(x))$$

with probability at least $1 - \gamma$. ■

Appendix E. Proofs of ERM Sampling (Lemma 16)

Proof [Proof of Lemma 16] Consider the definitions in (7), (8), (9). Note the following easy observation.

$$F(x) = \mathbb{E}_{k\sim\mathcal{D}}\tilde{f}_k(x)$$

Consider the following Lemma 20 which connects the optima of two convex functions F and G.

Lemma 20 *Let $F(x), G(y) : \mathbb{R}^d \to R$ be twice differentiable and strictly convex. Define*

$$x_* = \text{argmin}_x F(x) \text{ and } y_* = \text{argmin}_y G(y)$$

Then we have that

$$F(y_*) - F(x_*) = \|\nabla G(x_*)\|^2_{\mathbf{H}_G^{-1}\mathbf{H}_F\mathbf{H}_G^{-1}}.$$

where $\mathbf{H}_F \stackrel{\text{def}}{=} \int_0^1 \nabla^2 F(t.y_* + (1-t)x_*)dt$ and $\mathbf{H}_G \stackrel{\text{def}}{=} \int_0^1 \nabla^2 G(t.y_* + (1-t)x_*)dt$.

We wish to invoke Lemma 20 by setting $F = F(x), G = F_m(x)$. In this setting we have that

$$\mathbf{H}_F \stackrel{\text{def}}{=} \int_0^1 \nabla^2 F(t.x_m + (1-t)x_*)dt \text{ and } \mathbf{H}_G \stackrel{\text{def}}{=} \int_0^1 \nabla^2 F_m(t.x_m + (1-t)x_*)dt$$

Firstly note that the definition of F and Assumption (4) gives us that

$$\mathbf{H}_F \preceq M \cdot \mathbf{A}^\top\mathbf{A} \tag{23}$$

Using Definition 10 and Assumption (4) gives us that

$$\mathbf{H}_G \preceq M \cdot \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}$$

Combining the above two and noting that the event $\mathcal{E}_1$ happens with probability at least p we get that

$$\mathbf{H}_G^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{H}_G^{-1} \preceq 4M^2[\mathbf{A}^\top\mathbf{A}]^{-1} \text{ w.p. } p \tag{24}$$

Also note that for any fixed matrix $\mathbf{R}$, we have that

$$\mathbb{E}[\|\nabla F_m(x_*)\|^2_{\mathbf{R}}] = \frac{\mathbb{E}_{k\sim\mathcal{D}}[\|\nabla\tilde{f}_k(x_*)\|^2_{\mathbf{R}}]}{m}$$

which implies via Markov's inequality that with probability at least $9/10$ we have that

$$\|\nabla F_m(x_*)\|_{\mathbf{R}}^2 \leq \frac{10\mathbb{E}_{k\sim\mathcal{D}}[\|\nabla \tilde{f}_k(x_*)\|_{\mathbf{R}}^2]}{m} \tag{25}$$

Putting (24) and (25) together and using a union bound we get that

$$\|F_m(x_*)\|_{\mathbf{H}_G^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{H}_G^{-1}}^2 \leq \frac{40M^2\mathbb{E}_{k\sim\mathcal{D}}[\|\nabla \tilde{f}_k(x_*)\|_{[\mathbf{A}^\top\mathbf{A}]^{-1}}^2]}{m} \text{ w.p. } p - 1/10$$

Using Lemma 20 and (23) we get that with probability at least $p - 1/10$

$$F(x_m) - F(x_*) \leq \frac{40M^3\mathbb{E}_{k\sim\mathcal{D}}[\|\nabla \tilde{f}_k(x_*)\|_{[\mathbf{A}^\top\mathbf{A}]^{-1}}^2]}{m} \tag{26}$$

We will now connect $\mathbb{E}_{k\sim\mathcal{D}}[\|\nabla \tilde{f}_k(x_*)\|_{[\mathbf{A}^\top\mathbf{A}]^{-1}}^2]$ with the error at x_i.

Lemma 21 *Consider an ERM function $F(x) = \sum_{i=1}^m f_i(x)$ where $f_i(x) = \psi_i(a_i^\top x)$ with $\psi_i'' \in [\frac{1}{M}, M]$. Define a distribution $\mathcal{D}(j)$ over $[n]$ such that $Pr(j = k) = p_k \stackrel{\text{def}}{=} \frac{\tau_k}{\sum \tau_k}$ for numbers $\tau_k \stackrel{\text{def}}{=} \min(1, 20u_k \log(d))$ where $u_k \geq \sigma_i(\mathbf{A})$[6] are overestimates of leverage scores. Given a point $\tilde{x}$ consider the variance reduced reformulation*

$$F(x) = \mathbb{E}_{k\sim D}[\tilde{f}_k(x)]$$

where

$$\tilde{f}_k(x) \stackrel{\text{def}}{=} \frac{1}{p_k}\left[f_k(x) - \nabla f_k(\tilde{x})^\top x\right] + \nabla F(\tilde{x})^\top x$$

Then we have that

$$\mathbb{E}_{k\sim\mathcal{D}}\left[\|\nabla \tilde{f}_k(x_*)\|_{[\mathbf{A}^\top\mathbf{A}]^{-1}}^2\right] \leq 2(\sum_j \tau_j)\cdot M\cdot(F(\tilde{x}) - F(x_*))$$

Putting together (26) and Lemma 21 we get that

$$F(x_m) - F(x_*) \leq 80\left(\frac{(\sum_j \tau_j)\cdot M^4}{m}\cdot(F(x_0) - F(x_*))\right) \text{ w.p } p - \frac{1}{10}$$

Lemma 16 now follows from the choice of m. ■

We finish this section with proofs of Lemma 20 and 21

Proof [Proof of Lemma 20] For all $t \in [0,1]$ let $z(t) \stackrel{\text{def}}{=} t\cdot y_* + (1-t)\cdot x_*$ for $t \in [0,1]$ and $\mathbf{H}_F \stackrel{\text{def}}{=} \int_0^1 \nabla^2 F(z(t))dt$. By Taylor series expansion we have that

$$\begin{aligned} F(y_*) &= F(x_*) + \nabla F(x_*)^\top(y_* - x_*) + \int_0^1 \frac{1}{2}(y_* - x_*)^\top\nabla^2 F(z(t))(y_* - x_*)dt \\ &= F(x_*) + \frac{1}{2}\|y_* - x_*\|_{\mathbf{H}_F}^2. \end{aligned}$$

6. σ_i are leverage scores defined in Definition 9

Here we used that $\nabla F(x_*) = 0$ and $\nabla^2 F(z(t)) \succeq \mathbf{0}$ by the convexity of F. We also have by definition that

$$\nabla G(y_*) = \mathbf{0}$$

and therefore

$$\nabla G(y_*) - \nabla G(x_*) = \int_0^1 \nabla^2 G(z(t))(y_* - x_*) \cdot dt$$

and

$$(y_* - x_*) = -\mathbf{H}_G^{-1} \nabla G(x_*)$$

where $\mathbf{H}_G \stackrel{\text{def}}{=} \int_0^1 \nabla^2 G(z(t))$. We now have that

$$F(y_*) - F(x_*) = \frac{1}{2}\|y_* - x_*\|_{\mathbf{H}_F}^2 = \|\nabla G(x_*)\|_{\mathbf{H}_G^{-1}\mathbf{H}_F\mathbf{H}_G^{-1}}^2 \tag{27}$$

■

Proof [Proof of Lemma 21] For the purpose of this proof it will be convenient to perform a change of basis. Define the function

$$G(x) = \mathbb{E}_{k\sim\mathcal{D}} g_i(x) \text{ where } g_i(x) = \frac{1}{p_i} f_k((\mathbf{A}^\top \mathbf{A})^{-1/2} x)$$

Note that $G(x) = F((\mathbf{A}^\top \mathbf{A})^{-1/2})x)$. We will first note that

$$\nabla^2 g_i(x) = \frac{1}{p_i} \cdot \left[(\mathbf{A}^\top \mathbf{A})^{-1/2} a_i a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1/2} \cdot \psi_i''(a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1/2} x)\right]$$

and now by the cyclic property of trace and the fact that $\psi_i'' \leq M$ we have

$$\mathrm{tr}(\nabla^2 g_i(x)) = \frac{(\sum_j \tau_j)}{\tau_i} \cdot a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} a_i \cdot M$$

Note that $a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} a_i \leq 1$. Now either $\tau_i = 1$ or $\tau_i \geq 20 a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} a_i \log(d)$. In both cases we see that RHS above ≤ 1. Therefore we get that g_i is $(\sum_j \tau_j)M$ smooth. We now have the following lemma.

Lemma 22 *Let $\mathcal{D}$ be any distribution over $[n]$ and define $G = \mathbb{E}_{i\sim\mathcal{D}}[g_i(x)]$ for component convex functions g_i each of which is L smooth. Let $x_* \stackrel{\text{def}}{=} \operatorname{argmin} G(x)$. We have that*

$$\mathbb{E}_{i\sim D}\|\nabla g_i(x) - \nabla g_i(x_*)\|_2^2 \leq 2L(G(x) - G(x^*))$$

The proof of the above Lemma is identical to the proof of Equation 8 in Johnson and Zhang (2013b) and we provide the proof for completeness. Assuming the Lemma, note that

$$\begin{aligned}
2(\sum_j \tau_j) M(F(\tilde{x}) - F(x_*)) &= 2(\sum_j \tau_j) \cdot M \cdot (G((\mathbf{A}^\top \mathbf{A})\tilde{x}) - G((\mathbf{A}^\top \mathbf{A})x_*)) \\
&\geq \mathbb{E}_{i\sim\mathcal{D}}\|\nabla g_i((\mathbf{A}^\top \mathbf{A})\tilde{x}) - \nabla g_i((\mathbf{A}^\top \mathbf{A})x_*)\|_2^2 \\
&= \mathbb{E}_{i\sim\mathcal{D}}\|(\mathbf{A}^\top \mathbf{A})^{-1/2} \frac{1}{p_i}(\nabla f_i(\tilde{x}) - \nabla f_i(x_*))\|_2^2 \\
&= \mathbb{E}_{i\sim D}\|\nabla \tilde{f}_i(x_*)\|_{(\mathbf{A}^\top \mathbf{A})^{-1}}^2 - 2\mathbb{E}_{i\sim D}\left[\frac{1}{p_i}(\nabla f_i(x_*) - \nabla f_i(\tilde{x}))^\top [\mathbf{A}^\top \mathbf{A}]^{-1} \nabla F(\tilde{x})\right] \\
&\qquad - \|\nabla F(\tilde{x})\|_{[\mathbf{A}^\top \mathbf{A}]^{-1}}^2 \\
&\geq \mathbb{E}_{i\sim D}\|\nabla \tilde{f}_i(x_*)\|_{(\mathbf{A}^\top \mathbf{A})^{-1}}^2
\end{aligned}$$

The first line follows by definition. The second line by 22 and by noting that g is $(\sum_j \tau_j)M$ smooth. The third and fourth line follows by definition. The fifth line follows by noting that $\nabla F(x_*) = 0$. ■

Proof [Proof of Lemma 22] Let $x_* \stackrel{\text{def}}{=} \operatorname{argmin} g(x)$. Define auxiliary functions

$$h_i(x) \stackrel{\text{def}}{=} g_i(x) - g_i(x_*) - \nabla g_i(x_*)^\top (x - x_*)$$

We know that $h_i(x_*) = \min h_i(x)$ since $\nabla h_i(x_*) = 0$. Using smoothness of h and that $h_i(x_*) = 0$, we now have that

$$\|\nabla h_i(x)\|_2^2 \leq 2L h_i(x)$$

A simple substitution gives us that for all i

$$\|\nabla g_i(x) - \nabla g_i(x_*)\|_2^2 \leq 2L \left(g_i(x) - g_i(x_*) - \nabla g_i(x_*)^\top (x - x_*) \right)$$

Taking expectations and using the fact that $g(x_*) = 0$ gives us that

$$\mathbb{E}_{i \sim D} \|\nabla g_i(x) - \nabla g_i(x_*)\|_2^2 \leq 2L(g(x) - g(x^*))$$

■

Appendix F. Accelerated Coordinate Descent for ERM - Proof of Theorem 15

Proof To remind the reader

$$f(x) = \sum_{i=1}^n \psi_i(a_i^\top x) \text{ where } \psi_i''(x) \in [\mu_i, L_i] \, .$$

Following is a well known theorem. For a proof see Kakade et al. (2009).

Theorem 23 (Strong / Smooth Duality) *A closed and convex function f is β-strongly convex with respect to a norm $\|\cdot\|$ if and only if f^* is $\frac{1}{\beta}$-strongly smooth w.r.t the dual norm of $\|\cdot\|$.*

A direct application of the above theorem gives us that $\psi_i^{*\prime\prime}(y) \in [\frac{1}{L_i}, \frac{1}{\mu_i}]$. Consider the function

$$g_s(y) = \sum_{i=1}^n \psi_i^*(y_i) + \frac{1}{2\lambda} \|\mathbf{A}^\top y\|_2^2 - s^\top \mathbf{A}^\top y$$

Consider the following modified function $\tilde{g}_s(y) \stackrel{\text{def}}{=} g_s(\mathbf{D}y)$ where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = L_i$. We will equivalently minimize the function $\tilde{g}_s(y_i)$. We now immediately get that the function $\tilde{g}_s(y)$ is 1 strongly convex. Moreover we have that

$$\frac{d^2}{dy_i^2} g_s(y) = \frac{L_i}{\mu_i} + \frac{1}{\lambda} \|a_i\|^2 L_i \, .$$

Hence, Theorem 17 finds y satisfying (19) in time

$$O\left(s(\mathbf{A})\cdot\sum_{i\in[n]}\sqrt{\frac{L_i}{\mu_i}+\frac{1}{\lambda}\|a_i\|^2L_i}\log(\epsilon^{-1})\right)=O\left(\left(\sum_{i=1}^n\sqrt{\frac{L_i}{\mu_i}}+\frac{1}{\sqrt{\lambda}}\sum_{i=1}^n\|a_i\|\sqrt{L_i}\right)s(\mathbf{A})\log(\epsilon^{-1})\right)$$

A direct application of Theorem 18 gives that the total running time is

$$O\left(\left(\sum_{i=1}^n\sqrt{\frac{L_i}{\mu_i}}+\frac{1}{\sqrt{\lambda}}\sum_{i=1}^n\|a_i\|\sqrt{L_i}\right)s(\mathbf{A})\log(n\kappa)\log(\kappa/\epsilon)\right)$$

The above equation assumes that the inner iterations of accelerated coordinate descent can be implemented in $O(s(\mathbf{A}))$. This is easy to see because diagonal scaling is linear in sparsity. Therefore the only bottleneck is computing the gradient of the dual function ψ^*. We can assume that ψ is explicit and therefore the gradient of ψ^* is easily computed. ■

Appendix G. A Matrix Concentration Inequality for Sampling with Replacement

Lemma 24 *Given an error parameter $0\le\epsilon\le1$, let u be a vector of leverage score overestimates, i.e. $\sigma_i(\mathbf{A})\le u_i$ for all i. Let $\alpha=\epsilon^{-2}$ be a sampling rate parameter and c be a fixed constant. For each row we define a number $\gamma_i=\min\{1,\alpha cu_i\log(d)\}$ and a probability $p_i=\frac{\gamma_i}{\sum\gamma_i}$. Let Y_j be a random variable which is sampled by picking a vector a_i with probability p_i and setting $Y_j=\frac{a_ia_i^\top}{p_i}$. Now consider the random variable $Y=\frac{1}{m}\sum_jY_j$. We have that as long as $m\ge\sum_i\gamma_i$ then*

$$\Pr((1-\epsilon)\mathbf{A}^\top\mathbf{A}\preceq Y\preceq(1+\epsilon)\mathbf{A}^\top\mathbf{A})\ge1-d^{-c/3}$$

Proof The proof of the lemma follows the proof of Lemma 4 in Cohen et al. (2015). We only state the differences. We use the inequality given in Harvey (2012).

Lemma 25 *Let $Y_1\dots Y_k$ be independent random positive semidefinite matrices of size $d\times d$. Let $Y=\sum Y_i$ and let $Z=\mathbb{E}[Y]$. If $Y_i\preceq R.Z$ then*

$$\Pr\left[\sum Y_i\preceq(1-\epsilon)Z\right]\le de^{-\frac{\epsilon^2}{2R}}\quad\textit{and}\quad\Pr\left[\sum Y_i\succeq(1+\epsilon)Z\right]\le de^{-\frac{\epsilon^2}{3R}}\ .$$

Note that the expectation of $Y_j/m=a_ia_i^\top/m$. Moreover note that each

$$\frac{Y_j}{m}\preceq\max_i\frac{a_ia_i^\top\sum_k\gamma_k}{m\gamma_i}\preceq\frac{\mathbf{A}^\top\mathbf{A}}{c\log d\epsilon^{-2}}$$

The inequality follows from noting that $m\ge\sum\gamma_i$ and Equation 10 in Cohen et al. (2015). The calculations now follow exactly in the same way as in the proof in Cohen et al. (2015). ■