Nearly Linear Row Sampling Algorithm for Quantile Regression

Yi Li [1] Ruosong Wang [2] Lin Yang [3] Hanrui Zhang [4]

Abstract

We give a row sampling algorithm for the quantile loss function with sample complexity nearly linear in the dimensionality of the data, improving upon the previous best algorithm whose sampling complexity has at least cubic dependence on the dimensionality. Based upon our row sampling algorithm, we give the fastest known algorithm for quantile regression and a graph sparsification algorithm for balanced directed graphs. Our main technical contribution is to show that Lewis weights sampling, which has been used in row sampling algorithms for ℓ_p norms, can also be applied in row sampling algorithms for a variety of loss functions. We complement our theoretical results by experiments to demonstrate the practicality of our approach.

1. Introduction

Linear regression is a fundamental problem in machine learning and statistics. For a data matrix $A \in \mathbb{R}^{n\times d}$ and a response vector $b \in \mathbb{R}^n$ with $n \gg d$, the overconstrained linear regression problem can be formulated as solving the optimization problem $\min_{x\in\mathbb{R}^d} \phi(Ax-b)$ where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a loss function. Via the technique of row sampling, we have obtained remarkable speedups for solving linear regression for a wide range of loss functions. Such techniques involve designing an importance sampling scheme and showing that if one samples the data matrix A and the response vector b accordingly, $\phi(Ax-b)$ is approximately preserved for all $x \in \mathbb{R}^d$. Thus, one can obtain an approximate solution to the original linear regression problem by solving a linear regression instance on the sampled data matrix and response vector, which is usually much smaller in size.

The technique of row sampling has received much attention in randomized numerical linear algebra and machine learning. It is known that if one samples $O(d \log d/\varepsilon^2)$ rows of A according the to leverage scores, the resulting matrix A' would satisfy

$$(1-\varepsilon)\|Ax\|_2 \le \|A'x\|_2 \le (1+\varepsilon)\|Ax\|_2$$

for all $x \in \mathbb{R}^d$ with high probability (Tropp, 2012). Moreover, such row sampling algorithm can be implemented in $\widetilde{O}(\mathrm{nnz}(A) + \mathrm{poly}(d))$ time[1] (Clarkson & Woodruff, 2013; Meng & Mahoney, 2013; Nelson & Nguyen, 2013; Li et al., 2013; Cohen et al., 2015), where $\mathrm{nnz}(A)$ is the number of non-zero entries of A. Building upon the result of Clarkson (2005) for $p = 1$, Dasgupta et al. (2009) obtain the first row sampling algorithm for general ℓ_p norms. They show that by sampling $\widetilde{O}(d^{\max\{p/2+1,p\}+1}/\varepsilon^2)$ rows of A according to the ℓ_p leverage scores, the resulting matrix A' satisfies

$$(1-\varepsilon)\|Ax\|_p \le \|A'x\|_p \le (1+\varepsilon)\|Ax\|_p$$

for all $x \in \mathbb{R}^d$ simultaneously with high probability. The algorithm in (Dasgupta et al., 2009) runs in $\widetilde{O}(nd^5)$ time, and the running time is further improved to $\widetilde{O}(\mathrm{nnz}(A) + \mathrm{poly}(d))$ in (Sohler & Woodruff, 2011; Clarkson et al., 2016; Meng & Mahoney, 2013; Woodruff & Zhang, 2013; Wang & Woodruff, 2019). Later, based on results in Banach space theory (Talagrand, 1990; 1995; Bourgain et al., 1989), Cohen & Peng (2015) show that by sampling according to ℓ_p Lewis weights, the number of sampled rows can be further reduced to $\widetilde{O}(d/\varepsilon^2)$ when $p < 2$ and $\widetilde{O}(d^{p/2}/\varepsilon^2)$ when $p \ge 2$ which is nearly optimal as recently shown by Li et al. (2020). Row sampling algorithms are also obtained for other loss functions, including M-estimators (Clarkson & Woodruff, 2015b;a), the Tukey loss function (Clarkson et al., 2019) and Orlicz norms (Song et al., 2019).

Yang et al. (2013) study row sampling algorithms for the quantile loss function. Let $\tau \in (0, 1]$ be a parameter and $y = Ax - b$. The quantile loss function is defined to be

*Equal contribution [1]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore [2]Department of Computer Science, Carnegie Mellon University, USA [3]Department of Electrical and Computer Engineering, University of California Los Angeles, USA [4]Department of Computer Science, Duke University, USA. Correspondence to: Yi Li <yili@ntu.edu.sg>, Ruosong Wang <ruosongw@andrew.cmu.edu>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

[1]Throughout the paper, we use $\widetilde{O}(f)$ to denote $O(f \,\mathrm{polylog}\, f)$.

$h_\tau(y) = \sum_{i=1}^n h_\tau(y_i)$, where

$$h_\tau(y_i) = \begin{cases} \tau y_i & \text{if } y_i \geq 0 \\ (\tau - 1) y_i & \text{if } y_i \leq 0 \end{cases}. \qquad (1)$$

The corresponding quantile regression problem is of particular interest, due to its wide applications in medicine (Cole & Green, 1992; Royston & Altman, 1994; Royston & Wright, 2000), in hydrology (Pandey & Nguyen, 1999), in economics (Koenker & Hallock, 2001; Koenker & Bassett Jr, 1978), in survival analysis (Koenker & Geling, 2001), etc. See, e.g., (Koenker & Hallock, 2001; Davino et al., 2013) for surveys on theory and applications of quantile regression. Quantile regression is also an active research topic in statistical machine learning. See, e.g., (Chen et al., 2019; Chowdhury et al., 2019; Ota et al., 2019; Zheng et al., 2018; Zhao et al., 2017) for recent developments. Using the notion of the ℓ_1 leverage score (Dasgupta et al., 2009) and known ℓ_1 oblivious subspace embeddings (Sohler & Woodruff, 2011; Meng & Mahoney, 2013), Yang et al. (2013) obtain the first quantile regression algorithm that runs in $\widetilde{O}(\mathrm{nnz} + \mathrm{poly}(d/\varepsilon))$ time. More specifically, they show that by sampling $O(\tau/(1-\tau) \cdot d^3/\varepsilon^2)$ rows according to the ℓ_1 leverage scores, $h_\tau(Ax)$ is preserved for all $x \in \mathbb{R}^d$ up to a factor of $1 \pm \varepsilon$. Furthermore, quantile regression can be formulated as a linear program, and thus one can readily solve the smaller instance returned by the sampling process in $\mathrm{poly}(d/\varepsilon)$ time.

Although the results in (Yang et al., 2013) are interesting from a theoretical point of view, the d^3 factor in the number of sampled rows required by their algorithm is far from being practical due to the rise of large-scale datasets. Even for a moderate-sized dataset with $d = 300$, the algorithm in (Yang et al., 2013) requires at least $d^3 = 2.7 \times 10^7$ samples, which is too large to gain any speedup over the naïve implementation without using row sampling. For modern large-scale datasets, e.g., the ImageNet dataset (Deng et al., 2009), d is much larger than 10^5 even after pre-processing. In that case, even quadratic number of samples is too large to be practical. Thus, one may naturally ask whether it is possible to obtain a row sampling algorithm for the quantile loss function which requires only nearly linear number of samples, similar to the case of ℓ_p norms when $1 \leq p \leq 2$. In this paper, we provide an affirmative answer to this question.

Our Contributions. We provide the first row sampling algorithm with nearly linear number of sampled rows for the quantile loss function. Note that we can write $Ax - b = A'x'$ for $A' = [A\ b] \in \mathbb{R}^{n \times (d+1)}$ (concantenation of A and b) and $x' = \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right] \in \mathbb{R}^{d+1}$, we overload the notations by renaming A' to A and x' to x and replacing d with $d+1$, and consider instead preserving loss functions on Ax. For notational simplicity, we consider the equivalent problem of preserving

$$\rho_\tau(Ax) = \sum_{i=1}^n \rho_\tau((Ax)_i), \qquad (2)$$

where $\tau \in [0, 1]$ and $\rho_\tau : \mathbb{R} \to \mathbb{R}$ is defined as

$$\rho_\tau(t) = \begin{cases} t & t \geq 0 \\ -\tau t & t < 0 \end{cases}. \qquad (3)$$

The problem of preserving $h_\tau(\cdot)$ is equivalent to that of preserving $\rho_\tau(\cdot)$, since $h_\tau(t) = \tau \cdot \rho_{(1-\tau)/\tau}(t)$ for $\tau > 0$. Now we formally state our sampling result for the loss function $\rho_\tau(\cdot)$.

Theorem 1.1. *There is an algorithm that receives a matrix $A \in \mathbb{R}^{n \times d}$ and a parameter $\tau \in (0, 1]$, and outputs a matrix $\widetilde{A} \in \mathbb{R}^{N \times d}$ with $N = \widetilde{O}(d/(\varepsilon^2\tau^2))$ such that with probability at least $1 - 1/\mathrm{poly}(d)$, for all $x \in \mathbb{R}^d$,*

$$(1 - \varepsilon)\rho_\tau(Ax) \leq \rho_\tau(\widetilde{A}x) \leq (1 + \varepsilon)\rho_\tau(Ax).$$

The algorithm runs in $\widetilde{O}(\mathrm{nnz}(A) + d^\omega)$ time.[2]

Our assumption that $\tau \leq 1$ is without loss of generality, since $\rho_{1/\tau}(t) = \frac{1}{\tau}\rho_\tau(-t)$, and thus our results can be directly translated to the case where $\tau > 1$.

Compared with (Yang et al., 2013), the number of rows required by our row sampling algorithm has a nearly linear dependence on d, whereas the algorithm in (Yang et al., 2013) requires at least $O(d^3/(\varepsilon^2\tau))$ sampled rows, in our notation. Thus, our result is better by a factor of d^2 when τ is constant (which is usually the case for applications in practice). Furthermore, the running time of the row sampling algorithm in (Yang et al., 2013) is $\widetilde{O}(\mathrm{nnz}(A) + \mathrm{poly}(d))$, where the exponent hidden in the $\mathrm{poly}(d)$ term is much larger than 3 due to the use of the ellipsoid rounding routine in (Clarkson et al., 2016).

We then apply our sampling results to two problems: *quantile regression* and *balanced directed graph sparsification*.

Quantile Regression. For the quantile regression problem, notice that Theorem 1.1 already implies an algorithm that runs in input-sparsity time. Suppose that there is an algorithm that runs in time $\phi(s, d)$ for solving a quantile regression problem of size $s \times d$, then one can first use Theorem 1.1 to reduce the problem size from $n \times d$ to $\widetilde{O}(d/\varepsilon^2\tau^2) \times d$, and thus the overall time complexity of the algorithm would be $\widetilde{O}(\mathrm{nnz}(A) + d^\omega) + \phi\left(\widetilde{O}(d/\varepsilon^2\tau^2), d\right)$. As a comparison, the algorithm in (Yang et al., 2013) runs in $\widetilde{O}(\mathrm{nnz}(A) + \mathrm{poly}(d)) + \phi(\widetilde{O}(d^3/\varepsilon^2\tau), d)$ time (see Theorem 1 in (Yang et al., 2013)). Again, the exponent hidden in the $\mathrm{poly}(d)$ term is much larger than 3.

[2]We use ω to denote the matrix multiplication exponent. It is known that $\omega \leq 2.373$ (Williams, 2012).

To further demonstrate the flexibility and versatility of Theorem 1.1, we develop an algorithm for solving quantile regression using accelerated first-order methods. Here, our goal is to optimize the lower-order term that polynomially depends on d in the time complexity while keeping the high-order term to be $\widetilde{O}(\mathrm{nnz}(A))$. To achieve this goal, we adopt the celebrated sketch-and-solve paradigm (Woodruff, 2014), together with recently developed powerful tools for ℓ_1 regression (Durfee et al., 2018). As a result, we give the fastest known input-sparsity time algorithm for the quantile regression problem.

Theorem 1.2 (Informal version of Theorem 4.4)**.** *For a given $\tau \in (0, 1]$, the quantile regression problem can be solved up to a factor of $(1 + \varepsilon)$ in time $\widetilde{O}(\mathrm{nnz}(A) + d^{2.5}/(\varepsilon^2\tau^2))$.*

Balanced Directed Graph Sparsification. Another unique advantage of our sampling results over previous works is that, for constant ε and τ, the number of sampled rows required by our algorithm is roughly linear in the dimensionality d. When $\tau = 0$, Theorem 1.1 does not apply; nevertheless, a similar sampling scheme, when applied to the edge-vertex incidence matrix of a *balanced directed graph* (see Definition 5.1), yields a graph sparsifier with nearly linear size. Recall Eqn. (2) for loss functions applied to a vector.

Theorem 1.3 (Informal version of Corollary 5.2)**.** *For a given α-balanced directed graph $G = (V, E, w)$ with n vertices and m edges, there is an algorithm that outputs $G' = (V, E', w')$ with $E' \subseteq E$ and $|E'| = O(n(\alpha/\varepsilon)^2 \log(n\alpha/\varepsilon))$ such that with probability at least $1 - 1/\mathrm{poly}(n)$, for all $x \in \mathbb{R}^n$,*

$$(1 - \varepsilon)\rho_0(Bx) \le \rho_0(B'x) \le (1 + \varepsilon)\rho_0(Bx). \quad (4)$$

Here B is the edge-vertex incidence matrix of G and B' is the edge-vertex incidence matrix of G'.

We note that a sampling scheme gives meaningful sparsifiers for graphs of n vertices only if the number of sampled rows is $o(n^2)$, since the largest possible number of edges in a graph is $O(n^2)$. Hence the sampling scheme by Yang et al. (2013), for example, would sample $\Omega(n^3)$ rows and is not an option for graph sparsification. Our sparsification result yields sparsifiers of roughly the same size as the state-of-the-art sparsification algorithm for balanced graphs by Ikeda & Tanigawa (2018).[3] However, the sparsifier produced by Ikeda & Tanigawa (2018) is only guaranteed to be a cut sparsifier, meaning that (4) holds only for $x \in \{0, 1\}^n$ (see Remark 1), whereas for our sparsifier, (4) holds for all $x \in \mathbb{R}^n$, which is much stronger. Furthermore, the algorithm in (Ikeda & Tanigawa, 2018) is specially tailored for balanced graphs, based on a number of highly combinatorial subroutines; in contrast, our algorithm follows a general-purpose sampling framework and utilizes the properties of balanced graphs in a rather black-box manner. We believe our algorithm provides an alternative view of sparsification of (balanced) directed graphs, which also illustrates the power of our sampling scheme.

[3]The two results are not directly comparable: for example, our bound has better dependence on the number of vertices n, while their bound has better dependence on the balance parameter α.

In Section 7, we conduct an empirical evaluation of our row sampling algorithm for the quantile loss function to demonstrate the practicality of our approach.

Organization. This paper is organized as follows. In Section 2, we introduce necessary notations and definitions and also review known results regarding Lewis weights. In Section 3, we present our main row sampling results. In Section 4, we present our algorithm for solving quantile regression. In Section 5, we give our results on balanced graph sparsification. In Section 6, we generalize our row sampling results in Section 3 to a broader class of loss functions. In Section 7, we present our experimental results. We conclude and discuss future directions in Section 8. Most technical proofs are omitted and can be found in the supplementary material[4].

2. Preliminaries

Notations. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For a vector $x \in \mathbb{R}^n$, we use $\|x\|_p$ to denote its ℓ_p norm i.e., $\|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$. For a matrix $A \in \mathbb{R}^{n\times d}$, we use $A^\dagger$ to denote its Moore-Penrose inverse. For a matrix $A \in \mathbb{R}^{n\times d}$, we use A_i to denote its i-th row, viewed as a column vector. The *leverage score* of A_i is defined to be $\tau_i(A) = A_i^\top (A^\top A)^\dagger A_i$. For two vectors u and v, we use $\langle u, v\rangle$ to denote their inner product. For a function $\phi : \mathbb{R} \to \mathbb{R}$ and a vector $y \in \mathbb{R}^n$, we use $\phi(y)$ to denote $\sum_{i=1}^n \phi(y_i)$.

Graph Theory. We give some background on graph theory, which provides the context for our graph sparsification result. In this paper, we focus on directed graphs $G = (V, E, w)$ with edge set $E \subseteq V \times V$, and edge weights $w : E \to \mathbb{R}_+$. Throughout this paper, we assume that the directed graph G is strongly connected. For two sets of vertices S and T, we use $w_G(S, T)$ to denote the total weight of edges from S to T, i.e., $w_G(S, T) = \sum_{(u,v)\in(S\times T)\cap E} w(u, v)$. We also say $w_G(S, T)$ is the *capacity* of the *directed cut* from S to T. We use $w(S, T)$ to denote $w_G(S, T)$ when G is clear from the context. For a directed graph $G = (V, E, w)$ with $n = |V|$ vertices and $m = |E|$ edges, the *edge-vertex incidence matrix*

[4]A full version is also available at `arXiv:2006.08397 [cs.DS]`

$B \in \mathbb{R}^{m\times n}$ is defined to be

$$B_{e,u} = \begin{cases} w(e) & \text{if } e = (u,v) \\ -w(e) & \text{if } e = (v,u) \\ 0 & \text{otherwise} \end{cases}.$$

In words, each row of B corresponds to an edge $e = (u,v)$ in G, where the u-th entry is $w(e)$, v-th entry is $-w(e)$, and all other entries are 0.

Lewis Weights. We now briefly review the definition and some basic properties of *Lewis weights* (Cohen & Peng, 2015), which play a key role in our sampling results.

Definition 2.1 (ℓ_p Lewis weights (Cohen & Peng, 2015))**.** For a given matrix $A \in \mathbb{R}^{n\times d}$ and $p \geq 1$, we say $\{w_i\}_{i=1}^n$ are the ℓ_p *Lewis weights* of A if they satisfy $\tau_i(W^{1/2-1/p}A) = w_i$ for all $i \in [n]$, where $W = \operatorname{diag}(w_1,\ldots,w_n)$ is the $n\times n$ diagonal matrix whose i-th diagonal entry is w_i.

We need the following fact regarding lewis weights proved in (Cohen & Peng, 2015).

Lemma 2.1 ((Cohen & Peng, 2015))**.** *The ℓ_p Lewis weights $\{w_i\}_{i=1}^n$ always exist and are unique, and $\sum_{i=1}^n w_i \leq d$.*

The following theorem enables efficient (approximate) computation of Lewis weights, which we use as a subroutine in our sampling results.

Theorem 2.2 ((Cohen & Peng, 2015))**.** *There is an algorithm that receives a matrix $A \in \mathbb{R}^{n\times d}$ and $p \geq 1$, and outputs $\{\overline{w}_i\}_{i=1}^n$, such that with probability at least $1 - 1/\operatorname{poly}(d)$, for all $i \in [n]$, $w_i \leq \overline{w}_i \leq 2w_i$, where $\{w_i\}_{i=1}^n$ are the ℓ_p Lewis weights of A. The algorithm runs in $\widetilde{O}(\operatorname{nnz}(A)+d^{p/2+O(1)})$ time. When $p < 4$, the algorithm runs in $\widetilde{O}(\operatorname{nnz}(A)+d^{\omega})$ time.*

For graph sparsification specifically, we are interested in approximating Lewis weights for edge-vertex incidence matrices. The following theorem states that in such cases, the running time stated in Theorem 2.2 can be further improved.

Theorem 2.3. *When the matrix A is the edge-vertex incidence matrix of a graph G and $p < 4$, the algorithm in Theorem 2.2 runs in $\widetilde{O}(m)$ time, where m is the number of edges in G.*

3. Row Sampling Based on Lewis Weights

In this section, we present our row sampling results. Our results are built upon the Lewis weights sampling framework introduced in (Cohen & Peng, 2015). In particular, given a matrix $A \in \mathbb{R}^{n\times d}$, they show that if one samples $O(d\log(d/\varepsilon)/\varepsilon^2)$ rows of A based on the ℓ_1 Lewis weights (see Definition 2.1) and scales the rows properly, then one obtains a matrix $\widetilde{A}$ such that with high probability, for all $x \in \mathbb{R}^d$, $(1-\varepsilon)\|Ax\|_1 \leq \|\widetilde{A}x\|_1 \leq (1+\varepsilon)\|Ax\|_1$.

In the following theorem, we show that for any function $\phi(\cdot)$ of the form $\phi(x) = a|x| + bx$ for some $0 \leq b \leq a$, sampling according to ℓ_1 Lewis weights still approximately preserves the value of $\phi(Ax)$ for all $x \in \mathbb{R}^d$. Our result generalizes that of (Cohen & Peng, 2015), in the sense that if we set $a = 1$ and $b = 0$, we recover exactly their bound for the ℓ_1 norm. Later, based on Theorem 3.1, we shall give a row sampling algorithm for the quantile loss function, and also provide an algorithm for producing sparsifiers for balanced directed graphs.

Theorem 3.1. *For a given matrix $A \in \mathbb{R}^{n\times d}$, let $\{w_i\}_{i=1}^n$ be its ℓ_1 Lewis weights, and suppose the function $\phi(x) = a|x| + bx$ $(0 \leq b \leq a)$ satisfies that $\|Ax\|_1 \leq B\cdot\phi(Ax)$ for all $x \in \mathbb{R}^d$ for some $B > 0$. There exists an absolute constant C such that the following holds.*

For any values $p_1,\ldots,p_n \geq 0$ such that $\sum_{i\in[n]} p_i = N$ and $p_i \geq C(aB/\varepsilon)^2 w_i \log N$, if we generate a matrix $\widetilde{A}$ with N rows, where each row is chosen independently as A_i/p_i with probability p_i/N, then with probability at least $1 - 1/\operatorname{poly}(d)$, it holds for all $x \in \mathbb{R}^d$ simultaneously that

$$(1-\varepsilon)\phi(Ax) \leq \phi(\widetilde{A}x) \leq (1+\varepsilon)\phi(Ax).$$

Proof (sketch). We follow the framework in (Cohen & Peng, 2015). Suppose that $i_1, i_2, \ldots, i_N$ are the indices chosen by the sampling process. Our goal is to derive an upper bound on

$$F = \sup_{x:\phi(Ax)=1} \left|\phi(\widetilde{A}x) - 1\right| = \sup_{x:\phi(Ax)=1} \left|\sum_{k=1}^N \frac{\phi(\langle A_{i_k}, x\rangle)}{p_{i_k}} - 1\right|.$$

We shall show it holds for some ℓ that

$$M = \mathop{\mathbb{E}}_{i_1,\ldots,i_N} F^\ell \leq \frac{\varepsilon^\ell}{\operatorname{poly}(d)}, \tag{5}$$

and thus the result would follow from Markov's inequality. To prove the moment bound above, we first apply the symmetrization technique and obtain that

$$\begin{aligned} M &= \mathop{\mathbb{E}}_{i} \sup_{\phi(Ax)=1} \left|\sum_{k=1}^N \frac{\phi(\langle A_{i_k}, x\rangle)}{p_{i_k}} - 1\right|^\ell \\ &\leq 2^\ell \mathop{\mathbb{E}}_{i,\sigma}\left[\left(\sup_{\phi(Ax)=1}\left|\sum_{k=1}^N \sigma_k \frac{\phi(\langle A_{i_k}, x\rangle)}{p_{i_k}}\right|\right)^\ell\right], \end{aligned}$$

where $i = (i_1, i_2, \ldots, i_N)$, and $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N)$ is a Rademacher sequence. Using the comparison theorem of

Rademacher processes, we can show that

$$\begin{aligned}&\mathop{\mathbb{E}}_{\sigma}\left(\sup_{\phi(Ax)=1}\left|\sum_{k=1}^{N}\sigma_k\frac{\phi(\langle A_{i_k},x\rangle)}{p_{i_k}}\right|\right)^{\ell}\\ \le & 3a^{\ell}2^{\ell-1}\mathop{\mathbb{E}}_{\sigma}\sup_{\phi(Ax)=1}\left|\sum_{k=1}^{N}\sigma_k\frac{\langle A_{i_k},x\rangle}{p_{i_k}}\right|^{\ell}.\end{aligned}$$

The supremum on the right-hand side is similar to the Rademacher process considered in Cohen & Peng (2015). Analogously we can show that, conditioned on the event that for all $x\in\mathbb{R}^d$, $\|\widetilde{A}x\|_1\le C_1\|Ax\|_1$, which holds with high probability for some absolute constant $C_1>0$, the expectation can be upper bounded as

$$\begin{aligned}&\mathop{\mathbb{E}}_{\sigma}\sup_{\phi(Ax)=1}\left|\sum_{k=1}^{N}\sigma_k\frac{\langle A_{i_k},x\rangle}{p_{i_k}}\right|^{\ell}\\ \le\ & B^{\ell}\mathop{\mathbb{E}}_{\sigma}\sup_{\|Ax\|_1=1}\left|\sum_{k=1}^{N}\sigma_k\frac{\langle A_{i_k},x\rangle}{p_{i_k}}\right|^{\ell}\\ \le\ & B^{\ell}\frac{(\varepsilon/(C'aB))^{\ell}}{\mathrm{poly}(d)}\end{aligned}$$

for some $\ell=\Theta(\log(N+d^2))$, where C' is another absolute constant. Thus (5) holds if the expectation is conditioned on the event defined above. The failure probability of the event would be added to the overall failure probability, concluding the proof. □

Now Theorem 1.1 immediately follows from Theorem 3.1. See Section B in the supplementary material for formal proofs.

We have successfully applied the Lewis weights sampling framework to provide row sampling algorithms for the quantile loss function. In general, one may wonder if it is possible to apply Lewis weights sampling to more general loss functions. Cohen & Peng (2015) show that ℓ_p Lewis weights sampling works gracefully when the loss function is the ℓ_p norm. In Section 6, we show that when the loss function $\phi(x)$ can be approximated by $\|x\|_p^p$, one can preserve $\phi(Ax)$ approximately for all $x\in\mathbb{R}^d$, by sampling proportional to ℓ_p Lewis weights and using a weighted version of the loss function $\phi(\cdot)$.

4. Solving Quantile Regression

Based on the row sampling result in Theorem 1.1 and recent advances for ℓ_1 regression (Durfee et al., 2018), we obtain a faster algorithm for solving the quantile regression problem. The description of the algorithm is given in Algorithm 1, and we prove its running time and correctness in Theorem 4.4.

Algorithm 1 Algorithm for solving quantile regression

```
1: Obtain approximate $\ell_1$ Lewis weights for each row of $[A,b]$ using Theorem 2.2.
2: Sample $N=O\left(\frac{d}{\varepsilon^2\tau^2}\log\frac{n}{\varepsilon\tau}\right)$ rows of $[A,b]$ according to $\ell_1$ Lewis weights and obtain $[\widetilde{A},\widetilde{b}]\in\mathbb{R}^{N\times(d+1)}$.
3: Let $\widetilde{A}=Q\cdot R$ be the QR-decomposition, where $Q\in\mathbb{R}^{N\times d}$ and $R\in\mathbb{R}^{d\times d}$.
4: Run the Katyusha algorithm[5] with $x_0=(\widetilde{A}R^{-1})^T b$ to obtain $\overline{x}$ such that
   $$\rho_\tau\Big(\widetilde{A}R^{-1}\overline{x}-\widetilde{b}\Big)\le\Big(1+\frac{\varepsilon}{3}\Big)\min_{x\in\mathbb{R}^d}\rho_\tau\Big(\widetilde{A}R^{-1}x-\widetilde{b}\Big). \quad (6)$$
5: Return $x^*=R^{-1}\overline{x}$.
```

The following lemma, which is an easy corollary of Theorem 1.1, shows that if $\overline{x}$ satisfies the condition in (6), then it is indeed an approximate solution to $\min_{x\in\mathbb{R}^d}\rho_\tau(Ax-b)$.

Lemma 4.1. *For sufficiently small $\varepsilon>0$, if $\overline{x}$ satisfies the condition in* (6)*, with probability at least* 0.9*, the solution $x^*\in\mathbb{R}^d$ returned by Algorithm 1 satisfies*

$$\rho_\tau(Ax^*-b)\le(1+\varepsilon)\min_{x\in\mathbb{R}^d}\rho_\tau(Ax-b).$$

Thus, we focus on obtaining an approximate solution to $\min_{x\in\mathbb{R}^d}\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$ hereinafter. Our general strategy is to apply the accelerated SGD to $\min_{x\in\mathbb{R}^d}\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$. To this end, we formulate $\min_{x\in\mathbb{R}^d}\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$ as a finite-sum problem. The following lemma shows that after applying Lewis weights sampling, with constant probability, each summand in the finite-sum problem has a bounded Lipschitz constant (see Definition A.1 in the supplementary material for the definition), and $x_0=(\widetilde{A}R^{-1})^T b$ is close to the optimal solution.

Lemma 4.2. *Let $x_0=(\widetilde{A}R^{-1})^T b$ and $\widetilde{x^{\mathrm{opt}}}=\operatorname{argmin}_{x\in\mathbb{R}^d}\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$. With probability at least* 0.9*,*

(i) the function $f(x)=\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$ can be written as $f(x)=\frac{1}{N}\sum_{i=1}^{N}f_i(x)$, where each $f_i(\cdot)$ is $O(\sqrt{N}d)$-Lipschitz; and

(ii) it holds that

$$\|x_0-\widetilde{x^{\mathrm{opt}}}\|_2\le\sqrt{\frac{d}{N\tau^2}}\cdot\rho_\tau(\widetilde{A}R^{-1}\widetilde{x^{\mathrm{opt}}}-\widetilde{b})$$

[5]Katyusha is an accelerated stochastic gradient descent algorithm by Allen-Zhu (2017) for minimization of a convex function. Here the convex function is $x\mapsto\rho_\tau(\widetilde{A}R^{-1}x-\widetilde{b})$ and we use the variant in Corollary 3.7 of (Allen-Zhu, 2017). The justification of applying this variant of Katyusha is given by Lemma 4.2(i).

and

$$\rho_\tau(\widetilde{A}R^{-1}x_0 - \widetilde{b}) \le \sqrt{\frac{N}{\tau^2}} \cdot \rho_\tau(\widetilde{A}R^{-1}\widetilde{x^{\mathrm{opt}}} - \widetilde{b}).$$

Now we apply Katyusha, an accelerated SGD algorithm in due to Allen-Zhu (2017). The correctness and running time is summarized in the following lemma.

Lemma 4.3. *In Algorithm 1, Step 4 obtains $\overline{x}$ which satisfies that*

$$\rho_\tau(\widetilde{A}R^{-1}\overline{x} - \widetilde{b}) \le (1 + \varepsilon/3) \min_{x \in \mathbb{R}^d} \rho_\tau(\widetilde{A}R^{-1}x - \widetilde{b})$$

with probability at least 0.8*, and Step 4 runs in* $\widetilde{O}(d^{2.5}/(\tau^2\varepsilon^2))$ *time.*

Now we give the efficiency and correctness of Algorithm 1.

Theorem 4.4. *Algorithm 1 runs in time* $\widetilde{O}(\mathrm{nnz}(A) + d^{2.5}/(\varepsilon^2\tau^2))$ *and with probability at least* 0.7 *returns a vector* $x^* \in \mathbb{R}^d$ *such that*

$$\rho_\tau(Ax^* - b) \le (1 + \varepsilon) \cdot \min_{x \in \mathbb{R}^d} \rho_\tau(Ax - b).$$

We note that the failure probability of Algorithm 1 can be reduced to an arbitrarily small constant by independent repetitions and taking the best solution found among all repetitions.

5. Balanced Graph Sparsification

In this section, we demonstrate how our sampling results in Section 3 can be applied to graph sparsification. In particular, we prove that our sampling scheme can be used to find a sparsifier of *balanced* directed graphs, a notion defined in (Ene et al., 2016) as follows.

Definition 5.1. For a strongly connected directed graph $G = (V, E, w)$ and $\alpha \ge 1$, we say G is *α-balanced* if for every subset $S \subset V$, we have $w(S, V \setminus S) \le \alpha \cdot w(V \setminus S, S)$.

Recall that $w(S, T)$ is the capacity of the directed cut between S and T. Therefore, the preceding definition essentially says that a graph is balanced if the outgoing capacity from any set of vertices S is roughly the same (up to a factor of α) as the ingoing capacity to S. The parameter α measures how balanced the graph is. In particular, any undirected graph is 1-balanced.

Our plan is to apply our sampling results on the edge-vertex incidence matrix of the balanced graph. In order to do that, we need first to relate balanced graphs to the quantile loss function, and show that the incidence matrix indeed satisfies the conditions required by our sampling results. In particular, we shall need the following lemma regarding balanced directed graphs.

Lemma 5.1. *For a given strongly connected directed graph $G = (V, E, w)$ with n vertices and m edges, let $B \in \mathbb{R}^{m \times n}$ be the edge-vertex incidence matrix of G. If G is α-balanced, then it holds for all $x \in \mathbb{R}^n$ that $\|Bx\|_1 \le (1 + \alpha)\rho_0(Bx)$.*

Now we are ready to apply our sampling schemes to balanced graphs. As another application of Theorem 3.1, we give a randomized construction of a sparsifier of $O(n(\alpha/\varepsilon)^2 \log(\alpha n/\varepsilon))$ edges for α-balanced graphs. The algorithm is given in Algorithm 2, and we prove its correctness and running time in Corollary 5.2.

Algorithm 2 Algorithm for sparsifying balanced directed graphs.

```
1: Obtain approximate $\ell_1$ Lewis weights for each row of the edge-vertex incidence matrix $B$ of the given $\alpha$-balanced graph $G$ using Theorem 2.3.
2: Sample $m' = O(n(\alpha/\varepsilon)^2 \log(\alpha n/\varepsilon))$ rows of $B$ according to approximate $\ell_1$ Lewis weights and obtain $B' \in \mathbb{R}^{m' \times n}$.
3: Return the directed graph $G'$ that corresponds to the edge-vertex incidence matrix $B'$.
```

Corollary 5.2. *For a given α-balanced strongly connected directed graph $G = (V, E, w)$ with n vertices and m edges, let $B \in \mathbb{R}^{m \times n}$ be the edge-vertex incidence matrix of G. Algorithm 2 outputs $G' = (V, E', w')$ with $E' \subseteq E$ and $|E'| = m' = O(n(\alpha/\varepsilon)^2 \log(\alpha n/\varepsilon))$ in $\widetilde{O}(m)$ time. Let $B' \in \mathbb{R}^{m' \times n}$ be the edge-vertex incidence matrix of G'. With probability at least $1 - 1/\operatorname{poly}(n)$, for all $x \in \mathbb{R}^d$,*

$$(1 - \varepsilon)\rho_0(Bx) \le \rho_0(B'x) \le (1 + \varepsilon)\rho_0(Bx).$$

Remark 1. The sparsifier G' obtained in Corollary 5.2 is a *cut sparsifier*, meaning that it approximately preserves the capacities of all cuts of G. To see this, for any given $S \subseteq V$, let $x_S \in \{0, 1\}^n$ be the indicator vector of S, i.e., $(x_S)_i = 1$ when $i \in S$ and $(x_S)_i = 0$ otherwise, then $\rho_0(Bx) = w_G(S, V \setminus S)$. Similarly, $\rho_0(B'x) = w_{G'}(S, V \setminus S)$. Since for all $x \in \mathbb{R}^d$, $(1-\varepsilon)\rho_0(Bx) \le \rho_0(B'x) \le (1+\varepsilon)\rho_0(Bx)$, it follows that for all $S \subseteq V$, $(1 - \varepsilon)w_G(S, V \setminus S) \le w_{G'}(S, V \setminus S) \le (1 + \varepsilon)w_G(S, V \setminus S)$.

6. Row Sampling for Loss Functions Approximated by ℓ_p Norms

In Section 3, we have shown that ℓ_1 Lewis weights sampling approximately preserves the quantile loss function $\rho_\tau(Ax)$ for all $x \in \mathbb{R}^d$. A natural question is whether it is possible to preserve other loss functions using ℓ_p Lewis weights sampling. The following theorem shows that it is indeed possible. However, owing to the lack of homogeneity in the loss function, we have to use a weighted version of the loss function as the estimator. The proof is similar to that of Theorem 3.1 and is provided in Section C in the supplementary material.

Theorem 6.1. *Let $p \geq 1$. Suppose $\phi : \mathbb{R} \to \mathbb{R}$ satisfies $|\phi(x) - \phi(y)| \leq L\,||x|^p - |y|^p|$ for all $x, y \in \mathbb{R}$ for some $L > 0$ and $\phi(x) \geq \gamma|x|^p$ for all $x \in \mathbb{R}$ for some $\gamma > 0$.*

Let matrix $A \in \mathbb{R}^{n\times d}$ and $\{w_i\}_{i=1}^n$ be its ℓ_p Lewis weights. For any values $p_1, \ldots, p_n \geq 0$ such that $\sum_{i\in[n]} p_i = N$ and $p_i \geq f\left(p, d, \frac{\gamma}{L}\varepsilon, \delta\right) w_i$, the following holds.

Let $i_1, \ldots, i_N$ be i.i.d. random variables such that $\Pr[i_k = j] = p_j/N$ for all $k \in [N]$. Define $w_i = 1/p_{i_k}$. Construct $\widetilde{A} \in \mathbb{R}^{N\times d}$ so that $\widetilde{A}_k = A_{i_k}$ for each $k \in [N]$. With probability at least $1 - \delta$, it holds for all $x \in \mathbb{R}^d$ simultaneously that

$$(1-\varepsilon)\phi(Ax) \leq \phi^w(\widetilde{A}x) \leq (1+\varepsilon)\phi(Ax),$$

where $\phi^w(y) = \sum_{i=1}^N w_i\phi(y_i)$ for $y \in \mathbb{R}^N$.

Bounds on $f(p, d, \varepsilon, \delta)$ are given in the table below, where C_p is a constant depending only on p and C is an absolute constant.

p	δ	$f(p, d, \varepsilon, \delta)$
$p = 1$	$1/\operatorname{poly}(d)$	$C(1/\varepsilon^2)\log(d/\varepsilon)$
$1 < p < 2$	*constant*	$C(1/\varepsilon^2)\log(d/\varepsilon)\log^2\log(d/\varepsilon)$
$p > 2$	$1/\operatorname{poly}(d)$	$C_p(1/\varepsilon^2)d^{p/2}\log^2 d\log(d/\varepsilon)$

Therefore, to obtain a row sampling algorithm for loss functions that satisfy the conditions in Theorem 6.1, we first invoke the algorithm in Theorem 2.2 to obtain approximate ℓ_p Lewis weights of the input matrix A and then sample rows of A according to Theorem 6.1. The total number of sampled rows $N = \sum_{i=1}^n p_i = O(d \cdot f(p, d, \varepsilon, \delta))$ by Lemma 2.1.

For solving linear regression, $\phi(\cdot)$ is usually convex, in which case $\phi^w(\cdot)$ is also convex. Therefore, to obtain an approximate solution to the linear regression problem $\min_{x\in\mathbb{R}^d} \phi(Ax - b)$, one just needs to solve a much smaller instance $\min_{x\in\mathbb{R}^d} \phi^w(\widetilde{A}x - \widetilde{b})$, which can still be formulated as a convex program and thus can be efficiently solved.

We remark that by using the notion of ℓ_p leverage score and techniques in (Dasgupta et al., 2009), we may obtain a results similar to Theorem 6.1. However, the number of samples required by such an approach will be much larger than that of Theorem 6.1.

7. Empirical Evaluation of Row Sampling Algorithms

In this section, we conduct an empirical evaluation[6] of our sampling scheme against the three algorithms proposed by Yang et al. (2013), which are the current state of the art. The three algorithms, codenamed SPC1, SPC2 and SPC3, are all based on row sampling, but the weights are very different from Lewis weights. We also compare with uniform sampling as a baseline, which is widely used in practice for its simplicity. Throughout this section, we use all sampling algorithms only for preprocessing, and solve the reduced problem optimally using linear programming. This allows us to remove the possible bias introduced by different optimization methods and focus on the comparison of the sampling schemes.

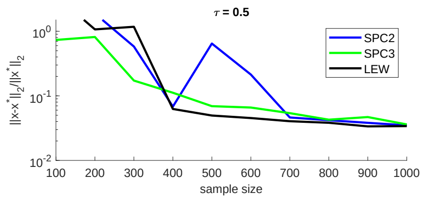
Synthetic Data. We first evaluate our sampling scheme on a synthetic dataset using the same setup as in (Yang et al., 2013). Here, we only compare with SPC2 and SPC3, which, as demonstrated in (Yang et al., 2013), significantly outperform all other sampling schemes on this dataset.

As described in (Yang et al., 2013), the data set is generated in the following manner. Each row of the matrix $A \in \mathbb{R}^{n\times d}$ is a canonical basis vector. Suppose the number of occurrences of e_j is c_j, where $c_j = qc_{j-1}$, for $j = 2, \ldots, d$ and some $q \in (1, 2]$ — such distributions of rows introduce imbalanced measurements and therefore increase the difficulty of sampling. The true vector $x^* \in \mathbb{R}^d$ has i.i.d. standard normal coordinates. Let $b^* = Ax^*$. The noise vector ν is generated with independent Laplacian entries, scaled such that $\|\nu\|_1/\|b^*\|_1 = 0.2$. The response vector b is given by $b_i = 500\nu_i$ with probability 0.001 and $b_i = b_i^* + \nu_i$ with probability 0.999. Here, we adhere to the same generation process so that our experimental results are comparable to those in (Yang et al., 2013).

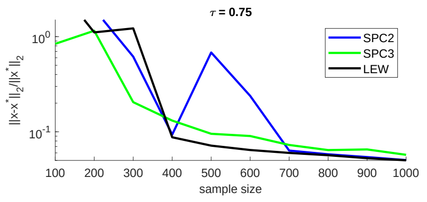
In a similar manner to (Yang et al., 2013), rather than determining the sample size from a given distortion parameter ε, we vary the quantile parameter $\tau \in \{0.5, 0.75, 0.95\}$, and for each τ, vary the sample size among $\{100, 200, \ldots, 1000\}$. Here the parameter τ corresponds to the parameter in the function $h_\tau(t)$ defined in Equation (1) rather than the renormalized function $\rho_\tau(t)$ defined in Equation (3). We choose $n = 10^5$ and $d = 50$.

See Figure 1 for results on the synthetic dataset and Figure 3 for the running time. When the same number of samples is at least 400, our sampling scheme (LEW) achieves the best approximation to x, outperforming both SPC2 and SPC3 in all three norms (ℓ_∞, ℓ_1 and ℓ_2) under the setting $\tau = 0.5$ and 0.75. When $\tau = 0.95$, our algorithm achieves

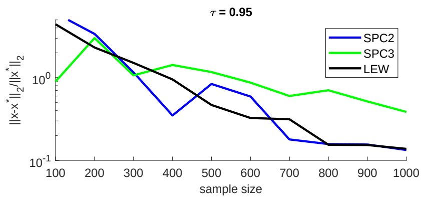
[6]All tests are run under MATLAB 2019b on a machine of Intel Core i7-6550U CPU@2.20GHz with 2 cores.



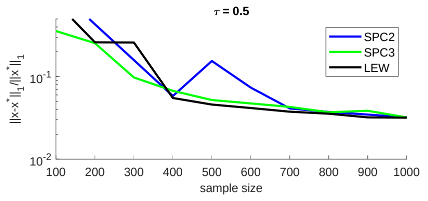
(a) $\tau = 0.5$, $\|x - x^*\|_2/\|x^*\|_2$



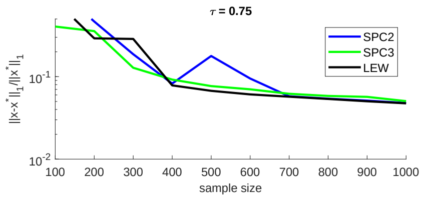
(b) $\tau = 0.75$, $\|x - x^*\|_2/\|x^*\|_2$



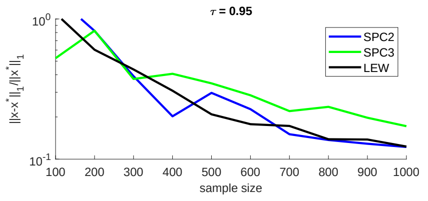
(c) $\tau = 0.95$, $\|x - x^*\|_2/\|x^*\|_2$



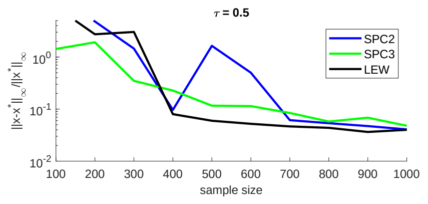
(d) $\tau = 0.5$, $\|x - x^*\|_1/\|x^*\|_1$



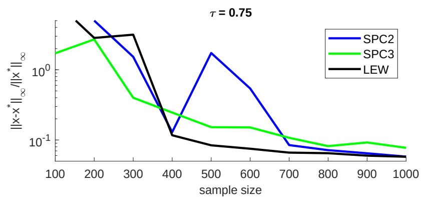
(e) $\tau = 0.75$, $\|x - x^*\|_1/\|x^*\|_1$



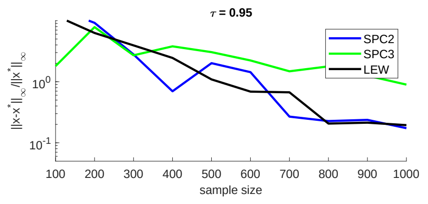
(f) $\tau = 0.95$, $\|x - x^*\|_1/\|x^*\|_1$



(g) $\tau = 0.5$, $\|x - x^*\|_\infty/\|x^*\|_\infty$

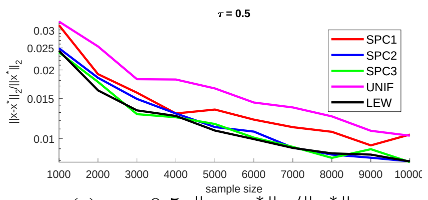


(h) $\tau = 0.75$, $\|x - x^*\|_\infty/\|x^*\|_\infty$

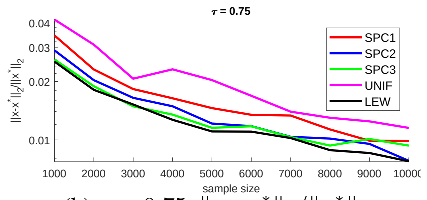


(i) $\tau = 0.95$, $\|x - x^*\|_\infty/\|x^*\|_\infty$

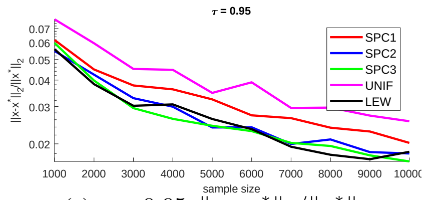
Figure 1: Mean of the relative errors of the solution vector (measured in three different norms, the ℓ_2 norm, the ℓ_1 norm and the ℓ_∞ norm) of 3 different methods on synthetic data. The plotted lines represent the mean of the errors of 50 independent trials.



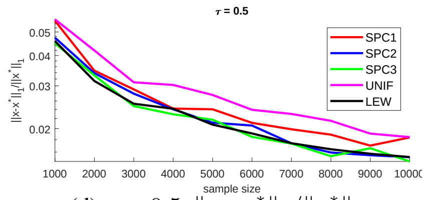
(a) $\tau = 0.5$, $\|x - x^*\|_2/\|x^*\|_2$



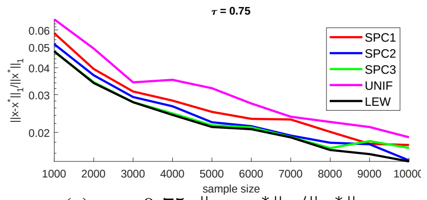
(b) $\tau = 0.75$, $\|x - x^*\|_2/\|x^*\|_2$



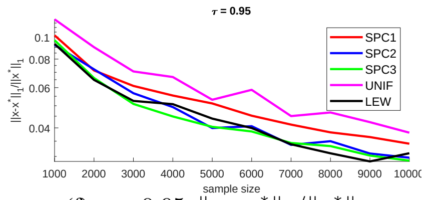
(c) $\tau = 0.95$, $\|x - x^*\|_2/\|x^*\|_2$



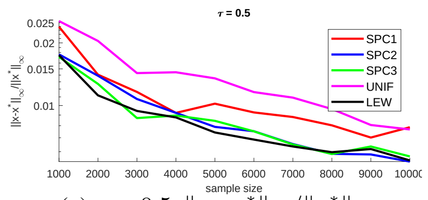
(d) $\tau = 0.5$, $\|x - x^*\|_1/\|x^*\|_1$



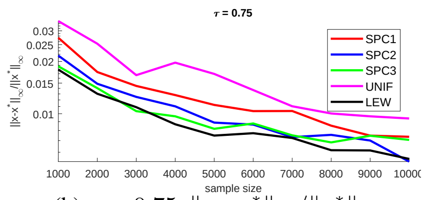
(e) $\tau = 0.75$, $\|x - x^*\|_1/\|x^*\|_1$



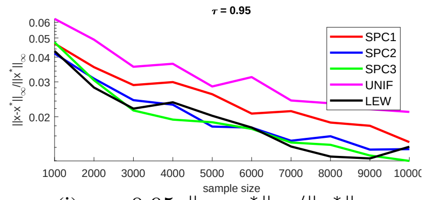
(f) $\tau = 0.95$, $\|x - x^*\|_1/\|x^*\|_1$



(g) $\tau = 0.5$, $\|x - x^*\|_\infty/\|x^*\|_\infty$



(h) $\tau = 0.75$, $\|x - x^*\|_\infty/\|x^*\|_\infty$



(i) $\tau = 0.95$, $\|x - x^*\|_\infty/\|x^*\|_\infty$

Figure 2: Mean of the relative errors of the solution vector (measured in three different norms, the ℓ_2 norm, the ℓ_1 norm and the ℓ_∞ norm) of 5 different methods on census data. The plotted lines represent the mean of the errors of 50 independent trials.

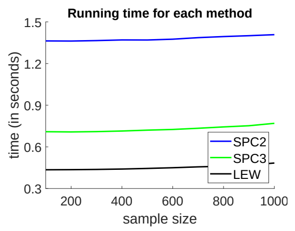


Figure 3: Mean running time of 50 independent trials on synthetic data.

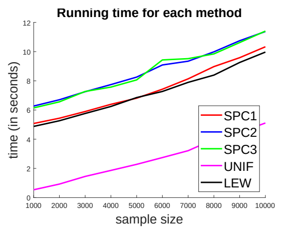


Figure 4: Mean running time of 50 independent trials on census data.

a similar approximation error to that of SPC2, which are both better than SPC3. This suggests that when the input is highly imbalanced and therefore more difficult to process, our method is accurate and stable with a faster running time.

Census Data. We also evaluate our algorithm on a real-world dataset, the U.S. 2000 Census data,[7] which consists of salary and related information on people who met certain criteria. As in (Yang et al., 2013), we conduct an experiment on the same 5% sample of the census data. In the corresponding regression problem, the matrix $A \in \mathbb{R}^{5048299\times 11}$ is the feature matrix and the vector $b \in \mathbb{R}^{5048299}$ records the salary of the sampled people. Our goal is to solve the regression problem $\min_x h_\tau(Ax-b)$, whose minimizer is denoted by x^*. We conduct a similar experiment as above with same values of τ and varying sample sizes in $\{1000, 2000, \ldots, 10000\}$, which are at most 0.2% of the data size.

See Figure 2 for approximation errors to x^* on the census data, where we consider all methods in (Yang et al., 2013). Our method, quite naturally, outperforms uniform sampling and SCP1 by a considerable margin, and achieves almost the same relative error as, or a marginally smaller relative error in certain cases than, SPC2 and SPC3. The running time of different methods are compared in Figure 4. Our method runs significantly faster than SPC2 and SPC3, suggesting that even on relatively easy-to-process real-world data, our method is robust with a superior running time.

8. Conclusion

In this paper, we give the first row sampling algorithm for the quantile loss function with sample complexity nearly linear in the dimensionality of the data. Technically, we show that Lewis weights sampling can be applied in row sampling algorithms for a variety of loss functions.

There are a few interesting problems that remain open. Is it possible to prove lower bounds on sample complexity of row sampling algorithms for the quantile loss function using techniques in (Li et al., 2020)? Furthermore, is it possible to use Lewis weights sampling to obtain nearly-linear sample complexity for other loss functions, e.g. the M-estimators studied in (Clarkson & Woodruff, 2015b;a)? Answering these problems would lead to a better understanding of the power and limitation of row sampling algorithms.

[7] http://www.census.gov/census2000/PUMS5.html

Acknowledgements

Y. Li was supported in part by a Singapore Ministry of Education (AcRF) Tier 2 grant MOE2018-T2-1-013. H. Zhang was supported by an National Science Foundation (NSF) grant IIS-1814056. The authors would like to thank Tianyi Zhang for insightful discussions.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Bourgain, J., Lindenstrauss, J., and Milman, V. Approximation of zonoids by zonotopes. *Acta mathematica*, 162(1): 73–141, 1989.

Chen, X., Liu, W., Zhang, Y., et al. Quantile regression under memory constraint. *The Annals of Statistics*, 47(6): 3244–3273, 2019.

Chowdhury, J., Chaudhuri, P., et al. Nonparametric depth and quantile regression for functional data. *Bernoulli*, 25 (1):395–423, 2019.

Clarkson, K. L. Subgradient and sampling algorithms for ℓ_1 regression. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 257–266. Society for Industrial and Applied Mathematics, 2005.

Clarkson, K. L. and Woodruff, D. P. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 81–90. ACM, 2013.

Clarkson, K. L. and Woodruff, D. P. Input sparsity and hardness for robust subspace approximation. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 310–329. IEEE, 2015a.

Clarkson, K. L. and Woodruff, D. P. Sketching for m-estimators: A unified approach to robust regression. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pp. 921–939. Society for Industrial and Applied Mathematics, 2015b.

Clarkson, K. L., Drineas, P., Magdon-Ismail, M., Mahoney, M. W., Meng, X., and Woodruff, D. P. The fast cauchy transform and faster robust linear regression. *SIAM Journal on Computing*, 45(3):763–810, 2016.

Clarkson, K. L., Wang, R., and Woodruff, D. P. Dimensionality reduction for tukey regression. In *Proceedings of the Thirty-sixth International Conference on Machine Learning*, 2019.

Cohen, M. B. and Peng, R. ℓ_p row sampling by lewis weights. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 183–192. ACM, 2015.

Cohen, M. B., Lee, Y. T., Musco, C., Musco, C., Peng, R., and Sidford, A. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pp. 181–190. ACM, 2015.

Cole, T. J. and Green, P. J. Smoothing reference centile curves: the lms method and penalized likelihood. *Statistics in medicine*, 11(10):1305–1319, 1992.

Dasgupta, A., Drineas, P., Harb, B., Kumar, R., and Mahoney, M. W. Sampling algorithms and coresets for ℓ_p regression. *SIAM Journal on Computing*, 38(5):2060–2078, 2009.

Davino, C., Furno, M., and Vistocco, D. *Quantile regression: theory and applications*, volume 988. John Wiley & Sons, 2013.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Drineas, P. and Mahoney, M. W. Effective resistances, statistical leverage, and applications to linear equation solving. *arXiv preprint arXiv:1005.3097*, 2010.

Durfee, D., Lai, K. A., and Sawlani, S. ℓ_1 regression using Lewis weights preconditioning and stochastic gradient descent. In *Conference On Learning Theory*, pp. 1626–1656, 2018.

Ene, A., Miller, G., Pachocki, J., and Sidford, A. Routing under balance. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pp. 598–611, New York, NY, USA, 2016. ACM. doi: 10.1145/2897518.2897654.

Ikeda, M. and Tanigawa, S.-i. Cut sparsifiers for balanced digraphs. In Epstein, L. and Erlebach, T. (eds.), *Approximation and Online Algorithms*, pp. 277–294, Cham, 2018. Springer International Publishing.

Koenker, R. and Bassett Jr, G. Regression quantiles. *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

Koenker, R. and Geling, O. Reappraising medfly longevity: a quantile regression survival analysis. *Journal of the American Statistical Association*, 96(454):458–468, 2001.

Koenker, R. and Hallock, K. F. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

Ledoux, M. and Talagrand, M. Comparison theorems, random geometry and some limit theorems for empirical processes. *Ann. Probab.*, 17(2):596–631, 04 1989.

Ledoux, M. and Talagrand, M. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 1991.

Li, M., Miller, G. L., and Peng, R. Iterative row sampling. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pp. 127–136. IEEE, 2013.

Li, Y., Wang, R., and Woodruff, D. P. Tight bounds for the subspace sketch problem with applications. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1655–1674. SIAM, 2020.

Meng, X. and Mahoney, M. W. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 91–100. ACM, 2013.

Nelson, J. and Nguyen, H. L. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pp. 117–126. IEEE, 2013.

Ota, H., Kato, K., Hara, S., et al. Quantile regression approach to conditional mode estimation. *Electronic Journal of Statistics*, 13(2):3120–3160, 2019.

Pandey, G. R. and Nguyen, V.-T.-V. A comparative study of regression based methods in regional flood frequency analysis. *Journal of Hydrology*, 225(1-2):92–101, 1999.

Royston, P. and Altman, D. G. Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 43(3):429–453, 1994.

Royston, P. and Wright, E. Goodness-of-fit statistics for age-specific reference intervals. *Statistics in medicine*, 19 (21):2943–2962, 2000.

Sohler, C. and Woodruff, D. P. Subspace embeddings for the l1-norm with applications. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 755–764. ACM, 2011.

Song, Z., Wang, R., Yang, L., Zhang, H., and Zhong, P. Efficient symmetric norm regression via linear sketching. In *Advances in Neural Information Processing Systems*, pp. 828–838, 2019.

Spielman, D. A. and Srivastava, N. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6): 1913–1926, 2011.

Talagrand, M. Embedding subspaces of L_1 into ℓ_1^n. *Proceedings of the American Mathematical Society*, 108(2): 363–369, 1990.

Talagrand, M. Embedding subspaces of L_p into ℓ_p^n. In *Geometric aspects of functional analysis*, pp. 311–326. Springer, 1995.

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

Wang, R. and Woodruff, D. P. Tight bounds for ℓ_p oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1825–1843. SIAM, 2019.

Williams, V. V. Multiplying matrices faster than coppersmith-winograd. In *STOC*, volume 12, pp. 887–898. Citeseer, 2012.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Woodruff, D. P. and Zhang, Q. Subspace embeddings and ℓ_p-regression using exponential random variables. In *Conference on Learning Theory*, pp. 546–567, 2013.

Yang, J., Meng, X., and Mahoney, M. Quantile regression for large-scale applications. In *International Conference on Machine Learning*, pp. 881–887, 2013.

Zhao, W., Lian, H., Liang, H., et al. Quantile regression for the single-index coefficient model. *Bernoulli*, 23(3): 1997–2027, 2017.

Zheng, Q., Peng, L., and He, X. High dimensional censored quantile regression. *Annals of statistics*, 46(1):308, 2018.