Online Active Regression

Cheng Chen [*1] **Yi Li** [*1] **Yiming Sun** [*1]

Abstract

Active regression considers a linear regression problem where the learner receives a large number of data points but can only observe a small number of labels. Since online algorithms can deal with incremental training data and take advantage of low computational cost, we consider an online extension of the active regression problem: the learner receives data points one by one and immediately decides whether it should collect the corresponding labels. The goal is to efficiently maintain the regression of received data points with a small budget of label queries. We propose novel algorithms for this problem under ℓ_p loss where $p \in [1, 2]$. To achieve a $(1+\epsilon)$-approximate solution, our proposed algorithms only require $\tilde{\mathcal{O}}(d/\operatorname{poly}(\epsilon) \cdot \log(n\kappa))$ queries of labels, where n is the number of data points and κ is a quantity, called the condition number, of the data points. The numerical results verify our theoretical results and show that our methods have comparable performance with offline active regression algorithms.

1. Introduction

Linear regression is a simple method to model the relationship between the data points in an Euclidean space and their scalar labels. A typical formulation is to solve the minimization problem $\min_x \|Ax - b\|_p$ for $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, where each row A_i is a data point in $\mathbb{R}^d$ and b_i is its corresponding scalar label. When $p = 2$, the linear regression is precisely the least-squares regression, which admits a closed-form solution and is thus a classical choice due to its computational simplicity. When $p \in [1, 2)$, it is more robust than least-squares as the solution is less sensitive to outliers. A popular choice is $p = 1$ because the regression can be cast as a linear programme though other values of p are recommended depending on the distribution of the noise in the labels. Interested readers may refer to Section 1.3 of (Gonin & Money, 1989) for some discussion.

One harder variant of linear regression is *active regression* (Sabato & Munos, 2014), in which the data points are easy to obtain but the labels are costly. Here one can query the label of any chosen data point and the task is to minimize the number of queries while still being able to solve the linear regression problem approximately. Specifically, one constructs an index set $S \subset [n]$ as small as possible, queries b_S (the restriction of b on S) and computes a solution $\tilde{x}$ based on A, S and b_S such that

$$\|A\tilde{x} - b\|_p^p \le (1+\epsilon) \min_x \|Ax - b\|_p^p. \tag{1}$$

For $p = 2$, the classical approach is to sample the rows of A according to the leverage scores. This can achieve (1) with large constant probability using $|S| = O(d \log d + d/\epsilon)$ queries. Chen & Price (2019) reduced the query complexity to the optimal $O(d/\epsilon)$, based on graph sparsifiers. When $p = 1$, Chen & Derezinski (2021) and Parulekar et al. (2021) showed that $O((d \log d)/\epsilon^2)$ queries suffices with large constant probability, based on sampling according to Lewis weights. More recently, Musco et al. (2021b) solved the problem for all values of p with query complexity $\tilde{O}(d/\epsilon)$ for $1 \le p < 2$ and $\tilde{O}(d^{p/2}/\epsilon^p)$ for $p > 2$, where the dependence on d is optimal up to logarithmic factors.

Another common setting of linear regression is the *online setting*, which considers memory restrictions that prohibit storing the inputs A and b in their entirety. In such a case, each pair of data points and their labels (i.e. each row of $[A\ b]$) arrives one by one, and the goal is to use as little space as possible to solve the linear regression problem. Again, the case of $p = 2$ has the richest research history, with the state-of-the-art results due to Cohen et al. (2020) and Jiang et al. (2022), which retain only $O(\epsilon^{-1} d \log d \log(\epsilon \|A\|_2^2))$ rows of A (where $\|A\|_2$ denotes the operator norm of A). The idea of the algorithms is to sample according to the online leverage scores, which was first employed in (Kapralov et al., 2017). The online leverage score of a row is simply the leverage score of the row in the submatrix of A consisting of all the revealed rows so far. The algorithm of Jiang et al. (2022) is based on that of Cohen et al. (2020) with further optimized runtime. The case of $p = 1$ was solved by (Braverman et al., 2020), who generalized the notion of

[*]Equal contribution [1]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Correspondence to: Yi Li <yili@ntu.edu.sg>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

online leverage score to online Lewis weights and sampled the rows of A according to the online Lewis weights.

In this paper, we consider the problem of *online active regression*, a combination of the two variants above. In a similar vein to (Cohen et al., 2020) and (Jiang et al., 2022), the rows of A arrive one by one, and upon receiving a row, one must decide whether it should be kept or discarded and whether to query the corresponding label, without ever retracting these decisions. The problem was considered by Riquelme et al. (2017), who assumed an underlying distribution of the data points together with a noise model of the labels and only considered ℓ_2-regression. Here we do not make such assumptions and need to handle arbitrary input data. To the best of our knowledge, our work is the first to consider the online active regression in the general ℓ_p-norm. Our approach is largely based upon the existing techniques for online regression and active regression. A technical contribution in our work is to show that one can compress a fraction of rows in a matrix by sampling these rows according to their Lewis weights while preserving the Lewis weights of the uncompressed rows (see Lemma 4.5 for the precise statement), which may be of independent interest.

Our Contributions. We show that the online active regression problem can be solved, attaining the error guarantee (1) with constant probability, using $m = \tilde{\mathcal{O}}(\epsilon^{-(2p+5)} d \log(n\kappa^{\mathrm{OL}}(A)))$ queries for $p \in [1,2)$ (where $\kappa^{\mathrm{OL}}(A)$ is the online condition number of A, see Definition 2) and $m = \tilde{\mathcal{O}}(\epsilon^{-9} d \log(n\|A\|_2/\sigma))$ queries for $p = 2$ (where $\|A\|_2$ is the operator norm of A and σ the smallest singular value of the first d rows of A). Our algorithms are sublinear in space complexity, using $m + \mathcal{O}(\epsilon^{-2} d \operatorname{poly}(\log n))$ words.

The query complexity for $p \in [1,2)$ depends on $\log n$ and $\log \kappa^{\mathrm{OL}}(A))$, which are not present in the offline counterpart (Musco et al., 2021b). But this is not unexpected, given that the $\log n \log \kappa^{\mathrm{OL}}(A)$ factor appears in the sketch size for the ℓ_1-subspace embedding under the sliding window model (Braverman et al., 2020).

We also demonstrate empirically the superior accuracy of our algorithm to online uniform sampling on both synthetic and real-world data. We vary the allotted number of queries and compare the relative error in the objective function of the regression (with respect to the minimum error, namely $\min_x \|Ax - b\|_p$). For active ℓ_1-regression, our algorithm achieves almost the same relative error as the offline active regression algorithm on both the synthetic and real-world data. For active ℓ_2-regression, our algorithm significantly outperforms the online uniform sampling algorithm on both synthetic and real-world data and is comparable with the offline active regression algorithm on the synthetic data.

2. Preliminaries

Notation. We use $[n]$ to denote the integer set $\{1, \ldots, n\}$. For a matrix A, we denote by $A^\dagger$ its Moore–Penrose inverse.

For two matrices A and B of the same number of columns, we denote by $A \circ B$ the vertical concatenation of A and B.

A matrix S is a called a sampling matrix if each row and each column has at most one nonzero entry. Associated with S are indicator variables $\{(\mathbb{1}_S)_i\}_{i=1,\ldots,n}$ (where n is the number of columns of S) defined as follows. For each i, we define $(\mathbb{1}_S)_i = 1$ if the i-th column of S is nonzero, and $(\mathbb{1}_S)_i = 0$ otherwise.

Suppose that $A \in \mathbb{R}^{n\times d}$. We define the operator norm of A, denoted by $\|A\|_2$, to be $\max_{\|x\|_2=1} \|Ax\|_2$. We also define an *online condition number* $\kappa^{\mathrm{OL}}(A) = \|A\|_2 \max_i \|(A^{(i)})^\dagger\|_2$, where $A^{(i)}$ is the submatrix consisting of the first i rows of A.

Suppose that $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^d$ and $p \geq 1$. We define $\textsc{Reg}(A, b, p)$ to be an $x \in \mathbb{R}^d$ that minimizes $\|Ax - b\|_p$. We remark that when $p > 1$, the minimizer is unique.

Lewis weights. A central technique to solve $\min_x \|Ax - b\|_p$ is to solve a compressed version $\min_x \|SAx - Sb\|_p$, where S is a sampling matrix. This sampling is based on Lewis weights (Cohen & Peng, 2015), which are defined below.

Definition 2.1 (Lewis weights)**.** Suppose that $A \in \mathbb{R}^{n\times d}$ and $p \geq 1$. The ℓ_p Lewis weights of A, denoted by $w_1(A), \ldots, w_n(A)$, are the unique real numbers such that $w_i(A) = (a_i^\top (A^\top W^{1-2/p} A)^{-1} a_i)^{p/2}$, where W is the diagonal matrix with diagonal elements $w_1(A), \ldots, w_n(A)$ and a_i is the i-th row of A.

For notational convenience, when A has n rows, we also write $w_n(A)$ as $w_{\mathrm{last}}(A)$. The ℓ_2 Lewis weight is also called the *leverage score*.

Definition 2.2. Given $p_1, \ldots, p_n \in [0,1]$, the *rescaled sampling matrix* S with respect to $p_1, \ldots, p_n$ is a random $n \times n$ diagonal matrix in which $S_{i,i} = p_i^{-1/p}$ with probability p_i and $S_{i,i} = 0$ with probability $1 - p_i$.

Lemma 2.3 (Lewis weights sampling (Cohen & Peng, 2015))**.** *Let $A \in \mathbb{R}^{n\times d}$. Choose $\beta = \Theta(\log(d/\delta)/\epsilon^2)$ and $p_1, \ldots, p_n$ such that $\min\{\beta w_i(A), 1\} \leq p_i \leq 1$. Let S be the rescaled sampling matrix with respect to $p_1, \ldots, p_n$. Then it holds with probability at least $1 - \delta$ that $(1-\epsilon)\|Ax\|_p \leq \|SAx\|_p \leq (1+\epsilon)\|Ax\|_p$ (i.e., S is an ϵ-subspace embedding for A in the ℓ_p-norm) and that the number of nonzero rows in S is $\mathcal{O}(\beta \sum_i w_i(A)) = \mathcal{O}(\beta d)$.*

In the light of the preceding lemma, one can choose an ϵ-subspace embedding matrix S for $[A\ b]$ and retain only the nonzero rows of S so that S has only $\tilde{O}(d/\epsilon^2)$ rows

and $\min_x \|SAx - Sb\|_p = (1 \pm \epsilon) \min \|Ax - b\|_p$. The remaining question is how to compute the Lewis weights of a given matrix. Cohen & Peng (2015) showed that, for a given matrix $A \in \mathbb{R}^{n \times d}$, the following iterations

$$W_{i,i}^{(j)} \leftarrow \left(a_i^\top \left(A^\top (W^{(j-1)})^{1-2/p} A \right)^{-1} a_i \right)^{p/2}, \quad (2)$$

with the initial point $W^{(0)} = I_n$, will converge to some diagonal matrix W, whose diagonal elements are exactly $w_1(A), \ldots, w_n(A)$.

Definition 2.4 (Online Lewis Weights)**.** Let $p \in [1, 2)$ and $A \in \mathbb{R}^{n \times d}$. The online ℓ_p Lewis weights, denoted by $w_1^{\text{OL}}(A), \ldots, w_n^{\text{OL}}(A)$, are defined to be $w_i^{\text{OL}}(A) = w_i(A^{(i)})$, where $A^{(i)}$ is the submatrix consisting of the first i rows of A.

We shall need the Johnson-Lindenstrauss matrix and an assumption on the input matrix A for the online active ℓ_2-regression.

Definition 2.5 (Johnson-Lindenstrauss Matrix)**.** Let $X \subseteq \mathbb{R}^d$ be a point set. A matrix J is said to be a Johnson-Lindenstrauss matrix for X of distortion parameter ϵ (or, an ϵ-JL matrix for X) if $(1-\epsilon)\|x\|_2^2 \le \|Jx\|_2^2 \le (1+\epsilon)\|x\|_2^2$ for all $x \in X$.

It is a classical result (Kane & Nelson, 2014) that when $|X| = T$, there exists a random matrix $J \in \mathbb{R}^{m \times d}$ with $m = \mathcal{O}(\epsilon^{-2} \log(T/\delta))$ such that (i) J is an ϵ-JL matrix for T with probability at least $1 - \delta$, (ii) each column of J contains $\mathcal{O}(\epsilon^{-1} \log(T/\delta))$ nonzero entries and (iii) J can be generated using $\mathcal{O}(\log^2(|T|/\delta) \log d)$ bits.

3. Algorithms and Main Results

The high-level approach follows (Musco et al., 2021a) and we give a brief review below. We sample A twice but with different sampling parameters β, getting $\tilde{A}$ of $\mathcal{O}(d \log d)$ rows and $\tilde{A}_1$ of $\mathcal{O}(d^2 \operatorname{poly}(\epsilon^{-1} \log d))$ rows, respectively. We use $\tilde{A}$ to solve $\min_{x \in \mathbb{R}^d} \|Ax - b\|_p$, obtaining a constant-factor approximation solution x_c. The problem is then reduced to solving $\min_{x \in \mathbb{R}^d} \|Ax - z\|_p$ with $z = b - Ax_c$, for which we shall solve $\min_{x \in \mathbb{R}^d} \|\tilde{A}_1 - \tilde{z}_1\|_p$ instead. Since $\tilde{A}_1$ has $\Omega(d^2)$ rows, we repeat the idea above and further subsample $\tilde{A}_1$ twice with different sampling parameters, getting $\tilde{A}_2$ of $\mathcal{O}(d \log d)$ rows and $\tilde{A}_3$ of $\mathcal{O}(d \operatorname{poly}(\epsilon^{-1} \log d))$ rows. The sampled matrix $\tilde{A}_2$ is used to obtain a constant-factor approximation solution $\hat{x}_c$ to $\min_{x \in \mathbb{R}^d} \|\tilde{A}_1 x - \tilde{z}_1\|_p$ and $\tilde{A}_3$ is used to solve $\min_{x \in \mathbb{R}^d} \|\tilde{A}_1 x - (\tilde{z}_1 - \tilde{A}_1 x_c')\|_p$ with a near-optimal solution $\bar{x}'$. The near-optimal solution to $\min_{x \in \mathbb{R}^d} \|\tilde{A}_1 x - \tilde{z}_1\|_p$ is then $\bar{x} = \hat{x}_c + \bar{x}'$. Finally, the solution to the original problem is $\tilde{x} = x_c + \bar{x}$. Note that in the algorithms, we use $\tilde{A}$ to denote the nonzero rows of SA where S is the rescaled sampling matrix. Hence, the sampled matrices $\tilde{A}$, $\tilde{A}_1$, $\tilde{A}_2$ and $\tilde{A}_3$ are SA, $S_1 A$, $S_2 S_1 A$ and $S_3 S_1 A$ respectively.

Algorithm 1 Online Active Regression for $p \in (1, 2)$

Initialize: Let $\tilde{A}^{(d)}, \tilde{A}_1^{(d)}, \tilde{A}_2^{(d)}, \tilde{A}_3^{(d)}$ be the first d rows of A and $\tilde{b}^{(d)}$ be the first d rows of b.

$\beta \leftarrow \Theta(\log d)$
$\beta_1 \leftarrow \Theta(d/\epsilon^{2+p})$
$\beta_2 \leftarrow \Theta(\log d)$
$\beta_3 \leftarrow \Theta(\log^2 d \log(d/\epsilon)/\epsilon^{2p+5})$
Retain the first d rows of A
while there is an additional row a_t **do**
  $\tilde{w}_t \leftarrow w_t(A^{(t)})$
  $p_t \leftarrow \min\{\beta \tilde{w}^{(t)}, 1\}$
  $(\tilde{A}^{(t)}, \tilde{b}^{(t)}) \leftarrow \textsc{Sample}(a_t, p_t, \tilde{A}^{(t-1)}, \tilde{b}^{(t-1)}, p)$
  $\tilde{w}_{1,t} \leftarrow w_t(A^{(t)})$
  $p_{1,t} \leftarrow \min\{\beta_1 \tilde{w}_{1,t}, 1\}$
  Sample a_t with probability $p_{1,t}$
  if a_t is sampled **then**
    $\tilde{A}_1^{(t)} \leftarrow \tilde{A}_1^{(t-1)} \circ a_t^\top p_{1,t}^{-\frac{1}{p}}$
    $\tilde{w}_{2,t} \leftarrow w_{\text{last}}(\tilde{A}_1^{(t)})$
    $p_{2,t} \leftarrow \min\{\beta_2 \tilde{w}_{2,t}, 1\}$
    $(\tilde{A}_2^{(t)}, \tilde{b}_2^{(t)}) \leftarrow \textsc{Sample}(a_t p_{1,t}^{-\frac{1}{p}}, p_{2,t}, \tilde{A}_2^{(t-1)}, \tilde{b}_2^{(t-1)}, p)$
    $\tilde{w}_{3,t} \leftarrow w_{\text{last}}(\tilde{A}_1^{(t)})$
    $p_{3,t} \leftarrow \min\{\beta_3 \tilde{w}_{3,t}, 1\}$
    $(\tilde{A}_3^{(t)}, \tilde{b}_3^{(t)}) \leftarrow \textsc{Sample}(a_t p_{1,t}^{-\frac{1}{p}}, p_{3,t}, \tilde{A}_3^{(t-1)}, \tilde{b}_3^{(t-1)}, p)$
  end if
end while
$x_c \leftarrow \textsc{Reg}(\tilde{A}, \tilde{b}, p)$
$\tilde{z}_2 \leftarrow \tilde{b}_2 - \tilde{A}_2 x_c$
$\hat{x}_c \leftarrow \textsc{Reg}(\tilde{A}_2, \tilde{z}_2, p)$
$\tilde{z}_3 \leftarrow \tilde{b}_3 - \tilde{A}_3 x_c$
$\bar{x}' \leftarrow \textsc{Reg}(\tilde{A}_3, \tilde{z}_3 - \tilde{A}_3 \hat{x}_c, p)$
$\bar{x} \leftarrow \hat{x}_c + \bar{x}'$
$\tilde{x} \leftarrow x_c + \bar{x}$
return $\tilde{x}$

Algorithm 2 $\textsc{Sample}(a_t, p_t, \tilde{A}^{(t-1)}, \tilde{b}^{(t-1)}, p)$

Sample a_t with probability p_t
if a_t is sampled **then**
  Query b_t
  $(\tilde{A}^{(t)}, \tilde{b}^{(t)}) \leftarrow (\tilde{A}^{(t-1)} \circ a_t^\top p_t^{-\frac{1}{p}}, \tilde{b}^{(t-1)} \circ b_t p_t^{-\frac{1}{p}})$
else
  $(\tilde{A}^{(t)}, \tilde{b}^{(t)}) \leftarrow (\tilde{A}^{(t-1)}, \tilde{b}^{(t-1)})$
end if

3.1. The case $p \in [1, 2)$

We present our main algorithm for $p \in (1, 2]$ in Algorithm 1. The following is the guarantee of the algorithm.

Theorem 3.1. *Let $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$. Algorithm 1 outputs a solution $\tilde{x}$ which satisfies that*

$$\|A\tilde{x} - b\|_p \le \min_{x\in\mathbb{R}^d} \|Ax - b\|_p \tag{3}$$

with probability at least 0.94 and makes $\mathcal{O}\left(\frac{d\log^2 d}{\epsilon^{2p+5}}\log\frac{d}{\epsilon}\cdot\log\frac{d\log n\log\kappa^{OL}(A)}{\epsilon}\cdot\log(n\kappa^{OL}(A))\right)$ queries overall in total.

A major drawback of Algorithm 1 is the cost of calculating the online Lewis weights. Recall that the online Lewis weight of a_t is defined with respect to the first t rows of A. A naïve implementation would require storing the entire matrix A, partly defying the purpose of an online algorithm. Furthermore, the iterative procedure described after Definition 2.1 takes $\mathcal{O}(\log t)$ iterations to reach a constant-factor approximation to the Lewis weights (Cohen & Peng, 2015), where each iteration takes $\mathcal{O}(td^2 + d^3)$ time, which would become intolerable as t becomes large. To address this issue, we adopt the compression idea in (Braverman et al., 2020), which maintains $\mathcal{O}(\log n)$ rescaled row-sampled submatrices of A, each having a small number of rows. The 'compression' algorithm is presented in Algorithm 3.

Algorithm 3 Compression algorithm for calculation of online Lewis weights

Initialize: B_0 contains the first d rows of A; $B_1, \dots, B_{\log n}$ are empty matrices; $Q = \Theta(\epsilon^{-2}d\log^3 n)$.

1: $\beta \leftarrow \Theta(\epsilon^{-2}d\log^3 n)$
2: **while** there is an additional row a_t **do**
3: $\quad B_0 \leftarrow B_0 \circ a_t$
4: $\quad$ **if** the size of B_0 exceeds Q **then**
5: $\quad\quad j \leftarrow$ the smallest index i such that B_i is empty
6: $\quad\quad M \leftarrow B_{i-1} \circ B_{i-2} \circ \cdots \circ B_0$
7: $\quad\quad p_i \leftarrow \min\{\beta w_i(M), 1\}$ for all i
8: $\quad\quad S \leftarrow$ rescaled sampling matrix with respect to probabilities $\{p_i\}_i$
9: $\quad\quad B_i \leftarrow SM$
10: $\quad\quad B_0, B_1, \dots, B_{i-1} \leftarrow$ empty matrix
11: $\quad$ **end if**
12: **end while**

With the compression algorithm for A which maintains $B_0, \dots, B_{\log n}$, we can replace Line 7 of Algorithm 1 with

$$\tilde{w}_t \leftarrow w_{\text{last}}(B_{\log n} \circ B_{\log n-1} \circ \cdots \circ B_0). \tag{4}$$

Similarly, we run an additional compression algorithms for each of $\tilde{A}_1$ and replace Lines 10, 15 and 18 with updates analogous to (4). In addition, we change the value of β and β_1 to

$$\beta = \Theta(\epsilon^{-2}\log d\log^2 n) \text{ and } \beta_1 = \Theta(\epsilon^{-4}\log d\log^4 n),$$

respectively.

By the construction of the blocks B_i, each B_i contains at most $R = \mathcal{O}(\epsilon^{-2}d\log^3 n)$ rows with probability at least $1 - 1/\operatorname{poly}(n)$, sufficient for taking a union bound over all the blocks throughout the process of reading all n rows of A. Hence we may assume that each block B_i always contains at most R rows. Now, $\tilde{w}_t$ is calculated to be the Lewis weight of a matrix of $R' = \mathcal{O}(Q + R\log n) = \mathcal{O}(R\log n)$ rows, which can be done in $\mathcal{O}((R'd^2 + d^3)\log R') = \mathcal{O}(\epsilon^{-2}d^3\operatorname{poly}(\log(n/\epsilon)))$ time for a constant-factor approximation, where the dependence on n is only polylogarithmic. The remaining question is correctness and the following theorem is the key to proving the correctness.

Theorem 3.2. *Let $A \in \mathbb{R}^{n\times d}$. With Algorithm 3 maintaining $B_0, \dots, B_{\log n}$, let $\tilde{w}_t$ be as in (4) for each $t \le n$. Then it holds with probability at least $1 - 1/\operatorname{poly}(n)$ that*

$$(1-\epsilon)w_t(A^{(t)}) \le \tilde{w}_t \le (1+\epsilon)w_t(A^{(t)}), \quad \forall t \le n,$$

where $A^{(t)}$ be the submatrix consisting of the first t rows of A. The weights $\tilde{w}_t$ can be calculated in $\mathcal{O}(\epsilon^{-2}d^3\operatorname{poly}(\log(n/\epsilon)))$ time and Algorithm 3 needs $\mathcal{O}(\epsilon^{-2}d\operatorname{poly}(\log n))$ words of space overall in total.

The proof of Theorem Theorem 3.2 is deferred to Section 4.2. Now we can strengthen Theorem 3.1 as follows.

Theorem 3.3. *Let $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$. When implemented using the compression technique as explained above, Algorithm 1 outputs a solution $\tilde{x}$ which satisfies (3) with probability at least $0.94 - o(1)$, making $m = \mathcal{O}\left(\frac{d\log^2 d}{\epsilon^{2p+5}}\log\frac{d}{\epsilon}\cdot\log\frac{d\log n\log\kappa^{OL}(A)}{\epsilon}\cdot\log(n\kappa^{OL}(A))\right)$ queries. Furthermore, it uses $m + \mathcal{O}(\frac{d}{\epsilon^2}\operatorname{poly}(\log n))$ words of space overall in total.*

3.2. The case $p = 2$

In this subsection, we assume that the first d rows of the input matrix A is not singular.

Assumption 3.4. The minimal singular value of the first d rows of A is $\sigma > 0$.

As mentioned in the preceding subsection, it is computationally expensive to compute Lewis weights in general. A special case is $p = 2$, where the Lewis weights are leverage scores and thus much easier to compute. In this case, $w_i(A) = a_i^\top(A^\top A)^{-1}a_i$, and correspondingly, the online Lewis weights become online leverage scores, which are $w_i^{\text{OL}}(A) = a_i^\top((A^{(i)})^\top A^{(i)})^{-1}a_i$. It is much easier to compute $w_i^{\text{OL}}(A)$ in the online setting because one can simply maintain $(A^{(i)})^\top A^{(i)}$ by adding $a_i a_i^\top$ when reading a new

Algorithm 4 Online Active Regression for $p = 2$

Initialize: Let $\tilde{A}^{(d)}, \tilde{A}_1^{(d)}, \tilde{A}_2^{(d)}, \tilde{A}_3^{(d)}$ be the first d rows of A and $\tilde{b}^{(d)}, \tilde{b}_2^{(d)}, \tilde{b}_3^{(d)}$ be the first d rows of b. Let $x_c^{(d)} = \textsc{Reg}(\tilde{A}^{(d)}, \tilde{b}^{(d)}, 2)$, $\tilde{z}_2^{(d)} = \tilde{z}_3^{(d)} = \tilde{b}^{(d)} - \tilde{A}^{(d)} x_c^{(d)}$, $\hat{x}_c^{(d)} = \textsc{Reg}(\tilde{A}_2^{(d)}, \tilde{z}_2^{(d)}, 2)$ and $\bar{x}'_d = \textsc{Reg}(\tilde{A}_3^{(d)}, \tilde{z}_3^{(d)} - \tilde{A}_3^{(d)} \hat{x}_c^{(d)}, 2)$. Let $\mathring{G}^{(d)} = ((\tilde{A}^{(d)})^\top \tilde{A}^{(d)})^{-1}$ and $H^{(d)} = \tilde{A}^{(d)} \mathring{G}^{(d)}$. Also let $\mathring{G}_i^{(d)} = ((\tilde{A}_i^{(d)})^\top \tilde{A}_i^{(d)})^{-1}$ and $H_i^{(d)} = \tilde{A}_i^{(d)} \mathring{G}_i^{(d)}$ for $i = 1, 2, 3$.

$\beta \leftarrow \Theta(\log d)$
$\beta_1 \leftarrow \Theta(d/\epsilon^4)$
$\beta_2 \leftarrow \Theta(\log d)$
$\beta_3 \leftarrow \Theta((\log^2 d)\log(d/\epsilon)/\epsilon^9)$
retain the first d rows of A
while there is an additional row a_t **do**
  $\tilde{w}_t \leftarrow \|H^{(t-1)} a_t\|_2^2$
  $(x_c^{(t)}, \tilde{A}^{(t)}, \tilde{b}^{(t)}, \mathring{G}^{(t)}, H^{(t)}) \leftarrow$ $\textsc{SampleQuery}(a_t, \tilde{b}^{(t-1)}, \perp, \perp, \tilde{A}^{(t-1)}, \beta, \tilde{w}_t, \mathring{G}^{(t-1)}, 1)$
  $\tilde{w}_{1,t} \leftarrow \|H_1^{(t)} a_t\|_2^2$
  Sample a_t with pr. $p_{1,t} = \min\{\beta_1 \tilde{w}_{1,t}, 1\}$
  if a_t is sampled **then**
    $\tilde{A}_1^{(t)} \leftarrow \tilde{A}_1^{(t-1)} \circ \frac{a_t^\top}{\sqrt{p_{1,t}}}$
    $(\mathring{G}_1^{(t)}, H_1^{(t)}) \leftarrow \textsc{Update}(\frac{a_t}{\sqrt{p_{1,t}}}, \perp, \perp, \perp, \tilde{A}_1^{(t-1)}, \mathring{G}_1^{(t-1)})$
    $\tilde{w}_{2,t} \leftarrow \|H_2^{(t)} \frac{a_t}{\sqrt{p_{1,t}}}\|_2^2$
    $(\hat{x}_c^{(t)}, \tilde{A}_2^{(t)}, \tilde{b}_2^{(t)}, \mathring{G}_2^{(t)}, H_2^{(t)}) \leftarrow$ $\textsc{SampleQuery}(\frac{a_t}{\sqrt{p_{1,t}}}, \tilde{b}_2^{(t-1)}, x_c^{(t)}, \perp, \tilde{A}_2^{(t-1)}, \beta_2, \tilde{w}_{2,t}, \mathring{G}_2^{(t-1)}, 2)$
    $\tilde{w}_{3,t} = \|H_3^{(t)} \frac{a_t}{\sqrt{p_{1,t}}}\|_2^2$
    $(\bar{x}'^{(t)}, \tilde{A}_3^{(t)}, \tilde{b}_3^{(t)}, \mathring{G}_3^{(t)}, H_3^{(t)}) \leftarrow$ $\textsc{SampleQuery}(\frac{a_t}{\sqrt{p_{1,t}}}, \tilde{b}_3^{(t-1)}, x_c^{(t)}, \hat{x}_c^{(t)}, \tilde{A}_3^{(t-1)}, \beta_3, \tilde{w}_{3,t}, \mathring{G}_3^{(t-1)}, 3)$
  end if
  $\bar{x}^{(t)} \leftarrow \hat{x}_c^{(t)} + \bar{x}'^{(t)}$
  $\tilde{x}^{(t)} \leftarrow \bar{x}^{(t)} + x_c^{(t)}$
end while
return $\tilde{x}^{(t)}$

row a_i (viewed as a column vector). A naïve implementation of this algorithm would require inverting a $d \times d$ matrix at each step and we can further optimize the running time by noticing that $((A^{(i)})^\top A^{(i)})^{-1}$ receives a rank-one update at each step. This is the approach taken by (Cohen et al., 2020) and (Jiang et al., 2022) for computing the online leverage scores in the online setting. Adopting this approach, we present our fast algorithm for $p = 2$ in Algorithm 4 and its guarantee below.

Algorithm 5 $\textsc{SampleQuery}(a_t, \tilde{b}^{(t-1)}, x_c^{(t)}, \hat{x}_c^{(t)}, \tilde{A}^{(t-1)}, \beta, \tilde{w}_t, \mathring{G}^{(t-1)}, \chi)$ in Algorithm 4

$p_t \leftarrow \min\{\beta(1+\epsilon)^2 \tilde{w}_t, 1\}$
Sample a_t with probability p_t
if a_t is sampled **then**
  $\tilde{A}^{(t)} \leftarrow \tilde{A}^{(t-1)} \circ \frac{a_t^\top}{\sqrt{p_t}}$
  Query b_t
  if $\chi = 1$ **then**
    $\tilde{b}^{(t)} \leftarrow \tilde{b}^{(t-1)} \circ \frac{b_t}{\sqrt{p_t}}$
  else
    $b^{(t)} \leftarrow b^{(t-1)} \circ \frac{b_t}{\sqrt{p_{1,t} p_t}}$
    $z^{(t)} \leftarrow b^{(t)} - \tilde{A}^{(t)} x_c^{(t)}$
  end if
  $(x^{(t)}, \mathring{G}^{(t)}, H^{(t)}) \leftarrow \textsc{Update}(a_t, \tilde{b}^{(t)}, \hat{x}_c^{(t)}, \tilde{A}^{(t)}, \mathring{G}^{(t-1)})$
else
  $(\tilde{A}^{(t)}, \tilde{b}^{(t)}) \leftarrow (\tilde{A}^{(t-1)}, \tilde{b}^{(t-1)})$
  $(x^{(t)}, \mathring{G}^{(t)}, H^{(t)}) \leftarrow (x^{(t-1)}, \mathring{G}^{(t-1)}, H^{(t-1)})$
end if
return $(x^{(t)}, \tilde{A}^{(t)}, \tilde{b}^{(t)}, \mathring{G}^{(t)}, H^{(t)})$

Algorithm 6 $\textsc{Update}(a_t, \tilde{b}^{(t)}, \hat{x}_c^{(t)}, \tilde{A}^{(t)}, \mathring{G}^{(t-1)})$

$g \leftarrow a_t^\top \mathring{G}^{(t-1)} a_t / p_t$
$\mathring{G}^{(t)} \leftarrow \mathring{G}^{(t-1)} - \frac{1}{1+g} \mathring{G}^{(t-1)} \frac{a_t a_t^\top}{p_t} \mathring{G}^{(t-1)}$
$s_t \leftarrow$ the number of rows in $\tilde{A}^{(t)}$
Update the ϵ-JL matrix $J^{(t+1)}$ of size $\frac{\log n}{\epsilon^2} \times s_t$
$F^{(t)} \leftarrow J^{(t+1)} \tilde{A}^{(t)}$
$H^{(t)} \leftarrow F^{(t)} \mathring{G}^{(t)}$
if $\tilde{b}^{(t)} = \perp$ **then**
  return $(\mathring{G}^{(t)}, H^{(t)})$
else if $\hat{x}_c^{(t)} = \perp$ **then**
  $x^{(t)} \leftarrow \mathring{G}^{(t)} \tilde{A}^{(t)\top} \tilde{b}^{(t)}$
else
  $x^{(t)} \leftarrow \mathring{G}^{(t)} \tilde{A}^{(t)\top} (\tilde{b}^{(t)} - \tilde{A}^{(t)} \hat{x}_c^{(t)})$
end if
return $(x^{(t)}, \mathring{G}^{(t)}, H^{(t)})$

Theorem 3.5. *Let $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$. Suppose that A satisfy Assumption 3.4. With probability at least 0.94, Algorithm 4 makes*

$$\mathcal{O}\left(\frac{d}{\epsilon^9} \log^2 d \cdot \log \frac{d}{\epsilon} \cdot \log\left(\frac{d}{\epsilon} \log \frac{\|A\|_2}{\sigma}\right) \cdot \log \frac{n\|A\|_2}{\sigma}\right)$$

queries in total and maintains for each $T = d+1, \ldots, n$ a solution $\tilde{x}^{(T)}$ which satisfies that

$$\left\|A^{(T)} \tilde{x}^{(T)} - b^{(T)}\right\|_2 \le (1+\epsilon) \min_{x \in \mathbb{R}^d} \left\|A^{(T)} x - b^{(T)}\right\|_2.$$

With probability at least 0.97*, Algorithm 4 runs in a total of*

$$\mathcal{O}\left(\frac{1}{\epsilon^2}\operatorname{nnz}(A)\log n + \frac{d^3}{\epsilon^4}\left(\log\frac{\|A\|_2}{\sigma}\right)\left(\log\frac{d}{\epsilon}\right)\left(\frac{\log n}{\epsilon^2}+d\right)\right)$$

time for processing the entire matrix A*.*

Remark 3.6. The theoretical guarantees of Theorem 3.3 and Theorem 3.5 can be extended to δ failure probability with an additional $\log(1/\delta)$ factor in the query complexities, using the same boosting procedure in (Musco et al., 2021a).

Remark 3.7. We only consider the case $p \in [1,2]$ because adding an extra row to a matrix never increases the Lewis weights of existing rows of that matrix. This property is used in upper bounding the sum of online Lewis weights (see Lemmas 4.1 and 4.2 below). However, this property does not necessarily hold when $p > 2$ and we leave open the problem of upper bounding the sum of online Lewis weights in this case.

4. Proofs of the Main Results

The framework of our Algorithm 4 and Algorithm 1 follows from the algorithm in (Musco et al., 2021a). Hence, in order to prove Theorem 3.5 and 3.3, it suffices to verify all the conditions and lemmata needed by the proof in (Musco et al., 2021a). (Small modifications are needed for $p = 2$ because the sampling matrices do not have independent rows and the details are postponed in Appendix B.) In particular, it suffices to show that

(i) the online ℓ_p Lewis weights calculated in Algorithms 4 and 1 are within an absolute constant factor of the corresponding true ℓ_p online Lewis weights, and
(ii) the sum of approximate ℓ_p online Lewis weights are bounded.

4.1. Sum of Online Lewis Weights

Suppose that (i) holds, (ii) would follow from that the sum of true ℓ_p online Lewis weights are bounded, which are exactly the following two lemmas, for $p \in [1,2)$ and $p = 2$, respectively.

Lemma 4.1. *Let* $p \in [1,2)$*. It holds that* $\sum_{i=1}^n w_i^{OL}(A) = \mathcal{O}(d\log n \cdot \log \kappa^{OL}(A))$.

Lemma 4.2 (Lemma 2.2 of (Cohen et al., 2020))**.** *Let* $p = 2$*. Suppose that* A *satisfy Assumption 3.4. It holds that* $\sum_{i=1}^n w_i^{OL}(A) = \mathcal{O}(d\log(\|A\|_2/\sigma))$.

The case $p = 1$ of Lemma 4.1 appeared in (Braverman et al., 2020). We generalize the result to $p \in (1,2)$, following their approach. The proof can be found in Appendix A.1, where we also note an omission in the proof of Braverman et al. (2020).

In the analysis of Algorithm 1, we shall apply Lemma 4.1 to $\tilde{A}_1 = S_1 A$, where S_1 is a sampling matrix w.r.t. the online Lewis weights of A. To upper bound $\kappa^{\mathrm{OL}}(S_1 A)$, we shall need the following auxiliary lemma, whose proof is postponed to Appendix A.2.

Lemma 4.3. *Let* $p \in [1,2)$ *and* S *is a rescaled sampling matrix w.r.t. the online Lewis weights of* A *and the oversampling parameter* β*. With probability at least* 0.99*, it holds that* $\log \kappa^{OL}(SA) = O(\log(n\kappa^{OL}(A)/\beta))$.

4.2. Approximating Online Lewis Weights

Now, it remains to prove (i) in order to prove the guarantee of $\tilde{x}$ in Theorems 3.3 and 3.5.

First, the guarantee of approximate ℓ_2 online Lewis weights follows from the works of Cohen et al. (2020) or Jiang et al. (2022), which we cite below.

Lemma 4.4 (Theorem 2.3 in (Cohen et al., 2020), Lemma 3.4 in (Jiang et al., 2022))**.** *Let* $\{\tilde{w}_i\}_i$ *be the approximate Lewis weights in Algorithm 4 and* $\beta = \Theta(\log n/\epsilon^2)$*. Let* S *be the rescaled sampling matrix with respect to* $\{\tilde{w}_i\}_i$*. It holds with probability at least* 0.99 *that*

$$(1-\epsilon)(A^{(t)})^\top A^{(t)} \preceq (SA^{(t)})^\top (SA^{(t)}) \preceq (1+\epsilon)(A^{(t)})^\top A^{(t)}$$

for all $t \in \{d+1,\ldots,n\}$ *and the number of non-zero rows of* S *is* $\mathcal{O}(\beta(\sum_{i=1}^n \tilde{w}_i))$.

As a consequence, $\tilde{w}_t \geq \frac{1-\epsilon}{1+\epsilon}\cdot a_i^\top((A^{(t)})^\top A^{(t)})^{-1}a_i \geq (1-2\epsilon)w_t^{\mathrm{OL}}(A)$ for all $t \in \{d+1,\ldots,n\}$. This establishes (i) when $p = 2$.

The case of general p follows from Theorem 3.2. The following lemma is the key to the proof.

Lemma 4.5. *Let* $A_i \in \mathbb{R}^{n_i\times d}$ $(i = 1,\ldots,r)$*,* $B \in \mathbb{R}^{k\times d}$ *and* $M = A_1 \circ A_2 \circ \cdots \circ A_r \circ B$*. For each* $i \in [r]$*, let* $S_i \in \mathbb{R}^{m_i\times n_i}$ *be the rescaled sampling matrix with respect to* $p_{i,1},\ldots,p_{i,n_i}$ *with* $\min\{\beta w_j(A_i),1\} \leq p_{i,j} \leq 1$ *for each* $j \in [n_i]$*, where* $\beta = \mathcal{O}(\epsilon^{-2}\log(d/\delta))$*. Let* $M' = S_1A_1 \circ \cdots \circ S_rA_r \circ B$*. Then, with probability at least* $1-\delta$*, it holds*

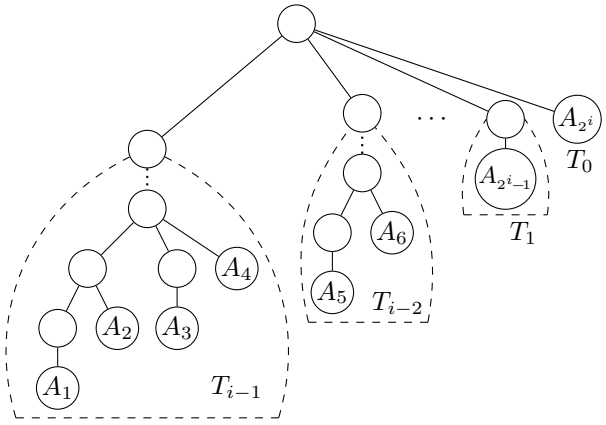
$$(1-\epsilon)w_{n_1+\cdots+n_r+j}(M) \leq w_{m_1+\cdots+m_r+j}(M') \leq (1+\epsilon)w_{n_1+\cdots+n_r+j}(M).$$

for all $j = 1,\ldots,k$.

A full version of the preceding lemma and its proof are postponed to Lemma A.3. Now we turn to prove Theorem 3.2.

Proof of Theorem 3.2. Observe that each block B_i is the compressed version of 2^i smaller matrices, say, $A_1,\ldots,A_{2^i}$, and each smaller matrix is compressed at most i times. The compression scheme inside B_i can be represented by a tree T_i, which satisfies that the root of T_i has

i children $T_{i-1}, T_{i-2}, \ldots, T_0$. Every internal node of the tree represents a compression operation, which subsamples (with rescaling) the vertical concatenation of its children. The following diagram shows an illustration of T_i.



Consider a decompression process which begins at the root and goes down the tree level by level. When going down a level, we decompress each internal node on that level into the vertical concatenation of its children. When the decompression process is completed, we will have a vertical concatenation of the leaves, namely, $A_1 \circ A_2 \circ \cdots \circ A_{2^i}$, which is a submatrix of $A^{(t)}$.

Let i^* be the largest i such that B_i is nonempty. Consider the decompression process of all blocks $B_{\log n} \circ \cdots \circ B_0$. This process will terminate in i^* steps,

$$A^{(t,i^*)} \to A^{(t,i^*-1)} \to \cdots \to A^{(t,0)},$$

where $A^{(t,i^*)} = B_{\log n} \circ \cdots \circ B_0$ and $A^{(t,0)} = A^{(t)}$. Let $\tilde{w}_{t,j} = w_{\text{last}}((A^{t,j}))$. Note that $\tilde{w}_{t,0} = w_t(A^{(t)})$. By Theorem 3.2 and our choices of parameters, it holds that

$$\left(1 - \frac{\epsilon}{2\log n}\right) \tilde{w}_{t,j} \le \tilde{w}_{t,j+1} \le \left(1 + \frac{\epsilon}{2\log n}\right) \tilde{w}_{t,j}$$

with probability at least $1 - 1/\operatorname{poly}(n)$. Iterating yields that

$$\left(1 - \frac{\epsilon}{2\log n}\right)^{i*} w_t(A^{(t)}) \le \tilde{w}_{t,i^*} \le \left(1 + \frac{\epsilon}{2\log n}\right)^{i^*} w_t(A^{(t)}).$$

Note that $\tilde{w}_{t,i^*} = \tilde{w}_t$ per (4). Since $i^* \le \log n$, we have

$$(1-\epsilon) w_t(A^{(t)}) \le \tilde{w}_{t,i^*} \le (1+\epsilon) w_t(A^{(t)}).$$

Taking a union bound over all t gives the claimed result. □

4.3. Time Complexity for $p = 2$

Lemma 4.6. *With probability at least* 0.98*, the running time of Algorithm 4 over n iterations is $\mathcal{O}(\epsilon^{-2} \log n \operatorname{nnz}(A) + \epsilon^{-4} d^3 (\epsilon^{-2} \log n + d) \log \frac{d}{\epsilon} \log \frac{\|A\|_2}{\sigma})$.*

Proof. We analyze the time complexity following Lemma 3.8 in (Jiang et al., 2022). Note that total running time is dominated by calculating Lewis weights and calls to UPDATE. The approximate Lewis weights are calculated by $\tilde{w}_t = \|H^{(t-1)} a_t\|_2^2$, which takes $\mathcal{O}(\epsilon^{-2} \operatorname{nnz}(A) \log n)$ time over n iterations. Observe that the runtime of each call to UPDATE is dominated by the time calculating $F^{(t)}$ and $H^{(t)}$, which takes $\mathcal{O}(\epsilon^{-2} d \log n + d^2)$ time. Calls to UPDATE only happen when there is a new row a_t is sampled and the number of samples is dominated by the maximum of the number of rows of S and that of S_1, which with probability at least 0.98 are $\mathcal{O}(d \log d)$ and $\mathcal{O}(\epsilon^{-4} d^2 \log \frac{d}{\epsilon} \log \frac{\|A\|_2}{\sigma})$, respectively. Hence, the total running time is $\mathcal{O}(\epsilon^{-2} \operatorname{nnz}(A) \log n + \epsilon^{-4} d^4 \log \frac{d}{\epsilon} \log \frac{\|A\|_2}{\sigma} + \epsilon^{-6} d^3 \log n \log \frac{d}{\epsilon} \log \frac{\|A\|_2}{\sigma})$. □

5. Experiments

In this section, we provide empirical results on online active ℓ_p regression with $p = 1$, $p = 1.5$ and $p = 2$. We compare our methods with online uniform sampling, the offline active regression algorithms (Musco et al., 2021a; Chen & Derezinski, 2021; Parulekar et al., 2021) for all values of p and, additionally, the thresholding algorithm in (Riquelme et al., 2017) for $p = 2$. The quantity we compare is the relative error, which is defined as $(err - err_{\text{opt}})/err_{\text{opt}}$, where $err = \|A\tilde{x} - b\|_p$ is the error of the algorithm's output $\tilde{x}$ and $err_{\text{opt}} = \min_x \|Ax - b\|_p$ is the minimum error of the ℓ_p regression. Below we explain the online uniform sampling algorithm, the thresholding algorithm and the adaptation of online and offline active regression algorithms to the budget-constrained setting. All algorithms are prescribed with a budget for querying the labels.

- Online Uniform Sampling: In the t-th round, we sample the new data point $[a_t, b_t]$ with probability $B_t/(n-t)$, where B_t is the remaining budget.
- Regression via Thresholding (for $p = 2$ only): We use the Algorithm 1.b in (Riquelme et al., 2017) and assign the weights $\xi_i = 1$ for all $i \in [n]$.
- Online Active Regression: We sample each data point with probability proportional to $\tilde{w}_t$, where $\tilde{w}_t$ is the approximate online Lewis weight calculated with the compression technique for $p = 1$ and $p = 1.5$.
- Offline Active Regression: For $p = 1$, the algorithms in (Chen & Derezinski, 2021; Parulekar et al., 2021) are under the budget setting and no modification is needed. For $p = 2$, the offline algorithm (Musco et al., 2021a) involves parallel sampling. Since it expects to sample $\mathcal{O}(d)$ data points for a constant-factor approximation, we allocate a budget of size d to the part of the constant-

factor approximation and allocate the remaining budget to the regression on residuals.

We perform experiments on both synthetic and real-world data sets to demonstrate the efficacy of our approaches.

- Synthetic Data: We generate the synthetic data as follows. Each row of $A \in \mathbb{R}^{n \times d}$ is a random Gaussian vector, i.e., $a_i \sim \mathcal{N}(0, I_d)$. The label is generated as $b = Ax^* + \xi$ where x^* is the ground truth vector and ξ is the Gaussian noise vector, i.e., $\xi \sim \mathcal{N}(0, 1)$. To make the rows of A have nonuniform Lewis weights, we enlarge d data points by a factor of $n^{\frac{1}{p}}$. We choose $n = 10000$ and $d = 100$.

- Real-world Data: We evaluate our algorithm on a real-world dataset, the gas sensor data (Vergara et al., 2012; Rodriguez-Lujan et al., 2014) from the UCI Machine Learning Repository[1]. The dataset contains 13910 measurements of chemical gases characterized by 128 features and their concentration levels.

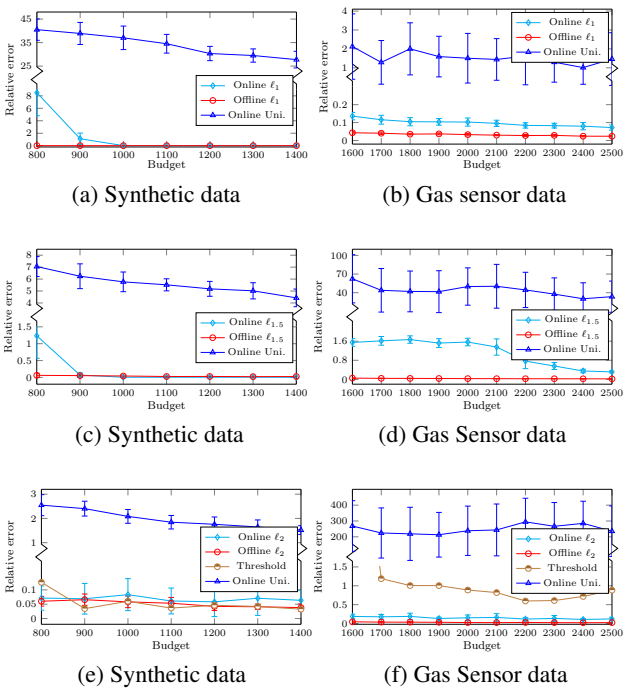
We vary the budget sizes for the synthetic data between 800 and 1400 (8%–14% of the data size) and for the real-world data between 1600 and 2500 (12%–18% of the data size). For each budget size, we run 20 independent trials and calculate the mean relative error and standard deviation. All our experiments are conducted in MATLAB on a Macbook Pro with an i5 2.9GHz CPU and 8GB of memory.

Below we discuss the experiments results for the online active ℓ_p regression, $p = 1, 1.5, 2$. The budget-versus-error plots are shown in Figure 1.

- $p = 1$: For the synthetic data, we see that the online regression algorithm achieves a relative error comparable to that of the offline regression algorithm when the budget is at least 1000 and always significantly outperforms the online uniform sampling algorithm. For the real-world data, the online regression algorithm's performance is again significantly better than the online uniform sampling algorithm and comparable to that of the offline active regression algorithm.

- $p = 1.5$: The online $\ell_{1.5}$ regression algorithm significantly outperforms the online uniform sampling on both data sets. It achieves a relative error comparable to that of the offline active regression algorithm on the real-world data and is only slightly worse than the offline algorithm when the budget size is at least 2300 (14.3% of the data size).

[1] https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations



(a) Synthetic data (b) Gas sensor data

(c) Synthetic data (d) Gas Sensor data

(e) Synthetic data (f) Gas Sensor data

Figure 1. Performance of algorithms for online ℓ_p active regression on both synthetic data and Gas Sensor data for $p = 1, 1.5, 2$.

- $p = 2$: The online ℓ_2 regression algorithm significantly outperforms the online uniform sampling algorithm on both datasets and performs much better than the thresholding algorithm on real-world data. It achieves a relative error comparable to that of the offline active regression algorithm on the synthetic data and is only slightly worse than the offline algorithm on real-world data.

6. Conclusion

We provably show an online active regression algorithm which uses sublinear space for the ℓ_p-norm, $p \in [1, 2]$. Our experiments demonstrate the superiority of the algorithm over online uniform sampling on both synthetic and real-world data and a comparable performance with the offline active regression algorithm.

Acknowledgements

C. Chen was supported by and Y. Li was partially supported by Singapore Ministry of Education (AcRF) Tier 2 grant MOE2018-T2-1-013 and Singapore Ministry of Education (AcRF) Tier 1 grant RG75/21. Y. Sun was supported by Singapore Ministry of Education (AcRF) Tier 2 grant MOE2018-T2-1-013.

References

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

Braverman, V., Drineas, P., Musco, C., Musco, C., Upadhyay, J., Woodruff, D. P., and Zhou, S. Near optimal linear algebra in the online and sliding window models. In Irani, S. (ed.), *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pp. 517–528. IEEE, 2020. doi: 10.1109/FOCS46700.2020.00055.

Chen, X. and Derezinski, M. Query complexity of least absolute deviation regression via robust uniform convergence. In Belkin, M. and Kpotufe, S. (eds.), *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pp. 1144–1179. PMLR, 2021. URL http://proceedings.mlr.press/v134/chen21d.html.

Chen, X. and Price, E. Active regression via linear-sample sparsification. In Beygelzimer, A. and Hsu, D. (eds.), *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pp. 663–695. PMLR, 2019. URL http://proceedings.mlr.press/v99/chen19a.html.

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, 2015.

Cohen, M. B., Musco, C., and Pachocki, J. Online row sampling. *Theory of Computing*, 16(15):1–25, 2020. APPROX-RANDOM 2016 Special Issue.

Gonin, R. and Money, A. H. *Nonlinear L_p-Norm Estimation*. CRC Press, 1989. doi: 10.1201/9780203745526.

Jiang, S., Peng, B., and Weinstein, O. Dynamic least-squares regression. arXiv:2201.00228 [cs.DS], 2022.

Kane, D. M. and Nelson, J. Sparser Johnson-Lindenstrauss transforms. *J. ACM*, 61(1), jan 2014. ISSN 0004-5411. doi: 10.1145/2559902. URL https://doi.org/10.1145/2559902.

Kapralov, M., Lee, Y. T., Musco, C., Musco, C., and Sidford, A. Single pass spectral sparsification in dynamic streams. *SIAM J. Comput.*, 46(1):456–477, 2017. doi: 10.1137/141002281.

Musco, C., Musco, C., Woodruff, D. P., and Yasuda, T. Active sampling for linear regression beyond the l_2 norm. arXiv:2111.04888v1 [cs.LG], 2021a.

Musco, C., Musco, C., Woodruff, D. P., and Yasuda, T. Active sampling for linear regression beyond the l_2 norm. arXiv:2111.04888v3 [cs.LG], 2021b.

Parulekar, A., Parulekar, A., and Price, E. L1 regression with lewis weights subsampling. In Wootters, M. and Sanità, L. (eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pp. 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.APPROX/RANDOM.2021.49.

Riquelme, C., Johari, R., and Zhang, B. Online active linear regression via thresholding. In Singh, S. P. and Markovitch, S. (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 2506–2512. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14599.

Rodriguez-Lujan, I., Fonollosa, J., Vergara, A., Homer, M., and Huerta, R. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems*, 130:123–134, 2014.

Sabato, S. and Munos, R. Active regression by stratification. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 469–477, 2014.

Tropp, J. Freedman's inequality for matrix martingales. *Electronic Communications in Probability*, 16:262–270, 2011.

Vergara, A., Vembu, S., Ayhan, T., Ryan, M. A., Homer, M. L., and Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

A. Some Facts of Lewis Weights

Lemma A.1. *Given $A \in \mathbb{R}^{n\times d}$ with ℓ_p Lewis weights $w_i, i \in [n]$, let S be the rescaled sampling matrix with respect to $p_1, \ldots, p_n$ satisfying that $\min\{\beta w_i, 1\} \le p_i \le 1$, where $\beta = O(\epsilon^{-2}\log(d/\delta))$. With probability at least $1-\delta$, it holds that*

$$(1-\epsilon)\sum_{i=1}^{n} w_i^{1-\frac{2}{p}} a_i a_i^\top \preceq \sum_{i=1}^{n} \frac{(\mathbb{1}_S)_i}{p_i} w_i^{1-\frac{2}{p}} a_i a_i^\top \preceq (1+\epsilon)\sum_{i=1}^{n} w_i^{1-\frac{2}{p}} a_i a_i^\top.$$

Proof. We prove the lemma by the matrix Chernoff bound. Without loss of generality, we assume that $p_i \le 1/\beta$ for all i, otherwise we can restrict the sum to the i's such that $p_i \le 1/\beta$. We further assume that $A^\top W^{1-\frac{2}{p}} A = I_d$, where $W = \mathrm{diag}\{w_1, \ldots, w_n\}$. Let $X_i = \frac{(\mathbb{1}_S)_i}{p_i} \cdot w_i^{1-\frac{2}{p}} a_i a_i^\top - w_i^{1-\frac{2}{p}} a_i a_i^\top$, then $\mathbb{E}X_i = 0$. By the definition of Lewis weights, we have $w_i^{\frac{2}{p}} = a_i^\top (A^\top W^{1-\frac{2}{p}} A)^{-1} a_i$. Hence, we have $\|a_i\|_2^2 = w_i^{\frac{2}{p}}$. Next, it holds that $\|X_i\|_2 \le \frac{w_i^{1-\frac{2}{p}}}{p_i}\|a_i\|_2^2 = \frac{1}{\beta} w_i^{-\frac{2}{p}} \|a_i\|_2^2 = \frac{1}{\beta}$ and

$$\left\|\sum_{i=1}^{n} \mathbb{E}\left(X_i X_i^\top\right)\right\|_2 \le \left\|\sum_{i=1}^{n} \frac{1}{p_i} w_i^{2(1-\frac{2}{p})} \|a_i\|_2^2 a_i a_i^\top\right\|_2 \le \left\|\sum_{i=1}^{i=n} w_i^{1-\frac{2}{p}} \cdot \frac{a_i a_i^\top}{\beta}\right\|_2 = \frac{w_i}{\beta} \le \frac{1}{\beta}.$$

Applying the matrix Chernoff inequality, we have

$$\Pr\left\{\left\|\sum_{i=1}^{n} X_i\right\|_2 \ge \epsilon\right\} \le 2d\exp\left(\frac{-\epsilon^2}{\frac{1}{\beta} + \frac{\epsilon}{3\beta}}\right) = 2d\exp\left(-\Omega(\beta\epsilon^2)\right) \le \delta. \qquad \square$$

Lemma A.2. *Suppose that $A \in \mathbb{R}^{n\times d}$ and $\overline{w_1}, \ldots, \overline{w_n}$ are the Lewis weights of A. Let $w_1, \ldots, w_n$ be weights such that*

$$\alpha w_i^{2/p} \le a_i^\top \left(\sum_i w_i^{1-2/p} a_i a_i^\top\right)^{-1} a_i \le \beta w_i^{2/p}, \quad \forall i = 1, \ldots, n,$$

then $\alpha w_i \le \overline{w_i} \le \beta w_i$ for all i.

Proof. Let $\gamma = \sup\{c > 0 : w_i \ge c\overline{w_i} \text{ for all } i\}$. It then holds for all i that

$$\begin{aligned}
w_i^{2/p} &\ge \frac{1}{\beta} a_i^\top \left(\sum_i w_i^{1-2/p} a_i a_i^\top\right)^{-1} a_i \\
&\ge \frac{1}{\beta} a_i^\top \left(\sum_i w_i^{1-2/p} a_i a_i^\top\right)^{-1} a_i \\
&\ge \frac{1}{\beta} a_i^\top \left(\sum_i (\gamma\overline{w_i})^{1-2/p} a_i a_i^\top\right)^{-1} a_i \\
&= \frac{\gamma^{2/p-1}}{\beta} a_i^\top \left(\sum_i \overline{w_i}^{1-2/p} a_i a_i^\top\right)^{-1} a_i \\
&= \frac{\gamma^{2/p-1}}{\beta} \overline{w_i}^{2/p}.
\end{aligned}$$

This implies that

$$\gamma^{2/p} \ge \frac{\gamma^{2/p-1}}{\beta},$$

and thus

$$\gamma \ge \frac{1}{\beta},$$

that is, $w_i \ge \overline{w_i}/\beta$ for all i. Similarly one can show that $w_i \le \overline{w_i}/\alpha$. $\square$

Combining Lemma A.1 and Lemma A.2 we have the following lemma. We assume that we retain only nonzero rows of any sampling matrix S in the following lemma.

Lemma A.3. *Let $A_i \in \mathbb{R}^{n_i \times d}$ $(i = 1, \dots, r)$, $B \in \mathbb{R}^{k \times d}$ and $M = A_1 \circ A_2 \circ \cdots \circ A_r \circ B$. For each $i \in [r]$, let $S_i \in \mathbb{R}^{m_i \times n_i}$ be the rescaled sampling matrix with respect to $p_{i,1}, \dots, p_{i,n_i}$ with $\min\{\beta w_j(A_i), 1\} \le p_{i,j} \le 1$ for each $j \in [n_i]$, where $\beta = O(\epsilon^{-2} \log(d/\delta))$. Let $M' = S_1 A_1 \circ \cdots \circ S_r A_r \circ B$. The following statements hold with probability at least $1 - \delta$.*

1. *For each $i \in [r]$ and each $j \in [m_i]$, it holds that*

$$(1-\epsilon)\frac{w_{n_1+\cdots+n_{i-1}+s_i(j)}(M)}{p_{i,s_i(j)}} \le w_{m_1+\cdots+m_{i-1}+j}(M') \le (1+\epsilon)\frac{w_{n_1+\cdots+n_{i-1}+s_i(j)}(M)}{p_{i,s_i(j)}},$$

   where $s_i(j) \in [n_i]$ is the row index such that $(S_i)_{j,s_i(j)} \neq 0$.

2. *For each $j = 1, \dots, k$, it holds that*

$$(1-\epsilon) w_{n_1+\cdots+n_r+j}(M) \le w_{m_1+\cdots+m_r+j}(M') \le (1+\epsilon) w_{n_1+\cdots+n_r+j}(M).$$

Proof. Define partial sums $\mu_i = m_1 + \cdots + m_i$ with $\mu_0 = 0$ and $\nu_i = n_1 + \cdots + n_i$ with $\nu_0 = 0$. For each $j \in [\mu_r + k]$,

$$w'_j = \begin{cases} w_{\nu_{i-1}+s_i(j)}(M)/p_{i,s_i(j)}, & \mu_{i-1} < j \le \mu_i; \\ w_{j-\mu_r+\nu_r}(M), & j \ge \mu_r. \end{cases}$$

and

$$L = \sum_{i=1}^{r} \sum_{j=1}^{m_i} \frac{(S_i A_i)_j (S_i A_i)_j^\top}{(w'_{\mu_{i-1}+j})^{p/2-1}} + \sum_{j=1}^{k} \frac{b_i b_i^\top}{(w'_{\mu_r+j})^{p/2-1}}.$$

Then we have

$$L = \sum_{i=1}^{r} \sum_{j=1}^{m_i} \frac{(A_i)_{s_i(j)} (A_i)_{s_i(j)}^\top}{p_{i,s_i(j)} (w_{s_i(j)}(A_i))^{p/2-1}} + \sum_{j=1}^{k} \frac{b_i b_i^\top}{(w_{\nu_r+j}(M))^{p/2-1}}.$$

Let $W_M = \operatorname{diag}\{w_1(M), \dots, w_{\nu_r+k}(M)\}$. Let $p_i = 1$ for $i = \nu_r + 1, \dots, \nu_r + k$. Also note that $p_{i,j} \ge \min\{\beta w_{\nu_{i-1}+j}(M), 1\}$ since $w_j(A_i) \ge w_{\nu_{i-1}+j}(M)$. It follows from Lemma A.1 that

$$(1-\epsilon)(M^\top W_M^{1-2/p} M) \preceq L \preceq (1+\epsilon)(M^\top W_M^{1-2/p} M),$$

with probability at least $1 - \delta$.

Next we verify that $\{w'_j\}_j$ are good weights for M'. When $\mu_{i-1} < j \le \mu_i$,

$$\begin{aligned}(w'_j)^{2/p} = \frac{(w_{\nu_{i-1}+s_i(j)}(M))^{2/p}}{p_{i,s_i(j)}^{2/p}} &= \frac{(A_i)_{s_i(j)} (M^\top W_M^{1-2/p} M)^{-1} (A_i)_{s_i(j)}^\top}{p_{i,s(j)}^{2/p}} \\ &= \frac{1}{1 \pm \epsilon} \cdot \frac{(A_i)_{s_i(j)} L^{-1} (A_i)_{s_i(j)}^\top}{p_{i,s_i(j)}^{2/p}} = \frac{1}{1 \pm \epsilon} (S_i A_i)_j L^{-1} (S_i A_i)_j^\top,\end{aligned}$$

where $(S_i A_i)_j$ denotes the j-th row of $S_i A_i$. Similarly, one can show that for $j > \mu_r$,

$$(w'_{\mu_r+j})^{2/p} = \frac{1}{1 \pm \epsilon} b_{j-\mu_r} L^{-1} b_{j-\mu_r}^\top.$$

The result follows from Lemma A.2. □

A.1. Online ℓ_p Lewis Weights

The goal of this section is to show Lemma 4.1, which states that the sum of the online ℓ_p Lewis weights of a matrix $A \in \mathbb{R}^{n\times d}$ is upper bounded by $O(d \log n \log(\kappa^{\text{OL}}(A)))$ for $p \in [1, 2)$. This is a generalization of Lemma 5.15 of (Braverman et al., 2020) from $p = 1$ and we follow the same approach in (Braverman et al., 2020).

Lemma A.4 (Monotonicity, Lemma 5.5 in (Cohen & Peng, 2015))**.** *For any matrix $A \in \mathbb{R}^{n\times d}$ and vector $x \in \mathbb{R}^d$, for every $i \in [n]$ we have $w_i(A) \geq w_i(B)$ where $B = [A^\top, x^\top]^\top$.*

Lemma A.5. *If the leverage scores of A are at most $C > 0$, then the ℓ_p Lewis weights of A are at most C for $p \in [1, 2]$.*

Proof. This is the generalization of Lemma 5.12 in (Braverman et al., 2020) and we follow the same proof approach.

By the assumption, we have $a_i^\top (A^\top A)^{-1} a_i \leq C$ for $i \in [n]$. We prove by induction that for iteration j in the Lewis weight iteration, we have $W^{(j)} \preceq C^{1-(1-p/2)^j} I_n$.

For the base case $j = 1$, we have $W^{(j)}_{i,i} = (a_i^\top (A^\top A)^{-1} a_i)^{p/2} \leq C^{p/2}$. Thus $W^{(1)} \preceq C^{p/2} I_n$ as desired.

For iteration j, by the induction hypothesis, we have $W^{(j-1)} \preceq C^{1-(1-p/2)^{j-1}} I_n$, which implies that $(W^{(j-1)})^{1-2/p} \succeq C^{(1-(1-p/2)^{j-1})(1-2/p)} I_n$ since $1 - 2/p \leq 0$. Thus,

$$A^\top (W^{(j-1)})^{1-2/p} A \succeq C^{(1-(1-p/2)^{j-1})(1-2/p)} A^\top A,$$

and

$$(A^\top (W^{(j-1)})^{1-2/p} A)^{-1} \preceq C^{(1-(1-p/2)^{j-1})(2/p-1)} (A^\top A)^{-1}.$$

It then follows from (2) that

$$(W^{(j)}_{i,i})^{2/p} = a_i^\top (A^\top (W^{(j-1)})^{1-2/p} A)^{-1} a_i \leq C^{(1-(1-p/2)^{j-1})(2/p-1)} a_i^\top (A^\top A)^{-1} a_i \leq C^{(1-(1-p/2)^{j-1})(2/p-1)+1}.$$

Notice that $((1 - (1 - p/2)^{j-1})(2/p - 1) + 1)p/2 = 1 - (1 - p/2)^j$, we have obtained that $W^{(j)}_{i,i} \leq C^{1-(1-p/2)^j}$ for all i, i.e., $W^{(j)} \preceq C^{1-(1-p/2)^j} I_n$. The induction step is established.

The claim follows the convergence of Lewis weight iteration (Cohen & Peng, 2015). □

Lemma A.6. *Given $A = [a_1, \ldots, a_n]^\top \in \mathbb{R}^{n\times d}$, let $B \in \mathbb{R}^{(n+1)\times d} = [a_1, \ldots, a_{j-1}, b_j, a_{j+1}, \ldots, a_n, b_{n+1}]^\top$ where $b_j = (1 - \gamma)^{1/p} a_j$ and $b_{n+1} = \gamma^{1/p} a_j$ for some $\gamma \in [0, 1]$ and $j \in [n]$. Then we have $w_i(A) = w_i(B)$ for $i \neq j, n + 1$, $w_j(B) = (1 - \gamma) w_j(A)$ and $w_{n+1}(B) = \gamma w_j(A)$.*

Proof. Without loss of generality, we suppose $j = n$. Let $W \in \mathbb{R}^{n\times n}$ be the diagonal Lewis weight matrix of A, i.e., $W_{i,i} = w_i(A)$. Let $\overline{W}^{(n+1)\times(n+1)}$ be a diagonal matrix where $\overline{W}_{i,i} = w_i(A)$ for $i = 1, \ldots, n - 1$, $\overline{W}_{n,n} = (1 - \gamma) w_n(A)$ and $\overline{W}_{n+1,n+1} = \gamma w_n(A)$. According to the uniqueness of Lewis weights, it suffices to show that $\tau_i(\overline{W}^{1/2-1/p} B) = \overline{W}_{i,i}$ for $i \in [n + 1]$.

Notice that the first $n - 1$ rows of $\overline{W}^{1/2-1/p} B$ are the same as those of $\overline{W}^{1/2-1/p} A$. The last two rows of $\overline{W}^{1/2-1/p} B$ are $w_n(A)^{1/2-1/p}(1 - \gamma)^{1/2-1/p}(1 - \gamma)^{1/p} a_n = w_n(A)^{1/2-1/p}(1 - \gamma)^{1/2} a_n$ and $w_n(A)^{1/2-1/p} \gamma^{1/2} a_n$, respectively. Thus we have $\|W^{1/2-1/p} A y\|_2^2 = \|\overline{W}^{1/2-1/p} B y\|_2^2$ for any vector y, which indicates that the leverage scores of the first $n - 1$ rows of $W^{1/2-1/p} A$ are the same as those of $\overline{W}^{1/2-1/p} B$, i.e., $\tau_i(\overline{W}^{1/2-1/p} B) = W_{i,i} = \overline{W}_{i,i}$ for $1 \leq i \leq n - 1$.

For the last two rows, we have $\tau_n(\overline{W}^{1/2-1/p} B) = (1 - \gamma)\tau_n(W^{1/2-1/p} A) = \overline{W}_{n,n}$ and $\tau_{n+1}(\overline{W}^{1/2-1/p} B) = \gamma \cdot \tau_n(W^{1/2-1/p} A) = \overline{W}_{n+1,n+1}$. Thus we have $\tau_i(\overline{W}^{1/2-1/p} B) = \overline{W}_{i,i}$ for all $i \in [n + 1]$. □

Corollary A.7. *For any matrix $A \in \mathbb{R}^{n\times d}$. Let $B \in \mathbb{R}^{n\times d}$ have the same rows but with the j-th row reweighted by a factor $\alpha \in [0, 1]$. Then for all $i \neq j$, $w_i(B) \geq w_i(A)$.*

Proof. Let $\gamma = 1 - \alpha^p$ and $\bar{B} \in \mathbb{R}^{(n+1)\times d} = [a_1, \ldots, a_{j-1}, (1 - \gamma)^{1/p} a_j, a_{j+1}, \ldots, a_n, \gamma^{1/p} a_j]^\top$. By Lemma A.6, we have $w_i(\bar{B}) = w_i(A)$ for $i \neq j$. Then by Lemma A.4 we have $w_i(B) \geq w_i(\bar{B}) = w_i(A)$. □

We are now ready to prove Lemma 4.1, which we restate below as Lemma A.8.

Lemma A.8. *For $A \in \mathbb{R}^{n\times d}$ and each $i \in [n]$, we denote $w_i^{OL}(A)$ be the online Lewis weight of a_i with respect to A. Then $\sum_{i=1}^n w_i^{OL}(A) = \mathcal{O}(d \log n \log \kappa^{OL}(A))$.*

Proof. The first part of our proof is similar to the proof of Lemma 5.15 in (Braverman et al., 2020). Suppose that $\lambda > 0$. Let $B_0 = \lambda I_d$, $B = \underbrace{B_0 \circ \cdots \circ B_0}_{n \text{ times}}$ and $X \triangleq B \circ A$. Following the proof of Lemma 5.15 of (Braverman et al., 2020), we have $\sum_{i=1}^n w_i^{\mathrm{OL}}(X) = \mathcal{O}(d \log n \log \kappa^{\mathrm{OL}}(A))$.

Now, let W_A be the Lewis weight matrix of A and $L = A_i^\top W_A^{1-2/p} A$. Let $\sigma = \lambda_{\min}(L)$, the smallest eigenvalue of L, and $\rho = \min_i (L^{-1})_{ii}$, the smallest diagonal element of L_i^{-1}. Choose $\lambda \le \left(\frac{\sigma}{n}\right)^{1/p} \rho^{(2-p)/(2p)}$, $\mu = (\frac{n\lambda^2}{\sigma})^{p/(2-p)}$, $U_X = \mu I_{nd}$ and $W_X = \begin{bmatrix} U_X & \\ & W_A \end{bmatrix}$. We claim that

$$\frac{1}{2}\mu^{2/p} \le B_j^\top \left(A^\top W_A^{1-2/p} A + B^\top U_X^{1-2/p} B\right)^{-1} B_j, \tag{5}$$

$$\frac{1}{2}(w_i(A))^{2/p} \le a_i^\top \left(A^\top W_A^{1-2/p} A + B^\top U_X^{1-2/p} B\right)^{-1} a_i \tag{6}$$

for all $j \in [nd]$ and all $i \in [n]$. Observe that $B^\top U_X^{1-2/p} B = n\lambda^2 \mu^{1-2/p} I_d \le \sigma I_d \preceq L$. Thus,

$$a_i^\top (L + n\lambda^2 \mu^{1-2/p} I_d)^{-1} a_i \ge \frac{1}{2} a_i^\top L^{-1} a_i = \frac{1}{2}(w_i(A))^{2/p},$$

establishing (6). Similarly, since $B_j = \lambda e_i$ for some i,

$$B_j^\top (L + n\lambda^2 \mu^{1-2/p} I_d)^{-1} B_j \ge \frac{1}{2}\lambda^2 (L^{-1})_{i,i} \ge \frac{1}{2}\lambda^2 \rho \ge \frac{1}{2}\mu^{2/p},$$

establishing (5). It then follows from Lemma A.2 that $w_i(A) \le 2 w_{nd+i}(X)$. Applying the argument above to the n submatrices which consist of the first i rows of A for each $i = 1, \ldots, n$, we see that we can choose λ to be sufficiently small such that $w_i^{\mathrm{OL}}(A) \le 2 w_{nd+i}^{\mathrm{OL}}(X)$ for all i. Therefore, $\sum_i w_i^{\mathrm{OL}}(A) = \mathcal{O}(d \log n \log \kappa^{OL}(A))$. □

Finally, we note an omission in the proof of Lemma 5.15 of (Braverman et al., 2020). In the arXiv version of (Braverman et al., 2020)[2], on page 42 it states that $U_X^{(j-1)} \preceq I_d$ and so $B^\top (U_X^{(j-1)})^{-1} B \preceq n\lambda^2 I_d$. The first inequality does not seem to imply the second one, as the latter requires a lower bound on $U_X^{(j-1)}$. We have used a different argument in our proof above.

A.2. Proof of Lemma 4.3

First, we note the following facts. For any two matrices A and B, $\|AB\|_2 \le \|A\|_2 \|B\|_2$, and when A has full row rank and $B \neq 0$, $\sigma_{\min}(AB) \ge \sigma_{\min}(A)\sigma_{\min}(B)$, where $\sigma_{\min}(\cdot)$ denotes the smallest nonzero singular value of a matrix.

It is clear that S, which is a rescaled sampling matrix, has full row rank. By the definition of the online condition number,

$$\begin{aligned}
\kappa^{\mathrm{OL}}(SA) = \|SA\|_2 \max_i \frac{1}{\sigma_{\min}(SA^{(i)})} &\le \|S\|_2 \|A\|_2 \max_i \frac{1}{\sigma_{\min}(SA^{(i)})_i} \\
&\le \|S\|_2 \|A\|_2 \max_i \frac{1}{\sigma_{\min}(S)\sigma_{\min}(A)_i} \\
&= \frac{\sigma_{\max}(S)}{\sigma_{\min}(S)} \kappa^{\mathrm{OL}}(A).
\end{aligned}$$

Now, observe that $\sigma_{\max}(S) = \max_i p_i^{-1/p} = (\min_i p_i)^{-1/p}$ and $\sigma_{\min}(S) = \min_i p_i^{-1/p} = (\max_i p_i)^{-1/p}$, where $\min\{\beta w_i^{\mathrm{OL}}(A), 1\} \le p_i \le 1$. It is clear that $\sigma_{\min}(S) \ge 1$. For the upper bound of $\sigma_{\max}(S)$, note that a row i with

[2] arXiv:1805.03765v4 [cs.DS], 19 Apr 2020.

$w_i^{\mathrm{OL}}(A) \le 1/(100n)$ will be sampled with probability

$$1 - \left(1 - \frac{1}{100n}\right)^n \le \frac{1}{100}.$$

Hence, with probability at least 0.99, none of the rows i with $w_i^{\mathrm{OL}}(A) \le 1/(100n)$ is sampled and so $\min_i p_i \ge \beta/(100n)$ and $\sigma_{\max}(S) \le (100n/\beta)^{1/p}$. Therefore, we conclude that with probability at least 0.99,

$$\kappa^{\mathrm{OL}}(SA) \le \left(\frac{100n}{\beta}\right)^{1/p} \kappa^{\mathrm{OL}}(A).$$

B. Omitted proofs of Theorem 3.5 and 3.1

In this section we highlight the modifications needed to prove Theorem 3.5 and Theorem 3.1, based on (Musco et al., 2021a). When $p = 2$, our sampling matrices does not have independent rows, since the online leverage scores are calculated with respect to sampled rows instead of all the rows that have been revealed. Hence, we cannot use a Bernstein bound, which is exactly where we need modify in the proof of Theorem 4.1 in (Musco et al., 2021a). This problem does not exist for $p \in [1, 2)$ and the original proof in (Musco et al., 2021a) applies. Below we shall reprove a key technical lemma in (Musco et al., 2021a) for $p = 2$ but state the auxiliary lemmas with a general p whenever possible. It was originally proved in the offline setting and we shall need to make small modifications to its proof so that it can be applied to the online setting.

Lemma B.1 (Lemma 3.7 in (Musco et al., 2021a))**.** *Consider the same setting in Lemma B.2. With probability at least 0.99, for all $x \in \mathbb{R}^d$ with $\|Ax\|_p = \mathcal{O}(\mathsf{OPT})$,*

$$\left|\|SAz - S\bar{z}\|_p^p - \|Ax - \bar{z}\|_p^p\right| = \mathcal{O}(\epsilon)\, \mathsf{OPT}^p\,.$$

Its proof depends on a series of lemmas, namely Lemmas B.2 to B.8. Lemmas B.2 to B.4, B.6 and B.7 are identical to those in (Musco et al., 2021a) and so we only cite the statements. The modification occurs in the proof of Lemma B.8 as well as in the proof of Lemma B.1 when given Lemma B.8.

Lemma B.2 (Constant factor approximation, Theorem 3.2 in (Musco et al., 2021a))**.** *Let $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$, $p \in [1, 2]$ and $\mathsf{OPT} = \min_{x\in\mathbb{R}^d} \|Ax - b\|_p$. If we sample A and obtain x_c by Algorithm 4 or Algorithm 1 with $\beta = \mathcal{O}(\log(d/\delta))$ then with probability at least $1 - \delta$,*

$$\|Ax_c - b\|_p \le 2^{1+\frac{1}{p}} 3/\delta^{\frac{1}{p}}\, \mathsf{OPT}.$$

When δ is constant then $\|Ax_c\|_p \le C \cdot \mathsf{OPT}$ for constant C.

Lemma B.3 (Lemma 3.5 in (Musco et al., 2021a))**.** *Considering the same setting in Lemma B.2, let $z = b - Ax_c$. Let $\mathcal{B}$ be an index set such that $\mathcal{B} = \{i \in [n] : \frac{|z_i|^p}{\mathsf{OPT}^p} \ge \frac{d^{\frac{2}{p}-1} w_i}{\epsilon^p}\}$. Let $\bar{z}$ be equal to z but with all entries in $\mathcal{B}$ set to 0. Then for all $x \in \mathbb{R}^d$ with $\|Ax\|_p = \mathcal{O}(\mathsf{OPT})$,*

$$\left|\|Ax - z\|_p^p - \|Ax - \bar{z}\|_p^p - \|z - \bar{z}\|_p^p\right| = \mathcal{O}(\epsilon)\, \mathsf{OPT}^p\,.$$

Lemma B.4 (Lemma 3.6 in (Musco et al., 2021a))**.** *Consider the same setting in Lemma B.2. With probability at least 0.99, $\|Sz\|_p = \mathcal{O}(\mathsf{OPT})$ and for any $x \in \mathbb{R}^d$ with $\|Ax\|_p = \mathcal{O}(\mathsf{OPT})$,*

$$\left|\|SAx - S\bar{z}\|_p^p - \|Ax - \bar{z}\|_p^p - \|Sz - S\bar{z}\|_p^p\right| = \mathcal{O}(\epsilon)\, \mathsf{OPT}^p\,.$$

Lemma B.5 (Bound over Net)**.** *Let $\mathcal{N}_\epsilon$ be an ϵ-net of the l_p unit ball $\{Ax \mid \|Ax\|_p \le 1\}$. If $|\|SAx - S\bar{z}\|_p - \|Ax - z\|_p| = \mathcal{O}(\epsilon)$ holds for all $x \in \mathcal{N}_\epsilon$. Then for any $x \in \mathbb{R}^d$ with $\|Ax\|_p \le 1$, we have $|\|SAx - S\bar{z}\|_p - \|Ax - z\|_p| = \mathcal{O}(\epsilon)$.*

Proof. To simplify the writing, without loss of generality, we assume $\mathsf{OPT} = 1$. For any x in the unit ball, by the definition of $\mathcal{N}_\epsilon$, there exists a vector $y \in \mathcal{N}_\epsilon$ such that $\|Ax - Ay\|_p \le \epsilon$. We have proved that S is a subspace embedding matrix for A, so $\|SAx - SAy\|_2 \le \mathcal{O}(\epsilon)$. Hence, we have

$$\begin{aligned}|\|SAx - Sz\|_2 - \|Ax - z\|_2| &\le |\|SAy - Sz\|_2 - \|Ay - z\|_2| + \|SAx - SAy\|_2 + \|Ax - Ay\|_2\\ &\le \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon) + \epsilon.\end{aligned}$$

□

Lemma B.6 (Compact rounding, Lemma 3.10 in (Cohen et al., 2020) and Theorem 7.3 in (Bourgain et al., 1989))**.** *Consider $A \in \mathbb{R}^{n\times d}$, $v \in \mathbb{R}^n$ with $|v(i)|^p \le \frac{d^{\frac{p}{2}-1}w_i}{\epsilon^p}$. Let l be $(1+\epsilon)^l = d^{\frac{1}{p}}$ and $\mathcal{N}_\epsilon$ be an ϵ-net in Lemma B.5. For any $y \in \mathcal{N}_\epsilon$, let $r = y - v$. Then we have $r' = e + \sum_{k=0}^{k=l} d_k$ such that*

1. *$|r'(i) - r(i)| \le \epsilon|v(i)|$, for any $i \in [n]$,*
2. *$|d_k(i)| \le \frac{2(1+\epsilon)^k w_i^{\frac{1}{p}}}{\epsilon d}$, for any $i \in [n]$ and $k \in \{0\} \cup [l]$,*
3. *$d_0, \ldots, d_l, e$ have mutually disjoint supports,*
4. *e is a single fixed vector with $|e(i)| \le \frac{w_i^{1/p}}{\epsilon}$, for any $i \in [n]$,*
5. *Each d_k is drawn from a set of vectors $\mathcal{D}_k$ with $\log|\mathcal{D}_k| \le c(p)\frac{d\log(n)}{\epsilon^{1+p}(1+\epsilon)^{pk}}$, for any $k \in [l]$.*

Lemma B.7. *For any $y = Ax$ with $\|Ax\|_p \le 1$, let $r = y - \bar{z}$ and r' be the rounding of r shown to exist in in Lemma B.6, with ϵ set to $\epsilon^{\frac{1}{p}}$. If $\|Sr'\|_p^p = (1\pm\epsilon)\|r'\|_p^p$, then $\|Sr\|_p^p = (1\pm\epsilon)\|r\|_p^p$.*

The next lemma is where we need to modify for the online setting, for which we shall use Freedman's inequality instead of Bernstein's inequality.

Lemma B.8. *For all roundings r' produced by Lemma B.7, with probability at least 0.99, we have*

$$\|Sr'\|_p^p = (1\pm\epsilon)\|r'\|_p^p$$

Proof. we analyze the sampling process via a martingale. To make modifications more clear, we use the notation in Claim 3.14 of (Musco et al., 2021a). We write $r' = e + \sum_{k=0}^{l} d_k$. We have $\|e\|_p = \mathcal{O}(1)$ and $\|d_k\| = \mathcal{O}(1)$ for $k \in \{0\}\cup[l]$. Let S be the rescaled sampling matrix with respect to p_i. Let $Sd_{k,(i)}$ and $d_{k,(i)}$ be the first i coordinates of Sd_k and d_k respectively. Let $Y_i = |Sd_{k,(i+1)}|^p - |d_{k,(i+1)}|^p$, $Y_0 = 0$ and $X_i = Y_i - Y_{i-1}$. By the second condition of Lemma B.6, we have $|d_k(i)| \le \frac{2(1+\epsilon)^k w_i^{\frac{1}{p}}}{\epsilon d^{\frac{1}{p}}}$. Since S rescales d_k by the sampling probability, we have $|Sd_k(i)|^p \le \frac{1}{\beta\tilde{w}_i}\cdot\frac{2^p(1+\epsilon)^{kp}w_i}{\epsilon^p d} = \mathcal{O}\left(\frac{(1+\epsilon)^{kp}}{\beta\epsilon^p d}\right)$. Hence, $|X_i| = |Sd_k(i+1)|^p - |d_k(i+1)|^p \le \mathcal{O}(\frac{(1+\epsilon)^{kp}}{\beta\epsilon^p d})$ and $\mathbb{E}_{i-1}X_i^2 \le \frac{1}{\beta\tilde{w}_i}|d_k(i+1)|^{2p} \le \mathcal{O}(\frac{(1+\epsilon)^{kp}}{\beta\epsilon^p d})|d_k(i+1)|^p$. Thus, we have $\sum_{i=1}^n \mathbb{E}_{i-1}X_i^2 \le \mathcal{O}(\frac{(1+\epsilon)^{kp}}{\beta\epsilon^p d})$. By Freedman inequality,

$$\Pr\left(\left|\|Sd_k\|_p^p - \|d_k\|_p^p\right| \ge \epsilon/(l+2)\right) \le \exp\left(-\frac{\epsilon^2/(2(l+2)^2)}{\mathcal{O}(\frac{(1+\epsilon)^{kp}}{\beta\epsilon^p d})}\right).$$

Therefore, if $\beta = \mathcal{O}(\log|\mathcal{D}_k|\cdot\frac{l^2\epsilon^{p+2}}{d(1+\epsilon)^{kp}}) = \mathcal{O}(\frac{\log^2 d\log n}{\epsilon^{2p+5}})$, we can take a union bound over all k and e. This completes the proof. □

Now we are ready to prove Lemma B.1.

Proof of Lemma B.1. Let $p = 2$. We prove the lemma by the matrix Freedman inequality. Let $Y_i = \|(SA)_i x - S\bar{z}_i\|_2^2 - \|A_i x - z_i\|_2^2$, $Y_0 = 0$ and $X_i = Y_i - Y_{i-1}$. Then, $|X_i|$ is uniformly bounded.

$$|X_i| = \left|\left\|\frac{\mathbb{1}_i}{\sqrt{p_i}}(a_i x - \bar{z}_i)\right\|_2^2 - \|a_i x - \bar{z}_i\|_2^2\right| \le \frac{1}{p_i}\cdot\|a_i x - \bar{z}_i\|_2^2.$$

If $i \in \mathcal{B}$, $\bar{z}_i = 0$, then, by Cauchy-Schwarz inequality, we have $\|a_i x\|_2^2 \le w_i^2\|Ax\|_2^2 = w_i\mathcal{O}(\mathsf{OPT}^2)$. Otherwise, $\|a_i x - \bar{z}_i\|_2^2 \le (\frac{1}{\epsilon}+1)^2 w_i^2\mathcal{O}(\mathsf{OPT})^2$. Hence, since $p_i = \min(\beta w_i, 1)$, we have $\|Y_i - Y_{i-1}\| \le \frac{1}{\beta\epsilon^2}\mathcal{O}(\mathsf{OPT}^2)$. $\mathbb{E}(X_i^2|Y_i,\ldots,Y_1)$

is denoted by $\mathbb{E}_{i-1}X_i^2$, so we have

$$\begin{aligned}\mathbb{E}_{i-1}X_i^2 &= \mathbb{E}\left(\|\frac{1}{\sqrt{p_i}}(a_ix-\bar{z}_i)\|_2^2 - \|a_ix-\bar{z}_i\|_2^2\right)^2\\ &= \mathbb{E}(\frac{1}{p_i}-1)^2\|a_ix-\bar{z}_i\|_2^4\\ &= (\frac{1}{p_i}-1)\|a_ix-\bar{z}_i\|_2^4\\ &\le \frac{w_i}{p_i}(\frac{1}{\epsilon}+1)^2\mathcal{O}(\mathsf{OPT}^2)\|a_ix-\bar{z}_i\|_2^2\\ &\le \frac{1}{\beta\epsilon^2}\mathcal{O}(\mathsf{OPT}^2)\|a_ix-\bar{z}_i\|_2^2.\end{aligned}$$

Therefore, $\sum_{i=1}^n \mathbb{E}_{i-1}X_i^2 \le \frac{1}{\beta\epsilon^2}\mathcal{O}(\mathsf{OPT}^2)\cdot\sum_{i=1}^n\|a_ix-\bar{z}_i\|_2^2$. Since $\|Ax\|_2^2=\mathcal{O}(\mathsf{OPT}^2)$ and $\|z\|_2^2=\mathcal{O}(\mathsf{OPT}^2)$, we can get $\sum_{i=1}^n\mathbb{E}_{i-1}X_i^2 \le \frac{1}{\beta\epsilon^2}\mathcal{O}(\mathsf{OPT}^4)$.

Then, by the matrix Freedman inequality (Tropp, 2011) and $\beta=\frac{d\log(\frac{1}{\delta})}{\epsilon^4}$, it follows that

$$\Pr(\|Y_n\|\ge C\epsilon\,\mathsf{OPT}^2) \le \exp\left(\frac{-C^2\epsilon^2\,\mathsf{OPT}^4}{\frac{1}{\beta\epsilon^4}\mathcal{O}(\mathsf{OPT}^4)+\frac{\mathcal{O}(\mathsf{OPT}^4)}{3\beta\epsilon}}\right) \le \exp\left(\frac{-\beta\epsilon^4}{2}\right)$$

for C large enough. This implies that with probability at least $1-\frac{\delta}{2^d}$, $\left|\|S(Ax-z)\|_2^2-\|Ax-z\|_2^2\right|\le\mathcal{O}(\epsilon)\,\mathsf{OPT}^2$, for a fixed $x\in\mathbb{R}^d$.

To simplify the writing, without loss of generality, now we assume $\mathsf{OPT}=1$. We apply a union bound over an ϵ-net $\mathcal{N}$ of the ball $\mathbf{B}=\{x\in\mathbb{R}^d|\,\|Ax-z\|_2^2=1)\}$. Note that there are at most $(\frac{3}{\epsilon})^d$ points in the ϵ-net. After applying a union bound over the net, according to Lemma B.5 $\left|\|S(Ax-z)\|_2^2-\|Ax-z\|_2^2\right|\le\mathcal{O}(\epsilon)$ holds for each $x\in\mathbb{R}^d$ with $\|Ax\|=1$ with probability at least $1-\delta$.

For any $x\in\mathbb{R}^d$ with $\|Ax\|_2^2=1$, by the definition of ϵ-net, there exists a vector $y\in\mathcal{N}$ such that $\|Ax-Ay\|_2\le\epsilon$. We have proved that S is a subspace embedding for A, so $\|S(Ax-y)\|_2\le\mathcal{O}(\epsilon)$. Hence, we have

$$\begin{aligned}|\|S(Ax-z)\|_2-\|Ax-z\|_2| &\le |\|S(Ay-z)\|_2-\|Ay-z\|_2| + \|S(Ax-Ay)\|_2+\|Ax-Ay\|_2\\ &\le \mathcal{O}(\epsilon)+\mathcal{O}(\epsilon)+\epsilon,\end{aligned}$$

which completes our proof. □