2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-5519-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS54457.2022.00076

Active Linear Regression for ℓ_p Norms and Beyond

Cameron Musco
University of Massachusetts Amherst
cmusco@cs.umass.edu

Christopher Musco
New York University
cmusco@nyu.edu

David P. Woodruff
Carnegie Mellon University
dwoodruf@cs.cmu.edu

Taisuke Yasuda
Carnegie Mellon University
taisukey@cs.cmu.edu

***Abstract*—We study active sampling algorithms for linear regression, which aim to query only a small number of entries of a target vector and output a near minimizer to the objective function.**

For ℓ_p norm regression for any $0 < p < \infty$, we give an algorithm based on Lewis weight sampling which outputs a $(1+\epsilon)$-approximate solution using just $\tilde{O}(d/\epsilon^2)$ queries to $\mathbf{b}$ for $p \in (0,1)$, $\tilde{O}(d/\epsilon)$ queries for $p \in (1,2)$, and $\tilde{O}(d^{p/2}/\epsilon^p)$ queries for $p \in (2,\infty)$. For $p \in (0,2)$, our bounds are optimal up to logarithmic factors, thus settling the query complexity for this range of p. For $p \in (2,\infty)$, our dependence on d is optimal, while our dependence on ϵ is off by at most a single ϵ factor, up to logarithmic factors. Our result resolves an open question of Chen and Dereziński, who gave near optimal bounds for the ℓ_1 norm, but required at least d^2/ϵ^2 samples for ℓ_p regression with $p \in (1,2)$, and gave no bounds for $p \in (2,\infty)$ or $p \in (0,1)$.

We also provide the first total sensitivity upper bound for loss functions with at most degree p polynomial growth. This improves a recent result of Tukan, Maalouf, and Feldman. By combining this with our techniques for ℓ_p regression, we obtain the first active regression algorithms for such loss functions, including the important cases of the Tukey and Huber losses. This answers another question of Chen and Dereziński. Our sensitivity bounds also give improvements to a variety of previous results using sensitivity sampling, including Orlicz norm subspace embeddings, robust subspace approximation, and dimension reduction for smoothed p-norms.

Finally, our active sampling results give the first sublinear time algorithms for Kronecker product regression under every ℓ_p norm. Previous results required reading the entire $\mathbf{b}$ vector in the kernel feature space.[1]

***Index Terms*—active learning, linear regression**

I. Introduction

We consider a classic active learning problem: given a design matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and query access to entries of an unknown target (measurement) vector $\mathbf{b} \in \mathbb{R}^n$, how can we compute an approximate minimizer of the regression problem $\min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{Ax}-\mathbf{b}\|$ while querying *as few entries of* $\mathbf{b}$ *as possible*? This problem arises in applications where labeled data is expensive: viewing a single entry of $\mathbf{b}$ might require running a survey, physical experiment, or time-intensive computer simulation [44], [45]. Concretely, we study the following problem for general vector norms[2] $\|\cdot\|$:

Problem I.1. *For $\mathbf{A} \in \mathbb{R}^{n\times d}, \mathbf{b} \in \mathbb{R}^n$, and accuracy parameter $0 < \epsilon \leq 1$, find $\tilde{\mathbf{x}} \in \mathbb{R}^d$ satisfying:*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq (1+\epsilon)\cdot \min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{Ax}-\mathbf{b}\|,$$

while reading as few of the entries $\{\mathbf{b}(1),\ldots,\mathbf{b}(n)\}$ of the target vector $\mathbf{b}$ as possible.[3]

Notably, the formulation of Problem I.1 makes no assumptions on $\mathbf{A}$ and $\mathbf{b}$. For example, we do not assume that there exists a ground truth $\bar{\mathbf{x}}$ and that $\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}$ is bounded in magnitude, or follows some distribution (e.g., has random Gaussian entries). Under these stronger assumptions, much is known about the problem, which has been studied for decades in the statistics literature on "optimal design of experiments", as well as in machine learning [9], [33], [44].

In contrast, progress on the assumption-free version of the problem has only come in recent years, thanks to advances in random matrix theory and randomized numerical linear algebra. This is for good reason: solving Problem I.1 inherently requires choosing which entries of $\mathbf{b}$ to query in a *randomized way*: an adversary can easily "fool" any deterministic algorithm by concentrating error in $\mathbf{Ax}-\mathbf{b}$ on the indices of $\mathbf{b}$ that will be deterministically queried.

A. Prior Work

Euclidean Norm. Problem I.1 is fully understood when the error is measured in the ℓ_2 norm, $\|\mathbf{w}\|_2 = \left(\sum_{i=1}^n |\mathbf{w}_i|^2\right)^{1/2}$ – i.e., for least squares regression. The typical approach is to subsample and reweight rows (i.e., constraints) of the regression problem and to let $\tilde{\mathbf{x}}$ be the minimizer of this sampled problem, which only involves a fraction of the entries in $\mathbf{b}$. I.e., letting $\mathbf{S} \in \mathbb{R}^{m\times n}$ be a sampling matrix with $m < n$ rows ($\mathbf{S}$ has one non-zero entry per row), set $\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{SAx}-\mathbf{Sb}\|$. When constraints are selected with probability proportional to the *statistical leverage scores* of $\mathbf{A}$'s rows, Problem I.1 can be solved with $O(d/\epsilon \cdot \log d)$

Cameron Musco's work on this project was supported in part by NSF Grants 2046235 and 1763618, along with an Adobe Research Grant. Christopher Musco was supported by NSF Grant 2045590. David P. Woodruff and Taisuke Yasuda were supported by ONR grant N00014-18-1-2562 and a Simons Investigator Award.

[1]Extended abstract; full version available at https://arxiv.org/abs/2111.04888.

[2]Our work will also extend to other loss functions of the form $\sum_{i=1}^n M([\mathbf{Ax}-\mathbf{b}]_i)$ that are not necessarily norms.

[3]In principal, entries of $\mathbf{b}$ can be read *adaptively* – i.e., we can select indices to query based on the results of other queries. However, the benefits of adaptivity appear limited. Most methods for solving Problem I.1 and those studied in this paper are non-adaptive.

978-1-6654-5519-0/22/$31.00 ©2022 IEEE
DOI 10.1109/FOCS54457.2022.00076

samples, and thus $O(d/\epsilon \cdot \log d)$ queries to $\mathbf{b}$ [21], [46], [53].[4] Using tools from spectral graph sparsification [6], [35], Chen and Price recently improved the leverage score sampling result to $O(d/\epsilon)$, which is optimal [12].

In practice, methods based on leverage score sampling (also known as "coherence motivated sampling") have found many applications. They are widely used in high-dimensional function fitting problems arising in the solution of parametric partial differential equations, where even mild assumptions on $\mathbf{A}$ and $\mathbf{b}$ are undesirable [16], [17], [30]. Methods for solving Problem I.1 in the ℓ_2 norm also yield robust methods for interpolating sparse Fourier functions, bandlimited and multiband functions, and for data-efficient kernel learning [4], [11], [23].

Other Norms. Much less was known about Problem I.1 beyond the ℓ_2 norm until recent work of Chen and Dereziński [10], which proves an upper bound of $O(d/\epsilon^2 \cdot \log d)$ queries for the ℓ_1 norm, $\|\mathbf{w}\|_1 = \sum_{i=1}^n |\mathbf{w}_i|$. This result is tight up to the $\log d$ factor. A similar result is obtained in [43]. Chen and Dereziński also prove a result for ℓ_p norms, $\|\mathbf{w}\|_p = (\sum_{i=1}^n |\mathbf{w}_i|^p)^{1/p}$, for $p \in (1,2)$, in which they show that $O(d^2/\epsilon^2 \cdot \log d)$ queries suffice to solve Problem I.1. As for the ℓ_2 norm, the results for ℓ_1 and ℓ_p are obtained by subsampling rows of the regression problem independently at random. However, instead of sampling with probabilities proportional to the leverage scores, [10], [43] employ a natural generalization of these scores known as the ℓ_p *Lewis weights* [18]. They left open the question of whether a linear in d dependence is possible for $1 < p < 2$, and any bounds at all for $p > 2$.

Beyond norms, if $\mathbf{b}$ is a $\{-1, 1\}$ label vector, and the error is measured via the logistic loss, Munteanu et al. [42] show that $\mathrm{poly}(d, \mu, 1/\epsilon)$ samples suffice, where μ is a complexity measure of $\mathbf{A}$. This bound has recently been tightened to $\tilde{O}(d\mu^2/\epsilon^2)$ [40], using Lewis weight sampling.[5] For other loss functions, such as the Tukey loss and Huber's M-estimators for robust regression [27], we are not aware of any known results solving Problem I.1. Chen and Dereziński also pose the open question of obtaining active regression bounds for other loss functions, in particular the Tukey and Huber losses, which are important in practice.

B. Our Contributions

a) ℓ_p Active Regression.: Our first main result is a new algorithm for solving Problem I.1 for the ℓ_p norm for any $0 < p < \infty$[6]. While near-optimal bounds are known for $p \in \{1, 2\}$ [10], [12], [43], the problem is far from settled for all other p. Previously, active ℓ_p regression for $p > 2$ and $0 < p < 1$ had no known nontrivial algorithms with $(1+\epsilon)$ relative error, and the only known approach was to read all n entries of $\mathbf{b}$ and solve the problem using offline results. A natural question is whether a sublinear query complexity is possible in these regimes. For $p \in (1, 2)$, [10] achieved an algorithm making $O(d^2/\epsilon^2 \cdot \log d)$ queries, thus achieving the first sublinear query complexity. One of their main open questions is whether the dependence on d can be improved to linear or not. Our main result answers all of these questions.

Theorem I.2 (Main Result for Active ℓ_p Regression)**.** *Given $0 < p < \infty$, $\mathbf{A} \in \mathbb{R}^{n \times d}$, and query access to $\mathbf{b} \in \mathbb{R}^n$, there is an algorithm that solves Problem I.1 for the ℓ_p-norm with probability $99/100$ which makes m queries in $\mathbf{b}$, where*

$$m = \begin{cases} O\left(\frac{d}{\epsilon^2}(\log d)^2(\log(d/\epsilon))\right) & p \in (0,1) \\ O\left(\frac{d}{\epsilon}(\log d)^2(\log(d/\epsilon))\right) & p \in (1,2) \\ O\left(\frac{d^{p/2}}{\epsilon^p}(\log d)^2(\log(d/\epsilon))^{p-1}\right) & p \in (2,\infty) \end{cases}.$$

We complement our algorithmic result with various new lower bounds which show the tightness of our algorithm. For $p \in (0, 2)$, our dependence on d and ϵ in the query complexity are simultaneously tight up to polylogarithmic factors; we show an $\Omega(d/\epsilon^2)$ lower bound for $p \in (0, 1)$ and an $\Omega(d/\epsilon)$ lower bound for $p \in (1, 2)$. For $p > 2$, our dependence on d is tight due to a lower bound of $\Omega(d^{p/2})$ which we show, while our ϵ dependence is off by at most factor of ϵ due to an $\Omega(\epsilon^{1-p})$ lower bound for the one-dimensional ℓ_p *power means* problem in Theorem 3 of [19]. Note that our active regression lower bounds for $p \in (0, 2)$ improve this previous power means lower bound.

Notably, we achieve a linear dependence on ϵ for $p \in (1, 2)$, which is perhaps surprising given that all previous known approaches to dimension reduction for ℓ_p regression relied on preserving the ℓ_p norm of all vectors in a subspace up to $(1 \pm \epsilon)$ factors [18], which requires $\Omega(d/\epsilon^2)$ dimensions [37]. It also demonstrates a separation in the query complexity for $p \leq 1$ and $1 < p < 2$, due to a lower bound of $\Omega(d/\epsilon^2)$ for $p = 1$ [10], [43] as well as for $p \in (0, 1)$ which we show.

Note that Theorem I.2 is stated to solve Problem I.1 with constant probability, $99/100$. In general, we show how to obtain $1 - \delta$ probability with dependence on δ that is only polylogarithmic in $1/\delta$. In fact, we show that any algorithm that simply samples rows of the regression problem and solves the sampled problem must suffer a $1/\delta^{p-1}$ dependence. Indeed, such a loss is seen in the algorithm of [10] for $p \in (1, 2)$. Thus, a success probability boosting routine, as we give in our work, is required to obtain an $O(\log(1/\delta))$ dependence.

b) Sensitivity Bounds and Active Regression for General Losses.: We show that our approach to solving Problem I.1 for ℓ_p norms generalizes to a broad class of loss functions known as M*-estimators* [15], which take the form $\sum_{i=1}^n M([\mathbf{A}\mathbf{x} - \mathbf{b}]_i)$. The only properties that we require are that we can (1) compute a constant factor approximation to Problem I.1 (2) the loss function obeys approximate variants of the triangle

[4] All query complexity bounds in this section are stated for solving Problem I.1 with high constant probability – e.g., probability $99/100$. In later sections we will include an explicit dependence on a failure probability δ.

[5] Throughout, $\tilde{O}$ is used to suppress polylogarithmic factors in the argument.

[6] Note that for $p \in (0, 1)$, $\|\cdot\|_p$ is not a norm, but we refer to it as a norm by a standard abuse of notation.

TABLE I: Upper and lower bounds for Problem I.1 for various norms and loss functions. New results are highlighted in blue. For simplicity, we suppress leading constants depending only on p, as well as $\operatorname{poly}\log n$ factors for M-estimator results. Our results significantly strengthen and generalize prior work, providing the first query complexity result with a tight d dependence for ℓ_p norms. We also give the first results for M-estimators as well as ℓ_p norms for $p > 2$ and $p \in (0, 1)$, and matching lower bounds in many cases.

Loss	Prior Work	Our Work	Lower Bound
$p = 2$	d/ϵ [12]	–	d/ϵ [12]
$p = 1$	d/ϵ^2 [10]	–	d/ϵ^2 [10]
$p \in (1, 2)$	d^2/ϵ^2 [10]	d/ϵ	d/ϵ
$p > 2$	–	$d^{\frac{p}{2}}/\epsilon^p$	$d^{\frac{p}{2}} + \epsilon^{1-p}$
$p \in (0, 1)$	–	d/ϵ^2	d/ϵ^2
M-est.	–	$d^{\frac{p}{2}+O(1)}/\epsilon^c$	d
Huber	–	$d^{4-2\sqrt{2}}/\epsilon^c$	d
Tukey	–	$d^{\frac{p}{2}+O(1)}/\epsilon^c$	d

inequality and (3) we can bound the so-called *sensitivities* of the loss, which bound the fraction of the total loss that can be concentrated at any coordinate $i \in [n]$ (see Equation (1)).

To the best of our knowledge, the only prior result achieving sensitivity bounds for general loss functions is [52]. However, this work makes use of Löwner-John ellipsoids, which leads to practically inefficient algorithms, and loses a factor of $\sqrt{d}$ in the total sensitivity due to the ellipsoidal rounding. As our second main result, we develop new sensitivity bounds for M-estimators that significantly simplify and improve this result.

Theorem I.3 (Main Result for Sensitivity Bounds)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and let M be an M-estimator loss with at most degree p growth. There is an algorithm which, with probability at least $99/100$, computes M-sensitivity upper bounds which sum to at most $O(d^{1\vee(p/2)}\log^2 n + \tau)$[7] in time at most $\tilde{O}(\operatorname{nnz}(\mathbf{A}) + nd^{O(1)}/\tau)$.*

Our approach to sensitivity bounds only relies on hashing and the computation of ℓ_p Lewis weights [18], [24], and avoids the computation of Löwner-John ellipsoids. This allows for input sparsity time algorithms, and answers an open question of [52] on avoiding Löwner–John ellipsoids in the computation of sensitivities. Note that our dependence on d matches the sensitivity bounds for the ℓ_p loss and is thus tight. We also show that the dependence on n is necessary for loss functions such as the Huber and Tukey losses. Furthermore, our algorithm can be turned into a non-algorithmic proof that the sensitivities sum to at most $O(d^{1\vee(p/2)}\log n)$ for these M-estimators; this is in fact tight for the Tukey loss by our lower bound of $\Omega(d\log n)$. Thus, we obtain the first tight bounds on the sum of sensitivities, for losses other than ℓ_p. Overall, we make significant progress on generalizing the theory of matrix approximation beyond ℓ_p losses to handle general M-estimators, which is a direction that has recently received much attention [13]–[15], [26], [51], [52].

Combined with our active regression techniques, our sensitivity bounds yield active regression algorithms for general loss functions, including the Huber and Tukey losses, answering an open question of Chen and Dereziński [10]. Note that prior to our work, no sublinear query complexity was known for any M-estimator regression, besides the ℓ_2 and ℓ_1 losses.

Furthermore, our new sensitivity bounds imply significant improvements in previous results using sensitivity sampling, beyond active regression, including Orlicz norm subspace embeddings [50] and robust subspace approximation [14]. We believe that our general technique here will find other further applications, and leave it as an open question to do so.

c) Subspace Embeddings for Orlicz Norms.: Orlicz norms can be viewed as scale-invariant extensions of M-estimators, and have recently attracted attention as a general class of norms that admit efficient dimensionality reduction results [3], [50]. In particular, [50] apply sensitivity sampling to obtain *subspace embeddings* for Orlicz norms, which yields a small weighted subset $\tilde{\mathbf{A}}$ of rows of a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|$[8] for all $\mathbf{x} \in \mathbb{R}^d$. However, the number of rows required by [50] is a large polynomial in d, and is also restricted to Orlicz norms of at most quadratic growth. We show that by applying our new sensitivity bounds, we can obtain subspace embeddings for Orlicz norms with $d^{2\vee(p/2+1)}\operatorname{poly}(\log n, \epsilon^{-1})$ rows, for any Orlicz norm with a polynomial growth bound of degree p.

d) Robust Subspace Approximation.: The robust subspace approximation problem generalizes the classical low rank approximation problem of finding a rank k projection $\mathbf{X}$ minimizing $\|\mathbf{A}\mathbf{X} - \mathbf{A}\|_F$ by replacing the Frobenius norm with an extension of M-estimators to matrix norms. [14] showed the first dimensionality reduction results for this problem for a general class of M-estimators of at most quadratic growth via a recursive sampling scheme using the sensitivity sampling framework. However, due to the use of looser sensitivity bounds, they suffer an undesirable factor of $(\log n)^{O(\log k)}$ in their sample complexities. Our new sensitivity bounds allow us to remove this factor, giving a dimension reduction result into a $\operatorname{poly}(k, \log n, \epsilon^{-1}) \times \operatorname{poly}(k, \log n, \epsilon^{-1})$ instance. We also extend their method beyond quadratic growth, to any degree p polynomial growth.

e) Active Regression for the Huber Loss.: Our active regression result for general M-estimators discussed above is loose by a factor of d in the sample complexity, compared to our ℓ_p active regression results. This is attributed to the use of our net argument for general M-estimators, whereas our ℓ_p active regression results can make use of more sophisticated chaining arguments of [7], [34], [48]. A natural question is if this gap can be improved.

We consider the important special case of the Huber loss, which is defined as follows:

[7]Here, $a \vee b$ denotes $\max(a, b)$, and $a \wedge b$ denotes $\min(a, b)$.

[8]For $a, b \geq 0$, $a \pm b$ denotes a number c such that $a - b \leq c \leq a + b$.

Definition I.4 (Huber loss [31])**.** *The Huber loss of width $\tau \geq 0$ is defined as*

$$H(x) \coloneqq \begin{cases} x^2/2\tau & \text{if } |x| \leq \tau \\ |x| - \tau/2 & \text{otherwise} \end{cases}$$

and the Huber norm[9] *is defined as $\|\mathbf{y}\|_H \coloneqq \sqrt{\sum_{i=1}^n H(\mathbf{y}(i))}$.*

The Huber loss is "arguably one of the most widely used M-estimators" [15], owing its popularity to its convexity and differentiability properties of ℓ_2, which allows for efficient algorithms (see, e.g., [41] for algorithms), in combination with its robustness properties of ℓ_1 [29]. This makes it widely applicable in practical big data settings (see, e.g., [5] for a list of popular software packages implementing Huber regression as well as references that make use of Huber regression). Variations on Huber regression have also recently been shown to hold theoretical guarantees in the robust statistics literature (see, e.g., [38], [39] and references therein).

For the Huber loss, we show that it is indeed possible to leverage the chaining techniques in order to obtain improved sample complexity bounds for active regression. We show that we can improve beyond the d^2 bound obtained by our general M-estimator algorithm as applied to the Huber loss, and obtain a sample complexity of $O(d^{4-2\sqrt{2}}\operatorname{poly}(\log n, \epsilon^{-1}))$ queries to $\mathbf{b}$, where $4 - 2\sqrt{2} \approx 1.17157$. For this result, we use the chaining techniques of [7], which provides a more flexible alternative to [34], but requires more technical effort to adapt to the active setting.

Theorem I.5 (Main Result for Huber Active Regression)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{b} \in \mathbb{R}^n$. There is an algorithm which, with probability at least $99/100$, returns a $\tilde{\mathbf{x}}$ satisfying*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_H \leq (1+\epsilon) \cdot \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_H$$

Furthermore, the algorithm reads at most $d^{4-2\sqrt{2}}\operatorname{poly}(\log n, \epsilon^{-1})$ entries of $\mathbf{b}$.

Our techniques also yield a subspace embedding result, which constructs a weighted subset $\tilde{\mathbf{A}}$ of $O(d^{4-2\sqrt{2}}\operatorname{poly}(\log n, \epsilon^{-1}))$ rows such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_H = (1 \pm \epsilon)\|\tilde{\mathbf{A}}\mathbf{x}\|_H$ for all $\mathbf{x} \in \mathbb{R}^d$. Previously, the best known dimension reduction bound for Huber regression, even in the non-active setting, was d^4 due to [15].

Furthermore, this is, to the best of our knowledge, the first example of a loss function other than ℓ_p which achieves a sensitivity sampling bound of better than d^2, despite the fact that such results have been sought in many works [13]–[15], [28], [50], [52]. The reason for this is that d^2 is a natural bound for sensitivity sampling, attributed to one d factor from the sum of sensitivities and one d factor from carrying out a union bound over a net of $\exp(d)$ vectors. For ℓ_p norms, the arguments of [7], [48] and their subsequent improvements avoid this problem by using a more sophisticated chaining argument. However, these arguments use the structure of ℓ_p spaces in crucial ways, such as isometric changes of density using Lewis weights [32], and do not generalize easily to other loss functions.

It is an interesting open question to determine whether our dimension reduction bound for the Huber loss can be improved all the way down to d.

f) Dimension Reduction for Gamma Functions for Faster ℓ_p Regression.: One particularly important application of sampling-based dimension reduction for loss functions beyond ℓ_p losses is, perhaps surprisingly, in the design of algorithms for ℓ_p regression. The work of [8] introduces *gamma functions* γ_p, which are generalizations of the Huber loss which behave quadratically near the origin and like $|x|^p$ away from the origin, in the context of algorithms for ℓ_p regression. Subsequently, [2] obtained even faster algorithms by using constant factor approximations of γ_p regression as a subroutine, in which the γ_p loss is minimized over a subspace. Dimension reduction for this loss function has been a crucial ingredient for recent results in fast algorithms for ℓ_p regression [1], [28]. In particular, [1] highlighted the open question of designing sparsification methods for γ_p functions for $p \in (1, 2)$, and [28] designed a sampling algorithm which samples $\tilde{O}(d^3)$ rows. By generalizing our dimension reduction techniques for the Huber loss, we obtain an algorithm which samples at most $O(d^{4-2\sqrt{2}}\operatorname{poly}(\log n, \epsilon^{-1}))$ rows for any $p \in [1, 2)$, and improves to $O(d\operatorname{poly}(\log n, \epsilon^{-1}))$ rows as $p \to 2$ (see Figure 1 for the trade-off curve).

g) Kronecker Product Regression.: Beyond applications in data-efficient regression, Theorem I.2 implies the first sublinear time algorithm for Kronecker product regression in any ℓ_p norm, where explicitly constructing the vector $\mathbf{b}$ is a computational bottleneck. In q-th order Kronecker product regression, one is given matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_q$, where $\mathbf{A}_i \in \mathbb{R}^{n_i \times d_i}$, as well as a vector $\mathbf{b} \in \mathbb{R}^{n_1 n_2 \cdots n_q}$, and the goal is to solve: $\min_{\mathbf{x}\in\mathbb{R}^{d_1 d_2 \cdots d_q}} \|(\mathbf{A}_1 \otimes \mathbf{A}_2 \cdots \otimes \mathbf{A}_q)\mathbf{x} - \mathbf{b}\|_p$, where $\otimes$ denotes the Kronecker product. Typically $\prod_{i=1}^q d_i$ is much less than $\prod_{i=1}^q n_i$, and the goal is to obtain algorithms that do not explicitly form $\mathbf{A}_1 \otimes \mathbf{A}_2 \otimes \cdots \otimes \mathbf{A}_q$ or $\mathbf{b}$, which is too expensive. Our results yield the first algorithm for Kronecker product regression, for every $p \geq 1$, whose running time *does not depend on* $\operatorname{nnz}(\mathbf{b})$, whereas previous results had a linear dependence on $\operatorname{nnz}(\mathbf{b})$, which can be as large as $\prod_{i=1}^q n_i$ [22].

Theorem I.6. *Let $q \geq 1$, $p \geq 1$ be constant, and $\epsilon > 0$. Kronecker product regression can be solved up to a $(1+\epsilon)$-factor with constant probability in $\tilde{O}(\sum_{i=1}^q \operatorname{nnz}(\mathbf{A}_i) + \operatorname{poly}(\prod_{i=1}^q d_i/\epsilon))$ time.*

C. Technical Approach

1) ℓ_p Active Regression: Our algorithm for solving Problem I.1 uses a novel variation on the "sample-and-solve" approach. In particular, we randomly select a row sampling matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ and return $\tilde{\mathbf{x}} = \arg\min_x \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|$, which only requires querying m entries of $\mathbf{b}$ (those that appear in $\mathbf{S}\mathbf{b}$). To get tight bounds for ℓ_p regression, we select $\mathbf{S}$ using ℓ_p Lewis weight sampling, a generalization of leverage score sampling for ℓ_2.

[9] Again, this is a standard abuse of notation, and the Huber norm is not an actual norm.

It can be shown that the ℓ_p Lewis weights upper bound the ℓ_p *sensitivities* of $\mathbf{A}$, a measure of importance for the rows of $\mathbf{A}$. The ℓ_p sensitivity of the i^{th} row of $\mathbf{A}$ is defined as

$$\mathbf{s}_i^p(\mathbf{A}) \coloneqq \max_{\mathbf{x}\in\mathbb{R}^d\setminus\{0\}} \frac{|[\mathbf{A}\mathbf{x}](i)|^p}{\|\mathbf{A}\mathbf{x}\|_p^p},$$

where $[\mathbf{A}\mathbf{x}](i)$ denotes the i^{th} entry of the vector $\mathbf{A}\mathbf{x}$, and captures how large the i^{th} entry of any $\mathbf{A}\mathbf{x} \in \text{span}(\mathbf{A})$ can be, relative to the ℓ_p norm. A standard scalar Bernstein bound shows that if $\mathbf{S}$ samples rows with probabilities that upper bound the sensitivities, then $\|\mathbf{S}\mathbf{A}\mathbf{x}\|_p^p = (1\pm\epsilon)\|\mathbf{A}\mathbf{x}\|_p^p$ with high probability, for each $\mathbf{x} \in \mathbb{R}^d$. An ϵ-net argument can extend this to a *for all* claim.

a) Prior Approaches to ℓ_p Active Regression: While the above ideas give an approach for standard ℓ_p regression, this bound does not suffice for *active* ℓ_p regression. To solve Problem I.1, we actually want that $\|\mathbf{S}(\mathbf{A}\mathbf{x}-\mathbf{b})\|_p^p = (1\pm\epsilon)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p^p$ for any $\mathbf{x}$. Will $\mathbf{S}$ provide such a guarantee? The main problem, as discussed in [10], [43] is that the translation by $\mathbf{b}$ may introduce outliers, i.e., entries with high sensitivity which are not captured by the sensitivity scores of $\mathbf{A}$. As shown by [10], [43], in the case of ℓ_1, the special structure of the loss function provides a solution. Indeed, by the triangle inequality,

$$|(|[\mathbf{A}\mathbf{x}-\mathbf{b}](i)| - |[\mathbf{A}\mathbf{x}^*-\mathbf{b}](i)|)| \le |[\mathbf{A}(\mathbf{x}-\mathbf{x}^*)](i)|$$

where $\mathbf{x}^*$ is the optimal solution. This fact can be used to show that sampling by the sensitivities of $\mathbf{A}$ preserves the *differences* between the cost of any $\mathbf{x}$ and the optimal $\mathbf{x}^*$. However, such a proof cannot work for $p \neq 1$, in which case we do not have such a nice inequality. For $p \in (1,2)$, [10] take the approach of bounding the residual error terms from the above approach by using a Taylor approximation, but this leads to a sample complexity of at least d^2.

b) Our Solution: Partitions by Sensitivity: Instead of relying on the technique of "cancelling out the outliers", we take a conceptually different approach. We proceed in two stages, where we (1) first find a constant factor solution $\mathbf{x}_c$ such that $\|\mathbf{A}\mathbf{x}_c-\mathbf{b}\|_p^p \le O(1)\cdot\min_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p^p$ using an idea of [20] and replace $\mathbf{b}$ by the residual vector $\mathbf{b}-\mathbf{A}\mathbf{x}_c$, and then (2) conceptually partition the target vector $\mathbf{b}$ into two sets of coordinates, the coordinates $i \in [n]$ that are small enough to be comparable to the sensitivity $\mathbf{s}_i^p(\mathbf{A})$ and those that are much larger. That is, we consider the coordinates $i \in [n]$ such that $|\mathbf{b}(i)|^p/\|\mathbf{b}\|_p^p \le C\cdot\mathbf{s}_i^p(\mathbf{A})$ for some $C > 0$, and all other coordinates. For the former set of coordinates, one can check that the Bernstein bound still applies, and $\mathbf{S}$ *does* preserve the norm of $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p^p$, when restricted to these coordinates. On the other hand, for the latter set of coordinates, we show that *no vector* of the form $\mathbf{A}\mathbf{x}$ can both be close to $\mathbf{b}(i)$ in its i^{th} entry, and still close to the remainder of $\mathbf{b}$ – the i^{th} entry is simply too large in magnitude. In particular, to have $[\mathbf{A}\mathbf{x}](i)$ close to $\mathbf{b}(i)$, we would require $\|\mathbf{A}\mathbf{x}\|_p^p$ to be much larger than $\|\mathbf{b}\|_p^p$, which by our preprocessing step, is on the order of the optimal cost $\min_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p$. Via the triangle inequality, this implies that $\mathbf{A}\mathbf{x}$ must be far from an optimal solution. Thus, we can argue that any near-optimal solution to $\min_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p$ *does not need to fit* $\mathbf{b}(i)$ *with* $|\mathbf{b}(i)|^p/\|\mathbf{b}\|_p^p$ *much larger than* $\mathbf{s}_i^p(\mathbf{A})$. We can effectively ignore the contribution of these rows.

Another technical challenge remains: to obtain an optimal dimension dependence, we need a refined ϵ-net argument to make *for all* statements about $\mathbf{x} \in \mathbb{R}^d$. To do so, we adapt the chaining arguments of Bourgain, Lindenstrauss, and Milman [7] and Ledoux and Talagrand [34] to the active regression setting, avoiding the standard ϵ-net and union bound argument used by, e.g., [47]. Although both [7] and [34] provide such approaches, we adapt the (slightly) more complex recursive Lewis weight sampling algorithm of [34] in order to obtain tighter dependencies on ϵ. The streamlined proof of [34] also adapts nicely to the active regression setting with minimal changes to the original argument. We note here that we will later also need to adapt the much more involved [7] argument to handle the Huber loss, in which case the proof of [7] allows for more fine-grained control over bounding the sensitivity sampling algorithm, but requires a more complex argument based on carefully partitioning the coordinates of the target vector $\mathbf{b}$ based on sensitivity weight classes. Aside from our new application of [7], [34], we hope that by translating the arguments of [7], [34] to the language of theoretical computer science and matrix approximation, they will find further applications to randomized algorithm design.

We note that our algorithm is quite a bit more involved than a simple scheme of sampling proportionally to Lewis weights and solving. This is for good reasons. Not only is it not clear that such an approach works at all, we show that for any $p > 1$, any algorithm which simply samples reweighted rows and solves the system must have a polynomial dependence on $1/\delta$ in the query complexity, while our algorithm achieves a $\log\frac{1}{\delta}$ dependence, by solving residual problems of a constant factor solution. Thus, our two-stage approach is necessary to achieve our δ dependence. Furthermore, the best known analysis of a simple "one-shot" Lewis weight sampling scheme suffers in ϵ dependencies for $p > 2$, where the one-shot approach is only known to give a $\tilde{O}(d^{p/2}/\epsilon^5)$ bound for subspace embeddings [7], [18], whose losses translate to losses for our active regression algorithms as well, while the recursive approach can achieve $\tilde{O}(d^{p/2}/\epsilon^2)$ [34]. While the [34] result is an existential result, we provide an analysis of the [34] proof to turn it into a randomized algorithm with logarithmic dependencies on the failure rate δ, which achieves the best known dependence on d, ϵ, and δ, up to logarithmic factors. We further modify this subspace embedding result for active regression, to optimize our ϵ dependence.

c) Optimized ϵ Dependence for $p \in (1,2)$: For $p \in (1,2)$, the above argument gives a bound of $\tilde{O}(d/\epsilon^2)$. While the linear dependence on d is optimal, confirming the conjecture of [10], it has a quadratic dependence on ϵ, which is in fact *not* optimal. We now show how to improve our bound to $\tilde{O}(d/\epsilon)$, which requires additional ideas. We first use strong convexity to show that a $(1+\gamma)$-approximate solution $\hat{\mathbf{x}} \in \mathbb{R}^d$ satisfying $\|\mathbf{A}\hat{\mathbf{x}}-\mathbf{b}\|_p^p \le (1+\gamma)\|\mathbf{A}\mathbf{x}^*-\mathbf{b}\|_p^p$, for the

optimal solution $\mathbf{x}^* \in \mathbb{R}^d$, in fact satisfies $\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{A}\mathbf{x}^*\|_p \leq O(\sqrt{\gamma})\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_p$. Then, by using that $\hat{\mathbf{x}}$ is close to the optimal solution, we show an improved bound on the difference in the objective values of $\hat{\mathbf{x}}$ and $\mathbf{x}^*$, i.e., that $\hat{\mathbf{x}}$ actually has an approximation ratio better than $(1+\gamma)$. We then iterate this argument until we obtain a $(1+\epsilon)$-approximation using $\tilde{O}(d/\epsilon)$ queries, at which point we can no longer get improvements. The chaining argument used in this proof, while similar to the previous proofs, has a different geometry than the previous chaining arguments, and requires additional ideas.

Our upper bound is tight up to polylogarithmic factors due to a lower bound we show. This improves an ℓ_p power means lower bound of [19], who only showed a lower bound of $\Omega(\epsilon^{1-p})$ queries. Unfortunately, we are unable to port our algorithmic techniques to the ℓ_p power means problem in high dimensions, due to difficulties in adapting their chaining argument.

2) Sensitivity Bounds: The notion of ℓ_p sensitivities, as discussed above, naturally generalizes to loss functions that take the form of coordinate-wise sums. Consider a loss function M and an $n \times d$ matrix $\mathbf{A}$. Then, the *sensitivity* of the ith coordinate with respect to the loss function M is defined as

$$\mathbf{s}_i^M(\mathbf{A}) \coloneqq \sup_{\mathbf{x}\in\mathbb{R}^d\setminus\{0\}} \frac{M([\mathbf{A}\mathbf{x}](i))}{\sum_{j=1}^n M([\mathbf{A}\mathbf{x}](j))}. \tag{1}$$

It is well-established that sensitivities provide a general framework for sampling rows of $\mathbf{A}$ that approximate $\mathbf{A}$ well under the loss function M [25]. While a rich literature exists for ℓ_p [18], [20], [49], little was known about the approximation of sensitivities for general loss functions until [52], which used Löwner-John ellipsoids to obtain sensitivity bounds for a general family of near-convex losses. However, the computation of Löwner-John ellipsoids has running time that is a large polynomial in n and d, and is impractical for large datasets, and [52] raise the open question of obtaining general sensitivity bounds without this expensive subroutine.

Our approach to new sensitivity bounds significantly generalizes the approach of [13], whose algorithm can be seen as a way to use hashing and Lewis weights to compute sensitivities for the Tukey loss, but heavily uses the properties of the Tukey loss in their analysis.

Suppose that a coordinate $i \in [n]$ has M-sensitivity $\alpha \in (0, 1]$, that is,

$$\mathbf{s}_i^M(\mathbf{A}) = \sup_{\mathbf{x}\in\mathbb{R}^d\setminus\{0\}} \frac{M([\mathbf{A}\mathbf{x}](i))}{\sum_{j=1}^n M([\mathbf{A}\mathbf{x}](j))} = \alpha,$$

and let $\mathbf{y} = \mathbf{A}\mathbf{x}$ witness this supremum, and assume for simplicity that $\sum_{i=1}^n M([\mathbf{A}\mathbf{x}](j)) = 1$. Note then that there can be at most $1/\alpha$ entries $j \in [n]$ of $\mathbf{y}$ that have coordinate value $M(\mathbf{y}_j) \geq M(\mathbf{y}_i) = \alpha$. Then, if we randomly hash the n coordinates into $O(1/\alpha)$ buckets, then with constant probability, coordinate i will be isolated from any other entry with $M(\mathbf{y}_j) \geq M(\mathbf{y}_i) = \alpha$. Now if M is monotonic, then this means that $\mathbf{y}_i$ is the largest coordinate in its hash bucket. Furthermore, the sum of the M-mass of all of the other coordinates in i's hash bucket is only an α fraction of the total M-mass, so entry i carries a constant fraction of the M-mass in its bucket. In this case, it can be shown that entry i must in fact carry a constant fraction of the ℓ_2 *mass* inside its hash bucket, if M is a function of at most quadratic growth. This is because when we switch the error metric from M to ℓ_2, then the largest entry will have the largest increase in its normalized contribution. This means that row i must have an ℓ_2 leverage score of $\Omega(1)$, in this hash bucket.

This leads to the following algorithm: (1) hash the n coordinates into $O(1/\alpha)$ buckets (2) compute ℓ_2 leverage scores for each bucket (3) assign an M-sensitivity of α for any coordinate that has ℓ_2 leverage score $\Omega(1)$. In each of the $O(1/\alpha)$ buckets, we will find at most $O(d)$ coordinates with leverage score at least $\Omega(1)$, so we assign an M-sensitivity of α to at most $O(d/\alpha)$ coordinates, which has a total sensitivity contribution of $O(d)$. By repeating this for $O(\log n)$ guesses of α in powers of 2, this gives a total sensitivity bound of $O(d \log n)$. The constant probability events in the hashing process can be boosted to probability $1 - 1/\operatorname{poly}(n)$ by repeating the procedure $O(\log n)$ times, which increases the total sensitivity to roughly $O(d \log^2 n)$.

By sampling according to these sensitivities and applying a union bound over a net, we obtain the first active regression algorithms for general loss functions. Note that this result is made possible by a combination of both our new sensitivity bounds for M-estimators and our new active regression techniques as discussed in Section I-C1. Furthermore, we demonstrate other applications of our sensitivity bound result, showing how to improve Orlicz norm subspace embeddings and robust subspace approximation.

3) Subspace Embeddings and Active Regression for the Huber Loss: As discussed previously, we tackle the question of leveraging the theory of [7] nets in order to obtain sample complexities for the Huber loss beyond d^2. Our algorithmic framework for active regression is based on the earlier idea of partitioning the entries of $\mathbf{b}$ by sensitivity and then applying sensitivity sampling, so we focus on the problem of preserving the Huber norm using an improved sensitivity sampling technique. Note that unlike the ℓ_p losses, the Huber loss is not scale-invariant. Furthermore, perhaps the largest obstacle in designing row sampling algorithms for the Huber loss going beyond standard ϵ-net arguments is that there is no analogue of the chaining constructions of [7], [48] for the Huber loss. This can also be attributed to the fact that the Huber loss is not scale-invariant, which precludes an isometric change-of-density type theorem for the Huber loss as done in [36], [48]. We show how to overcome these obstacles in the following discussion.

a) A Sharp Huber Inequality.: Our algorithmic framework follows the Huber algorithm of [15], which is a recursive sampling algorithm which reduces the number of rows from n to roughly $n^{1/2}d^2$ in each recursive application of the algorithm. To show this result, [15] first show in their Lemma 2.1 that the Huber norm is within a factor of $O(n^{1/2})$ of the smaller of the ℓ_1 and ℓ_2 norms:

Lemma I.7 (Huber Inequality version 1 ([15], Lemma 2.1))**.** *Let* $\mathbf{y} \in \mathbb{R}^n$*. Then,*

$$\|\mathbf{y}\|_H^2 = \sum_{i=1}^{n} H(\mathbf{y}_i) \geq \Omega(n^{-1/2}) \min\{\|\mathbf{y}\|_1, \|\mathbf{y}\|_2^2\}$$

It can be shown that the above lemma implies that the Huber sensitivities are within a factor of $O(n^{1/2})$ of the sum of the ℓ_1 and ℓ_2 sensitivities. This motivates the idea of sampling the rows of $\mathbf{A}$ with probability proportional to the sum of the ℓ_1 and ℓ_2 Lewis weights, oversampled by a factor of $O(n^{1/2})$. This is indeed how [15] proceeds.

The recursion $n \to n^{1/2}d^2$ solves to a final row count of around d^4, which is quadratically worse than our general loss function result of d^2 using our new sensitivity upper bounds and our general framework. To improve this further, first note that two improvements can be made to the above argument. First, by using the Huber inequality in a different way, we can use it in conjunction with the [7] net bounds, which reduces the row count in one recursive application to roughly $n^{1/2}d$ rather than $n^{1/2}d^2$. This reduces the overall row count to d^2 after solving for the recursion, but this still does not beat our general purpose sensitivity sampling algorithm, despite the use of the [7] nets. The second improvement is that the Huber inequality as proved in [15] is in fact loose by a polynomial factor in n, and can be improved to the following:

Lemma I.8 (Huber Inequality version 2)**.** *Let* $\mathbf{y} \in \mathbb{R}^n$*. Then,*

$$\|\mathbf{y}\|_H^2 = \sum_{i=1}^{n} H(\mathbf{y}_i) \geq \Omega(n^{-1/3}) \min\{\|\mathbf{y}\|_1, \|\mathbf{y}\|_2^2\}$$

This lemma is tight up to constant factors[10], and gives a recursion of roughly $n \to n^{1/3}d$, giving

$$O(d^{3/2} \operatorname{poly}(\epsilon^{-1}, \log n))$$

rows, which shaves a factor of approximately $\sqrt{d}$ over the naïve Bernstein bound over a net.

b) Storing Large Huber Sensitivities.: In order to further improve upon this bound, we crucially make use of our improved sensitivity bounds and a generalized version of the above Huber inequality lemma that is parameterized by an upper bound on the size of the entries of $\mathbf{y}$.

Lemma I.9 (Huber Inequality version 3)**.** *Let* $\mathbf{y} \in \mathbb{R}^n$ *and let* $0 < \gamma \leq 1$*. Let*

$$T \supseteq \left\{ i \in [n] : H(\mathbf{y}_i) \leq \gamma \|\mathbf{y}\|_H^2 \right\}.$$

Then, for some constant $c > 0$*, at least one of the following bounds holds:*

$$\|\mathbf{y}\ |_T\|_H^2 = \sum_{i \in T} H(\mathbf{y}_i) \geq c \frac{1}{(\gamma n)^{1/3}} \min\{\|\mathbf{y}\ |_T\|_1, \|\mathbf{y}\ |_T\|_2^2\}$$

$$\|\mathbf{y}\|_H^2 = \sum_{i=1}^{n} H(\mathbf{y}_i) \geq c\gamma \min\{\|\mathbf{y}\|_1, \|\mathbf{y}\|_2^2\}.$$

[10]Consider the vector with one coordinate with $n^{1/3}$ and $(n-1)$ coordinates with $n^{-1/3}$.

By directly including the rows of $\mathbf{A}$ with Huber sensitivity at least γ, we exactly preserve the Huber norm inside $\overline{T} = [n] \setminus T$ for every $\mathbf{y}$. On the remaining coordinates inside T, we then have an improved Huber inequality, which implies an improved sampling bound. By balancing the number of rows which we directly include, which is roughly d/γ, and the sampling bound inside T, which is roughly $(1/\gamma + (\gamma n)^{1/3})d$, we obtain a bound of roughly $n^{1/4}d$ rows by choosing $\gamma = n^{-1/4}$ at each step. By recursively applying this result, we obtain an improved sampling bound of

$$O(d^{4/3} \operatorname{poly}(\epsilon^{-1}, \log n)).$$

c) Comparing to Every $p \in [1, 2]$*.:* Finally, to achieve our final optimization, we further drive down the ratio between $\|\mathbf{y}\|_H^2$ and $\|\mathbf{y}\|_p^p$ *by choosing the best* $p \in [1, 2]$ *for each* $\mathbf{y}$:

Lemma I.10 (Huber Inequality ver. 4)**.** *Let* $\mathbf{y} \in \mathbb{R}^n$*,* $\alpha \in [0, 1/2]$*,* $\gamma = n^{-\alpha}$ *with* $2/n \leq \gamma \leq 1$*. Let*

$$T \supseteq \left\{ i \in [n] : H(\mathbf{y}_i) \leq \gamma \|\mathbf{y}\|_H^2 \right\}.$$

Then, for some $c > 0$ *and* $\beta = 3 - 2\sqrt{2} \approx 0.17157$*, at least one of the following bounds holds:*

$$\|\mathbf{y}\ |_T\|_H^2 \geq c \frac{1}{(\gamma n)^{\beta}} \min_{p \in [1,2]} \|\mathbf{y}\ |_T\|_p^p$$

$$\|\mathbf{y}\|_H^2 \geq c\gamma \min_{p \in \{1,2\}} \|\mathbf{y}\|_p^p$$

In fact, we prove a generalized bound for the ℓ_2-ℓ_q loss for any $q \in (0, 2)$ in Lemma I.11. The interval $p \in [1, 2]$ can be discretized in increments of $\frac{1}{\log n}$, so with $O(\log n)$ applications of [7] nets, we can always find a p within an additive $\frac{1}{\log n}$ of the optimal p for every net vector $\mathbf{y}$, which only affects Lemma I.10 by constant factors whenever $\mathbf{y}$ has entries bounded by $\operatorname{poly}(n)$. By proceeding as previously discussed, we arrive at our final bound of

$$O(d^{4-2\sqrt{2}} \operatorname{poly}(\epsilon^{-1}, \log n)).$$

d) Extensions to ℓ_2-ℓ_q *Loss..:* We generalize our results to the ℓ_2-ℓ_q loss for $q \in (0, 2)$. As q ranges from 0 to 1 to 2, the ℓ_2-ℓ_q interpolates between the Tukey, Huber, and ℓ_2 losses up to constant factors, and provides a natural generalization of these loss function.

Lemma I.11 (ℓ_2-ℓ_q Inequality)**.** *Let* $q \in (0, 2)$ *and define*

$$M(x) = \begin{cases} |x|^2 & \text{if } |x| \leq 1 \\ |x|^q & \text{if } |x| > 1 \end{cases}.$$

Let $\mathbf{y} \in \mathbb{R}^n$ *and let* $\alpha \in [0, q/2]$ *and* $\gamma = n^{-\alpha}$ *with* $2/n \leq \gamma \leq 1$*. Let*

$$T \supseteq \left\{ i \in [n] : M(\mathbf{y}_i) \leq \gamma \|\mathbf{y}\|_M^2 \right\}.$$

Then, for some constant $c > 0$*, at least one of the following bounds holds:*

$$\begin{aligned} \|\mathbf{y}\|_M^2 &\geq c\gamma^{2/q-1} \min_{p \in \{q,2\}} \|\mathbf{y}\|_p^p \\ \|\mathbf{y}\ |_T\|_M^2 &\geq c \frac{1}{(\gamma n)^{\beta}} \min_{p \in [q,2]} \|\mathbf{y}\ |_T\|_p^p \end{aligned} \tag{2}$$

where

$$\beta = \frac{1}{2/q-1}\left[(2/q+1) - 2\sqrt{2/q}\right].$$

For $q \in [1,2)$, the $1/\gamma^{2/q-1}$ distortion in Equation (2) is smaller than the $1/\gamma$ factor incurred from keeping M-sensitivities at least γ, so we can balance the parameters as $1/\gamma = (\gamma n)^{\beta}$ as before, which leads to a recursion that gives us a bound of $n = d^{1+\beta}\operatorname{poly}((\log n)/\epsilon)$. For $q \in (0,1)$, the $1/\gamma^{2/q-1}$ distortion in Equation (2) is worse than $1/\gamma$, which means we must balance $1/\gamma^{2/q-1} = (\gamma n)^{\beta}$, or $\gamma = n^{-\beta/(2/q-1+\beta)}$, which gives a worse bound of $n = d^{\gamma}\operatorname{poly}((\log n)/\epsilon)$ for

$$\gamma = \frac{2/q-1+\beta}{2/q-1+\beta(2-2/q)}.$$

This is better than a d^2 bound as long as $q \geq (\sqrt{5}-1)^2/8 \approx 0.19098$.

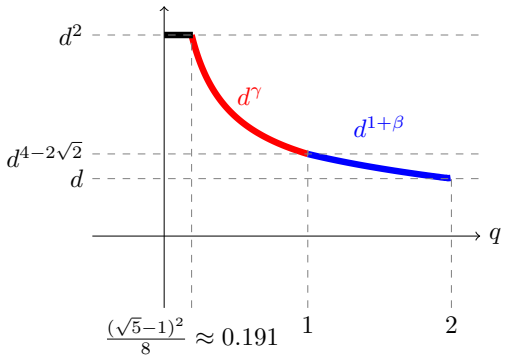

Fig. 1: Dependence on d for the active regression sample complexity for the ℓ_2-ℓ_q loss. Similar bounds apply to subspace embeddings as well.

D. Conclusions and Future Directions

In this work, we study the sample complexity of active linear regression for both the ℓ_p norm as well as general M-estimator losses.

For the ℓ_p norm, we provide optimal algorithms and lower bounds for $p \in (0,2)$, with $\tilde{\Theta}(d/\epsilon^2)$ samples for $p \in (0,1)$ and $\tilde{\Theta}(d/\epsilon)$ samples for $p \in (1,2)$. For $p > 2$, we provide an upper bound of $\tilde{O}(d^{p/2}/\epsilon^p)$, which is optimal in the d dependence and off by a single ϵ factor in the ϵ dependence, up to polylogarithmic factors. Our algorithms provide the first nontrivial bounds, i.e., sample complexity less than n, for $p \in (0,1) \cup (2,\infty)$, while for $p \in (1,2)$, we significantly improve upon the $\tilde{O}(d^2/\epsilon^2)$ upper bound of [10] and answer their main open question. We obtain these results via a two-stage algorithm and a novel sensitivity partitioning technique for every p, as well as an iterative improvement argument via strong convexity and Lewis bases to improve the ϵ dependence for $p \in (1,2)$. Our result is the first to achieve a linear dependence on ϵ for dimension reduction for ℓ_p regression for $p \in (1,2)$.

Next, we obtain a new sensitivity bound which achieves optimal total sensitivity bounds for M-estimators of at most polynomial growth, which runs in input sparsity time and avoids the use of Löwner–John ellipsoids. This answers an open question of [52] and makes significant progress in the general direction of matrix approximation beyond ℓ_p losses. By combining this with our new active regression techniques, we obtain active regression algorithms for general M-estimator losses, including the Tukey and Huber losses, which answers an open question of [10].

For the important special case of the Huber loss, we introduce new techniques which bound Huber sensitivities by the sum of ℓ_p Lewis weights, which allows us to take advantage of chaining arguments for ℓ_p in order to obtain an active regression algorithm making at most $O(d^{4-2\sqrt{2}}\operatorname{poly}(\log n, \epsilon^{-1}))$ queries. Our techniques also give subspace embeddings with the same number of rows. This is the first dimension reduction result for losses other than ℓ_p to approximate a d-dimensional subspace with fewer than d^2 dimensions. This improves over a previous bound of d^4 for the Huber loss, which held only for subspace embeddings, and not active regression, in [15].

Finally, our results and techniques give many applications in a wide variety of related problems. Our lower bounds for active regression give improved lower bounds for the sublinear power means problem [19]; our new sensitivity bounding techniques sharpen and generalize previous results on Orlicz norm subspace embeddings [51] and robust subspace approximation [14]; our techniques for dimension reduction for the Huber loss gives improved bounds for sparsification for γ_p functions for applications in fast algorithms for ℓ_p regression [1], [28]. We believe that our techniques will be applicable further, and hope to see more uses in future work.

We conclude with questions that are still left open by our work. Perhaps the most pressing is to resolve the query complexity of active ℓ_p regression for $p > 2$: our upper bound is $\tilde{O}(d^{p/2}/\epsilon^p)$, while the lower bound is $\Omega(d^{p/2}+\epsilon^{1-p})$. Closing this gap would be interesting. Our bounds are also loose by a factor of $\log\frac{1}{\delta}$ for all $p > 0$, while we can get an optimal dependence on δ if we assume knowledge of the optimal value and sacrifice a factor of ϵ. A natural question if one can achieve a simultaneously optimal dependence on d, ϵ, and δ, up to logarithmic factors, and without assumptions. Another gap to close is the query complexity of Huber regression, or more generally M-estimator regression, even for just the d dependence: our upper bound is $\tilde{O}(d^{4-2\sqrt{2}}\operatorname{poly}\log n)$ for constant ϵ, while only a trivial lower bound of $\Omega(d)$ is known.

Acknowledgment

T. Y. thanks Cody Johnson and Yi Li for helpful discussions. We thank anonymous reviewers for comments which helped improve the presentation of the paper.

References

[1] D. Adil, B. Bullins, R. Kyng, and S. Sachdeva. Almost-linear-time weighted ℓ_p-norm solvers in slightly dense graphs via sparsification. In N. Bansal, E. Merelli, and J. Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021,*

July 12-16, 2021, Glasgow, Scotland (Virtual Conference), volume 198 of *LIPIcs*, pages 9:1–9:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[2] D. Adil, R. Kyng, R. Peng, and S. Sachdeva. Iterative refinement for ℓ_p-norm regression. In T. M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1405–1424. SIAM, 2019.

[3] A. Andoni, C. Lin, Y. Sheng, P. Zhong, and R. Zhong. Subspace embedding and linear regression with Orlicz norm. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[4] H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh. A universal sampling method for reconstructing signals with simple fourier transforms. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, 2019.

[5] M. Baldauf and J. S. Silva. On the use of robust regression in econometrics. *Economics Letters*, 114(1):124–127, 2012.

[6] J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. Preliminary version in the 41st Annual ACM Symposium on Theory of Computing (STOC), 2009.

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

[8] S. Bubeck, M. B. Cohen, Y. T. Lee, and Y. Li. An homotopy method for $\mathrm{l_p}$ regression provably beyond self-concordance and in input-sparsity time. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1130–1137. ACM, 2018.

[9] K. Chaudhuri, S. M. Kakade, P. Netrapalli, and S. Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, 2015.

[10] X. Chen and M. Dereziński. Query complexity of least absolute deviation regression via robust uniform convergence. In *Proceedings of the 34th Annual Conference on Computational Learning Theory (COLT)*, 2021.

[11] X. Chen, D. M. Kane, E. Price, and Z. Song. Fourier-sparse interpolation without a frequency gap. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 741–750, 2016. Full version at arXiv:1609.01361.

[12] X. Chen and E. Price. Active regression via linear-sample sparsification active regression via linear-sample sparsification. In *Proceedings of the 32nd Annual Conference on Computational Learning Theory (COLT)*, 2019.

[13] K. Clarkson, R. Wang, and D. Woodruff. Dimensionality reduction for Tukey regression. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1262–1271. PMLR, 2019.

[14] K. L. Clarkson and D. P. Woodruff. Input sparsity and hardness for robust subspace approximation. In V. Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 310–329. IEEE Computer Society, 2015.

[15] K. L. Clarkson and D. P. Woodruff. Sketching for *M*-estimators: A unified approach to robust regression. In P. Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 921–939. SIAM, 2015.

[16] A. Cohen, M. A. Davenport, and D. Leviatan. On the stability and accuracy of least squares approximations. *Foundations of Computational Mathematics*, 13(5):819–834, 2013.

[17] A. Cohen and G. Migliorati. Optimal weighted least-squares methods. *SMAI Journal of Computational Mathematics*, 3:181–203, 2017.

[18] M. B. Cohen and R. Peng. l_p row sampling by Lewis weights. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 183–192, 2015.

[19] V. Cohen-Addad, D. Saulpic, and C. Schwiegelshohn. Improved coresets and sublinear algorithms for power means in euclidean spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

[21] M. Derezinski, M. K. K. Warmuth, and D. J. Hsu. Leveraged volume sampling for linear regression. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.

[22] H. Diao, R. Jayaram, Z. Song, W. Sun, and D. P. Woodruff. Optimal sketching for kronecker product regression and low rank approximation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4739–4750, 2019.

[23] T. Erdélyi, C. Musco, and C. Musco. Fourier sparse leverage scores and approximate kernel learning. *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[24] M. Fazel, Y. T. Lee, S. Padmanabhan, and A. Sidford. Computing Lewis weights to high precision. *arXiv:2110.15563*, 2021.

[25] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In L. Fortnow and S. P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011.

[26] D. Feldman and L. J. Schulman. Data reduction for weighted and outlier-resistant clustering. In Y. Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1343–1354. SIAM, 2012.

[27] J. Fox. Robust Regression: Appendix to an R and S-PLUS Companion to Applied Regression, 2002.

[28] M. Ghadiri, R. Peng, and S. S. Vempala. Faster p-norm regression using sparsity. *arXiv preprint arXiv:2109.11537*, 2021.

[29] A. Guitton and W. W. Symes. Robust and stable velocity analysis using the huber function. In *SEG Technical Program Expanded Abstracts 1999*, pages 1166–1169. Society of Exploration Geophysicists, 1999.

[30] J. Hampton and A. Doostan. Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. *Computer Methods in Applied Mechanics and Engineering*, 290:73–97, 2015.

[31] P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[32] W. B. Johnson and G. Schechtman. Finite dimensional subspaces of L_p. In *Handbook of the geometry of Banach spaces, Vol. I*, pages 837–870. North-Holland, Amsterdam, 2001.

[33] J. Kiefer and J. Wolfowitz. Optimum designs in regression problems. *Annals of Mathematical Statistics*, 30(2):271–294, 1959.

[34] M. Ledoux and M. Talagrand. *Probability in Banach spaces*. Classics in Mathematics. Springer-Verlag, Berlin, 1991.

[35] Y. T. Lee and H. Sun. Constructing linear-sized spectral sparsification in almost-linear time. *SIAM Journal on Computing*, 47(6):2315–2336, 2018.

[36] D. Lewis. Finite dimensional subspaces of L_p. *Studia Mathematica*, 63(2):207–212, 1978.

[37] Y. Li, R. Wang, and D. P. Woodruff. Tight bounds for the subspace sketch problem with applications. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1655–1674, 2020.

[38] P. Loh. Scale calibration for high-dimensional robust regression. *CoRR*, abs/1811.02096, 2018.

[39] P.-L. Loh. Statistical consistency and asymptotic normality for high-dimensional robust M-estimators. *Ann. Statist.*, 45(2):866–896, 2017.

[40] T. Mai, C. Musco, and A. B. Rao. Coresets for classification – simplified and strengthened. *arXiv:2106.04254*, 2021.

[41] O. L. Mangasarian and D. R. Musicant. Robust linear and support vector regression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(9):950–955, 2000.

[42] A. Munteanu, C. Schwiegelshohn, C. Sohler, and D. P. Woodruff. On coresets for logistic regression. *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.

[43] A. Parulekar, A. Parulekar, and E. Price. l_1 regression with Lewis weights subsampling. *arXiv:2105.09433*, 2021.

[44] F. Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006.

[45] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[46] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[47] G. Schechtman. More on embedding subspaces of L_p in l_r^n. *Compositio Math.*, 61(2):159–169, 1987.

[48] G. Schechtman and A. Zvavitch. Embedding subspaces of L_p into ℓ_p^n, $0 < p < 1$. *Mathematische Nachrichten*, 227(1):133–142, 2001.

[49] C. Sohler and D. P. Woodruff. Subspace embeddings for the l1-norm with applications. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 755–764, 2011.

[50] Z. Song, R. Wang, L. F. Yang, H. Zhang, and P. Zhong. Efficient symmetric norm regression via linear sketching. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 828–838, 2019.

[51] Z. Song, D. P. Woodruff, and P. Zhong. Towards a zero-one law for column subset selection. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6120–6131, 2019.

[52] M. Tukan, A. Maalouf, and D. Feldman. Coresets for near-convex functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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