Dimensionality Reduction for Tukey Regression*

Kenneth L. Clarkson
IBM Research - Almaden
klclarks@us.ibm.com

Ruosong Wang
Carnegie Mellon University
ruosongw@andrew.cmu.edu

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

Abstract

We give the first dimensionality reduction methods for the overconstrained Tukey regression problem. The Tukey loss function $\|y\|_M = \sum_i M(y_i)$ has $M(y_i) \approx |y_i|^p$ for residual errors $y_i$ smaller than a prescribed threshold $\tau$, but $M(y_i)$ becomes constant for errors $|y_i| > \tau$.

Our results depend on a new structural result, proven constructively, showing that for any $d$-dimensional subspace $L \subset \mathbb{R}^n$, there is a fixed bounded-size subset of coordinates containing, for every $y \in L$, all the large coordinates, with respect to the Tukey loss function, of $y$. We think of these as “residual leverage scores”, since the coordinates in $y$ itself may have very different magnitude even though they both contribute the same value $\tau$ to the $M$-function.

Our methods reduce a given Tukey regression problem to a smaller weighted version, whose solution is a provably good approximate solution to the original problem. Our reductions are simple and easy to implement, and we give empirical results demonstrating their practicality, using existing heuristic solvers for the small versions.

One of our reductions uses row sampling, for an instance $\min_{x \in \mathbb{R}^d} \|Ax-b\|_M$, where $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, with $n \gg d$. The algorithm takes $\tilde{O}(\text{nnz}(A)+\text{poly}(d\log n/\varepsilon))$ time to return a weight vector with poly$(d\log n/\varepsilon)$ non-zero entries, such that the solution of the resulting weighted Tukey regression problem is a $(1+\varepsilon)$-approximate solution. Here $\text{nnz}(A)$ is the number of non-zero entries of $A$. Another reduction uses a sketching matrix $S$, chosen independently of $A$ and $b$, such that $SA$ and $Sb$ yield an $O(\log n)$ approximation, so that the solution for a weighted version with inputs $SA, Sb$ is an $O(\log n)$-approximate solution. Here $S$ has poly$(d\log n)$ rows and $SA$ and $Sb$ are computable in $O(\text{nnz}(A))$ time.

We also give exponential-time algorithms giving provably good solutions, and hardness results suggesting that a significant speedup in the worst case is unlikely.

*Ruosong Wang and David P. Woodruff were supported in part by Office of Naval Research (ONR) grant N00014-18-1-2562. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.
1 Introduction

A number of problems in numerical linear algebra have witnessed remarkable speedups via the technique of linear sketching. Such speedups are made possible typically by reductions in the dimension of the input (here the number of rows of the input matrix), whereby a large scale optimization problem is replaced by a much smaller optimization problem, and then a slower algorithm is run on the small problem. It is then argued that the solution to the smaller problem provides an approximate solution to the original problem. We refer the reader to several recent surveys on this topic [18, 22, 34].

This approach has led to optimal algorithms for approximate overconstrained least squares regression: given an \( n \times d \) matrix \( A \), with \( n \gg d \), and an \( n \times 1 \) vector \( b \), output a vector \( x' \in \mathbb{R}^d \) for which \( \|Ax' - b\|_2 \leq (1 + \varepsilon) \min_x \|Ax - b\|_2 \). For this problem, one first samples a random matrix \( S \in \mathbb{R}^{k \times n} \) with a small number \( k \) of rows, and replaces \( A \) with \( S \cdot A \) and \( b \) with \( S \cdot b \). Then one solves (or approximately solves) the small problem \( \min_x \|SAx - Sb\|_2 \). The goal of this sketch and solve approach is to choose a distribution \( S \) so that if \( x' \) is the minimizer to this latter problem, then one has that \( \|Ax' - b\|_2 \leq (1 + \varepsilon) \min_x \|Ax - b\|_2 \) with high probability. Note that \( x' = (SA)^+Sb \), where \( (SA)^+ \) denotes the pseudoinverse of \( SA \) and can be computed in \( kd^2 \) time, see, e.g., [34] for a survey and further background. Consequently, the overall time to solve least squares regression is \( T + kd^2 \), where \( T \) is the time to compute \( S \cdot A \) and \( S \cdot b \). Thus, the goal is to minimize both the time \( T \) and the sketching dimension \( k \). Using this approach, Sárosi showed [28] how to achieve \( O(nd \log n) + \text{poly}(d/\varepsilon) \) overall time, which was subsequently improved to the optimal \( \text{nnz}(A) + \text{poly}(d/\varepsilon) \) time in [18, 23, 25].

Recently, a number of works have looked at more robust versions of regression. Sohler and Woodruff [29], building off of earlier work of Clarkson [6] (see also [13]), showed how to use the sketch and solve paradigm to obtain a \((1 + \varepsilon)\)-approximation to \( \ell_1 \) regression, namely, to output a vector \( x' \in \mathbb{R}^d \) for which \( \|Ax' - b\|_1 \leq (1 + \varepsilon) \min_x \|Ax - b\|_1 \). This version of regression, also known as least absolute deviations, is known to be less sensitive to outliers than least squares regression, since one takes the absolute values of the errors in the residuals rather than their squares, and so does not try to fit outliers as much. By now, we also have optimal \( \text{nnz}(A) + \text{poly}(d/\varepsilon) \) time algorithms for \( \ell_1 \) regression [20, 7, 35, 18, 23, 32], for the related quantile regression problem [36], and for \( \ell_p \) regression for every \( p \geq 1 \) [13, 35, 12].

In this paper we consider more general overconstrained regression problems: given an \( n \times d \) matrix \( A \) with \( n \gg d \), and an \( n \times 1 \) vector \( b \), output a vector \( x' \in \mathbb{R}^d \) for which \( \|Ax' - b\|_M \leq (1 + \varepsilon) \min_x \|Ax - b\|_M \), where for an \( n \)-dimensional vector \( y \) and a function \( M: \mathbb{R} \rightarrow \mathbb{R}^+ \), the notation \( \|y\|_M \) denotes \( \sum_{i=1}^n M(y_i) \). If \( M(x) = x^2 \) then we have the least squares regression problem, while if \( M(x) = |x| \) we have the least absolute deviations problem.

Clarkson and Woodruff showed [11] that for any function \( M(\cdot) \) which has at least linear (with positive slope) and at most quadratic growth, as well as some natural other properties, there is a distribution on sketching matrices \( S \) with \( k = \text{poly}(d \log n) \) rows and corresponding vector \( w \in \mathbb{R}^k \), with the following properties. The product \( S \cdot A \) can be computed in \( \text{nnz}(A) \) time, and if one solves a weighted version of the sketched problem, \( \min_x \sum_{i=1}^k w_i M((SAx - b)_i) \), then the minimizer is a constant-factor approximation to the original problem. This gives an algorithm with overall \( \text{nnz}(A) + \text{poly}(d) \) running time for the important Huber loss function: given a parameter \( \tau \), we
Figure 1: The Tukey loss function.

\[
M(x) = \begin{cases} 
  x^2/(2\tau) & |x| \leq \tau \\
  |x| - \tau/2 & \text{otherwise}
\end{cases}.
\]

Unlike least absolute deviations, the Huber loss function is differentiable at the origin, which is important for optimization. However, like least absolute deviations, for large values of \(x\) the function is linear, and thus pays less attention to outliers. Other loss functions similar to Huber, such as the \(\ell_1 - \ell_2\) and Fair estimators, were also shown to have \(\text{nnz}(A) + \text{poly}(d)\) time algorithms. These results were extended to \((1 + \varepsilon)\)-approximations via sampling-based techniques in [9].

Despite the large body of \(M\)-functions handled by previous work, a notable well-studied exception is the Tukey loss function [15], with

\[
M(x) = \begin{cases} 
  \frac{\tau^2}{6} \left(1 - \left(\frac{x}{\tau}\right)^2\right)^3 & |x| \leq \tau \\
  \frac{\tau^2}{2} & \text{otherwise}
\end{cases}.
\]

See Figure 1 for a plot of the Tukey loss function. By a simple Taylor expansion, it is easy to see that \(M(x) = \Theta(x^2)\) for \(|x| \leq \tau\) and \(M(x) = \Theta(\tau^2)\) otherwise. While similar to Huber in the sense that it is quadratic near the origin, it is even less sensitive to outliers than Huber since it is constant when one is far enough away from the origin. Thus, it does not satisfy the linear growth (with positive slope) requirement of [11, 9]. An important consequence of this distinction is that, while for \(M\)-functions with linear growth, a single outlier “at infinity” can radically affect the regression output, this is not the case for the Tukey loss function, due to the bound on its value.

Although the Tukey loss function is not convex, a local minimum of it can be found via iteratively reweighted least squares [15]. Also, the dimensionality reduction approach still makes sense for Tukey: here a large non-convex problem is reduced to a smaller non-convex one. Our reduction of a non-convex problem to a smaller one is arguably even more interesting than reducing the size of a convex problem, since inefficient algorithms may now be efficient on the much smaller problem.

**Notation.** For a matrix \(A \in \mathbb{R}^{n \times d}\), we use \(A_{i,*}\) to denote its \(i\)-th row, \(A_{*,j}\) to denote its \(j\)-th column, and \(A_{i,j}\) to denote a specific entry. For a set of indices \(\Gamma \subseteq [n]\), we use \(A_{\Gamma,*}\) to denote the submatrix of \(A\) formed by all rows in \(\Gamma\). Similarly, we use \(A_{*,\Gamma}\) to denote the submatrix formed by all columns in \(\Gamma\). For a matrix \(A \in \mathbb{R}^{n \times d}\) and a vector \(b \in \mathbb{R}^n\), we use \([A \ b] \in \mathbb{R}^{n \times (d+1)}\) to denote the matrix whose first \(d\) columns are \(A\) and the last column is \(b\). For a matrix \(A \in \mathbb{R}^{n \times d}\), we use \(\text{im}(A) = \{Ax \mid x \in \mathbb{R}^d\}\) to denote the column span of \(A\).
1.1 Our Assumptions

Before stating our results, we give the general assumptions our algorithms and analyses need. We need the following assumptions on the loss function.

**Assumption 1.** There exist real numbers $\tau \geq 0$, constants $p \geq 1$ and $0 < L_M \leq 1 \leq U_M$ such that the function $M : \mathbb{R} \to \mathbb{R}^+$ satisfies:

1. **Symmetric:** $M(a) = M(-a)$ for all $a$.
2. **Nondecreasing:** $M(a) \geq M(a')$ for $|a| \geq |a'|$.
3. **Growth condition:** for $|a| \geq |a'|$,
   \[ \frac{|a|^p}{|a'|^p} \geq \frac{M(a)}{M(a')} \]
4. **Nearly $p$th power:** for all $|a| \leq \tau$,
   \[ L_M |a|^p \leq M(a) \leq U_M |a|^p. \]
5. ** Mostly flat:** $M(a) = \tau^p$ for $|a| \geq \tau$.

The conditions in Assumption 1 state that our loss function essentially behaves as an $\ell_p$ loss function $M(a) = |a|^p$ for $a \leq \tau$, at which point $M(a) = \tau^p$. However, the conditions are more robust in that $M(a)$ just needs to agree with $|a|^p$ up to a fixed constant factor. This is a non-trivial extension of the $\ell_p$ loss function since we will obtain $(1 + \varepsilon)$-approximations in our algorithms, and consequently cannot simply replace the $M$ loss function in our problem with an $\ell_p$ loss function, as this would result in a constant factor loss. Moreover, such an extension is essential for capturing common robust loss functions, such as the Tukey loss function (1) above: $M(a) = a^2(1 - (1 - (a/\tau)^2)^3)$ if $|a| < \tau$, and $M(a) = \tau^2$ for $|a| \geq \tau$ (we note that sometimes this loss function is divided by the number 6, but this plays no role from a minimization perspective). Note that the Tukey loss function $M(a)$ does not coincide with the $\ell_2$ loss function for $|a| \leq \tau$, though it is within a constant factor of it so our conditions can handle it. For a recent use of this loss function for regression in applications to deep learning, see, e.g., [4].

For $\tau = \infty$, we indeed have $M(a) = |a|^p$ and $\ell_p$ loss functions are widely studied for regression. The case $p = 2$ is just ordinary least squares regression. For $1 \leq p < 2$, the $\ell_p$ loss function is less sensitive than least squares since one is not squaring the errors, and therefore for such $p$ the loss function is considered to be more robust than ordinary least squares. For $p > 2$, and in particular large values of $p$, the $\ell_p$ regression solution approaches the $\ell_\infty$ regression solution, which minimizes the maximum error. The $\ell_p$ loss functions for every $p \geq 1$ are well-studied in the context of regression, see, e.g., [6, 13, 7, 35, 12]. Loss functions with the polynomial growth condition (Assumption 1) are also well-studied [11, 9]. We also note that already for $\ell_p$ loss functions, all known dimensionality reduction techniques reduce the original regression problem to a problem of size at least $d^{\Omega(p)}$, which is recently shown to be necessary [21]. Consequently, it is impossible to obtain a considerable dimensionality reduction when $p$ is allowed to grow with $n$. Since this is a special case of our more general setting of arbitrary $\tau$, we restrict our attention to $p$ that does not grow with $n$.

For general $\tau$ and $p$, we obtain the Tukey $\ell_p$ loss function $M(a) = |a|^p$ for $a \leq \tau$, and $M(a) = \tau^p$ for $a \geq \tau$. Note that for $p = 1$, the loss function is the maximum likelihood estimator in the presence
of i.i.d. Laplacian noise, while for \( p = 2 \), the loss function is the maximum likelihood estimator in the presence of i.i.d. Gaussian noise. Thus we obtain the first dimensionality reduction for loss functions that correspond to maximum likelihood estimators of classical noise models where the loss function completely saturates beyond some threshold \( \tau \). This is particularly useful for handling large outliers.

We will also need the following assumptions on the input, which we justify below.

**Assumption 2.** We assume:

1. For given \( C_1 \leq \text{poly}(n) \), there is \( U = n^{O(d^2)} \) such that \( \|\hat{x}\|_2 \leq U \) for any \( C_1 \)-approximate solution \( \hat{x} \) of \( \min_x \|Ax - b\|_M \).

2. The columns of \( A \) and \( b \) have \( \ell_2 \) norms in \( n^{O(d)} \).

3. The threshold \( \tau = \Omega(1/n^{O(d)}) \).

As we will show, Assumption 2.1 holds when 2.2 and 2.3 hold, and the entries of \( A \) are integers. Alternatively, Assumption 2.1 might hold due to the particular input given, or to an additional explicit problem constraint, or as a consequence of regularization.

We need such a bound on the magnitude of the entries of the Tukey regression optimum, since as the following example shows, they can grow quite large, and behave quite differently from \( \ell_p \) regression solutions. Here we use the case \( p = 2 \) as an example.

Suppose \( A \) is the \( n \times 2 \) matrix

\[
A = \begin{bmatrix}
1 & 0 \\
1 & 1 \\
0 & \varepsilon \\
0 & \varepsilon \\
\vdots \\
0 & \varepsilon
\end{bmatrix}
\]

with \( n - 2 \) rows of \([0 \ \varepsilon]\) for \( \varepsilon > 0 \). Suppose \( b \in \mathbb{R}^n \) is the vector of all ones. Then the \( \ell_2 \) regression optimum \( x^* = \arg \min_x \|Ax - b\|_2^2 \) has \( \|Ax^* - b\|_2^2 \) at most \( n \), since \( x = 0 \) has that cost. So \( n \geq \|Ax^* - b\|_2^2 \geq (x^*_1 - 1)^2 + (x^*_2 - 1)^2 \), from the first two rows, so \( (x^*_1)^2 = O(n) \), implying that also \( (x^*_2)^2 = O(n) \).

It can also be shown that when \( \varepsilon < 1/n \), we have the entries of the \( \ell_2 \) regression optimum \( x^* \) in \( O(1) \): as \( x_1 \) and \( x_2 \) get larger, they contribute to the cost, due to the first two rows, more rapidly than the decrease in the \( (n - 2)(1 - x_2/n)^2 \) cost due to the \( n - 2 \) last rows.

However, for the Tukey loss function \( \|Ax - b\|_M \) with parameter \( \tau = 1/2 \) and \( p = 2 \), the cost for \( x = [1 \ 1/\varepsilon] \) is at most a constant, since the contributions for all rows of \( A \) but the second is zero, and contribution made by the second row is at most a constant. However if \( x_2 < 1/(2\varepsilon) \), the cost is \( \Omega(n) \), since all but the first two rows contribute \( \Omega(1) \) to the cost. Thus the optimal \( x \) for the Tukey regression has entries \( \Omega(1/\varepsilon) \), and no \( x \) with entries \( o(1/\varepsilon) \) is a constant-factor approximation.

Indeed, given an upper bound on the entries of \( x \), for any \( n' < n - 2 \) there is a large enough version of our example such that no \( x \) satisfying that bound can be within an \( n' \) factor of optimal for the Tukey regression problem. This example is in fact a near-optimal separation, as one can show that the \( \ell_2 \) regression solution always provides an \( O(n) \)-approximate solution when \( p = 2 \).
1.2 Our Contributions

We show that under the assumptions mentioned above, it is possible to obtain dimensionality reductions for Tukey regression. All of our results hold with arbitrarily large constant probability, which can be amplified by independent repetitions and taking the best solution found.

Row Sampling Algorithm. We first give a row sampling algorithm for Tukey regression.

**Theorem 1.1.** Given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$, there is an algorithm that constructs a weight vector $w$ in $O(\text{nnz}(A) + d^{p/2} \cdot \text{poly}(d \log n)/\epsilon^2)$ time with $\|w\|_0 \leq O(d^{p/2} \cdot \text{poly}(d \log n)/\epsilon^2)$, for which if $x^*_{M,w}$ is the minimizer to $\min \sum^n_{i=1} w_i M((Ax - b)_i)$, then

$$\|Ax^*_{M,w} - b\|_M \leq (1 + \epsilon) \min_x \|Ax - b\|_M,$$

where $M$ is the Tukey loss function.

Since one can directly ignore those rows $A_{i,*}$ with $w_i = 0$ when solving $\min \sum^n_{i=1} w_i M((Ax - b)_i)$, our row sampling algorithm actually reduces a Tukey regression instance to a weighted version of itself with $O(d^{p/2} \cdot \text{poly}(d \log n)/\epsilon^2)$ rows. Notably, the running time of the algorithm and the number of rows in the reduced problem match those given by Lewis weights sampling [12], up to $\text{poly}(d \log n)$ factors. However, Lewis weights sampling is designed specifically for the $\ell_p$ norm, which is a simple special case of the Tukey loss function where $\tau = \infty$.

Our reduction is from the Tukey regression problem to a smaller, weighted version of itself, and since known heuristics for Tukey regression can also handle weights, we can apply them to the reduced problem as well.

Oblivious Sketch. While the row sampling algorithm produces a $(1 + \epsilon)$-approximate solution, where $\epsilon$ can be made arbitrarily small, the algorithm does have some properties that can be a disadvantage in some settings: it makes $\text{polylog}(n)$ passes over the matrix $A$ and vector $b$, and the rows chosen depend on the input. In the setting of streaming algorithms and distributed computation, on the other hand, a sketch-and-solve approach can be more effective. We give such an approach to Tukey regression.

**Theorem 1.2.** When $1 \leq p \leq 2$, there is a distribution $S \in \mathbb{R}^{r \times n}$ over sketching matrices with $r = \text{poly}(d \log n)$, and a corresponding weight vector $w \in \mathbb{R}^r$, for which $S \cdot A$ and $S \cdot b$ can be computed in $O(\text{nnz}(A))$ time and for which if $x^*_{S,M,w}$ is the minimizer to $\min \sum^r_{i=1} w_i M((SAx - Sb)_i)$, then

$$\|Ax^*_{S,M,w} - b\|_M \leq O(\log n) \min_x \|Ax - b\|_M,$$

where $M$ is the Tukey loss function.

Our sketching matrices $S$ are oblivious, meaning that their distribution does not depend on the data matrix $A$ and the vector $b$. Furthermore, applying the sketching matrices requires only one pass over these inputs, and thus can be readily implemented in streaming and distributed settings.

We further show that the same distribution $S$ on sketching matrices gives a fixed constant $C \geq 1$ approximation factor if one slightly changes the regression problem solved in the reduced space.

We also remark that for oblivious sketches, the condition that $p \leq 2$ is necessary, as shown in [5].

\[1\] Throughout the paper we use $\tilde{O}(f)$ to denote $f \cdot \text{polylog } f$, and $\tilde{\Omega}(f)$ to denote $f/\text{polylog } f$. 
Hardness Results and Provable Algorithms. We give a reduction from \textsc{MAX}-3SAT to Tukey regression, which implies the NP-Hardness of Tukey regression. Under the Exponential Time Hypothesis [16], using Dinur’s PCP Theorem [14], we can strengthen the hardness result and show that even solving Tukey regression with approximation ratio $1+\eta$ requires $2^{\tilde{\Omega}(d)}$ time for some fixed constant $\eta > 0$.

We complement our hardness results by giving an exponential time algorithm for Tukey regression, using the polynomial system verifier [27, 3]. This technique has been used to solve a number of numerical linear algebra problems in previous works [30, 26, 10, 2, 24]. For the loss function defined in [11], the algorithm runs in $2^{O(n \log n)} \cdot \log(1/\varepsilon)$ time to find a $(1 + \varepsilon)$-approximate solution of an instance of size $n \times d$. By further applying our dimensionality reduction methods, the running time can be reduced to $2^{\text{poly}(d \log n)}$, which is significantly faster when $n \gg d$ and comes close to the $2^{\tilde{\Omega}(d)}$ running time lower bound.

Empirical Evaluation. We test our dimensionality reduction methods on both synthetic datasets and real datasets. Our empirical results quite clearly demonstrate the practicality of our methods.

2 Technical Overview

In this section, we give an overview of our technical contributions. For convenience, we state our high-level ideas in terms of the loss function

$$M(x) = \begin{cases} x^2 & |x| \leq 1 \\ 1 & |x| \geq 1 \end{cases}$$

where $p = 2$ and $\tau = 1$. We show how to generalize our ideas to other loss functions that satisfy Assumption [11] later.

2.1 Structural Theorem and Algorithms for Finding Heavy Coordinates

Our first main technical contribution is the following structural theorem, which is crucial for each of our algorithms.

\textbf{Theorem 2.1} (Informal). For a given matrix $A \in \mathbb{R}^{n \times d}$ and $\alpha \geq 1$, there exists a set of indices $I \subseteq [n]$ with size $|I| \leq O(\alpha)$, such that for all $y \in \text{im}(A)$, if $y$ satisfies $\|y\|_M \leq \alpha$ then $\{i \in n \mid |y_i| > 1\} \subseteq I$.

Intuitively, Theorem 2.1 states that for all vectors $y$ in the column space of $A$ with small $\|y\|_M$, the heavy coordinates of $y$ (coordinates with $|y_i| \geq 1$) must lie in a set $I$ with small cardinality. To prove Theorem 2.1, in Figure 2 we give an informal description of our algorithm for finding the set $I$. The formal description of the algorithm can be found in Section 4.1.
1. Let $I = \emptyset$.
2. Repeat the following for $\alpha$ times:
   (a) Calculate the leverage scores $\{u_i\}_{i \in [n] \setminus I, *}$ of the matrix $A_{[n] \setminus I, *}$.
   (b) For each $i \in [n] \setminus I$, if $u_i \geq \Omega(1/\alpha)$, then add $i$ into $I$.
3. Return $I$.

Figure 2: Polynomial time algorithm for finding heavy coordinates.

For correctness, we first notice that for a vector $y$ with $\|y\|_M \leq \alpha$, the number of heavy coordinates is at most $\alpha$, since $M(y_i) = 1$ for all $|y_i| > 1$. Now consider the coordinate $i$ with largest $|y_i|$ and $|y_i| > 1$. We must have $\|y\|_2^2 \leq \alpha + \alpha y_i^2$, since the contribution of coordinates with $|y_i| \leq 1$ to $\|y\|_M \leq \alpha$, and there are at most $\alpha$ coordinates with $|y_i| > 1$, each contributing at most $y_i^2$ to $\|y\|_2^2$. Now we claim that we must add the coordinate $i$ with largest $|y_i|$ into the set $I$, which simply follows from

$$\frac{y_i^2}{\|y\|_2^2} \geq \frac{y_i^2}{\alpha + \alpha y_i^2} \geq \Omega(1/\alpha)$$

(2)

and thus the leverage score of the row $A_{i,*}$ is at least $\Omega(1/\alpha)$. (Here we use that the $i$-th leverage score is at least as large as $y_i^2/\|y\|_2^2$ for all $y \in \im(A)$.) After adding $i$ into $I$, we consider the second largest $|y_i|$ with $|y_i| \geq 1$. A similar argument shows that we will also add $i$ into $I$ in the second repetition. After repeating $\alpha$ times we will add all coordinates $i$ with $|y_i| > 1$ into $I$, and all coordinates added to $I$ have leverage score $\Omega(1/\alpha)$.

The above algorithm has two main drawbacks. First of all, it returns a set with size $|I| \leq O(\alpha d^2)$ as opposed to $\tilde{O}(\alpha d)$. Moreover, the algorithm runs in $O(\mnz(A) \cdot \alpha)$ time since we need to calculate the leverage scores of $A_{[n] \setminus I, *}$ a total of $\alpha$ times. When $\alpha = \poly(d)$, such an algorithm does not run in input-sparsity time. An input-sparsity time algorithm for finding such a set $I$, on the other hand, is an important subroutine for our input-sparsity time row sampling algorithm. In Section 4.2 we give a randomized algorithm for finding a set $I$ with size $|I| \leq \tilde{O}(\alpha d)$ that runs in input-sparsity time, and we give an informal description of the algorithm in Figure 3. Notice that calculating leverage scores of the matrices $A_{[\Gamma_j,*]}$ can be done in $\tilde{O}(\mnz(A) + \poly(d))$ time using existing approaches [8, 25].
1. Let $I = \emptyset$.

2. Repeat the following for $O(\log(d\alpha))$ times:
   (a) Randomly partition $[n]$ into $\Gamma_1, \Gamma_2, \ldots, \Gamma_\alpha$.
   (b) For each $j \in [\alpha]$, calculate the leverage scores $\{u_i\}_{i \in \Gamma_j}$ of the matrix $A_{\Gamma_j,*}$.
   (c) For each $j \in [\alpha]$, for each $i \in \Gamma_j$, if $\hat{u}_i \geq \Omega(1)$, then add $i$ to $I$.

3. Return $I$.

Figure 3: Input-sparsity time algorithm for finding heavy coordinates.

For correctness, recall that we only need to find those coordinates $i$ for which there exists a vector $y \in \text{im}(A)$ with $\|y\|_M \leq \alpha$ and $|y_i| \geq 1$. Since $\|y\|_M \leq \alpha$, there are most $\alpha$ coordinates in $y$ with absolute value at least 1. Thus, with constant probability, the coordinate $i$ is in a set $\Gamma_j$ such that it is the only coordinate with $|y_i| \geq 1$ in $\Gamma_j$. Moreover, by Markov’s inequality, with constant probability the squared $\ell_2$ norm of coordinates in $\Gamma_j \setminus \{i\}$ is at most a constant. Conditioned on these events, using an argument similar to (2), the leverage score of the row $A_{i,*}$ in $A_{\Gamma_j,*}$ is at least a constant, in which case we will add $i$ into $I$. In order to show that we will add all such $i$ into $I$ with good probability, we repeat the whole procedure for $O(\log(d\alpha))$ times and apply a union bound over all such $i$. $O(\log(d\alpha))$ repetitions suffice since there are at most $O(\text{poly}(d\alpha))$ different such $i$, as implied by the existential result mentioned above.

The above algorithm also implies the existence of a set $I$ with better upper bounds on $|I|$, by the probabilistic method. These algorithms can be readily generalized to general $\tau > 0$, and any $p \geq 1$ using $\ell_p$ Lewis weights in place of leverage scores. We also give a brief overview of Lewis weights and related properties in Section 3.2 for readers unfamiliar with these topics.

2.2 The Net Argument

Our second technical contribution is a net argument for Tukey loss functions. Due to the lack of scale-invariance, the net size for the Tukey loss functions need not be $n^{\text{poly}(d)}$. While the $M$-functions in [11] also do not satisfy scale-invariance, the $M$-functions in [11] have at least linear growth and so for any value $c$, and for an orthonormal basis $U$ of $A$, the set of $x$ for which $\|Ux\|_M = c$ satisfy $c/\text{poly}(n) \leq \|x\|_2 \leq c \cdot \text{poly}(n)$, and so one could use $O(\log n)$ nested nets for the $\ell_2$ norm to obtain a net for the $M$-functions. This does not hold for the Tukey loss function $M$, e.g., if $c = \tau$, and if the first column of $U$ is $(1, 0, \ldots, 0)^T$, then if $x_1 = \infty$ and $x_2 = x_3 = \cdots = x_d = 0$, one has $\|Ux\|_M = c$. This motivates Assumption 2 above.

Using Assumption 2, we construct a net $\mathcal{N}_\varepsilon$ with size $|\mathcal{N}_\varepsilon| \leq (n/\varepsilon)^{\text{poly}(d)}$, such that for any $y = Ax - b$ with $\|x\|_2 \leq U = n^{\text{poly}(d)}$, there exists $y' \in \mathcal{N}_\varepsilon$ with $\|y' - y\|_M \leq \varepsilon$. The construction is based on a standard volume argument. Notice that such a net only gives an additive error guarantee. To give a relative error guarantee, we notice that for a vector $y = Ax - b$ with sufficiently small $\|y\|_M$, we must have $\|y\|_\infty \leq 1$, in which case the Tukey loss function $\| \cdot \|_M$ behaves similarly to the squared $\ell_2$ norm $\| \cdot \|_2^2$, and thus we can instead use the net construction for the $\ell_2$ norm. This
1. Use the algorithm in Figure 3 to find a set $I$ with $\alpha = \text{poly}(d \log n / \varepsilon)$.

2. Calculate the leverage scores $\{u_i\}$ of the matrix $A_{[n] \setminus I,*}$.

3. For each row $A_{i,*}$, we define its sampling probability $p_i$ to be

$$p_i = \begin{cases} 1, & i \in I \\ \min\{1, 1/2 + u_i \text{poly}(d/\varepsilon)\}, & i \notin I \end{cases}.$$ 

4. Sample each row with probability $p_i$.

5. Recursively call the algorithm on the resulting matrix until the number of remaining rows is at most $\text{poly}(d \log n / \varepsilon)$.

Figure 4: The row sampling algorithm.

idea can be easily generalized to general $p \geq 1$ and $\tau > 0$ if the loss function satisfies $M(x) = |x|^p$ when $|x| \leq \tau$.

To cope with other loss functions that satisfy Assumption 1 for which $M(x)$ can only be approximated by $|x|^p$ when $|x| \leq \tau$, we use the nested net construction in [11] when $\|y\|_M$ is sufficiently small. Our final net argument for Tukey loss functions is a careful combination of the two net constructions mentioned above. The full details are given in Section 5.3.

2.3 The Row Sampling Algorithm

Our row sampling algorithm proceeds in a recursive manner, and employs a combination of uniform sampling and leverage score sampling, together with the procedure for finding heavy coordinates. We give an informal description in Figure 4. See Section 6 for the formal description and analysis.

For a vector $y = Ax - b$, we conceptually split coordinates of $y$ into two parts: heavy coordinates (those with $|y_i| > 1$) and light coordinates (those with $|y_i| \leq 1$). Intuitively, we need to apply uniform sampling to heavy coordinates, since all heavy coordinates contribute the same to $\|y\|_M$, and leverage score sampling to light coordinates, since the Tukey loss function behaves similarly to the squared $\ell_2$ norm for light coordinates.

In the formal analysis given in Section 6.1 we show that if either the contribution from heavy coordinates to $\|y\|_M$ or the contribution from light coordinates to $\|y\|_M$ is at least $\Omega(\text{poly}(d \log n / \varepsilon))$, then with high probability, uniform sampling with sampling probability $1/2$ will preserve $\|y\|_M$ up to $\varepsilon$ relative error, for all vectors $y$ in the net. The proof is based on standard concentration inequalities.

If both the contribution from heavy coordinates and the contribution from light coordinates is $O(\text{poly}(d \log n / \varepsilon))$, uniform sampling will no longer be sufficient, and we resort to the structural theorem in such cases. By setting $\alpha = \text{poly}(d \log n / \varepsilon)$ in the algorithm for finding heavy coordinates, we can identify a set $I$ with size $|I| = \text{poly}(d \log n / \varepsilon)$, which includes the indices of all heavy
coordinates. We simply keep all coordinates in \( I \) by setting \( p_i = 1 \). The remaining coordinates must be light, and hence behave very similarly to the squared \( \ell_2 \) norm. Thus, we can use leverage score sampling to deal with the remaining light coordinates. This also explains why we need to use a combination of uniform sampling and leverage score sampling.

Our algorithm will eliminate roughly half of the coordinates in each round, and after \( O(\log n) \) rounds there are at most \( O(\text{poly}(d \log n/\varepsilon)) \) remaining coordinates, in which case we stop the sampling process and return our reduced version of the problem. In each round we calculate the leverage scores and call the algorithm in Figure 3 to find heavy coordinates. Since both subroutines can be implemented to run in \( \tilde{O}(\text{nnz}(A) + \text{poly}(d \log n/\varepsilon)) \) time, the overall running time of our row sampling algorithm is also \( \tilde{O}(\text{nnz}(A) + \text{poly}(d \log n/\varepsilon)) \).

The above algorithm can be readily generalized to any loss function \( M \) that satisfies Assumption 1. Our formal analysis in Section 6 is a careful combination of all ideas mentioned above.

### 2.4 The Oblivious Sketch

From an algorithmic standpoint, our oblivious sketch \( S \) is similar to that in [11]. The distribution on matrices \( S \) can be viewed roughly as a stack of \( h_{\max} = O(\log n) \) matrices, where the \( i \)-th such matrix is the product of a CountSketch matrix with \( \text{poly}(d \log n) \) rows with a diagonal matrix \( D \) which samples roughly \( 1/(d \log n)^i \) uniformly random coordinates of an \( n \)-dimensional vector. Thus, \( S \) can be viewed as applying CountSketch to a subsampled set of coordinates of a vector, where the subsampling is more aggressive as \( i \) increases. The weight vector \( w \) is such that \( w_j = (d \log n)^i \) for all coordinates \( j \) corresponding to the \( i \)-th matrix in the stack. Our main technical contribution here is showing that this simple sketch actually works for Tukey loss functions. One of the main ideas in [11] is that if there is a subset of at least \( \text{poly}(d) \log n \) coordinates of a vector \( y \) of similar absolute value, then in one of the levels of subsampling of \( S \), with probability \( 1 - 1/n^{\text{poly}(d)} \) there will be \( \Theta(\text{poly}(d \log n)) \) coordinates in this group which survive the subsampling and are isolated, that is, they hash to separate CountSketch buckets. Using that the \( M \)-function does not grow too quickly, which holds for Tukey loss functions as well if \( p \leq 2 \), this suffices for estimating the contribution to \( \| y \|_M \) from all large subsets of similar coordinates.

The main difference in this work is how estimates for small subsets of coordinates of \( y \) are made. In [11] an argument based on leverage scores sufficed, since, as one ranges over all unit vectors of the form \( y = Ax - b \), there is only a small subset of coordinates which could ever be large, which follows from the condition that the column span of \( A \) is a low-dimensional subspace. At first glance, for Tukey loss functions this might not be true. One may think that for any \( t \leq \text{poly}(d) \log n \), it could be that for a vector \( y = Ax - b \), any subset \( T \) of \( t \) of its \( n \) coordinates could have the property that \( M(y_i) = 1 \) for \( i \in T \), and \( M(y_i) < 1 \) otherwise. However, our structural theorem in fact precludes such a possibility. The structural theorem implies that there are only \( \text{poly}(d \log n) \) coordinates for which \( M(y_i) \) could be 1. For those coordinates with \( M(y_i) < 1 \), the Tukey loss function behaves very similarly to the squared \( \ell_2 \) norm, and thus we can again use the argument based on leverage scores. After considering these two different types of coordinates, we can now apply the perfect hashing argument as in [11].

These ideas can be readily generalized to general \( \tau > 0 \), and any \( 1 \leq p \leq 2 \), again using \( \ell_p \) Lewis weights in place of leverage scores. We formalize these ideas in Lemma 7.1.
3 Preliminaries

For two real numbers $a$ and $b$, we use the notation $a = (1 \pm \varepsilon)b$ if $a \in [(1 - \varepsilon)b, (1 + \varepsilon)b]$. We use $\| \cdot \|_p$ to denote the $\ell_p$ norm of a vector, and $\| \cdot \|_{p,w}$ to denote the weighted $\ell_p$ norm, i.e.,

$$\|y\|_{p,w} = \left( \sum_{i=1}^{n} w_i |y_i|^p \right)^{1/p}.$$

For a vector $y \in \mathbb{R}^n$, a weight vector $w \in \mathbb{R}^n$ whose entries are all non-negative and a loss function $M : \mathbb{R} \to \mathbb{R}^+$ that satisfies Assumption 1, $\|y\|_{M,w}$ is defined to be

$$\|y\|_{M,w} = \sum_{i=1}^{n} w_i \cdot M(y_i).$$

We also define $\|y\|_M$ to be

$$\|y\|_M = \sum_{i=1}^{n} M(y_i).$$

For a vector $y \in \mathbb{R}^n$ and a real number $\tau \geq 0$, we define $H_y$ to be the set $H_y = \{i \in [n] \mid |y_i| > \tau\}$, and $L_y$ to be the set $L_y = \{i \in [n] \mid |y_i| \leq \tau\}$.

3.1 Tail Inequalities

Lemma 3.1 (Bernstein’s inequality). Suppose $X_1, X_2, \ldots, X_n$ are independent random variables taking values in $[-b, b]$. Let $X = \sum_{i=1}^{n} X_i$ and $\text{Var}[X] = \sum_{i=1}^{n} \text{Var}[X_i]$ be the variance of $X$. For any $t > 0$ we have

$$\Pr[|X - \mathbb{E}[X]| > t] \leq 2 \exp \left( -\frac{t^2}{2 \text{Var}[X] + 2bt/3} \right).$$

3.2 Facts Regarding the Loss Function

Lemma 3.2. Under Assumption 1, there is a constant $C > 0$ that depends only on $p$, for which for any $a, b$ with $|b| \leq \varepsilon |a|$, we have $M(a + b) = (1 \pm C \varepsilon)M(a)$.

Proof. Without loss of generality we assume $a > 0$. When $b \geq 0$, by Assumption 1, we have

$$M(a) \leq M(a + b) \leq (1 + \varepsilon)^p M(a) \leq (1 + C \varepsilon)M(a).$$

When $b < 0$, we have

$$M(a) \geq M(a + b) \geq \left( \frac{a}{a + b} \right)^p M(a) \geq (1 - C \varepsilon)M(a).$$

Lemma 3.3. Under Assumption 1, there is a constant $C' > 0$ that depends only on $p$, for which for any $e, y \in \mathbb{R}^n$ and any weight vector $w$ with $\|e\|_{M,w} \leq \varepsilon 2^{p+1} \|y\|_{M,w}$,

$$\|y + e\|_{M,w} = (1 \pm C' \varepsilon)\|y\|_{M,w}.$$
Proof. Clearly, by Assumption 1.3
\[ \|e/\varepsilon^2\|_{M,w} \leq \varepsilon^{-2p}\|e\|_{M,w} \leq \varepsilon\|y\|_{M,w}. \]

Let \( S = \{i \in [n] \mid |e_i| \leq \varepsilon|y_i|\} \). By Lemma 3.2, for all \( i \in S \) we have \( M(y_i + e_i) = (1 \pm C\varepsilon)M(y_i) \). For all \( i \in [n] \setminus S \), we have \( |e_i| > \varepsilon|y_i| \). For sufficiently small \( \varepsilon \), by Assumption 1.2 and Lemma 3.2
\[ M(e_i + y_i) \leq M(e_i/\varepsilon^2 + y_i) \leq (1 + C\varepsilon)M(e_i/\varepsilon^2), \]
which implies
\[ \sum_{i \in [n] \setminus S} w_iM(y_i + e_i) \leq (1 + C\varepsilon)\|e/\varepsilon^2\|_{M,w} \leq (1 + C\varepsilon)\|y\|_{M,w}. \]

Furthermore,
\[ \sum_{i \in [n] \setminus S} w_iM(y_i) \leq \sum_{i \in [n] \setminus S} w_iM(e_i/\varepsilon) \leq \|e/\varepsilon^2\|_{M,w} \leq \varepsilon\|y\|_{M,w}. \]

Thus,
\[
\|y + e\|_{M,w} = \sum_{i \in S} w_iM(y_i + e_i) + \sum_{i \in [n] \setminus S} w_iM(y_i + e_i) \\
= (1 + C\varepsilon)\sum_{i \in S} w_iM(y_i) \pm (1 + C\varepsilon)\varepsilon\|y\|_{M,w} \\
= (1 + C'\varepsilon)\|y\|_{M,w}. 
\]

\[
\sum_{i \in S} \tau_i(A) = \sum_{i \in S} \tau_i(A_i^*) = (A_i^*)^T, 
\]

\[
\sum_{i \in [n] \setminus S} \tau_i(A) = \sum_{i \in [n] \setminus S} \tau_i(A_i^*) = U^{1/2 - 1/p}A. 
\]

3.3 Facts Regarding Lewis Weights

In this section we recall some facts regarding leverage scores and Lewis weights.

Definition 3.1. Given a matrix \( A \in \mathbb{R}^{n \times d} \). The leverage score of a row \( A_i, * \) is defined to be
\[ \tau_i(A) = A_i,*(A^TA)^\dagger(A_i,*)^T. \]

Definition 3.2 (12). For a matrix \( A \in \mathbb{R}^{n \times d} \), its \( \ell_p \) Lewis weights \( \{u_i\}_{i=1}^n \) are the unique weights such that for each \( i \in [n] \) we have
\[ u_i = \tau_i(U^{1/2 - 1/p}A). \]

Here \( \tau_i \) is the leverage score of the \( i \)-th row of a matrix and \( U \) is the diagonal matrix formed by putting the elements of \( u \) on the diagonal.

Theorem 3.4 (12). There is an algorithm that receives a matrix \( A \in \mathbb{R}^{n \times d} \) and outputs \( \{\hat{u}_i\}_{i=1}^n \) such that
\[ u_i \leq \hat{u}_i \leq 2u_i, \]
where \( \{u_i\}_{i=1}^n \) are the \( \ell_p \) Lewis weights of \( A \). Furthermore, the algorithm runs in \( \widetilde{O}(nnz(A) + dp/2 + O(1)) \) time.
Theorem 3.5 (Lewis’s change of density [19], see also [33, p. 113]). Given a matrix $A \in \mathbb{R}^{n \times d}$ and $p \geq 1$, there exists a basis matrix $H \in \mathbb{R}^{n \times d}$ of the column space of $A$, such that if we define a weight vector $\pi \in \mathbb{R}^n$ where $\pi_i = \|H_{i,\ast}\|_2$, then the following hold:

1. $\|\pi\|_p^p \leq d$;
2. $\bar{U}^{p/2-1}H$ is an orthonormal matrix.

Here $\bar{U}$ is the diagonal matrix formed by putting the elements of $\pi$ on the diagonal.

Lemma 3.6 (See, e.g., [33, p. 115]). Given a matrix $A \in \mathbb{R}^{n \times d}$, for the basis matrix $H$ and the weight vector $\pi$ defined in Theorem 3.5, for all $x \in \mathbb{R}^d$ we have

$$\|\bar{U}^{p/2-1}Hx\|_2 \leq \|Hx\|_p \leq d^{1/p-1/2}\|\bar{U}^{p/2-1}Hx\|_2$$

when $1 \leq p \leq 2$, and

$$\|Hx\|_p \leq \|\bar{U}^{p/2-1}Hx\|_2 \leq d^{1/2-1/p}\|Hx\|_p$$

when $p \geq 2$. Since $\bar{U}^{p/2-1}H$ is an orthonormal matrix, for all $x \in \mathbb{R}^d$ we have

$$\|x\|_2 \leq \|Hx\|_p \leq d^{1/p-1/2}\|x\|_2$$

when $1 \leq p \leq 2$, and

$$\|Hx\|_p \leq \|x\|_2 \leq d^{1/2-1/p}\|Hx\|_p$$

when $p \geq 2$.

Lemma 3.7. Given a matrix $A \in \mathbb{R}^{n \times d}$ and $p \geq 1$, the weight vector $u$ defined in Definition 3.2 and the weight vector $\pi$ defined in Theorem 3.5 satisfies

$$u_i = \pi_i^p.$$  

Proof. We show that substituting $u_i = \pi_i^p$ will satisfy

$$u_i = \tau_i(U^{1/2-1/p}A),$$

and thus the theorem follows by the uniqueness of Lewis weights.

Since leverage scores are invariant under change of basis (see, e.g., [34, p. 30]), we have

$$\tau_i(U^{1/2-1/p}A) = \tau_i(U^{1/2-1/p}H),$$

where $H$ is the basis matrix defined in Theorem 3.5. Substituting $u_i = \pi_i^p$ we have

$$\tau_i(U^{1/2-1/p}A) = \tau_i(\bar{U}^{p/2-1}H).$$

However, since $\bar{U}^{p/2-1}$ is an orthonormal matrix, and the leverage scores of an orthonormal matrix are just squared $\ell_2$ norm of rows (see, e.g., [34, p. 29]), we have

$$\tau_i(U^{1/2-1/p}A) = \left(\pi_i^{p/2-1}\|H_{i,\ast}\|_2\right)^2 = \pi_i^p.$$
Lemma 3.8. Given a matrix \( A \in \mathbb{R}^{n \times d} \) and \( p \geq 1 \), for all \( y \in \text{im}(A) \) and \( i \in [n] \), we have
\[
|y_i|^p \leq d^\max\left\{0, \frac{p}{2} - 1\right\} u_i \cdot \|y\|_p^p.
\]
Here \( \{u_i\}_{i=1}^n \) are the \( \ell_p \) Lewis weights defined in Definition 3.2.

Proof. For all \( y \in \text{im}(A) \), we can write \( y = Hx \) for some vector \( x \in \mathbb{R} \) and the basis matrix \( H \) in Theorem 3.5. By the Cauchy-Schwarz inequality,
\[
|y_i|^p = |\langle x, H_i, * \rangle|^p \leq \|x\|_2^p \cdot \|H_i, *\|_2^p,
\]
which implies
\[
|y_i|^p \leq d^\max\left\{0, \frac{p}{2} - 1\right\} \cdot \|y\|_p^p \cdot \|H_i, *\|_2^p
\]
by Lemma 3.6 which again implies
\[
|y_i|^p \leq d^\max\left\{0, \frac{p}{2} - 1\right\} u_i \cdot \|y\|_p^p
\]
since \( \overline{u_i} = \|H_i, *\|_2 \) and \( u_i = \overline{u_i} \) by Lemma 3.7.

Lemma 3.9. Under Assumption 7, given a matrix \( A \in \mathbb{R}^{n \times d} \), \( \delta_{\text{lewis}} \in (0, 1) \), and a weight vector \( w \in \mathbb{R}^n \) such that (i) \( w_i \geq 1 \) for all \( i \in [n] \) and (ii) \( \max_{i \in [n]} w_i \leq 2 \min_{i \in [n]} w_i \). Let \( w' \in \mathbb{R}^n \) be another weight vector which is defined to be
\[
w'_i = \begin{cases} w_i/p_i & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}
\]
and \( p_i \) satisfies
\[
p_i \geq \min\{1, \Theta(U_M/L_M \cdot d^\max\left\{0, \frac{p}{2} - 1\right\} u_i \cdot \log(1/\delta_{\text{lewis}})/\varepsilon^2)\},
\]
then for any fixed vectors \( x \in \mathbb{R}^d \) such that \( \|Ax\|_\infty \leq \tau \), with probability at least \( 1 - \delta_{\text{lewis}} \) we have
\[
\|Ax\|_{M, w} = (1 \pm \varepsilon) \|Ax\|_{M, w'}.
\]

Proof. Without loss of generality we assume \( 1 \leq w_i \leq 2 \) for all \( i \in [n] \). Let \( y = Ax \). We use the random variable \( Z_i \) to denote
\[
Z_i = w'_i M(y_i).
\]
Clearly \( \mathbb{E}[Z_i] = w_i M(y_i) \), which implies
\[
\mathbb{E}[\|y\|_{M, w'}] = \|y\|_{M, w}.
\]
Furthermore, \( Z_i \leq 2M(y_i)/p_i \). Since \( \|y\|_\infty \leq \tau \) and \( L_M |y_i|^p \leq M(y_i) \leq U_M |y_i|^p \) when \( |y_i| \leq \tau \), by Lemma 3.8 we have
\[
Z_i \leq 2U_M |y_i|^p/p_i \leq \Theta(L_M \cdot \|y\|_p^p \cdot \varepsilon^2 / \log(1/\delta_{\text{lewis}})) \leq \Theta(\|y\|_{M, w} \cdot \varepsilon^2 / \log(1/\delta_{\text{lewis}})).
\]
Moreover, \( \mathbb{E}[Z_i^2] \leq O((M(y_i))^2/p_i) \), which implies
\[
\sum_{i=1}^n \mathbb{E}[Z_i^2] \leq O \left( \sum_{i=1}^n (M(y_i))^2/p_i \right),
\]

14
By H"older’s inequality,
\[ \sum_{i=1}^{n} E[Z_i^2] \leq O(\|y\|_M) \cdot \max_{i \in [n]} M(y_i)/p_i \leq O(\|y\|_{M,w}^2 \cdot \varepsilon^2 / \log(1/\delta_{\text{lewis}})). \]

Furthermore, since
\[ \text{Var} \left[ \sum_{i=1}^{n} Z_i \right] = \sum_{i=1}^{n} \text{Var}[Z_i] \leq \sum_{i=1}^{n} E[Z_i^2], \]
Bernstein’s inequality in Lemma 3.1 implies
\[ \Pr \left[ \|y\|_{M,w'} - \|y\|_{M,w} > t \right] \leq \exp \left(-\Theta \left( \|y\|_{M,w} \cdot \varepsilon^2 / \log(1/\delta_{\text{lewis}}) \cdot t + \|y\|_{M,w}^2 \cdot \varepsilon^2 / \log(1/\delta_{\text{lewis}}) \right) \right). \]

Taking \( t = \varepsilon \cdot \|y\|_{M,w} \) implies the desired result. \( \square \)

**Theorem 3.10.** Given a matrix \( A \in \mathbb{R}^{n \times d} \), \( \delta_{\text{subspace}} \in (0, 1) \), and a weight vector \( w \in \mathbb{R}^n \) such that (i) \( w_i \geq 1 \) for all \( i \in [n] \) and (ii) \( \max_{i \in [n]} w_i \leq 2 \min_{i \in [n]} w_i \). Let \( w' \in \mathbb{R}^n \) be another weight vector which is defined to be
\[
w'_i = \begin{cases} w_i/p_i & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}
\]
and \( p_i \) satisfies
\[ p_i \geq \min \{1, \Theta(d_{\max(0, p/2-1)}u_i \cdot (d \log(1/\varepsilon) + \log(1/\delta_{\text{subspace}}))/\varepsilon^2)\}, \]
then with probability at least \( 1 - \delta_{\text{subspace}} \), for all vectors \( x \in \mathbb{R}^d \), we have
\[ \|Ax\|_{p,w} = (1 \pm \varepsilon)\|Ax\|_{p,w'}. \]

**Proof.** Let \( \mathcal{N} \) be an \( \varepsilon \)-net for \( \{Ax \mid \|Ax\|_{p,w} = 1\} \). Standard facts (see, e.g., [34, p. 48]) imply that \( \log|\mathcal{N}| \leq O(d \log(1/\varepsilon)) \). Now we invoke Lemma 3.9 with \( \delta_{\text{lewis}} = \delta_{\text{subspace}}/|\mathcal{N}| \). Notice that \( f(x) = \|x\|^p \) is also a loss function that satisfies Assumption 11 with \( L_M = U_M = 1 \) and \( \tau = \infty \). Thus, if \( p_i \) satisfies
\[ p_i \geq \Theta(d_{\max(0, p/2-1)}u_i \cdot (d \log(1/\varepsilon) + \log(1/\delta_{\text{subspace}}))/\varepsilon^2), \]
then with probability \( 1 - \delta_{\text{subspace}} \), simultaneously for all \( x \in \mathcal{N} \) we have
\[ \|Ax\|_{p,w} = (1 \pm \varepsilon)\|Ax\|_{p,w'}. \]
Now we can invoke the standard successive approximation argument (see, e.g., [34, p. 47]) to show that with probability \( 1 - \delta_{\text{subspace}} \), simultaneously for all \( x \in \mathbb{R}^d \) we have
\[ \|Ax\|_{p,w} = (1 \pm O(\varepsilon))\|Ax\|_{p,w'}. \]
Adjusting constants implies the desired result. \( \square \)
4 Finding Heavy Coordinates

4.1 A Polynomial Time Algorithm

1. Let $J = \emptyset$.

2. Repeat the following for $\alpha$ times:
   
   (a) Calculate $\{u_i\}_{i \in [n] \setminus J}$, which are the $\ell_p$ Lewis weights of the matrix $A_{[n] \setminus J,*}$.
   
   (b) For each $i \in [n] \setminus J$, if $d_{\max}\{0, p/2 - 1\} u_i \geq \frac{1}{2\alpha}$, then add $i$ into $J$.

Theorem 4.1. For a given matrix $A \in \mathbb{R}^{n \times d}$, $\tau \geq 0$ and $p \geq 1$, the algorithm in Figure 5 returns a set of indices $J \subseteq [n]$ with size $|J| \leq O\left(d_{\max}\{p/2, 1\} \cdot \alpha^2\right)$, such that for all $y \in \text{im}(A)$, if $y$ satisfies (i) $\|y\|_p \leq \alpha \cdot \tau$ and (ii) $|H_y| \leq \alpha$, then $H_y \subseteq J$.

Proof. Consider a fixed vector $y \in \text{im}(A)$ that satisfies (i) $\|y\|_p \leq \alpha \cdot \tau$ and (ii) $|H_y| \leq \alpha$. For ease of notation, we assume $|y_1| \geq |y_2| \geq \cdots \geq |y_n|$. Of course, this order is unknown and is not used by our algorithm. Under this assumption, $H_y = \{1, 2, \ldots, |H_y|\}$.

We prove $H_y \subseteq J$ by induction. For any $i < |H_y|$, suppose $[i] \subseteq J$ and $i+1 \notin J$ after the $i$-th repetition of Step 2, we show that we will add $i+1$ into $J$ in the $(i+1)$-th repetition of Step 2. Since, $[i] \subseteq J$ and $|y_1| \geq |y_2| \geq \cdots \geq |y_n|$, $\|y_{[n] \setminus J}\|_p \leq \|y_L_y\|_p + \alpha |y_{i+1}| \leq \alpha \tau + \alpha |y_{i+1}|^p$.

Since $i+1 \in H_y$, we must have $|y_{i+1}| \geq \tau$, which implies $\frac{|y_{i+1}|^p}{\|y_{[n] \setminus J}\|_p^p} \geq \frac{1}{2\alpha}$. By Lemma 3.8, this implies $d_{\max}\{0, p/2 - 1\} u_{i+1} \geq \frac{1}{2\alpha}$, where $u_{i+1}$ is the $\ell_p$ Lewis weight of the row $A_{i+1,*}$ in $A_{[n] \setminus J,*}$, in which case we will add $i+1$ into $J$. Thus, $H_y \subseteq J$ since $|H_y| \leq \alpha$.

Now we analyze the size of $J$. For the algorithm in Figure 5, we repeat the whole procedure $\alpha$ times. Each time, an index $i$ will be added into $I$ if and only if $d_{\max}\{0, p/2 - 1\} u_i \geq \frac{1}{2\alpha}$.
1. Let $|J| = O(d^{\max\{p/2,1\}} \cdot \alpha^2)$ as in Corollary 4.2.

2. Repeat the following for $O(\log(|J|/\delta_{\text{struct}}))$ times:
   
   (a) Randomly partition $[n]$ into $\Gamma_1, \Gamma_2, \ldots, \Gamma_\alpha$.
   
   (b) For each $j \in [\alpha]$, use the algorithm in Theorem 3.2 to obtain weights $\{\hat{u}_i\}_{i \in \Gamma_j}$ such that $u_i \leq \hat{u}_i \leq 2u_i$, where $\{u_i\}_{i \in \Gamma_j}$ are the $\ell_p$ Lewis weights of the matrix $A_{\Gamma, \cdot}$.
   
   (c) For each $j \in [\alpha]$, for each $i \in \Gamma_j$, if $d^{\max\{0,p/2-1\}} \hat{u}_i \geq \frac{1}{6}$,
      
      then add $i$ to $I$.

Figure 6: Algorithm for finding the set $I$.

However, since
\[
\sum_{i \in [n] \setminus J} u_i = \sum_{i \in [n] \setminus J} \hat{u}_i^p \leq d
\]
by Theorem 3.5, there are at most $O(d^{\max\{p/2,1\}} \cdot \alpha)$ such indices $i$. Thus, the total size of $J$ is upper bounded by $O(d^{\max\{p/2,1\}} \cdot \alpha^2)$.

The above algorithm also implies the following existential result.

**Corollary 4.2.** For a given matrix $A \in \mathbb{R}^{n \times d}$, $\tau \geq 0$ and $p \geq 1$, there exists a set of indices $J \subseteq [n]$ with size $|J| \leq O(d^{\max\{p/2,1\}} \cdot \alpha^2)$, such that for all $y \in \text{im}(A)$, if $y$ satisfies (i) $\|y_{L_y}\|_p \leq \alpha \cdot \tau^p$ and (ii) $|H_y| \leq \alpha$, then $H_y \subseteq J$.

### 4.2 An Input-sparsity Time Algorithm

To find a set of heavy coordinates, the algorithm in Theorem 4.1 runs in polynomial time. In this section we present an algorithm for finding heavy coordinates that runs in input-sparsity time. The algorithm is described in Figure 6.

**Theorem 4.3.** For a given matrix $A \in \mathbb{R}^{n \times d}$, $\tau \geq 0$, $\delta_{\text{struct}} \in (0,1)$, and $p \geq 1$, the algorithm in Figure 6 returns a set of indices $I \subseteq [n]$ with size $|I| \leq \tilde{O}(d^{\max\{p/2,1\}} \cdot \alpha \cdot \log(1/\delta_{\text{struct}}))$, such that with probability at least $1 - \delta_{\text{struct}}$, simultaneously for all $y \in \text{im}(A)$, if $y$ satisfies (i) $\|y_{L_y}\|_p \leq \alpha \cdot \tau^p$ and (ii) $|H_y| \leq \alpha$, then $H_y \subseteq I$. Furthermore, the algorithm runs in $\tilde{O}\left(\left(nnz(A) + d^{p/2+O(1)} \cdot \alpha\right) \cdot \log(1/\delta_{\text{struct}})\right)$ time.

**Proof.** Let $J$ be the set with size $|J| \leq O(d^{\max\{p/2,1\}} \cdot \alpha^2)$ whose existence is proved in Corollary 4.2. For all $y \in \text{im}(A)$, if $y$ satisfies (i) $\|y_{L_y}\|_p \leq \alpha \cdot \tau^p$ and (ii) $|H_y| \leq \alpha$, then $H_y \subseteq J$. We only consider those $c \in J$ for which there exists $y \in \text{im}(A)$ such that (i) $\|y_{L_y}\|_p \leq \alpha \cdot \tau^p$, (ii) $|H_y| \leq \alpha$ and (iii) $c \in H_y$, since we can remove other $c$ from $J$ and the properties of $J$ still hold.
For such \(c \in H_y\) and the corresponding \(y \in \text{im}(A)\), suppose for some \(j \in [\alpha]\) we have \(c \in \Gamma_j\). Since \(|H_y| \leq \alpha\), with probability \((1 - 1/\alpha)^{|H_y|-1} \geq 1/e\), we have \(\Gamma_j \cap H_y = \{c\}\). Furthermore, by Lemma 3.8, we must have \(E[\|y_{L_y \cap \Gamma_j}\|_p] = \|y_{L_y}\|_p/\alpha \leq \tau^p\). By Markov’s inequality, with probability at least 0.8, we have \(\|y_{L_y \cap \Gamma_j}\|_p \leq 5\tau^p\). Thus, by a union bound, with probability at least \(1/e - 0.2 > 0.1\), we have \(\|y_{L_y \cap \Gamma_j}\|_p \leq 5\tau^p\) and \(\Gamma_j \cap H_y = \{c\}\). By repeating \(O(\log(\langle J \rangle/\delta_{\text{struct}}))\) times, the success probability is at least \(1 - \delta_{\text{struct}}/|J|\). Applying a union bound over all \(c \in J\), with probability \(1 - \delta_{\text{struct}}\), the stated conditions hold for all \(c \in J\). We condition on this event in the rest of the proof.

Consider any \(c \in J\) and \(y \in \text{im}(A)\) with the properties stated above. Since \(|y_c| \geq \tau\), we have

\[
\frac{|y_c|^p}{\|y\|_p^p} \geq \frac{|y_c|^p}{\|y\cap L_y\|^p_p + |y_c|^p} \geq \frac{1}{6}.
\]

By Lemma 3.3, we must have

\[
d^{\max\{0,p/2-1\}}u_c \geq \frac{1}{6},
\]

where \(u_c\) is the \(\ell_p\) Lewis weight of the row \(A_{c,*}\) in the matrix \(A_{\Gamma_j,*}\), which also implies

\[
d^{\max\{0,p/2-1\}}\hat{u}_c \geq \frac{1}{6}
\]

since \(\hat{u}_c \geq u_c\), in which case we will add \(c\) to \(I\).

Now we analyze the size of \(I\). For each \(j \in [\alpha]\), we have

\[
\sum_{i \in \Gamma_j} \hat{u}_i \leq 2 \sum_{i \in \Gamma_j} u_i = 2 \sum_{i \in \Gamma_j} \hat{u}_i \leq 2d
\]

by Theorem 3.5. For each \(j \in [\alpha]\), there are at most \(O(d^{\max\{p/2,1\}})\) indices \(i\) which satisfy

\[
d^{\max\{0,p/2-1\}}\hat{u}_i \geq \frac{1}{6},
\]

which implies we will add at most \(O(\alpha \cdot d^{\max\{p/2,1\}})\) elements into \(I\) during each repetition. The bound on the size of \(I\) follows since there are only \(O(\log(\langle J \rangle/\delta_{\text{struct}})) = O(d \cdot \log(\langle J \rangle/\theta_{\text{struct}}))\) repetitions.

For the running time of the algorithm, since we invoke the algorithm in Theorem 3.4 for \(O(\log(\langle J \rangle/\delta_{\text{struct}}))\) times, and each time we estimate the \(\ell_p\) Lewis weights of \(A_{\Gamma_1,*}, A_{\Gamma_2,*}, \ldots, A_{\Gamma_{\alpha},*}\), which implies the running time for each repetition is upper bounded by

\[
\sum_{j=1}^{\alpha} \tilde{O}\left(\text{nnz}(A_{\Gamma_j,*}) + d^{p/2+O(1)}\right) = \tilde{O}\left(\text{nnz}(A) + d^{p/2+O(1)} \cdot \alpha\right).
\]

The bound on the running time follows since we repeat for \(O(\log(\langle J \rangle/\delta_{\text{struct}}))\) times. \(\Box\)

The above algorithm and the probabilistic method also imply the following existential result.

**Corollary 4.4.** For a given matrix \(A \in \mathbb{R}^{n \times d}, \tau \geq 0\) and \(p \geq 1\), there exists a set of indices \(I \subseteq [n]\) with size \(|I| \leq \tilde{O}(d^{\max\{p/2,1\}} \cdot \alpha)\), such that for all \(y \in \text{im}(A)\), if \(y\) satisfies (i) \(\|y_{L_y}\|_p \leq \alpha \cdot \tau^p\) and (ii) \(|H_y| \leq \alpha\), then \(H_y \subseteq I\).
5 The Net Argument

5.1 Bounding the Norm

We will generally assume that for product $Ax$, the $x$ involved is in $\text{im}(A^\top)$, which is the orthogonal complement of the nullspace of $A$; any nullspace component of $x$ would not affect $Ax$ or $SAx$, and so can be neglected for our purposes.

**Lemma 5.1.** When the entries of $A$ are integral, for any nonempty $S \subset [n]$, $\|A_{S,*}\|_2 \leq \|A\|_2 \text{CP}(A) \sqrt{d}$, and under also Assumption 2.2, $\|A_{S,*}\|_2 \leq n^{O(d^2)}$.

**Proof.** When $S$ is a nonempty proper subset of $[n]$, then since $\|A_{S,*}\|_2 \leq \|A\|_2$ and $\text{CP}(A_{S,*}) \leq \text{CP}(A)$, we have that if $\|A_{S,*}\|_2 \leq \|A\|_2 \text{CP}(A_{S,*}) \sqrt{d}$, then the lemma follows. So we can assume $S = [n]$.

First suppose $A$ has full column rank, so that $A^\top A$ is invertible. For any $y \in \mathbb{R}^n$, $A^+ y$ is the unique solution $x^*$ of $A^\top Ax = A^\top y$. Applying Cramer’s rule, the entries of $x^*$ have the form $x_i = \frac{\det B_i}{\det A \top A}$, where $B_i$ is the same as $A^\top A$, except that the $i$th column of $B_i$ is $A^\top y$. The integrality of $A$ implies $\det A \top A \geq 1$; using that together with Hadamard’s determinant inequality and the definition of the spectral norm, we have $\|x^*\|_2 \leq \|A\|_2 \text{CP}(A) \|y\|_2 \sqrt{d}$. Since this holds for any $y$, we have $\|A^+\|_2 \leq \|A\|_2 \text{CP}(A) \sqrt{d}$ as claimed.

Now suppose $A$ has rank $k < d$. Then there is $T \subset [d]$ of size $k$ whose members are indices of a set of $k$ linearly independent columns of $A$. Moreover, if $x^* = A^+ y$ is a solution to $\min_x \|Ax - y\|_2$, then there is another solution where the entries with indices in $[d] \setminus T$ are zero, since a given column not in $T$ is a linear combination of columns in $T$. That is, the solution to $\min_{x \in \mathbb{R}^k} \|A_{*,T} x - y\|_2$ can be mapped directly to a solution $x^*$ in $\mathbb{R}^k$ with the same Euclidean norm. Since $A_{*,T}$ has full column rank, the analysis above implies that

$$\|x^*\|_2 \leq \|A_{*,T}\|_2 \text{CP}(A_{*,T}) \|y\|_2 \sqrt{k} \leq \|A\|_2 \text{CP}(A) \|y\|_2 \sqrt{d},$$

so the bound on $\|A^+\|_2$ holds also when $A$ has less than full rank.

The last statement of the lemma follows directly, using the definitions of $\|A\|_2$, $\text{CP}(A)$, and Assumption 2.2.

**Lemma 5.2.** If $A$ has integral entries, and if Assumptions 1, 2, 3 hold, then Assumption 2.2 holds.

**Proof.** Let $x_{M}^{C_1}$ be a $C_1$-approximate solution of $\min_x \|Ax - b\|_M$, which Assumption 2.2 requires to have bounded Euclidean norm. Let $\hat{M}(a) \equiv \min \{\tau^p, |a|^p\}$, so that Assumptions 1 and 2 imply that $L_M \hat{M}(a) \leq M(a) \leq U_M \hat{M}(a)$ for all $a$. Letting $x_M^* \equiv \arg\min_x \|Ax - b\|_M$, and similarly defining $x_M^{*\hat{M}}$, this condition implies that

$$\|Ax_M^{C_1} - b\|_{\hat{M}} \leq \frac{1}{L_M} \|Ax_M^{C_1} - b\|_M \leq \frac{C_1}{L_M} \|Ax_M^* - b\|_M \leq C_2 \|Ax_M^* - b\|_{\hat{M}} \leq C_2 \|Ax_M^{*\hat{M}} - b\|_{\hat{M}}, \quad (3)$$

19
where \( C_2 \equiv C_1 U_M / L_M \).

Let \( S \) denote the set of indices at which \( |A_{i,s} x_{C_i}^M - b_i| \leq \tau \). If \( S \) is empty, then \( x_{C_i}^M \) can be assumed to be zero.

Similarly to our general assumption that \( x_{C_i}^M \in \text{im}(A^\top) \), we can assume that \( x_{C_i}^M \in \text{im}(A_{S,s}^T) \), since any component of \( x_{C_i}^M \) in the nullspace of \( A_{S,s} \) can be removed without changing \( A_{S,s} x_{C_i}^M \), and without increasing the \( n - |S| \) contributions of \( \tau^p \) from the remaining summands in \( \|Ax_{C_i}^M - b\|_M \).

(Here we used Assumption 1\([\ref{assumption2}]) that \( M(a) = \tau^p \) for \( |a| \geq \tau \).

From \( x_{C_i}^M \in \text{im}(A^\top) \) it follows that \( \|x_{C_i}^M\|_2 = \|A_{S,s}^+ A_{S,s} x_{C_i}^M\|_2 \leq \|A_{S,s}^+\|_2 \|A_{S,s} x_{C_i}^M\|_2 \), and since
\[
\|A_{S,s} x_{C_i}^M\|_2 \leq \sqrt{n} \|A_{S,s} x_{C_i}^M\|_p \\
\leq \sqrt{n} (\|A_{S,s} x_{C_i}^M - b_s\|_p + \|b_s\|_p) \\
\leq C_2 \sqrt{n} (\|Ax_{C_i}^M - b\|_{IM}^{1/p} + \|b_s\|_p) \quad \text{(by \((\ref{assumption3})\))} \\
\leq 2C_2 \sqrt{n} \|b\|_p,
\]
we have \( \|x_{C_i}^M\|_2 \leq \|A_{S,s}^+\|_2 \|A_{S,s} x_{C_i}^M\|_2 \leq \|A_{S,s}^+\|_2^2 C_2 \sqrt{n} \|b\|_p \), and so from Lemma 5.1 and Assumption 2\([\ref{assumption2}]) the bound on \( \|x_{C_i}^M\|_2 \) of Assumption 2\([\ref{assumption2}]) follows.

\[\square\]

### 5.2 Net Constructions

**Lemma 5.3.** Under the given assumptions, for \( U \) as in Assumption 2\([\ref{assumption2}]) there exists a set \( N_\varepsilon \subseteq \text{im}([A \ b]) \) with size \( |N_\varepsilon| \leq n^{O(d^2)} \cdot (1/\varepsilon)^{O(d)} \), such that for any \( x \) satisfying \( \|x\|_2 \leq U \), there exists \( y' \in N_\varepsilon \) such that
\[
\|(Ax - b) - y'\|_M \leq \varepsilon^p.
\]

**Proof.** Let \( \hat{M}(a) \equiv \min\{\tau^p, |a|^p\} \). Assume for now that \( \varepsilon \leq \tau / 2 \), so that if \( \|Ax\|_{\hat{M}} \leq \varepsilon^p \), then every entry of \( Ax \) is no more than \( \tau \) in magnitude, and so \( \|Ax\|_{\hat{M}}^p = \|Ax\|_{\hat{M}} \).

Let
\[
B_\varepsilon \equiv \{Ax - b \mid \|Ax - b\|_{\hat{M}} \leq \varepsilon^p\} = \{Ax - b \mid \|Ax - b\|_p \leq \varepsilon\}
\]
and
\[
B_U \equiv \{Ax - b \mid \|x\|_2 \leq U\} \subseteq \{Ax - b \mid \|Ax - b\|_p \leq \sqrt{n} \cdot (\|A\|_2 U + \|b\|_2)\}.
\]

From the scale invariance of the \( \ell_p \) norm, and the volume in at-most \( d \) dimensions, \( \text{Vol}(B_\varepsilon) \geq (\varepsilon / (\sqrt{n} \cdot (\|A\|_2 U + \|b\|_2)))^d \text{Vol}(B_U) \), so that at most \( (\sqrt{n} \cdot (\|A\|_2 U + \|b\|_2)/\varepsilon)^d \) translates of \( B_\varepsilon \) can be packed into \( B_U \) without intersecting. Thus the set \( N_\varepsilon \) of centers of such a maximal packing of translates is an \( \varepsilon^p \)-cover of \( B_U \), that is, for any point \( y \in B_U \), there is some \( y' \in N_\varepsilon \) such that \( \|y' - y\|_p \leq \varepsilon \), so that \( \|y' - y\|_{\hat{M}} \leq \varepsilon^p \).

If \( \varepsilon > \tau / 2 \), we just note that a \((\tau / 2)^p\)-cover is also an \( \varepsilon^p \)-cover, and so there is an \( \varepsilon^p \)-cover of size \( (\sqrt{n} \cdot (\|A\|_2 U + \|b\|_2)/\min\{\tau / 2, \varepsilon\})^d \).

Plugging in the bounds for \( U \) from Assumption 2\([\ref{assumption2}]) and for \( \tau, \|b\|_2, \text{ and } \|A\|_2 \leq \max_{i \in [d]} \|A_{i,s}\|_2 \) from Assumptions 2\([\ref{assumption2}]) and 2\([\ref{assumption3}]) the cardinality bound of the lemma follows.

This argument is readily adapted to more general \( \| \cdot \|_M \), by noticing that \( \|y - y'\|_M \leq U_M \cdot \|y - y'\|_{\hat{M}} \) using Assumption 1\([\ref{assumption1}]) and adjusting constants.

\[\square\]
Lemma 5.4. Under the given assumptions, there exists a set $\mathcal{M}_{e}^{\alpha,\beta} \subseteq \text{im}(A b)$ with size $|\mathcal{M}_{e}^{\alpha,\beta}| \leq O\left(\frac{\beta}{\alpha} \cdot n^{O(d^2)} \cdot (1/\varepsilon)^{O(d)}\right)$, such that for any $x$ satisfying $\alpha \leq \|Ax - b\|_p \leq \beta \leq \tau$, there exists $y' \in \mathcal{M}_{e}^{\alpha,\beta}$ such that

$$\|(Ax - b) - y'\|_M \leq \varepsilon^p \cdot \|Ax - b\|_M.$$ 

Proof. We assume $\varepsilon \leq \tau$, since otherwise we can take $\varepsilon$ to be $\tau$. By standard constructions (see, e.g., [34, p. 48]), there exists a set $\mathcal{M}_\gamma \subseteq \text{im}(A b)$ with size $|\mathcal{M}_\gamma| \leq (1/\varepsilon)^{O(d)}$, such that for any $y = Ax - b$ with $\|y\|_p = \gamma$, there exists $y' \in \mathcal{M}_\gamma$ such that $\|y - y'\|_p \leq \gamma \cdot \varepsilon$.

Let $\mathcal{M}_{e}^{\alpha,\beta}$ be

$$\mathcal{M}_{e}^{\alpha,\beta} = \mathcal{M}_{\alpha} \cup \mathcal{M}_{(1+\varepsilon)\alpha} \cup \mathcal{M}_{(1+\varepsilon)^2\alpha} \cup \cdots \cup \mathcal{M}_{\beta}.$$ 

Clearly, by Assumption 2

$$|\mathcal{M}_{e}^{\alpha,\beta}| \leq \log_{1+\varepsilon}(\beta/\alpha) \cdot n^{O(d^2)} \cdot (1/\varepsilon)^{O(d)} \leq O\left(\frac{\beta}{\alpha} \cdot n^{O(d^2)} \cdot (1/\varepsilon)^{O(d)}\right).$$

Now we show that $\mathcal{M}_{e}^{\alpha,\beta}$ satisfies the desired properties. For any $x \in \mathbb{R}^d$ such that $y = Ax - b$ satisfies $\alpha \leq \|y\|_p \leq \beta \leq \tau$, we must have $|y_i| \leq \tau$ for all entries of $y$. By normalization, there exists $\hat{y}$ such that $\|y - \hat{y}\|_p \leq \varepsilon \cdot \|y\|_p$ and $\|\hat{y}\|_p = (1 + \varepsilon)^i \cdot \alpha$ for some $i \in \mathbb{N}$. Furthermore, by the property of $\mathcal{M}_{(1+\varepsilon)^i\alpha}$, there exists $y' \in \mathcal{M}_{(1+\varepsilon)^i\alpha} \subseteq \mathcal{M}_{e}^{\alpha,\beta}$ such that $\|y - y'\|_p \leq \varepsilon \cdot \|y'\|_p \leq 2\varepsilon \cdot \|y\|_p$. Thus, by triangle inequality, we have $\|y - y'\|_p \leq 3\varepsilon \cdot \|y\|_p$. For sufficiently small $\varepsilon$, since $\|y\|_p \leq \tau$, we also have $\|y - y'\|_p \leq \tau$, which implies $\|y - y'\|_\infty \leq \tau$. Thus, using Assumption 3, we have

$$\|y - y'\|_M \leq U_M \|y - y'\|_p \leq U_M \cdot (3\varepsilon)^p \cdot \|y\|_p \leq U_M / L_M (3\varepsilon)^p \|y\|_M.$$ 

Adjusting constants implies the desired properties.

\[\square\]

5.3 The Net Argument

Theorem 5.5. For any $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, given a matrix $S \in \mathbb{R}^{r \times n}$ and a weight vector $w \in \mathbb{R}^n$ such that $w_i \geq 0$ for all $i \in [n]$. Let $c = \min_x \|Ax - b\|_p$. If there exist $U_O, U_A, L_A, L_N \leq \text{poly}(n)$ such that

1. $\|S(Ax^*_M - b)\|_{M,w} \leq U_O \|Ax^*_M - b\|_M$, where $x^*_M = \text{argmin}_x \|Ax - b\|_M$;
2. $L_A \|Ax - b\|_M \leq \|S(Ax - b)\|_{M,w} \leq U_A \|Ax - b\|_M$ for all $x \in \mathbb{R}^d$;
3. $\|Sy\|_{M,w} \geq L_N \|y\|_M$ for all $y \in \mathcal{N}_{\text{poly}(\varepsilon/\tau/\alpha)} \cup \mathcal{M}_{\text{poly}(\varepsilon/\tau/\alpha)}$,

then, any $C$-approximate solution of $\min_x \|S(Ax - b)\|_{M,w}$ with $C \leq \text{poly}(n)$ is a $C \cdot (1 + O(\varepsilon)) \cdot U_O / L_N$-approximate solution of $\min_x \|Ax - b\|_M$. Here $\mathcal{N}_{\text{poly}(\varepsilon/\tau/\alpha)}$ and $\mathcal{M}_{\text{poly}(\varepsilon/\tau/\alpha)}$ are as defined in Lemma 5.3 and Lemma 5.4, respectively.

Proof. We distinguish two cases in the proof.

21
Case 1: \((C \cdot U_M \cdot U_A/(L_M \cdot L_A)) \cdot c^p \leq \tau^p\). In this case, we prove that any C-approximate solution \(x_{S,M,w}^C\) of \(\min_x \|S(Ax - b)\|_{M,w}\) satisfies \(c \leq \|Ax_{S,M,w}^C - b\|_p \leq (C \cdot U_M \cdot U_A/(L_M \cdot L_A))^{1/p} \cdot c \leq \tau\). Let \(x_p = \arg\min_c \|Ax - b\|_p\), we have

\[
\|Ax_{S,M,w}^C - b\|_M \\
\leq \|S(Ax_{S,M,w}^C - b)\|_{M,w}/L_A \\
\leq C \cdot \|S(Ax_p^* - b)\|_{M,w}/L_A \\
\leq C \cdot \|Ax_p^* - b\|_M \cdot U_A/L_A \\
\leq C \cdot \|Ax_p^* - b\|^p (U_M \cdot U_A)/L_A \\
= C \cdot c^p \cdot (U_M \cdot U_A)/L_A.
\]

Since \(L_M \leq 1\), this implies \(\|Ax_{S,M,w}^C - b\|_M \leq \tau^p\), which implies \(\|Ax_{S,M,w}^C - b\|_\infty \leq \tau\). Thus, \(\|Ax_{S,M,w}^C - b\|_p \leq \|Ax_{S,M,w}^C - b\|_M / L_M \leq (C \cdot U_M \cdot U_A/(L_M \cdot L_A)) \cdot c^p\), which implies \(\|Ax_{S,M,w}^C - b\|_p \leq (C \cdot U_M \cdot U_A/(L_M \cdot L_A))^{1/p} \cdot c\). Moreover, by the definition of \(c\) we have \(\|Ax_{S,M,w}^C - b\|_p \geq c\).

Since \((C \cdot U_M \cdot U_A/(L_M \cdot L_A))^{1/p} \leq \text{poly}(n)\), by Lemma 2.4 there exists \(y' \in M_{c,\text{poly}(\varepsilon/n)}\) such that \(\|Ax_{S,M,w}^C - b - y'\|_M \leq \text{poly}(\varepsilon/n) \cdot \|Ax_{S,M,w}^C - b\|_M\). Notice that

\[
\|S(Ax_{S,M,w}^C - b)\|_{M,w} = \|Sy' + S((Ax_{S,M,w}^C - b) - y')\|_{M,w}.
\]

For \(Sy'\), since \(y' \in M_{c,\text{poly}(\varepsilon/n)}\), we have

\[
\|Sy'\|_{M,w} \geq L_N \|y'\|_M = L_N \|Ax_{S,M,w}^C - b + (y' - (Ax_{S,M,w}^C - b))\|_M.
\]

Since \(\|y' - (Ax_{S,M,w}^C - b)\|_M \leq \text{poly}(\varepsilon/n) \cdot \|Ax_{S,M,w}^C - b\|_M\), by Lemma 3.3 we have \(\|Ax_{S,M,w}^C - b + (y' - (Ax_{S,M,w}^C - b))\|_M \geq (1 - \varepsilon) \|Ax_{S,M,w}^C - b\|_M\), which implies \(\|Sy'\|_{M,w} \geq L_N (1 - \varepsilon) \|Ax_{S,M,w}^C - b\|_M\). On the other hand, \(\|S((Ax_{S,M,w}^C - b) - y')\|_{M,w} \leq U_A \|Ax_{S,M,w}^C - b - y'\|_M \leq \text{poly}(\varepsilon/n) \cdot \|Ax_{S,M,w}^C - b\|_M\). Again by Lemma 3.3 we have \(\|S(Ax_{S,M,w}^C - b)\|_{M,w} \geq L_N (1 - \varepsilon) \|Ax_{S,M,w}^C - b\|_M\). Furthermore, since \(x_{S,M,w}^C\) is a C-approximate solution of \(\min_x \|S(Ax - b)\|_{M,w}\), we must have

\[
\|Ax_{S,M,w}^C - b\|_M \leq (1 + O(\varepsilon)) / L_N \cdot \|S(Ax_{S,M,w}^C - b)\|_{M,w} \\
\leq C \cdot (1 + O(\varepsilon)) / L_N \cdot \|S(Ax_{S,M,w}^C - b)\|_{M,w} \\
\leq C \cdot (1 + O(\varepsilon)) \cdot U_A / L_N \cdot \|Ax_{S,M,w}^C - b\|_M.
\]

Case 2: \((C \cdot U_M \cdot U_A/(L_M \cdot L_A)) \cdot c^p \geq \tau^p\). In this case, we first prove that any C-approximate solution \(x_{S,M,w}^C\) of \(\min_x \|S(Ax - b)\|_{M,w}\) is a \(\text{poly}(n)\)-approximate solution of \(\min_x \|Ax - b\|_M\). By Assumption 2.1, this implies all C-approximate solution \(x_{S,M,w}^C\) of \(\min_x \|S(Ax - b)\|_{M,w}\) satisfies \(\|x_{S,M,w}^C\|_2^2 \leq U\).

Consider any C-approximate solution \(x_{S,M,w}^C\) of \(\min_x \|S(Ax - b)\|_{M,w}\), we have

\[
\|Ax_{S,M,w}^C - b\|_M \leq \|S(Ax_{S,M,w}^C - b)\|_{M,w}/L_A \leq C \cdot \|S(Ax_M^* - b)\|_{M,w}/L_A \\
\leq C \cdot U_A / L_A \cdot \|Ax_M^* - b\|_M \leq \text{poly}(n) \cdot \|Ax_M^* - b\|_M.
\]
We further show that \( \|Ax - b\|_M \geq \tau^n / \text{poly}(n) \) for all \( x \in \mathbb{R}^d \). If \( \|Ax - b\|_\infty \geq \tau \), then the statement clearly holds. Otherwise, \( \|Ax - b\|_M \geq L_M \|Ax - b\|_p^p \geq L_M^p \geq L_M^p \geq \tau^n / \text{poly}(n) \). Thus, for any \( C \)-approximate solution \( x^C \) of \( \min_{x} \|S(Ax - b)\|_{M,w} \), there exists \( y' \in \mathcal{N}_{\text{poly}(\tau/\varepsilon)} \) such that

\[
\|y' - (Ax^C_{M,w} - b)\|_M \leq \text{poly}(\varepsilon \cdot \tau/\varepsilon) \leq \text{poly}(\varepsilon/\varepsilon) \cdot \|Ax^C_{M,w} - b\|_M.
\]

The rest of the proof is exactly the same as that of Case 1.

\[\square\]

6 A Row Sampling Algorithm for Tukey Loss Functions

In this section we present the row sampling algorithm. The row sampling algorithm proceeds in a recursive manner. We describe a single recursive step in Section 6.1 and the overall algorithm in Section 6.2.

6.1 One Recursive Step

The goal of this section is to design one recursive step of the row sampling algorithm. For a weight vector \( w \in \mathbb{R}^n \), the recursive step outputs a sparser weight vector \( w' \in \mathbb{R}^n \) such that for any set \( \mathcal{N} \subseteq \text{im}(A) \) with size \(|\mathcal{N}|\), with probability at least \( 1 - \delta_\alpha \), simultaneously for all \( y \in \mathcal{N} \),

\[
\|y\|_{M,w'} = (1 \pm \varepsilon)\|y\|_{M,w}.
\]

We maintain that if \( w_i \neq 0 \), then \( w_i \geq 1 \) and \( \|w\|_\infty \leq n^2 \) as an invariant in the recursion. These conditions imply that we can partition the positive coordinates of \( w \) into \( 2 \log n \) groups \( P_j \), for which \( P_j = \{ i \mid 2^{j-1} \leq w_i < 2^j \} \).

Now we define one recursive step of our sampling procedure. We split the matrix \( A \) into \( A_{P_1}, A_{P_2}, \ldots, A_{P_{2\log n}} \), and deal with each of them separately. For each \( 1 \leq j \leq 2 \log n \), we invoke the algorithm in Theorem 4.3 to identify a set \( I_j \) for the matrix \( A_{P_j} \), for some parameters \( \alpha \) and \( \delta_\text{struct} \) to be determined. For each \( 1 \leq j \leq 2 \log n \), we also use the algorithm in Theorem 3.4 to calculate \( \{\hat{u}_i\}_{i \in P_j} \) such that \( u_i \leq \hat{u}_i \leq 2 u_i \) where \( \{u_i\}_{i \in P_j} \) are the \( \ell_p \) Lewis weights of the matrix \( A_{P_j} \).

Now for each \( i \in P_j \), we define its sampling probability \( p_i \) to be

\[
p_i = \begin{cases} 1 & i \in I_j \\ \min\{1, 1/2 + \Theta(d^\text{max}(0,p/2-1)\hat{u}_i \cdot Y)\} & i \notin I_j \end{cases},
\]

where \( Y \equiv d \log(1/\varepsilon) + \log(\log n/\delta_\alpha) + U_M/L_M \log(|\mathcal{N}| \cdot \log n/\delta_\alpha)/\varepsilon^2 \).

For each \( i \in [n] \), we set \( w'_i = 0 \) with probability \( 1 - p_i \), and set \( w'_i = w_i / p_i \) with probability \( p_i \). The finishes the definition of one step of the sampling procedure.

Let

\[
F \equiv \sum_{1 \leq j \leq 2 \log n} |I_j| + \sum_{1 \leq j \leq 2 \log n} \sum_{i \in P_j \cap I_j} \Theta(d^\text{max}(0,p/2-1)\hat{u}_i \cdot Y).
\]

Our first lemma shows that with probability at least \( 1 - \delta_\alpha \), the number of non-zero entries in \( w' \) is at most \( \frac{2}{3} \|w\|_0 \), provided \( \|w\|_0 \) is large enough.
Lemma 6.1. When \( \|w\|_0 \geq 10F \), with probability at least \( 1 - \delta_o \),
\[
\|w\|_0 \leq \frac{2}{3}\|w\|_0.
\]

Proof. Notice that
\[
E[\|w\|_0] \leq \|w\|_0/2 + F.
\]
By Bernstein’s inequality in Lemma 3.1, since \( F \geq \Omega(\log(1/\delta_o)) \), with probability at least \( 1 - \exp(-\Omega(F)) \) we have
\[
\|w\|_0 \leq \|w\|_0/2 + F + \|w\|_0/10 \leq \frac{2}{3}\|w\|_0.
\]

Our second lemma shows that \( \|w\|_\infty \) is upper bounded by \( 2\|w\|_\infty \).

Lemma 6.2. \( \|w\|_\infty \leq 2\|w\|_\infty \).

Proof. Since \( p_i \geq 1/2 \) for all \( i \in [n] \), we have \( \|w\|_\infty \leq 2\|w\|_\infty \).

We show that for sufficiently large constant \( C \), if we set
\[
\alpha = C \cdot \frac{U_M}{L_M} \cdot \log(|N| \cdot \log n/\delta_o) / \varepsilon^2
\]
and \( \delta_{\text{struct}} = \delta_o / (4 \log n) \), then with probability at least \( 1 - \delta_o \), simultaneously for all \( y \in N \) we have
\[
\|y\|_{M,w} = (1 \pm \varepsilon)\|y\|_{M,w}.
\]
By Theorem 4.3 and Theorem 3.5 since
\[
\sum_{1 \leq j \leq 2 \log n} \sum_{i \in P_j \setminus I_j} \hat{u}_i \leq O(d \log n),
\]
this also implies
\[
F = \tilde{O}(d^{\max(1,p/2)} \log n \cdot (\log(|N|/\delta_o) \cdot \log(1/\delta_o) + d)) / \varepsilon^2.
\]
Furthermore, for each \( 1 \leq j \leq 2 \log n \), we invoke the algorithm in Theorem 3.4 and the algorithm in Theorem 4.3 on \( A_{P_1,s}, A_{P_2,s}, \ldots, A_{P_{2\log n,s}} \), and thus the running time of each recursive step is thus upper bounded by
\[
\tilde{O}((\text{nnz}(A) + d^{p/2+O(1)} \cdot \alpha) \cdot \log(1/\delta_{\text{struct}})) = \tilde{O}((\text{nnz}(A) + d^{p/2+O(1)} \cdot \log(|N|/\delta_o) \cdot \varepsilon^2) \cdot \log(1/\delta_o)).
\]
Now we consider a fixed vector \( y \in \text{im}(A) \). We use the following two lemmas in our analysis.

Lemma 6.3. With probability \( 1 - \delta_o / O(|N| \cdot \log n) \), the following holds:

- If \( \|y_{H_y \cap P_j}\|_{M,w} \geq C \cdot U_M \cdot \tau^p \cdot 2^{j-1} \cdot \log(|N| \cdot \log n / \delta_o) / \varepsilon^2 \), then
  \[
  \|y_{H_y \cap P_j}\|_{M,w} = (1 \pm \varepsilon/2)\|y_{H_y \cap P_j}\|_{M,w};
  \]
• If \( \| y_{H_y \cap P_j} \|_{M,w} < C \cdot U_M \cdot \tau^p \cdot 2^{j-1} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon^2 \), then

\[
\| y_{H_y \cap P_j} \|_{M,w'} - \| y_{H_y \cap P_j} \|_{M,w} \leq C \cdot U_M \cdot \tau^p \cdot 2^{j-2} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon.
\]

**Proof.** For each \( i \in H_y \cap P_j \), we use \( Z_i \) to denote the random variable

\[
Z_i = \begin{cases} w_i M(y_i) / p_i & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i. \end{cases}
\]

Since \( Z_i = w_i M(y_i) \), we have

\[
\| y_{H_y \cap P_j} \|_{M,w'} = \sum_{i \in H_y \cap P_j} Z_i.
\]

It is clear that \( Z_i \leq 2^{j+1} \cdot U_M \cdot \tau^p \) since \( p_i \geq 1/2 \) and \( w_i \leq 2^j \), \( E[Z_i] = w_i M(y_i) \) and \( E[Z_i^2] = w_i^2 (M(y_i))^2 / p_i \). By Hölder’s inequality,

\[
\sum_{i \in H_y \cap P_j} E[Z_i^2] \leq 2^{j+1} \cdot \| y_{H_y \cap P_j} \|_{M,w} \cdot U_M \cdot \tau^p.
\]

Thus by Bernstein’s inequality in Lemma 3.1, we have

\[
\Pr \left[ \left| \sum_{i \in H_y \cap P_j} Z_i - \| y_{H_y \cap P_j} \|_{M,w} \right| \geq t \right] \leq 2 \exp \left( - \frac{t^2}{2 \cdot \tau^p \cdot \tau^p \cdot t / 3 + 2^{j+2} \cdot \| y_{H_y \cap P_j} \|_{M,w} \cdot U_M \cdot \tau^p} \right).
\]

When \( \| y_{H_y \cap P_j} \|_{M,w} \geq C \cdot U_M \cdot \tau^p \cdot 2^{j-1} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon^2 \), we take

\[
t = \varepsilon / 2 \cdot \| y_{H_y \cap P_j} \|_{M,w} \geq C \cdot U_M \cdot \tau^p \cdot 2^{j-2} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon.
\]

By taking \( C \) to be some sufficiently large constant, with probability at least \( 1 - \delta_o / O(\|N\| \cdot \log n) \),

\[
\| y_{H_y \cap P_j} \|_{M,w'} = (1 \pm \varepsilon / 2) \| y_{H_y \cap P_j} \|_{M,w}.
\]

When \( \| y_{H_y \cap P_j} \|_{M,w} < C \cdot U_M \cdot \tau^p \cdot 2^{j-1} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon^2 \), we take

\[
t = C \cdot U_M \cdot \tau^p \cdot 2^{j-2} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon.
\]

By taking \( C \) to be some sufficiently large constant, with probability at least \( 1 - \delta_o / O(\|N\| \cdot \log n) \),

\[
\| y_{H_y \cap P_j} \|_{M,w'} - \| y_{H_y \cap P_j} \|_{M,w} \leq C \cdot U_M \cdot \tau^p \cdot 2^{j-2} \cdot \log(\|N\| \cdot \log n / \delta_o) / \varepsilon.
\]

The proof of the following lemma is exactly the same as Lemma 6.3

**Lemma 6.4.** With probability \( 1 - \delta_o / O(\|N\| \cdot \log n) \), the following holds:
Lemma 6.5. If we set $\alpha = C \cdot U_M / L_M \cdot \log(|N| \cdot \log n / \delta_0) / \varepsilon^2$, $\delta_{\text{struct}} = \delta_0 / (4 \log n)$, then for each $1 \leq j \leq 2 \log n$, with probability at least $1 - \delta_0 / (2 \log n)$, simultaneously for all $y \in N$,

$$\|y_{P_j}\|_{M,w'} = (1 \pm \varepsilon)\|y_{P_j}\|_{M,w}.$$ 

Applying a union bound over all $1 \leq j \leq 2 \log n$, with probability at least $1 - \delta_0$, simultaneously for all $y \in N$,

$$\|y\|_{M,w'} = (1 \pm \varepsilon)\|y\|_{M,w}.$$

Proof. By Theorem 4.3, for each $1 \leq j \leq 2 \log n$, with probability $1 - \delta_0 / (4 \log n)$, simultaneously for all $y \in N \subseteq \text{im}(A)$, if $y$ satisfies (i) $\|y_{L_y \cap P_j}\|_p^p \leq \alpha \cdot \tau^p$ and (ii) $|H_y \cap P_j| \leq \alpha$, then we have $H_y \cap P_j \subseteq I_j$. We condition on this event in the remaining part of the proof.

Now we consider a fixed $y \in N$. We show that $\|y_{P_j}\|_{M,w'} = (1 \pm \varepsilon)\|y_{P_j}\|_{M,w}$ with probability at least $1 - \delta_0 / O(|N| \cdot \log n)$. The desired bound follows by applying a union bound over all $y \in N$.

We distinguish four cases in our analysis. We use $T$ to denote a fixed threshold

$$T = C \cdot U_M \cdot \tau^p \cdot 2^{j-1} \cdot \log(|N| \cdot \log n / \delta_0) / \varepsilon^2.$$ 

Case (i): $\|y_{H_y \cap P_j}\|_{M,w} < T$ and $\|y_{L_y \cap P_j}\|_{M,w} < T$. Since $\|y_{H_y \cap P_j}\|_{M,w} < T$, we must have

$$|H_y \cap P_j| < C \cdot U_M / L_M \cdot \log(|N| \cdot \log n / \delta_0) / \varepsilon^2 = \alpha.$$ 

Furthermore, we also have

$$\|y_{L_y \cap P_j}\|_p^p < C \cdot U_M / L_M \cdot \tau^p \cdot \log(|N| \cdot \log n / \delta_0) / \varepsilon^2 = \alpha \cdot \tau^p.$$ 

By Lemma 3.9 with probability at least $1 - \delta_0 / O(|N| \cdot \log n)$, we have

$$\|y_{P_j \setminus I_j}\|_{M,w'} = (1 \pm \varepsilon)\|y_{P_j \setminus I_j}\|_{M,w},$$

since $H_y \cap P_j \subseteq I_j$. Moreover, $\|y_{I_j}\|_{M,w} = \|y_{I_j}\|_{M,w}$ since $w_i = w'_i$ for all $i \in I_j$. Thus, we have $\|y_{P_j}\|_{M,w'} = (1 \pm \varepsilon)\|y_{P_j}\|_{M,w}$.

Case (ii): $\|y_{H_y \cap P_j}\|_{M,w} \geq T$ and $\|y_{L_y \cap P_j}\|_{M,w} \geq T$. By Lemma 6.3 and Lemma 6.4 with probability at least $1 - \delta_0 / O(|N| \cdot \log n)$,

$$\|y_{H_y \cap P_j}\|_{M,w'} = (1 \pm \varepsilon / 2)\|y_{H_y \cap P_j}\|_{M,w},$$

and

$$\|y_{L_y \cap P_j}\|_{M,w'} = (1 \pm \varepsilon / 2)\|y_{L_y \cap P_j}\|_{M,w},$$

which implies

$$\|y_{P_j}\|_{M,w'} = (1 \pm \varepsilon / 2)\|y_{P_j}\|_{M,w}$$

26
Case (iii): $\|y_{H_y \cap P_j}\|_{M,w} \geq T$ and $\|y_{L_y \cap P_j}\|_{M,w} < T$. By Lemma 6.3 and Lemma 6.4 with probability at least $1 - \delta_0 / O(|N| \cdot \log n)$,

$$\|y_{H_y \cap P_j}\|_{M,w'} = (1 \pm \epsilon/2)\|y_{H_g \cap P_j}\|_{M,w}$$

and

$$\|y_{L_y \cap P_j}\|_{M,w'} - \|y_{L_y \cap P_j}\|_{M,w} \leq C \cdot U_M \cdot \tau^p \cdot 2^{i-2} \cdot \log(|N| \cdot \log n / \delta_0) / \epsilon.$$

Since

$$\|y_{P_j}\|_{M,w} \geq \|y_{H_y \cap P_j}\|_{M,w} \geq T \geq C \cdot U_M \cdot \tau^p \cdot 2^{i-1} \cdot \log(|N| \cdot \log n / \delta_0) / \epsilon^2,$$

we have

$$\|y_{L_y \cap P_j}\|_{M,w'} - \|y_{L_y \cap P_j}\|_{M,w} \leq \epsilon/2 \cdot \|y_{P_j}\|_{M,w},$$

which implies

$$\|y_{P_j}\|_{M,w'} = (1 \pm \epsilon)\|y_{P_j}\|_{M,w}.$$

Case (iv): $\|y_{H_y \cap P_j}\|_{M,w} < T$ and $\|y_{L_y \cap P_j}\|_{M,w} \geq T$. By Lemma 6.3 and Lemma 6.4 with probability at least $1 - \delta_0 / O(|N| \cdot \log n)$,

$$\|y_{L_y \cap P_j}\|_{M,w'} = (1 \pm \epsilon/2)\|y_{L_y \cap P_j}\|_{M,w}$$

and

$$\|y_{H_y \cap P_j}\|_{M,w'} - \|y_{H_y \cap P_j}\|_{M,w} \leq C \cdot U_M \cdot \tau^p \cdot 2^{i-2} \cdot \log(|N| \cdot \log n / \delta_0) / \epsilon.$$

Since

$$\|y_{P_j}\|_{M,w} \geq \|y_{L_y \cap P_j}\|_{M,w} \geq T \geq C \cdot U_M \cdot \tau^p \cdot 2^{i-1} \cdot \log(|N| \cdot \log n / \delta_0) / \epsilon^2,$$

we have

$$\|y_{H_y \cap P_j}\|_{M,w'} - \|y_{H_y \cap P_j}\|_{M,w} \leq \epsilon/2 \cdot \|y_{P_j}\|_{M,w},$$

which implies

$$\|y_{P_j}\|_{M,w'} = (1 \pm \epsilon)\|y_{P_j}\|_{M,w}.$$

Now we show that with probability $1 - \delta_0$, simultaneously for all $x \in \mathbb{R}^d$, $\|Ax\|_{p,w'} = (1 \pm \epsilon)\|Ax\|_{p,w}$.

**Lemma 6.6.** For any $1 \leq j \leq 2 \log n$, with probability at least $1 - \delta_0 / (2 \log n)$, simultaneously for all $y = Ax$,

$$\|y_{P_j}\|_{p,w'} = (1 \pm \epsilon)\|y_{P_j}\|_{p,w}. $$

Applying a union bound over all $1 \leq j \leq 2 \log n$, this implies with probability at least $1 - \delta_0$,

$$\|y\|_{p,w'} = (1 \pm \epsilon)\|y\|_{p,w}.$$

**Proof.** For any fixed $1 \leq j \leq 2 \log n$, by Theorem 3.10 if we take $\delta_{\text{subspace}} = \delta_0 / (2 \log n)$, with probability at least $1 - \delta_0 / (2 \log n)$, simultaneously for all $y = Ax$, we have

$$\|y_{P_j \setminus I_j}\|_{p,w'} = (1 \pm \epsilon)\|y_{P_j \setminus I_j}\|_{p,w}. $$

Moreover, $\|y_{I_j}\|_{p,w} = \|y_{I_j}\|_{p,w'}$ since $w_i = w'_i$ for all $i \in I_j$. Thus, we have $\|y_{P_j}\|_{p,w'} = (1 \pm \epsilon)\|y_{P_j}\|_{p,w}$. 

\[ \square \]
6.2 The Recursive Algorithm

We start by setting \( w = 1^\circ \). In each recursive step, we use the sampling procedure defined in Section 6.1 to obtain \( w' \), by setting \( \delta_0 = \delta / O(\log n) \) and \( \varepsilon = \varepsilon' / O(\log n) \) for some \( \varepsilon' > 0 \). By Lemma 6.5, and a union bound over all the \( \log n \) recursive depths, with probability at least \( 1 - \delta / (10 \log n) \), we have \( \|w'\|_0 \leq 2 / 3 \|w\|_0 \).

We repeat the recursive step until \( \|w\|_0 \leq 10^F \).

By applying a union bound over all recursive steps, with probability \( 1 - \delta / 10 \), the recursive depth is at most \( \log_{3/2} n \). By Lemma 6.2, this also implies with probability \( 1 - \delta / 10 \), during the whole recursive algorithm, the weight vector \( w \) always satisfies \( \|w\|_\infty \leq 2^{\log_{1.5} n} \leq n^2 \). If we use \( w_{\text{final}} \) to denote the final weight vector, then we have

\[
\|w_{\text{final}}\|_0 \leq 10^F = \tilde{O}(d^{\max\{1, p/2\}} \log n \cdot (\log(|N|/\delta_0) \cdot \log(1/\delta_0) + d)/\varepsilon^2).
\]

By Lemma 6.3 and a union bound over all the \( \log_{1.5} n \) recursive depths, with probability \( 1 - \delta \), simultaneously for all \( y \in N \), we have

\[
\|Ax\|_{M, w_{\text{final}}} = (1 \pm O(\varepsilon \cdot \log n)) \|Ax\|_M = (1 \pm O(\varepsilon')) \|Ax\|_M.
\]

Moreover, by Lemma 6.6 and a union bound over all the \( \log_{1.5} n \) recursive depths, with probability \( 1 - \delta / 10 \), simultaneously for all \( y = Ax \) we have

\[
\|Ax\|_{p, w_{\text{final}}}^p = (1 \pm O(\varepsilon \cdot \log n)) \|Ax\|_{p, w}^p = (1 \pm O(\varepsilon')) \|Ax\|_{p, w}^p.
\]

We further show that conditioned on this event, simultaneously for all \( x \in \mathbb{R}^d \),

\[
\|Ax\|_{M, w_{\text{final}}} \geq \frac{L_M}{U_M \cdot n} \cdot \|Ax\|_M.
\]

Consider a fixed vector \( x \in \mathbb{R}^d \), if there exists a coordinate \( i \in H_Ax \) such that \( w_i > 0 \), since \( w_i \geq 1 \) if \( w_i > 0 \), we must have

\[
\|Ax\|_{M, w_{\text{final}}} \geq w_i M((Ax)_i) \geq M((Ax)_i) \geq L_M \cdot \tau^p.
\]

On the other hand,

\[
\|Ax\|_M \leq n \cdot U_M \cdot \tau^p,
\]

which implies

\[
\|Ax\|_{M, w_{\text{final}}} \geq \frac{L_M}{U_M \cdot n} \cdot \|Ax\|_M.
\]

Otherwise, \( i \in L_Ax \) for all \( i \in [n] \), which implies

\[
\|Ax\|_{M, w_{\text{final}}} \geq L_M \cdot \|Ax\|_{p, w_{\text{final}}}^p \geq (1 - O(\varepsilon')) L_M \|Ax\|_{p, w}^p \geq \frac{(1 - O(\varepsilon')) L_M}{U_M} \|Ax\|_M.
\]

Finally, since each recursive step runs in \( \tilde{O}((\text{nnz}(A) + d^{p/2+O(1)} \cdot \log(|N|/\delta) \cdot \varepsilon^2) \cdot \log(1/\delta)) \) time, and the number of recursive steps is upper bounded by \( \log_{1.5} n \) with probability \( 1 - \delta / 10 \), the total running time is also upper bounded by \( \tilde{O}((\text{nnz}(A) + d^{p/2+O(1)} \cdot \log(|N|/\delta) \cdot \varepsilon^2) \cdot \log(1/\delta)) \) with probability \( 1 - \delta / 10 \).

The following lemma can be proved by applying a union bound over all observations above, changing \( \varepsilon' \) to \( \varepsilon \) and changing \( A \) to \( [A \ b] \).
Lemma 6.7. The algorithm outputs a vector $w_{\text{final}} \in \mathbb{R}^n$, such that for any set $\mathcal{N} \subseteq \text{im}(A b)$ with size $|\mathcal{N}|$, with probability $1 - \delta$, the algorithm runs in $\tilde{O}((\text{nnz}(A) + d^{p/2} + O(1)/\varepsilon^2) \cdot \log(1/\delta))$ time and the following holds:

1. $\|w_{\text{final}}\|_0 \leq \tilde{O}(d^{\max\{1,p/2\}} \log^3 n \cdot (\log(|\mathcal{N}|/\delta) \cdot \log(1/\delta) + d)/\varepsilon^2)$;
2. $\|w_{\text{final}}\|_\infty \leq n^2$;
3. For all $x \in \mathbb{R}^d$, $\|A x - b\|_{M,w_{\text{final}}} \geq \frac{L_{\mathcal{N}}}{L_{Nn}} \cdot \|A x - b\|_M$.
4. For all $x \in \mathcal{N}$, $\|A x - b\|_{M,w_{\text{final}}} = (1 \pm \varepsilon)\|A x - b\|_M$.

Combining Lemma 6.7 with the net argument in Theorem 5.5, we have the following theorem.

Theorem 6.8. By setting $|\mathcal{N}| = n^{O(p^3)} \cdot (1/\varepsilon)^O(d)$, the algorithm outputs a vector $w_{\text{final}} \in \mathbb{R}^n$, such that with probability $1 - \delta$, the algorithm runs in $\tilde{O}((\text{nnz}(A) + d^{p/2} + O(1)/\varepsilon^2) \cdot \log(1/\delta))$ time, $\|w_{\text{final}}\|_0 \leq \tilde{O}(d^{p/2} + O(1)) \log^4 n \cdot \log^2(1/\delta)/\varepsilon^2)$ and any $C$-approximate solution of $\min_x \|A x - b\|_M$, with $C \leq poly(n)$, is a $C \cdot (1 + \varepsilon)$-approximate solution of $\min_x \|A x - b\|_M$.

Proof. Lemma 6.7 implies that $U_O = 1 + \varepsilon$, $L_N = 1 - \varepsilon$, $L_A = L_{\mathcal{N}}/L_{Nn}$ and $U_A \leq \|w_{\text{final}}\|_\infty \leq n^2$. Adjusting constants and applying Theorem 5.5 imply the desired result. □

7 The $M$-sketch

In this section we give an oblivious sketch for Tukey loss functions. Throughout this section we assume $1 \leq p \leq 2$ in Assumption 1.

For convenience and to set up notation, we first describe the construction.

The sketch. Each coordinate $z_p$ of a vector $z$ to be sketched is mapped to a level $h_p$, and the number of coordinates mapped to level $h$ is exponentially small in $h$: for an integer branching factor $b > 1$, we expect the number of coordinates at level $h$ to be about a $b^{-h}$ fraction of the coordinates. The number of buckets at a given level is $N = bc m$, where integers $m, c > 1$ are parameters to be determined later.

Our sketching matrix is $S \in \mathbb{R}^{N h_{\max} \times n}$, where $h_{\max} \equiv \lfloor \log_b(n/m) \rfloor$. Our weight vector $w \in \mathbb{R}^{N h_{\max}}$ has entries $w_{i+1} \leftarrow \beta b^h$, for $i \in \lfloor Nh, N(h+1) \rfloor$ and integer $h = 0, 1, \ldots, h_{\max}$, and $\beta \equiv (b - b^{-h_{\max}})/(b - 1)$. Our sketch is reminiscent of sketches in the data stream literature, where we hash into buckets at multiple levels of subsampling [17, 31]. However, the estimation performed in the sketch space needs to be the same as in the original space, which necessitates a new analysis.

The entries of $S$ are $S_{j,p} \leftarrow \Lambda_p$, where $p \in [n]$ and $j \leftarrow g_p + Nh_p$, and

$$\Lambda_p \leftarrow \pm 1 \text{ with equal probability}$$

$$g_p \in [N] \text{ chosen with equal probability}$$

$$h_p \leftarrow h \text{ with probability } 1/\beta b^h \text{ for integer } h \in [0, h_{\max}],$$

all independently. Let $L_h$ be the multiset $\{z_p | h_p = h\}$, and $L_{h,i}$ the multiset $\{z_p | h_p = h, g_p = i\}$; that is, $L_h$ is the multiset of values at a given level, $L_{h,i}$ is the multiset of values in a bucket. We can write $\|S z\|_{M,w}$ as $\sum_{h \in [0,h_{\max}], i \in [N]} \beta b^h M(\|L_{h,i}\|_\Lambda)$, where $\|L\|_\Lambda$ denotes $|\sum_{z_p \in L} \Lambda_p z_p|$. 

29
7.1 Accuracy Bounds for Sketching One Vector

We will show that our sketching construction has the property that for a given vector \( z \in \mathbb{R}^n \), with high probability, \( \|Sz\|_{M,w} \) is not too much smaller than \( \|z\|_M \). We assume that \( \|z\|_M = 1 \), for notational convenience.

Define \( y \in \mathbb{R}^n \) by \( y_p = M(z_p) \), so that \( \|y\|_1 = \|z\|_M = 1 \). Let \( Z \) denote the multiset comprising the coordinates of \( z \), and let \( Y \) denote the multiset comprising the coordinates of \( y \). For \( \hat{Z} \subset Z \), let \( M(\hat{Z}) \subset Y \) denote \( \{M(z_p) \mid z_p \in \hat{Z}\} \). Let \( \|Y\|_k \) denote \( \left( \sum_{y \in Y} |y|^k \right)^{1/k} \), so \( \|Y\|_1 = \|y\|_1 \). Hereafter multisets will just be called “sets”.

Weight classes. Fix a value \( \gamma > 1 \), and for integer \( q \geq 1 \), let \( W_q \) denote the multiset comprising weight class \( \{y_p \in Y \mid \gamma^{-q} \leq y_p \leq \gamma^{1-q}\} \). We have \( \beta b^h \mathbb{E}[\|M(L_h) \cap W_q\|_1] = \|W_q\|_1 \). For a set of integers \( Q \), let \( W_Q \) denote \( \cup_{q \in Q} W_q \).

Defining \( q_{\text{max}} \) and \( h(q) \). For given \( \varepsilon > 0 \), consider \( y' \in \mathbb{R}^n \) with \( y'_i \leftarrow y_i \) when \( y_i > \varepsilon/n \), and \( y'_i \leftarrow 0 \) otherwise. Then \( \|y'\|_1 \geq 1 - n(\varepsilon/n) = 1 - \varepsilon \). We can neglect \( W_q \) for \( q > q_{\text{max}} \equiv \log_\gamma(n/\varepsilon) \), up to error \( \varepsilon \). Moreover, we can assume that \( \|W_q\|_1 \geq \varepsilon/q_{\text{max}} \), since the contribution to \( \|y\|_1 \) of weight classes \( W_q \) of smaller total weight, added up for \( q \leq q_{\text{max}} \), is at most \( \varepsilon \).

Let \( h(q) \) denote \( \left[ \log_b(|W_q|/\beta m) \right] \) for \( |W_q| \geq \beta m \), and zero otherwise, so that
\[
m \leq \mathbb{E}[\|M(L_{h(q)}) \cap W_q\|] \leq bm
\]
for all \( W_q \) except those with \( |W_q| < \beta m \), for which the lower bound does not hold.

Since \( |W_q| \leq n \) for all \( q \), we have \( h(q) \leq [\log_b(n/\beta m)] \leq h_{\text{max}} \).

7.2 Contraction Bounds

Here we will show that \( \|Sz\|_{M,w} \) is not too much smaller than \( \|z\|_M \). We will need some weak conditions among the parameters. Recall that \( N = bcm \).

Assumption 3. We will assume \( b \geq m \), \( b > c \), \( m = \Omega(\log \log(n/\varepsilon)) \), \( \log b = \Omega(\log \log(n/\varepsilon)) \), \( \gamma \geq 2 \geq \beta \), an error parameter \( \varepsilon \in [1/10, 1/3] \), and for \( \log N \leq \varepsilon^2 m \). We will consider \( \gamma \) to be fixed throughout, that is, not dependent on the other parameters.

We need lemmas that allow lower bounds on the contributions of the weight classes. First, some notation. For \( h = 0, 1, \ldots, h_{\text{max}} \), let
\[
M_\leq \equiv \log_\gamma(m/\varepsilon) = O(\log_\gamma(b/\varepsilon))
Q_\leq \equiv \{q \mid |W_q| < \beta m, q \leq M_\leq\}
\hat{Q}_h \equiv \{q \mid h(q) = h, |W_q| \geq \beta m\}
M_\geq \equiv \log_\gamma(2(1+3\varepsilon)b/\varepsilon)
Q_h \equiv \{q \in \hat{Q}_h \mid q \leq M_\geq + \min q\}_{q \in Q_h}
Q^* \equiv Q_\leq \cup [\cup_{h} Q_h].
\]

Here \( Q_\leq \) is the set of indices of weight classes that have relatively few members, but contain relatively large weights. \( \hat{Q}_h \) gives the indices of \( W_q \) that are “large” and have \( h \) as the level at

30
which between $m$ and $bm$ members of $W_q$ are expected in $L_h$. The set $Q_h$ cuts out the weight classes that can be regarded as negligible at level $h$.

**Lemma 7.1.** If $N \geq \max\{O(|M|d^3\varepsilon), \tilde{O}(d^2m^2/\varepsilon^2)\}$, then with constant probability, for all $z \in \text{im}(A)$ and all $q \in Q_<$, the following event $E_{\omega}$ holds: there are sets $W_q^* \subset W_q$, with $|W_q^*| \geq (1 - \varepsilon)|W_q|$, such that for all $y \in W_q^*$:

1. they are isolated: they are the sole members of $W_{Q_<}$ in their bucket;

2. their buckets are low-weight: the set $L$ of other entries in bucket containing $y \in W_q^*$ has $\|L\|_1 \leq 1/\varepsilon^2m^3$.

**Proof.** Without loss of generality we assume $h(q)$ are the same for all $q \in M_<$, since otherwise we can deal with each $h(q)$ separately.

Let $\alpha = m/(L \cdot \varepsilon)$. By Lemma 4.4, there exists a set $I \subseteq [n]$ with size $|I| = \tilde{O}(d \cdot \alpha) = \tilde{O}(d \cdot m/\varepsilon)$ such that for any $z \in \text{im}(A)$, if $z$ satisfies (i) $\|z_L\|_p^p \leq \alpha \cdot \tau^p$ and (ii) $|H_z| \leq \alpha$, then $H_z \subseteq I$. Let $\{u_i\}_{i \in [n] \setminus I}$ be the $\ell_p$ Lewis weights of $A_{[n]} \setminus \tau^*$, and let $J \subseteq [n] \setminus I$ be the set of indices of the $d \cdot m/\varepsilon \cdot U_M/L_M$ largest coordinates of $u$. Thus, $|J| \leq O(d \cdot m/\varepsilon)$. Since $J$ contains the $d \cdot m/\varepsilon \cdot U_M/L_M$ largest coordinates of $u$ and

$$\sum_{i \in [n] \setminus I} u_i = \sum_{i \in [n] \setminus I} \overline{u}_i^p \leq d$$

by Theorem 3.5, for each $i \in [n] \setminus (I \cup J)$, we have $u_i \leq d/(d \cdot m/\varepsilon \cdot U_M/L_M) \leq \varepsilon/m \cdot L_M/U_M$.

If $\tau^p < \|z\|_M \cdot \varepsilon/m$, by Assumption 12 we have $M(z_i) \leq \tau^p < \|z\|_M \cdot \varepsilon/m$ for all $i \in [n]$. In this case, we have $W_{Q_<} = \emptyset$. Thus we assume $\tau^p \geq \|z\|_M \cdot \varepsilon/m$ in the remaining part of the analysis.

Since $\|z\|_M \geq |H_z| \cdot \tau^p$, we have $|H_z| \leq m/\varepsilon$. Furthermore, by Assumption 1.4, $\|z_L\|_p^p \leq \|z_L\|_M/L_M \leq \|z\|_M/L_M \leq \tau^p \cdot m/(L_M \cdot \varepsilon)$. Thus by setting $\alpha = m/(L_M \cdot \varepsilon)$ we have $H_z \subseteq I$.

For each $i \in [n] \setminus I$, we have $|z_i| \leq \tau$. By Lemma 3.8 and Assumption 1.4 for each $i \in [n] \setminus I$, $M(z_i) \leq \tau^p \cdot L_M \leq u_i \cdot \|z_i\|_I^p/\|L_M\| \leq u_i \cdot \|z_i\|_I \cdot M \cdot U_M/L_M < u_i \cdot \|z\|_M \cdot U_M/L_M$. Thus for each entry $i \in [n] \setminus (I \cup J)$, we have $M(z_i) < \varepsilon/m \cdot \|z\|_M$.

Thus, the indices of all members of $W_{Q_<}$ are in $I \cup J$. By setting $N \geq |I \cup J|/\kappa = \tilde{O}(d^2m^2/\varepsilon^2)/\kappa$, the expected number of total collisions in $I \cup J$ is $|I \cup J|^2/N \leq \kappa$. Thus, by Markov’s inequality, with probability $1 - 2\kappa$, the total number of collisions is upper bounded by $1/2$, i.e., there is no collision. This implies the first condition.

For the second condition, we use $\{u_i\}_{i \in [n] \setminus (I \cup J)}$ to denote the $\ell_p$ Lewis weights of $A_{[n] \setminus (I \cup J)}$.

Consider a fixed $q \in M_\kappa$. By the first condition, all elements in $W_q$ are the sole members of $W_{Q_<}$ in their buckets. For each bucket we define $B_{h,i}$ to be the multiset $\{u_p \mid h_p = h, g_p = i, p \in [n] \setminus (I \cup J)\}$.

By setting $N \geq \frac{\gamma M\cdot|M_\kappa|}{d^3\varepsilon}$, for each $y \in W_q$, $E[\|B_{h,i}\|_1] \leq d/N \leq \frac{L_M}{U_M} \cdot \frac{1}{\varepsilon^2m^2} \cdot \frac{\varepsilon^2}{|M_\kappa|}$ where $L_{h,i}$ is the bucket that contains $y$. This is simply because $\sum_{i \in N} B_{h,i} \leq \sum_{i \in [n] \setminus (I \cup J)} u_i \leq d$ by Theorem 3.5. We say a bucket is *good* if $\|B_{h,i}\|_1 \leq \frac{L_M}{U_M} \cdot \frac{1}{\varepsilon^2m^2}$. Notice that for $y \in W_q$, if $y$ is in a good bucket
Lemma 7.3 (Lemma 3.6 of [11]). For large enough $N$, the two conditions and setting $\kappa$ Lemma 7.2 (Lemma 3.8 of [11])

Let $M$ and if $q \in C$ and for a constant condition holds with probability at least $1 - \varepsilon^2 m$.

Thus, it suffices to show that at least $(1 - \varepsilon) |W_q|$ buckets associated with $y \in W_q$ are good.

By Markov's inequality, for each $y \in W_q$, with probability $1 - \varepsilon \cdot \kappa / |M_\prec|$, the bucket that contains $y$ is good. Thus, for the $|W_q|$ buckets associated with $y \in W_q$, the expected number of good buckets is at least $(1 - \varepsilon \cdot \kappa / |M_\prec|)|W_q|$. Again, by Markov's inequality, with probability at least $1 - \kappa / |M_\prec|$, at least $(1 - \varepsilon)|W_q|$ buckets associated with $y \in W_q$ are good, and we just take these $(1 - \varepsilon)|W_q|$ good buckets to be $W_q^*$. By applying a union bound over all $q \in M_\prec$, the second condition holds with probability at least $1 - \kappa$. The lemma follows by applying a union bound over the two conditions and setting $\kappa$ to be a small constant.

Lemma 7.2 (Lemma 3.8 of [11]). Let $Q_h' \equiv \{ q \mid q \leq M_h' \}$, where $M_h' \equiv \log \gamma (\beta b^{h+1} m^2 q_{\max})$. Then for large enough $N = O(m^2 b \varepsilon^{-1} q_{\max})$, with probability at least $1 - C^{-\varepsilon^2 m}$ for a constant $C > 1$, for each $q \in \cup_h Q_h$, there is $W_q^* \subset L_{h(q)} \cap W_q$ such that:

1. $|W_q^*| \geq (1 - \varepsilon) \beta^{-1} b^{-h(a)} |W_q|$.

2. each $x \in W_q^*$ is in a bucket with no other member of $W_{Q_h'}$.

3. $\|W_q^*\|_1 \geq (1 - 4 \gamma \varepsilon) \beta^{-1} b^{-h} \|W_q\|_1$.

4. each $x \in W_q^*$ is in a bucket with no member of $W_{Q_h'}$.

For $v \in T \subset Z$, let $T - v$ denote $T \setminus \{v\}$.

Lemma 7.3 (Lemma 3.6 of [11]). For $v \in T \subset Z$,

$$M(\|T\|_A) \geq \left(1 - \frac{\|T - v\|_A}{|v|}\right)^2 M(v),$$

and if $M(v) \geq \varepsilon^{-1} \|T - v\|_M$, then

$$\frac{\|T - v\|_2}{|v|} \leq \varepsilon^{1/2},$$

and for a constant $C$, $E_A[M(\|T\|_A)] \geq (1 - C \varepsilon^{1/2})M(v)$.
Lemma 7.4 (Lemma 3.9 of [11]). Assume Assumption \( \mathcal{A} \). There is \( N = O(\varepsilon^{-2}m^2bq_{\max}) \), so that for all \( 0 \leq h \leq h_{\max} \) and \( q \in Q_h \) with \( \|W_q\|_1 \geq \varepsilon/q_{\max} \), we have

\[
\sum_{y_p \in W_q^*} M(\|L(y_p)\|_A) \geq (1 - \varepsilon^{1/2})\|W_q\|_1
\]

with failure probability at most \( C^{-\varepsilon^2m} \) for fixed \( C > 1 \).

Lemma 7.5. Assume that \( \mathcal{E}_v \) of Lemma 7.1 holds, and Assumption \( \mathcal{A} \). Then for \( q \in Q_< \),

\[
\sum_{y_p \in W_q^*} M(\|L(y_p)\|_A) \geq (1 - \varepsilon^{1/2})\|W_q\|_1
\]

with failure probability at most \( C^{-\varepsilon^2m} \) for a constant \( C > 1 \).

Proof. Let \( v \equiv z_p \) where \( y_p = M(z_p) \), let \( L(v) \) denote the \( \{z_p \mid M(z_p) \in L\} \). Condition \( \mathcal{E}_v \) and \( M(v) \geq \varepsilon/m \) imply that

\[
\|L(v) - v\|^2_2 \leq \|L\|_1 \leq 1/\varepsilon^2m^3 < M(v)/\varepsilon m,
\]

so that using (6) we have

\[
\frac{\|L(v) - v\|^2_2}{\|v\|^2} \leq \frac{\|L(v) - v\|_M}{M(v)} \leq \frac{1}{\varepsilon^2m^3}.
\]

(7)

Since \( \|L\|_\infty \leq \|L\|_1 \), we also have, for all \( v' \in L(v) - v \), and using again \( M(v) \geq \varepsilon/m \),

\[
\frac{\|v'\|}{\|v\|} \leq \left(\frac{M(v')}{M(v)}\right)^{1/2} \leq \frac{1}{\varepsilon^2m}.\]

(8)

From (8), we have that the summands determining \( \|L(v) - v\|_A \) have magnitude at most \( |v|\varepsilon^{1/2}/\varepsilon^2m \). From (7), we have \( \|L(v) - v\|_2^2 \) is at most \( v^2\varepsilon/\varepsilon^2m \). It follows from Bernstein’s inequality that with failure probability \( \exp(-\varepsilon^2m) \), \( \|L(v) - v\|_A \leq \varepsilon^{1/2}|v| \). Applying the first claim of Lemma 7.3 we have \( M(\|L(v)\|_A) \geq (1 - 2\varepsilon^{1/2})M(v) \), for all \( v \in M^{-1}(W_q^*) \) with failure probability \( \beta m M_\lambda \exp(-\varepsilon^2m) \). Summing over \( W_q^* \), we have

\[
\sum_{v \in M^{-1}(W_q^*)} M(\|L(v)\|_A) \geq (1 - \varepsilon^{1/2})\|W_q^*\|_1 \geq (1 - 2\varepsilon)(1 - \varepsilon^{1/2})\|W_q\|_1.
\]

This implies the bound, using Assumption \( \mathcal{A} \) after adjusting constants. \( \square \)

The above lemmas imply that overall, with high probability, the sketching-based estimate of \( \|z\|_M \) of a single given vector \( z \) is very likely to not much smaller than \( \|z\|_M \), as stated next.

Theorem 7.6 (Theorem 3.2 of [11]). Assume Assumption \( \mathcal{A} \) and condition \( \mathcal{E}_v \) of Lemma 7.1. Then \( \|Sz\|_{M,w} \geq \|z\|_M(1 - \varepsilon^{1/2}) \), with failure probability no more than \( C^{-\varepsilon^2m} \), for an absolute constant \( C > 1 \).
7.3 A “Clipped” Version

For a vector $z$, we use $\|Sz\|_{M_c,w}$ to denote a “clipped” version of $\|Sz\|_{M,w}$, in which we ignore small buckets and use a subset of the coordinates of $Sz$ as follows: $\|Sz\|_{M_c,w}$ is obtained by adding in only those buckets in level $h$ that are among the top

$$M^* \equiv bmM_\geq + \beta mM_\leq$$

in $\|L_{h,i}\|_\Lambda$, recalling $M_\geq$ and $M_\leq$ defined in (3). Formally, we define $\|Sz\|_{M_c,w}$ to be

$$\|Sz\|_{M_c,w} = \sum_{h \in [0,h_{\max}], i \in [M^*]} \beta b^h M(\|L_{h,(i)}\|_\Lambda),$$

where $L_{h,(i)}$ denotes the level $h$ bucket with the $i$-th largest $\|L_{h,i}\|_\Lambda$ among all the level $h$ buckets.

The proof of the contraction bound of $\|Sz\|_{M,w}$ in Theorem 7.6 requires only lower bounds on $M(\|L_{h,i}\|_\Lambda)$ for those at most $M^*$ buckets on level $h$. Thus, the proven contraction bounds continue to hold for $\|Sz\|_{M_c,w}$, and in particular $\|Sz\|_{M_c,w} \geq (1 - \varepsilon)\|Sz\|_{M,w}$.

7.4 Dilation Bounds

We use two prior bounds of [11] on dilation; the first shows that the dilation is at most $O(\log n)$ in expectation, while the second shows that the “clipped” version gives $O(1)$ dilation with constant probability. Note that we need only expectations, since we need the dilation bound to hold only for the optimal solution as in Theorem 5.5.

**Theorem 7.7** (Theorem 3.3 of [11]). $E[\|Sz\|_{M_c,w}] = O(h_{\max})\|z\|_M$.

Better dilation is achieved by using the “clipped” version $\|Sz\|_{M_c,w}$, as described in [11].

**Theorem 7.8** (Theorem 3.4 of [11]). There is $c = O(\log \gamma(b/\varepsilon)/(\log_b(n/m)))$ and $b \geq c$, recalling $N = mbc$, such that

$$E[\|Sz\|_{M_c,w}] \leq C\|z\|_M$$

for a constant $C$.

7.5 Regression Theorem

**Lemma 7.9.** There is $N = O(d^2 h_{\max})$, so that with constant probability, simultaneously for all $x \in \mathbb{R}^d$,

$$0.9/(n \cdot U_M/L_M)\|Ax - b\|_M \leq \|S(Ax - b)\|_{M,w} \leq U_M/L_M \cdot n^2 \cdot \|Ax - b\|_M.$$

**Proof.** For the upper bound,

$$\|Sz\|_{M,w} = \sum_{h \in [0,h_{\max}], i \in [N]} \beta b^h M(\|L_{h,(i)}\|_\Lambda).$$
The weights $\beta b^h$ are less than $n$, and
\[
M(\|L_{h,i}\|_\Lambda) \\
\leq M(\|L_{h,i}\|_1) \\
\leq M(n^{1-1/p}\|L_{h,i}\|_p) \quad \text{(Assumption 1.2)} \\
\leq U_M \cdot n^{p-1}\|L_{h,i}\|_p \quad \text{(Assumption 1.4)} \\
\leq U_M / L_M \cdot n \cdot \sum_{z_p \in L_{h,i}} M(z_p). \quad \text{(Assumption 1.4)}
\]

Since any given $z_p$ contributes once to $\|S_z\|_{M,w}$, $\|S_z\|_{M,w} \leq U_M / L_M \cdot n^2 \cdot \|z\|_M$.

For the lower bound, notice that
\[
\|S_z\|^2_{2,w} = \sum_{h \in [0,h_{\max}], i \in [N]} \beta b^h \|L_{h,i}\|^2_\Lambda.
\]

For each $h \in [0,h_{\max}]$, since $N = O(d^2 h_{\max})$, with probability at least $1 - 1/(10 h_{\max})$, simultaneously for all $z \in \text{im}(A)$ we have
\[
\sum_{i \in [N]} \|L_{h,i}\|^2_\Lambda = (1 \pm 0.1) \sum_{z_p \in L_h} z_p^2,
\]
since the summation on the left-hand side can be equivalently viewed as applying CountSketch \cite{8, 25, 23} on $L_h$. Thus, by applying union bound over all $h \in [0, h_{\max}]$, we have
\[
\|S_z\|^2_{2,w} = \sum_{h \in [0,h_{\max}], i \in [N]} \beta b^h \|L_{h,i}\|^2_\Lambda \geq 0.9 \|z\|^2_2. \quad (9)
\]

If there exists some $i \in H_{S_z}$, since $w_i \geq 1$ for all $i$, we have
\[
\|S_z\|_{M,w} \geq w_i M((S_z)_i) \geq M((S_z)_i) \geq \tau^p.
\]

On the other hand,
\[
\|z\|_M \leq n \cdot U_M \cdot \tau^p,
\]
which implies
\[
\|S_z\|_{M,w} \geq \|z\|_M / (n \cdot U_M).
\]

If $H_{S_z} = \emptyset$, then
\[
\|S_z\|_{M,w} \\
\geq \sum_i w_i ((S_z)_i)^p \cdot L_M \quad \text{(Assumption 1.4)} \\
= \|S_z\|_{p,w}^p \cdot L_M \\
\geq \|S_z\|^p_{2,w} \cdot L_M \quad (p \leq 2) \\
\geq 0.9 \|z\|^p_2 \cdot L_M \quad (9) \\
\geq 0.9 \|z\|^p_{p} \cdot L_M / n \\
\geq 0.9 \|z\|_{M}/(n \cdot U_M / L_M). \quad \text{(Assumption 1.4)}
\]
The following theorem states that $M$-sketches can be used for Tukey regression, under the conditions described above.

**Theorem 7.10.** Under Assumption 1 and Assumption 2, there is an algorithm running in $O(\text{nnz}(A))$ time, that with constant probability creates a sketched regression problem $\min_x \|S(Ax - b)\|_{M,w}$ where $SA$ and $Sb$ have $\text{poly}(d \log n)$ rows, and any $C$-approximate solution $\hat{x}$ of $\min_x \|S(Ax - b)\|_{M,w}$ with $C \leq \text{poly}(n)$ satisfies

$$\|A\hat{x} - b\|_M \leq O(C \cdot \log_d n) \min_{x \in \mathbb{R}^d} \|Ax - b\|_M.$$ 

Moreover, any $C$-approximate solution $\hat{x}$ of $\min_x \|S(Ax - b)\|_{M,c,w}$ with $C \leq \text{poly}(n)$ satisfies

$$\|A\hat{x} - b\|_M \leq O(C) \min_{x \in \mathbb{R}^d} \|Ax - b\|_M.$$ 

**Proof.** We set $S$ to be an $M$-sketch matrix with large enough $N = \text{poly}(d \log n)$. We note that, up to the trivial scaling by $\beta$, $SA$ satisfies Assumption 2 if $A$ does. We also set $m = O(d^3 \log n)$, and $\varepsilon = 1/10$. We apply Theorem 5.5 to prove the desired result.

The given $N$ is large enough for Theorem 7.6 and Lemma 7.9 to apply, obtaining a contraction bound with failure probability $1 - m$. By Theorem 7.6 the needed contraction bound holds for all members of $N_{\text{poly}(\varepsilon \cdot \tau/n)} \cup M_{\text{poly}(\varepsilon/n)}$, with failure probability $n^{O(d^3)}C_1^{-m} < 1$, for $m = O(d^3 \log n)$, assuming the condition $\mathcal{E}_u$.

Thus, by Theorem 7.7 we have $U_0 = O(\delta \log_d n)$. By Lemma 7.9, $L_A = 0.9/(n \cdot U_M/L_M)$ and $U_A = U_M/L_M \cdot n^2$. By Theorem 7.6 $L_N = 1 - \varepsilon^{1/2} = \Omega(1)$. Thus, by Theorem 5.5 we have

$$\|A\hat{x} - b\|_M \leq O(C \cdot \log_d n) \min_{x \in \mathbb{R}^d} \|Ax - b\|_M.$$ 

A similar argument holds for $C$-approximate solution $\hat{x}$ of $\min_x \|S(Ax - b)\|_{M,c,w}$. \qed

8 Hardness Results and Provable Algorithms for Tukey Regression

8.1 Hardness Results

In this section, we prove hardness results for Tukey regression based on the Exponential Time Hypothesis [16]. We first state the hypothesis.

**Hypothesis 1** (Exponential Time Hypothesis [16]). For some constant $\delta > 0$, no algorithm can solve 3-SAT on $n$ variables and $m = O(n)$ clauses correctly with probability at least $2/3$ in $O(2^{\delta n})$ time.

Using Dinur’s PCP Theorem [14], Hypothesis 1 implies a hardness result for MAX-3SAT.

**Theorem 8.1** ([14]). Under Hypothesis 1, for some constant $\varepsilon > 0$ and $c > 0$, no algorithm can, given a 3-SAT formula on $n$ variables and $m = O(n)$ clauses, distinguish between the following cases correctly with probability at least $2/3$ in $2^{n/\log^c n}$ time:

- There is an assignment that satisfies all clauses in $\phi$;
• Any assignment can satisfy at most \((1 - \varepsilon)m\) clauses in \(\phi\).

We make the following assumptions on the loss function \(M : \mathbb{R} \to \mathbb{R}^+\). Notice that the following assumptions are more general than those in Assumption [1]

**Assumption 4.** There exist real numbers \(\tau \geq 0\) and \(C > 0\) such that

1. \(M(x) = C\) for all \(|x| \geq \tau\).
2. \(0 \leq M(x) \leq C\) for all \(|x| \leq \tau\).
3. \(M(0) = 0\).

Now we give an reduction that transforms a 3-SAT formula \(\phi\) with \(d\) variables and \(m = O(d)\) clauses to a Tukey regression instance

\[
\min_x \|Ax - b\|_M,
\]

such that \(A \in \mathbb{R}^{n \times d}\) and \(b \in \mathbb{R}^n\) with \(n = O(d)\), and all entries in \(A\) are in \(\{0, +1, -1\}\) and all entries in \(b\) are in \(\{\pm k\tau \mid k \in \mathbb{N}, k \leq O(1)\}\). Furthermore, there are at most three non-zero entries in each row of \(A\).

For each variable \(v_i\) in the formula \(\phi\), there is a variable \(x_i\) in the Tukey regression that corresponds to \(v_i\). For each variable \(v_i\), if \(v_i\) appears in \(\Gamma_i\) clauses in \(\phi\), we add \(2\Gamma_i\) rows into \([A\ b]\). These \(2\Gamma_i\) rows are chosen such that when calculating \(\|Ax - b\|_M\), there are \(\Gamma_i\) terms of the form \(M(x_i)\), and another \(\Gamma_i\) terms of the form \(M(x_i - 10\tau)\). This can be done by taking the \(i\)-th entry of the corresponding row of \(A\) to be 1 and taking the corresponding entry of \(b\) to be either 0 or \(10\tau\). Since \(\sum_{i=1}^d \Gamma_i = 3m\) in a 3-SAT formula \(\phi\), we have added \(6m = O(d)\) rows into \([A\ b]\). We call these rows Part I of \([A\ b]\).

Now for each clause \(C \in \phi\), we add three rows into \([A\ b]\). Suppose the three variables in \(C\) are \(v_i, v_j\) and \(v_k\). The first row is chosen such that when calculating \(\|Ax - b\|_M\), there is a term of the form \(M(a + b + c - 10\tau)\), where \(a = x_i\) if there is a positive literal that corresponds to \(v_i\) in \(C\) and \(a = 10\tau - x_i\) if there is a negative literal that corresponds to \(v_i\) in \(C\). Similarly, \(b = x_j\) if there is a positive literal that corresponds to \(v_j\) in \(C\) and \(b = 10\tau - x_j\) if there is a negative literal that corresponds to \(v_j\) in \(C\). The same holds for \(c, x_k\), and \(v_k\). The second and the third row are designed such that when calculating \(\|Ax - b\|_M\), there is a term of the form \(M(a + b + c - 20\tau)\) and another term of the form \(M(a + b + c - 30\tau)\). Clearly, this can also be done while satisfying the constraint that all entries in \(A\) are in \(\{0, +1, -1\}\) and all entries in \(b\) are in \(\{\pm k\tau \mid k \in \mathbb{N}, k \leq O(1)\}\). We have added \(3m\) rows into \([A\ b]\). We call these rows Part II of \([A\ b]\).

This finishes our construction, with \(6m + 3m = O(d)\) rows in total. It also satisfies all the restrictions mentioned above.

Now we show that when \(\phi\) is satisfiable, if we are given any solution \(\overline{x}\) such that

\[
\|A\overline{x} - b\|_M \leq (1 + \eta) \min_x \|Ax - b\|_M,
\]

then we can find an assignment to \(\phi\) that satisfies at least \((1 - 5\eta)m\) clauses.

We first show that when \(\phi\) is satisfiable, the regression instance we constructed satisfies

\[
\min_x \|Ax - b\|_M \leq 5C \cdot m.
\]
We show this by explicitly constructing a vector $x$. For each variable $v_i$ in $\phi$, if $v_i = 0$ in the satisfiable assignment, then we set $x_i$ to be 0. Otherwise, we set $x_i$ to be $10\tau$. For each variable $v_i$, since $x_i \in \{0, 10\tau\}$, for all the $2\Gamma_i$ rows added for it, there will be $\Gamma_i$ rows contributing 0 when calculating $\|Ax - b\|_M$, and another $\Gamma_i$ rows contributing $C$ when calculating $\|Ax - b\|_M$. The total contribution from this part will be $3C \cdot m$. For each clause $C \in \phi$, for the three rows added for it, there will be one row contributing 0 when calculating $\|Ax - b\|_M$, and another two rows contributing $C$ when calculating $\|Ax - b\|_M$. This is by construction of $[A \, b]$ and by the fact that $C$ is satisfied. Notice that $M(a + b + c - 10\tau) = 0$ if only one literal in $C$ is satisfied, $M(a + b + c - 20\tau) = 0$ if two literals are satisfied, and $M(a + b + c - 30\tau) = 0$ if all three literals in $C$ are satisfied. Thus, we must have $\min_x \|Ax - b\|_M \leq 5C \cdot m$, which implies $\|A\mathbf{x} - b\|_M \leq (1 + \eta)5C \cdot m$.

We first show that we can assume each $\bar{x}_i$ satisfies $\bar{x}_i \in [-\tau, \tau]$ or $\bar{x}_i \in [9\tau, 11\tau]$. This is because we can set $\bar{x}_i = 0$ otherwise without increasing $\|A\mathbf{x} - b\|_M$, as we will show immediately. For any $\bar{x}_i$ that is not in the two ranges mentioned above, its contribution to $\|A\mathbf{x} - b\|_M$ in Part I will be at most $C \cdot \Gamma_i$. Thus, by setting $\bar{x}_i = 0$ the total contribution to $\|A\mathbf{x} - b\|_M$ in Part I has been decreased by at least $C \cdot \Gamma_i$. Now we consider Part II of the rows in $[A \, b]$. The contribution to $\|A\mathbf{x} - b\|_M$ of all rows in $[A \, b]$ created for clauses that do not contain $v_i$, their contribution to $\|A\mathbf{x} - b\|_M$ is lower bounded by $C \cdot 2\Gamma_i$, and upper bounded by $C \cdot 3\Gamma_i$. The lower bound follows since for any three real numbers $a$, $b$ and $c$, at least two elements in $\{a + b + c - 10\tau, a + b + c - 20\tau, a + b + c - 30\tau\}$ have absolute value at least $\tau$, and $M(x) = C$ for all $|x| \geq \tau$. Thus, by setting $\bar{x}_i = 0$ the total contribution to $\|A\mathbf{x} - b\|_M$ in Part II will be increased by at most $C \cdot \Gamma_i$, which implies we can set $\bar{x}_i = 0$ without increasing $\|A\mathbf{x} - b\|_M$.

Now we show how to construct an assignment to the 3-SAT formula $\phi$ which satisfies at least $(1 - 5\eta)m$ clauses, using a vector $\bar{x} \in \mathbb{R}^d$ which satisfies (i) $\|A\bar{x} - b\|_M \leq (1 + \eta)5C \cdot m$ and (ii) $\bar{x}_i \in [-\tau, \tau]$ or $\bar{x}_i \in [9\tau, 11\tau]$ for all $\bar{x}_i$. We set $v_i = 0$ if $\bar{x}_i \in [-\tau, \tau]$ and set $v_i = 1$ if $\bar{x}_i \in [9\tau, 11\tau]$. To count the number of clauses satisfied by the assignment, we show that for each clause $C \in \phi$, $C$ is satisfied whenever $a + b + c \geq 7\tau$. Recall that $a = x_j$ if there is a positive literal that corresponds to $v_i$ in $C$ and $a = 10\tau - x_j$ if there is a negative literal that corresponds to $v_i$ in $C$. Similarly, $b = x_j$ if there is a positive literal that corresponds to $v_j$ in $C$ and $b = 10\tau - x_j$ if there is a negative literal that corresponds to $v_j$ in $C$. The same holds for $c, x_k$, and $v_k$. Since $a$, $b$ and $c$ are all in the range $[-\tau, \tau]$ or in the range $[9\tau, 11\tau]$, whenever $a + b + c \geq 7\tau$, we must have $a \geq 9\tau$, $b \geq 9\tau$ or $c \geq 9\tau$, in which case clause $C$ will be satisfied. Thus, at least $(1 - 5\eta)m$ clauses will be satisfied, since otherwise $\|A\mathbf{x} - b\|_M$ will be larger than $3C \cdot m + 2C \cdot m + 5\eta C \cdot m = (1 + \eta)5C \cdot m$. Here the first term $3C \cdot m$ corresponds to the contribution from Part I, since any $\bar{x}_i$ must satisfy $|\bar{x}_i| \geq \tau$ or $|\bar{x}_i| - 10\tau \geq \tau$. The second and the third term $2C \cdot m + 5\eta C \cdot m$ corresponds to the contribution from Part II when at least $5\eta m$ clauses are not satisfied.

Our reduction implies the following theorem.

**Theorem 8.2.** Suppose there is an algorithm that runs in $T(d)$ time and succeeds with probability $2/3$ for Tukey regression with approximation ratio $1 + \eta$ when the loss function $M$ satisfies Assumption [7] and the input data satisfies the following restrictions:

1. $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$ with $n = O(d)$.

2. All entries in $A$ are in $\{0, +1, -1\}$ and all entries in $b$ are in $\{\pm k\tau \mid k \in \mathbb{N}, k \leq O(1)\}$.
3. There are at most three non-zero entries in each row of $A$.

Then, there exists an algorithm that runs in $T(d)$ time for a 3-SAT formula on $d$ variables and $m = O(d)$ clauses which distinguishes between the following cases correctly with probability at least $2/3$:

- There is an assignment that satisfies all clauses in $\phi$.
- Any assignment can satisfy at most $(1 - 5\eta)m$ clauses in $\phi$.

Combining Theorem 8.1 and Theorem 8.2 with the Hypothesis 1, we have the following corollary.

**Corollary 8.3.** Under Hypothesis 1, for some constant $\eta > 0$ and $C > 0$, no algorithm can solve Tukey regression with approximation ratio $1 + \eta$ and success probability $2/3$, and runs in $2^{d/\log C}d$ time, when the loss function $M$ satisfies Assumption 4 and the input data satisfies the following restrictions:

1. $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$ with $n = O(d)$.
2. All entries in $A$ are in $\{0, +1, -1\}$ and all entries in $b$ are in $\{\pm k\tau \mid k \in \mathbb{N}, k \leq O(1)\}$.
3. There are at most three non-zero entries in each row of $A$.

**8.2 Provable Algorithms**

In this section, we use the polynomial system verifier to develop provable algorithms for Tukey regression.

**Theorem 8.4** ([27, 3]). Given a real polynomial system $P(x_1, x_2, \ldots, x_d)$ with $d$ variables and $n$ polynomial constraints $\{f_i(x_1, x_2, \ldots, x_d) \Delta_i 0\}_{i=1}^n$, where $\Delta_i$ is any of the “standard relations”: $\{>, \geq, =, \neq, \leq, <\}$, let $D$ denote the maximum degree of all the polynomial constraints and let $H$ denote the maximum bitsize of the coefficients of all the polynomial constraints. Then there exists an algorithm that runs in

$$(Dn)^{O(d)} \text{poly}(H)$$

time that can determine if there exists a solution to the polynomial system $P$.

Besides Assumption 1, we further assume that the loss function $M(x)$ can be approximated by a polynomial $P(x)$ with degree $D$, when $|x| \leq \tau$. Formally, we assume there exist two constants $L_P \leq 1 \leq U_P$ such that when $|x| \leq \tau$, we have

$$L_P P(|x|) \leq M(|x|) \leq U_P P(|x|).$$

Indeed, Assumption 1 already implies we can take $P(x) = x^p$, with $L_P = L_M$ and $U_P = U_M$ when $p$ is an integer. However, for some loss function (e.g., the one defined in 1), one can find a better polynomial to approximate the loss function. Since the approximation ratio of our algorithm depends on $U_P/L_P$, for those loss functions we can get an algorithm with better approximation ratio. We also assume Assumption 2 and all entries in $A$ and $b$ are integers.

We first show that under Assumption 2 and the assumption that all entries in $A$ and $b$ are integers, either $\|Ax - b\|_M = 0$ for some $x \in \mathbb{R}^d$, or $\|Ax - b\|_M \geq 1/2^{\text{poly}(nd)}$ for all $x \in \mathbb{R}^d$. 

39
Lemma 8.5. Suppose all entries in $A$ and $b$ are integers, under Assumption 1 and Assumption 2 either $\|Ax - b\|_M = 0$ for some $x \in \mathbb{R}^d$, or $\|Ax - b\|_M \geq 1/2^{\text{poly}(nd)}$ for all $x \in \mathbb{R}^d$.

Proof. We show that either there exists $x \in \mathbb{R}^d$ such that $Ax = b$, or $\|Ax - b\|_2 \geq 1/2^{\text{poly}(nd)}$ for all $x \in \mathbb{R}^d$. Notice that $\|Ax - b\|_2 \geq 1/2^{\text{poly}(nd)}$ implies $\|Ax - b\|_\infty \geq 1/2^{\text{poly}(nd)}/\sqrt{n}$, and thus the claimed bound follows from Assumption 1.

Without loss of generality we assume $A$ is non-singular. By the normal equation, we know $x^* = (A^TA)^{-1}(A^Tb)$ is an optimal solution to $\min_x \|Ax - b\|_2$. By Cramer’s rule, all entries in $x^*$ are either 0 or have absolute value at least $1/2^{\text{poly}(nd)}$. This directly implies either $Ax^* - b = 0$ or $\|Ax^* - b\|_2 \geq 1/2^{\text{poly}(nd)}$.

Lemma 8.5 implies that either $\|Ax - b\|_M = 0$ for some $x \in \mathbb{R}^d$, or $\|Ax - b\|_M \geq 1/2^{\text{poly}(nd)}$ for all $x \in \mathbb{R}^d$. The former case can be solved by simply solving the linear system $Ax = b$. Thus we assume $\|Ax - b\|_M \geq 1/2^{\text{poly}(nd)}$ for all $x \in \mathbb{R}^d$ in the rest part of this section.

To solve the Tukey regression problem $\min_x \|Ax - b\|_M$, we apply a binary search to find the optimal solution value $\text{OPT}$. Since $1/2^{\text{poly}(nd)} \leq \text{OPT} \leq n \cdot \tau^p \leq 2^{\text{poly}(nd)}$ by Assumption 1 and Assumption 2 the binary search makes at most $\log(2^{\text{poly}(nd)}/\varepsilon) = \text{poly}(nd) + \log(1/\varepsilon)$ guesses to the value of $\text{OPT}$ to find a $(1 + \varepsilon)$-approximate solution.

For each guess $\lambda$, we need to decide whether there exists $x \in \mathbb{R}^d$ such that $\|Ax - b\|_M \leq \lambda$ or not. We use the polynomial system verifier in Theorem 8.4 to solve this problem. We first enumerate a set of coordinates $S \subseteq [n]$, which are the coordinates with $|(Ax^* - b)_i| \geq \tau$, where $x^* = \text{argmin}_x \|Ax - b\|_M$, and then solve the following decision problem:

$$
\sum_{i \in [n] \setminus S} P(\sigma_i(Ax - b)_i) + |S| \cdot \tau^p \leq \lambda
$$

s.t. $\sigma_i^2 = 1, \forall i \in [n] \setminus S$

$$
0 \leq \sigma_i(Ax - b)_i \leq \tau, \forall i \in [n] \setminus S.
$$

Clearly, $\sigma_i(Ax - b)_i = |(Ax - b)_i|$, and thus $L_P P(\sigma_i(Ax - b)_i) \leq M((Ax - b)_i) \leq U_P P(\sigma_i(Ax - b)_i)$. Thus by Assumption 1 for all $x \in \mathbb{R}^d$ and $S \subseteq [n]$,

$$
L_P \|Ax - b\|_M \leq \sum_{i \in [n] \setminus S} P(\sigma_i(Ax - b)_i) + |S| \cdot \tau^p.
$$

Moreover,

$$
\sum_{i \in [n] \setminus S} P(\sigma_i(Ax^* - b)_i) + |S| \cdot \tau^p \leq U_P \|Ax^* - b\|_M
$$

when $S = \{i \in [n] \mid |(Ax^* - b)_i| \geq \tau\}$, which implies the binary search will return a $((1 + \varepsilon) \cdot U_P / L_P)$-approximate solution.

Now we analyze the running time of the algorithm. We make at most $\text{poly}(nd) + \log(1/\varepsilon)$ guesses to the value of $\text{OPT}$. For each guess, we enumerate a set of coordinates $S$, which takes $O(2^n)$ time. For each set $S \subseteq [n]$, we need to solve the decision problem mentioned above, which has $n + d$ variables and $O(n)$ polynomial constraints with degree at most $D$. By Theorem 8.4 this decision problem can be solved in $(nD)^{O(n)}$ time. Thus, the overall time complexity is upper bounded by $(nD)^{O(n)} \cdot \log(1/\varepsilon)$.

Notice that we can apply the row sampling algorithm in Theorem 6.8 to reduce the size of the problem before applying this algorithm. This reduces the running time from $(nD)^{O(n)} \cdot \log(1/\varepsilon) = 2^{O(n \cdot \log n + \log D)} \cdot \log(1/\varepsilon)$ to $2^{O(D \cdot \log D \cdot p/2 \cdot \text{poly}(d \cdot \log n)/\varepsilon^2)}$. Formally, we have the following theorem.
Theorem 8.6. Under Assumption 1 and 2 and suppose all entries in $A$ and $b$ are integers, and there exists a polynomial $P(x)$ with degree $D$ and two constants $L_P \leq 1 \leq U_M$ such that when $|x| \leq \tau$, we have

$$L_P P(|x|) \leq M(|x|) \leq U_P P(|x|).$$

Then there exists an algorithm that returns a $((1+\varepsilon)\cdot U_P/L_P)$-approximate solution to $\min_x \|Ax - b\|_M$ and runs in $2^{O(d\log D \cdot d^{p/2} \cdot \text{poly}(d\log n)/\varepsilon^2)}$ time.

Corollary 8.7. Under Assumption 2 and suppose all entries in $A$ and $b$ are integers, for the loss function $M$ defined in (1) there exists an algorithm that returns a $(1 + \varepsilon)$-approximate solution to $\min_x \|Ax - b\|_M$ and runs in $2^{O(\text{poly}(d\log n)/\varepsilon^2)}$ time.

9 Experiments

In this section we provide experimental results to illustrate the practicality of our dimensionality reduction methods. Figure 7 shows the approximation ratio of our dimensionality reduction methods, when applied to synthetic and real datasets. For all datasets, the number of data points is $n = 10000$. The dimension $d$ is different for different datasets and is marked in Figure 7. We adopt the loss function defined in (1) and use different values of $\tau$ for different datasets. To calculate the approximation ratio of our dimensionality reduction methods, we solve the full problems and their sketched counterparts by using the LinvPy software [1]. This software uses iteratively re-weighted least squares (IRLS), and we modify it for $\|\cdot\|_{M,w}$, which requires only to include a “fixed” weighting from $w$ into the IRLS solver.

The Random Gaussian dataset is a synthetic dataset, whose entries are sampled i.i.d. from the standard Gaussian distribution. The remaining datasets are chosen from the UCI Machine Learning Repository. The $\tau$ values were chosen roughly so that there would be significant clipping of the residuals. For each dataset, we also randomly select 5% of the entries of the $b$ vector and change them to $10^4$, to model outliers. Such modified datasets are marked as “with outliers” in Figure 7.

We tested both the row sampling algorithm and the oblivious sketch. We varied the size of the sketch from $2d$ to $10d$ ($d$ is the dimension of the dataset) and calculated the approximation ratio $\|Ax - b\|_M/\|Ax^* - b\|_M$ using the modified LinvPy software, where $x^*$ is the solution returned by solving the full problem and $\hat{x}$ is the solution returned by solving the sketched version. We repeated each experiment ten times and took the best result among all repetitions.

Discussions As can be seen from Figure 7 the row sampling algorithm has better approximation ratio as we increase the sketch size. The same is not always true for the oblivious sketch, since the oblivious sketch only guarantees an $O(\log n)$ approximation instead of a $(1 + \varepsilon)$-approximate solution, as returned by the row sampling algorithm. Moreover, the row sampling algorithm consistently outperforms the oblivious sketch in the experiments, except for extremely small sketch sizes (around $2d$). However, applying the oblivious sketch requires only one pass over the input, and the distribution of the sketching matrices does not depend on the input. These advantages make the oblivious sketch preferable in streaming and distributed settings. Another advantage of the oblivious sketch is its simplicity.

Our empirical results demonstrate the practicality of our dimensionality reduction methods. Our methods successfully reduce a Tukey regression instance of size $10000 \times d$ to another instance...
with $O(d)$ rows, without much sacrifice in the quality of the solution. In most cases, the row sampling algorithm reduces the size to $3d$ rows while retaining an approximation ratio of at most $2$.

## 10 Conclusions

We give the first dimensionality reduction methods for the overconstrained Tukey regression problem. We first give a row sampling algorithm which takes $\tilde{O}(\text{nnz}(A) + \text{poly}(d \log n/\varepsilon))$ time to return a weight vector with $\text{poly}(d \log n/\varepsilon)$ non-zero entries, such that the solution of the resulting weighted Tukey regression problem gives a $(1 + \varepsilon)$-approximation to the original problem. We further give another way to reduce Tukey regression problems to smaller weighted versions, via an oblivious sketching matrix $S$, applied in a single pass over the data. Our dimensionality reduction methods are simple and easy to implement, and we give empirical results demonstrating their practicality. We also give hardness results showing that the Tukey regression problem cannot be efficiently solved in the worst-case.

From a technical point of view, our algorithms for finding heavy coordinates and our structural theorem seem to be of independent interest. We leave it as an intriguing open problem to find more applications of them.

## Acknowledgements

The authors would like to thank Lijie Chen and Peilin Zhong for helpful discussions, and the anonymous ICML reviewers for their insightful comments.
References

[1] The linvpy package. https://github.com/LCAV/linvpy/.

[2] Sanjeev Arora, Rong Ge, Ravi Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization—provably. SIAM Journal on Computing, 45(4):1582–1611, 2016.

[3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. Journal of the ACM (JACM), 43(6):1002–1045, 1996.

[4] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In Proceedings of the IEEE International Conference on Computer Vision, pages 2830–2838, 2015.

[5] Vladimir Braverman and Rafail Ostrovsky. Zero-one frequency laws. In Proceedings of the forty-second ACM symposium on Theory of computing, pages 281–290. ACM, 2010.

[6] Kenneth L. Clarkson. Subgradient and sampling algorithms for $\ell_1$ regression. In SODA, 2005.

[7] Kenneth L Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, Xiangrui Meng, and David P Woodruff. The fast cauchy transform and faster robust linear regression. SIAM Journal on Computing, 45(3):763–810, 2016.

[8] Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In STOC, 2013. Full version at http://arxiv.org/abs/1207.6365.

[9] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015, pages 310–329, 2015.

[10] Kenneth L Clarkson and David P Woodruff. Input sparsity and hardness for robust subspace approximation. In Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on, pages 310–329. IEEE, 2015.

[11] Kenneth L. Clarkson and David P. Woodruff. Sketching for $M$-estimators: A unified approach to robust regression. In 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, 2015.

[12] Michael B. Cohen and Richard Peng. $\ell_p$ row sampling by Lewis weights. In Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015, pages 183–192, 2015.

[13] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for $\ell_p$ regression. SIAM J. Comput., 38(5):2060–2078, 2009.

[14] Irit Dinur. The pcp theorem by gap amplification. Journal of the ACM (JACM), 54(3):12, 2007.

[15] John Fox. Robust Regression (web appendix). Sage Publications, 2002.
[16] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[17] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC*, pages 202–208, 2005.

[18] Ravi Kannan and Santosh Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4(3-4):157–288, 2009.

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

[20] Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 127–136. IEEE, 2013.

[21] Yi Li, Ruosong Wang, and David P Woodruff. Tight bounds for the subspace sketch problem with applications. *arXiv preprint arXiv:1904.05543*, 2019.

[22] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[23] X. Meng and M. W. Mahoney. Low-distortion Subspace Embeddings in Input-sparsity Time and Applications to Robust Linear Regression. *ArXiv e-prints*, October 2012.

[24] Ankur Moitra. An almost optimal algorithm for computing nonnegative rank. *SIAM Journal on Computing*, 45(1):156–173, 2016.

[25] Jelani Nelson and Huy L. Nguyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. *CoRR*, abs/1211.1002, 2012.

[26] Ilya Razenshteyn, Zhao Song, and David P Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 250–263. ACM, 2016.

[27] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–299, 1992.

[28] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

[29] Christian Sohler and David P. Woodruff. Subspace embeddings for the $l_1$-norm with applications. In *STOC*, pages 755–764, 2011.

[30] Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise $l_1$-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 688–701. ACM, 2017.

[31] Elad Verbin and Qin Zhang. Rademacher-sketch: A dimensionality-reducing embedding for sum-product norms, with an application to earth-mover distance. In *ICALP (1)*, pages 834–845, 2012.
[32] Ruosong Wang and David P Woodruff. Tight bounds for $\ell_p$ oblivious subspace embeddings. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1825–1843. SIAM, 2019.

[33] Przemysław Wojtaszczyk. Banach spaces for analysts, volume 25. Cambridge University Press, 1996.

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

[35] David P. Woodruff and Qin Zhang. Subspace embeddings and $l_p$-regression using exponential random variables. CoRR, abs/1305.5580, 2013.

[36] Jiyan Yang, Xiangrui Meng, and Michael W. Mahoney. Quantile regression for large-scale applications. In ICML (3), pages 881–887, 2013.