Supplemental material: Conditional Linear Regression

Diego Calderon Brendan Juba Sirui Li Zongyi Li Lisa Ruan

1 Soft Regression and Outlier Removal

Our algorithm works primarily on terms: we consider the terms $\{t_j\}_{j=1}^m$ to be atomic sets of data, whose weights $|t_j|$ are the number of points (or probability mass) satisfying the terms. The ideal condition $\{x : \mathbf{c}^*(\mathbf{x}) = 1\}$ is denoted by I_{good}; we also use I_{good} to denote the collection of terms of the DNF $\mathbf{c}^*$: $\{t_i : t_i \text{ is a term of } \mathbf{c}^*\}$, so the number of terms in I_{good} is t. From the perspective of Charikar et al. (2017), we treat I_{good} as our "good data," with the other points being arbitrary bad data. The algorithm computes regression parameters $\mathbf{w}_j$ for each term t_j, and clusters these parameters. The parameters are iteratively recomputed in a coordinate system centered on each of the clusters; since the quality of the approximation we obtain scales with the radius of the parameter space we consider, this centering improves the quality of the estimates we obtain. Eventually, our algorithm is going to suggest a list of candidate parameters $\hat{\mathbf{w}}$, with one of them approximating $\mathbf{w}^*$. Then, using the residuals of each candidate $\hat{\mathbf{w}}$ as labels, we learn a corresponding $\hat{\mathbf{c}}$ and evaluate its quality in order to select a good pair to output. Towards realizing this strategy, we need to compute approximations to the regression parameters that are not too impacted by the presence of terms outside the desired DNF.

1.1 Preprocessing

In this section, we show how to convert the data into a suitable form: later, we will assume the terms are disjoint and that we have an adequate number of examples to estimate the loss on each term. We will ensure these conditions by introducing duplicate points when they are shared, and by deleting terms that are satisfied by too few examples.

1.1.1 Reduction to Disjoint Terms by Duplicating Points

Given N data points and m terms $t_1, \dots, t_m$, if we view terms as sets, our analysis will require these terms to be disjoint. A simple method is to duplicate the points for each term they are contained in. For example, if the i^{th} point $\mathbf{x}^{(i)} = (\mathbf{x}, \mathbf{y}, z)^{(i)}$ is contained in terms t_a and t_b, then we create two points $(\mathbf{x}, \mathbf{y}, z)^{(a,i)}$ and $(\mathbf{x}, \mathbf{y}, z)^{(b,i)}$, each with the same attributes $(\mathbf{x}, \mathbf{y}, z)$ as the original point $\mathbf{x}^{(i)}$. After duplication, the terms are disjoint, and there will be at most Nm points. We denote the resulting number of points by N'. The size of I_{good}, changing from $|\bigcup_{I_{good}} t_i|$ to $\sum_{I_{good}} |t_i|$, may also blow up with a factor ranging from 1 to t. Note that the proportion of good points N_{good}/N decreases by at most a factor of $1/m$ since $N'_{good}/N' \geq N_{good}/mN$. This double counting process may skew the empirical distribution of I_{good} by up to a factor of t. Consequently, it may result in up to a $t \leq n^k$-factor blow-up in the error, and this is ultimately the source of the increase in loss suffered by our algorithm in the main theorem. For convenience, we will use the same notation N, I_{good}, and μ for both before and after duplication when there is no confusion.

1.1.2 Reduction to Adequately Sampled Terms by Deleting Small Terms

The approach of Charikar et al. (2017) can only guarantee that we obtain satisfactory estimates of the parameters for sufficiently large subsets of the data. Intuitively, this is not a significant limitation as if a term has very small size, it will not contribute much to our empirical estimates. Indeed, with high probability, the small terms (terms with size $< \beta\mu N$ for $\beta \leq \gamma/t$) only comprise a γ fraction of I_{good}. Based on this motivation, if a term has size less than $\beta\mu N$, then we just delete it at the beginning. Especially for a t-term DNF, not many terms could be small, so it is safe to ignore these small terms. As before, we will continue to abuse notation, using t and m for the number of terms when there is no confusion.

1.2 Loss Functions

In this section we define our loss functions and analyze their properties.

Given N data points and m disjoint sets (terms) $t_1, \ldots, t_m$ with size (weight) $|t_1|, \ldots, |t_m|$, we can define a loss function for each point in the space of parameters. For each ith point, define $f^{(i)} : \mathcal{H} \to \mathbb{R}$ by

$$f^{(i)}(\mathbf{w}) = (z^{(i)} - \langle \mathbf{w}, \mathbf{y}^{(i)} \rangle)^2$$

Similarly, we define a loss function for each of the terms t_j, $f_1, \ldots, f_m : \mathcal{H} \to \mathbb{R}$, as the average loss over these data points $\{\mathbf{x}^{(i)} = (\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, z^{(i)})\}$ in the term t_j (beware we abuse the notation to let $\mathbf{x}^{(i)}$ denote the ith point).

$$\begin{aligned}
f_j(\mathbf{w}) &= \frac{1}{|t_j|} \sum_{\mathbf{x}^{(i)} \in t_j} f^{(i)}(\mathbf{w}) \\
&= \frac{1}{|t_j|} \sum_{\mathbf{x}^{(i)} \in t_j} (z^{(i)} - \langle \mathbf{w}, \mathbf{y}^{(i)} \rangle)^2 \\
&\underset{(*)}{=} \frac{1}{|t_j|} \|\mathbf{z} - Y\mathbf{w}^\top\|_2^2 \\
&= \frac{1}{|t_j|} (\mathbf{z} - Y\mathbf{w}^\top)^\top (\mathbf{z} - Y\mathbf{w}^\top) \\
&= \frac{1}{|t_j|} \left(\mathbf{z}^\top \mathbf{z} - \mathbf{z}^\top Y \mathbf{w}^\top - \mathbf{z} Y^\top \mathbf{w} + \mathbf{w} Y^\top Y \mathbf{w}^\top \right) \\
&= \frac{1}{|t_j|} [1, \mathbf{w}] \begin{bmatrix} \mathbf{z}^\top \mathbf{z} & -\mathbf{z}^\top Y \\ -Y^\top \mathbf{z} & Y^\top Y \end{bmatrix} [1, \mathbf{w}]^\top
\end{aligned}$$

Where at $(*)$, we write the formula in vectors and matrices. We treat $\mathbf{z}$ as a $|t_j| \times 1$ column vector, with each coordinate being the z for the corresponding point in the term t_j. Similarly, Y is a $|t_j| \times d$ matrix, with each row containing a point and $\mathbf{w}$ is $1 \times d$ row vector. One advantage of our formulation is that the loss function for each term can be eventually written as $f_j(\mathbf{w}) = [1, \mathbf{w}] A [1, \mathbf{w}]^\top$, where A is a $(d+1) \times (d+1)$ matrix. We can pre-compute this quadratic loss matrix A so that the running time of the main algorithm is independent of the number of data points, and is thus a function only of the number of terms and dimension for our regression problem.

Note that these loss functions are stochastic, depending on the sample from the distribution $(\mathbf{x}, \mathbf{y}, z) \sim D$. That is, the true loss for a fixed term t_j is:

$$\mathbb{E}[f_j(\mathbf{w})] = \mathbb{E}_{\mathbf{x}^{(i)}}[(z^{(i)} - \langle \mathbf{w}, \mathbf{y}^{(i)} \rangle)^2 | t_j].$$

Similarly, for I_{good}, we define the loss function

$$f_{I_{good}}(\mathbf{w}) = \frac{1}{|I_{good}|} \sum_{\mathbf{x}^{(i)} \in I_{good}} f^{(i)}(\mathbf{w}).$$

Let $\bar{f}$ denote the expected loss function for points averaged over I_{good},

$$\bar{f}(\mathbf{w}) = \mathbb{E}[f_{I_{good}}].$$

Then the optimal $\mathbf{w}^*$ is defined as

$$\mathbf{w}^* := \arg\min_{\mathbf{w}} \bar{f}(\mathbf{w}).$$

Our ultimate goal is to find $\hat{\mathbf{w}}$ that minimizes $\bar{f}(\hat{\mathbf{w}})$, but the difficulty is that $\bar{f}$ is unknown (since I_{good} is unknown). To overcome this barrier, instead of directly minimizing $\bar{f}(\hat{\mathbf{w}})$, we try to find a parameter $\hat{\mathbf{w}}$ such that $\bar{f}(\hat{\mathbf{w}}) - \bar{f}(\mathbf{w}^*)$ is small. Once we get a close approximation $\hat{\mathbf{w}}$, we can use the covering algorithm of Juba et al. (2018) to find a good corresponding condition $\hat{\mathbf{c}}$.

In summary, we reformulate our problem in terms of these new loss functions as follows:

Definition 1.1 (Restatement of conditional linear regression problem) *Given D a distribution over points $\{\mathbf{x}^{(i)} := (\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, z^{(i)})\}_{(i)=1}^{N}$, and $\{t_j\}_{j=1}^{m}$ predefined disjoint subsets (terms), let I_{good} be the (unknown) target collection corresponding to $\mathbf{c}^* = \bigcup_{t_j \in \mathbf{c}^*} t_j$ with probability mass $Pr[\mathbf{x} \in I_{good}] \geq \mu$, and $\bar{f}$ be the regression loss over I_{good}. If there exists a linear regression fit $\mathbf{w}^*$ such that:*

$$\mathbb{E}_D[\bar{f}(\mathbf{w}^*)] \leq \epsilon.$$

Then we want to find a $\hat{\mathbf{w}}$ that approximates $\mathbf{w}^*$:

$$\mathbb{E}_D[\bar{f}(\hat{\mathbf{w}})] \leq \gamma + \epsilon.$$

1.3 Main Optimization Algorithm

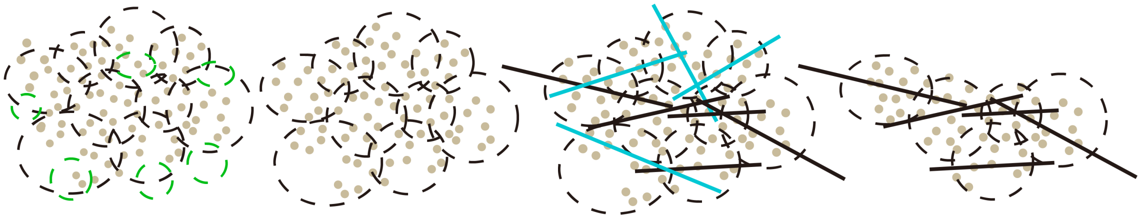

Figure 1: Algorithm 1

1: The original data space with the terms. **2:** Delete the small terms and duplicate points. **3:** Compute the best regression parameter $\hat{\mathbf{w}}_i$ for each term. **4:** Meanwhile iteratively downweight these terms whose $\hat{\mathbf{w}}_i$ have large error on their neighbor terms.

The main algorithm is an alternating-minimization-style algorithm: given a soft choice of which terms are outliers, we let each term choose a local set of regression parameters that are collectively regularized by the trace of their enclosing ellipsoid. Then, given these local regression parameters, we update our scoring of outliers by examining which terms find it difficult to assemble a coalition of sufficiently many "neighboring" terms whose parameters are, on average, close to the given term. We repeat the two until we obtain a sufficiently small enclosing ellipsoid for the collection of regression parameters.

1.3.1 Semidefinite Programming for Soft Regression

Following Charikar et al. (2017), we now present Algorithm 1 for approximating the regression parameters. We assign "local" regression parameters $\mathbf{w}_i$ for each term t_i, and use a semi-definite program (SDP) to minimize the total loss $\sum |t_i| f_i(\mathbf{w}_i)$ with regularization to force these parameters to be close to each other. Following each iteration, we use Algorithm 2 to remove outliers, by decreasing the weight factors c_i for those terms without enough neighbors. The process is illustrated in Figure 1. Intuitively, if there exists a good linear regression fit $\mathbf{w}^*$ on I_{good}, then for each term $t_i \in I_{good}$, $f_i(\mathbf{w}^*)$ should be small. Therefore, we can find a small ellipse Y (or $\mathcal{E}_Y$) centered at $\mathbf{o}$ bounding all parameters for the terms in I_{good} if the center $\mathbf{o}$ is close to $\mathbf{w}^*$. The SDP will find such an ellipse bounding the parameters while minimizing the weighted total loss.

Algorithm 1: Soft regression algorithm

```
Input: terms $t_{1:m}$, center $\mathbf{o}$.
Output: parameters $\hat{\mathbf{w}}_{1:m}$ and a matrix $\hat{Y}$
Initialize $c_{1:m} \leftarrow (1, \dots, 1)$, $\lambda \leftarrow \frac{\sqrt{8\mu}NtS}{r}$
repeat
  Let $\hat{\mathbf{w}}_{1:m}, \hat{Y}$ be the solution to SDP:
      $\underset{\mathbf{w}_1,\dots,\mathbf{w}_m,Y}{\text{minimize}} \quad \sum_{i=1}^{m} c_i |t_i| f_i(\mathbf{w}_i) + \lambda \operatorname{tr}(Y)$      (1)
      subject to $\quad (\mathbf{w}_i - \mathbf{o})(\mathbf{w}_i - \mathbf{o})^\top \preceq Y$ for all $i = 1, \dots, m.$
  if $\operatorname{tr}(\hat{Y}) > \frac{6r^2}{\mu}$ then
    $c \leftarrow \text{UpdateWeights}(c, \hat{\mathbf{w}}_{1:m}, \hat{Y})$
  end if
until $\operatorname{tr}(\hat{Y}) \le \frac{6r^2}{\mu}$
Return $\hat{\mathbf{w}}_{1:m}, \hat{Y}$
```

Formally, in the SDP 1 (in Algorithm 1), w are $1 \times d$ vectors and Y is a $d \times d$-dimensional matrix (recall d is the dimension for $\mathbf{y}$ and $\mathbf{w}$). We bound the parameters $\mathbf{w}_i$ with the ellipse Y by imposing the semidefinite constraint $\mathbf{w}_i {\mathbf{w}_i}^\top \preceq Y$, which is equivalent to letting $\begin{bmatrix} Y & \mathbf{w}_i \\ {\mathbf{w}_i}^\top & 1 \end{bmatrix} \succeq 0$, saying that $\mathbf{w}_i$ lies within the ellipse centered at 0 defined by Y. Similarly, when the center is $\mathbf{o}$, the constraints are $(\mathbf{w}_i - \mathbf{o})(\mathbf{w}_i - \mathbf{o})^\top \preceq Y$. The regularization $tr(Y)$ of the SDP penalizes the size of the ellipse, making the various parameter copies $\mathbf{w}_i$ lie close to each other.

1.3.2 Removing Outliers

The terms not in I_{good} may have large loss for the optimal parameters $\mathbf{w}^*$, and therefore make the total loss in SDP 1 large. To remove these bad terms, we assign a weight factor $c_i \in (0, 1)$ for each term t_i and down weight these terms with large loss, as shown in Algorithm 2.

Algorithm 2: Algorithm for updating outlier weights

```
**Input:** $c, \hat{\mathbf{w}}_{1:m}, \hat{Y}$.
**Output:** $c'$.
**for** $i = 1$ **to** $m$ **do**
    Let $\tilde{\mathbf{w}}_i$ be the solution to
        $$\begin{aligned} \underset{\tilde{\mathbf{w}}_i, a_{i1}, \ldots, a_{im}}{\text{minimize}} \quad & f_i(\tilde{\mathbf{w}}_i) \\ \text{subject to} \quad & \tilde{\mathbf{w}}_i = \sum_j^m a_{ij}\hat{\mathbf{w}}_j, \quad \sum_j^m a_{ij} = 1 \\ & 0 \le a_{ij} \le \frac{2}{\mu N}|t_j|, \quad \forall j \end{aligned} \qquad (2)$$
    $z_i \leftarrow f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}}_i)$
**end for**
$z_{max} \leftarrow max\{\ z_i \mid c_i \neq 0\}$
$c'_i \leftarrow c_i . \frac{z_{max} - z_i}{z_{max}}$ for $i = 1, \ldots, n$
**Return** $c'$
```

In Algorithm 2, we solve an SDP for each term to find its best μN neighbor points and compute the "average" parameter $\tilde{\mathbf{w}}_i$ over the neighborhood. $\tilde{\mathbf{w}}_i$ is a linear combination of its neighbors' parameters: $\tilde{\mathbf{w}}_i = \sum_j^m a_{ij}\hat{\mathbf{w}}_j$, minimizing the term's loss $f_i(\tilde{\mathbf{w}}_i)$. Intuitively, if a term is a good term, i.e. $t_i \in I_{good}$, then its parameter $\hat{\mathbf{w}}_i$ should be close to the average of parameters of all terms in I_{good}, $\mathbf{w}_i \approx \sum_{I_{good}} \frac{|t_j|}{|I_{good}|}\mathbf{w}_j$. In the SDP for t_i, we define coefficients a_{ij} to play the role of $\frac{|t_j|}{|I_{good}|}$. These coefficients $\{a_{ij}\}$ are required to sum to 1, i.e. $\sum_j^m a_{ij} = 1$, and each should not be larger than $\frac{|t_j|}{|I_{good}|} \sim 2\frac{|t_j|}{\mu N}$. At a high level, the SDP computes the best neighbors for t_i by assigning $\{a_{ij}\}$, so that the average parameter $\tilde{\mathbf{w}}_i$ over the neighbors minimizes f_i. If a term is bad, it is hard to find such good neighbors, so if the loss $f_i(\tilde{\mathbf{w}}_i)$ is much larger than the original loss, then we consider the term to be an outlier, and down-weight its weight factor c_i.

1.4 A Bound on the Loss That Is Linear in the Radius

Similarly to Charikar et al. (2017), we obtain a theorem saying the algorithm will return meaningful outputs on I_{good}. The main change is that we use terms instead of points. In other words, we generalize their arguments from unit-weight points to sets with different weights. And based on a spectral norm analysis, we show the bound will shrink linearly with the radius as long as we have enough data.

First, to estimate the losses by their inputs, we introduce the gradient ∇f. By the convexity of f, we have $(f(\mathbf{w}) - f(\mathbf{w}^*)) \le \langle \nabla f(\mathbf{w}), \mathbf{w} - \mathbf{w}^* \rangle$. Note that $\|\mathbf{w} - \mathbf{w}^*\|$ is bounded by $2r$, where

$r := \max \|\mathbf{w}\|_2$. We will need to bound the gradient as well.

To bound the loss functions, we use the spectral norm of gradients:

$$S := \max_{\mathbf{w} \in \mathcal{H}} \frac{1}{\sqrt{t}} \| \left[\nabla f_j(\mathbf{w}) - \nabla \bar{f}(\mathbf{w}) \right]_{j \in I_{good}} \|_{op}$$

Where $\| \cdot \|_{op}$ is the spectral norm (operator 2-norm) of the matrix, whose rows are gradients of loss functions in I_{good}: $(\nabla f_i(\mathbf{w}) - \nabla \bar{f}(\mathbf{w}))_{i \in I_{good}}$. S measures the difference between the gradient of loss functions of terms in I_{good}: $\nabla f_i(\mathbf{w})$ and gradient of average loss on I_{good}: $\nabla \bar{f}(\mathbf{w})$. At a high level, this bound tells us how bad these loss functions could be. We note that since the gradient is a linear operator, this quantity is invariant to regularization of the loss functions.

As shown by Charikar et al. (2017), if $\nabla f_i - \nabla \bar{f}$ is a $\sigma_{\nabla f}$ sub-gaussian distribution, then $S = \mathcal{O}(\sigma_{\nabla f})$, generally a constant. Although a constant bound $\mathcal{O}(\sigma_{\nabla f})$ is good for their purposes – mean estimation – it is too weak for linear regression. In the sequel we will show S is going to shrink as the radius of parameters r decreases.

For linear regression, $f_j(\mathbf{w}) := \frac{1}{|t_j|} \sum_{\mathbf{x}^{(i)} \in t_j} f^{(i)}(\mathbf{w})$, and $\nabla f_j(\mathbf{w}) = \frac{1}{|t_j|} \sum_{\mathbf{x}^{(i)} \in t_j} \nabla f^{(i)}(\mathbf{w})$, where for each point $\nabla f^{(i)}(\mathbf{w}) = 2(\mathbf{w}^\top \mathbf{y}^{(i)} - z^{(i)})\mathbf{y}^{(i)}$. If we assume $z^{(i)} = \mathbf{w}^{*\top} \mathbf{y}^{(i)} + \epsilon^{(i)}$, and the residual $\epsilon^{(i)}$ (a subgaussian, e.g., from $\mathcal{N}(0, \sigma_\epsilon^2)$) is independent of $\mathbf{y}^{(i)}$, then

$$\nabla f^{(i)}(\mathbf{w}) = 2(\mathbf{w}^\top - \mathbf{w}^{*\top})\mathbf{y}^{(i)}\mathbf{y}^{(i)\top} + \epsilon^{(i)}\mathbf{y}^{(i)}$$

$$\nabla f_j(\mathbf{w}) = \frac{1}{|t_j|} \left(2(\mathbf{w}^\top - \mathbf{w}^{*\top}) \sum_{\mathbf{x}^{(i)} \in t_j} \mathbf{y}^{(i)}\mathbf{y}^{(i)\top} + \sum_{\mathbf{x}^{(i)} \in t_j} \epsilon^{(i)}\mathbf{y}^{(i)} \right)$$

Similarly, we can write the target function as:

$$\nabla \bar{f}(\mathbf{w}) = 2(\mathbf{w}^\top - \mathbf{w}^{*\top})\mathbb{E}[\mathbf{y}\mathbf{y}^\top]$$

So the difference of the gradients is actually:

$$(\nabla f_j(\mathbf{w}) - \nabla \bar{f}(\mathbf{w})) = 2(\mathbf{w}^\top - \mathbf{w}^{*\top})(\sum_{\mathbf{x}^{(i)} \in t_j} \frac{\mathbf{y}^{(i)}\mathbf{y}^{(i)\top}}{|t_j|} - \mathbb{E}[\mathbf{y}\mathbf{y}^\top]) + \sum_{\mathbf{x}^{(i)} \in t_j} \frac{\epsilon^{(i)}\mathbf{y}^{(i)}}{|t_j|}$$

The first term is going to shrink as $(\mathbf{w}^\top - \mathbf{w}^{*\top})$ decreases. (If we draw enough data, $\frac{1}{|t_j|} \sum_{t_j} \mathbf{y}^{(i)}\mathbf{y}^{(i)\top} \to \mathbb{E}[\mathbf{y}^{(i)}\mathbf{y}^{(i)\top} | t_j]$, so we'll be able to regard the other factor as a fixed "scaling.") The second term approaches zero as we draw more data, $\frac{1}{|t_j|} \sum_{t_j} \epsilon^{(i)}\mathbf{y}^{(i)} \to 0$. So given that we have drawn enough data, we will be able to bound each row of S by the radius $r := \max_{\mathbf{w}} \|\mathbf{w}\|_2$ and similarly for the whole matrix.

More concretely, if we define $S_0 := \| \left[\frac{1}{|t_j|} \sum_{t_j} \mathbf{y}^{(i)}\mathbf{y}^{(i)\top} - \mathbb{E}[\mathbf{y}\mathbf{y}^\top] \right]_{I_{good}} \|_{op}$, then we find $S = \mathcal{O}(rS_0)$. Note, S_0 is fixed given the data, and thus remains constant across iterations. Furthermore, S_0 concentrates around $\| \left[\mathbb{E}[\mathbf{y}^{(i)}\mathbf{y}^{(i)\top} | t_j] - \mathbb{E}[\mathbf{y}\mathbf{y}^\top] \right]_{I_{good}} \|_{op}$ and can thus be bounded. Therefore, the bound on S we can guarantee will decrease when we take more points. We know $\frac{1}{|t_j|} \sum_{t_i} \epsilon^{(i)}$ can be bounded with a simple sub-gaussian tail bound : $\Pr[\frac{1}{|t_j|} \sum \epsilon \geq \tau] \leq \exp[-\frac{2\tau^2}{\sigma_\epsilon^2/|t_j|}]$. Plugging in $\tau \leftarrow r$, and fixing δ, we find that as long as the number of examples $|t_j| \geq \sigma_\epsilon^2 \log(1/\delta)/2r^2$, then $\sum_{t_j} \epsilon^{(i)} \leq r$ with probability $1 - \delta$. Taking a union bound over $\delta \leftarrow \delta/t$, it suffices to take $|t_j| \geq \sigma_\epsilon^2 \log(m/\delta)/2r^2$, and thus $N = \mathcal{O}(\sigma_\epsilon^2 \log(m/\delta)/\beta\mu r^2))$. In summary, we obtain

Lemma 1.2 *For $N = \mathcal{O}(\sigma_\epsilon^2 \log(m/\delta)/\beta\mu r^2))$ example points, with probability $1-\delta$ the spectral norm of the gradients S is bounded by a linear function of the radius $r := \max_{\mathbf{w}} \|\mathbf{w}\|_2$, i.e., $S = \mathcal{O}(rS_0)$.*

1.5 Analysis of Main Optimization Algorithms 1 and 2

Let $\hat{\mathbf{w}}_{1:m}$ be the outputs from Algorithm 1. We define the weighted average parameter of terms from I_{good} as $\hat{\mathbf{w}}_{avg} := (\sum_{i \in I_{good}} c_i|t_i|\hat{\mathbf{w}}_i)/(\sum_{i \in I_{good}} c_i|t_i|)$. In this section, we aim to prove a bound on $\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*)$ by controlling the optimization error $|f_i(\hat{\mathbf{w}}_i) - f_i(\mathbf{w}^*)|$ and the statistical error $|\bar{f}(\hat{\mathbf{w}}_{avg}) - f_i(\hat{\mathbf{w}})|$. Then we prove Algorithm 2 will not decrease the weight of the good terms too much.

Theorem 1.3 says that Algorithm 1 can find a small ellipse $\mathcal{E}_Y$ bounding its output, and the expected loss over $\hat{\mathbf{w}}_{avg}$ is close to the expected loss of $\mathbf{w}^*$.

Theorem 1.3 (Weighted Version of Theorem 4.1, Charikar et al. (2017)) *Let $\hat{\mathbf{w}}_{1:m}, \hat{Y}$ be the output of Algorithm 1. Then, $\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*) \leq 18\frac{tSr}{\sqrt{\mu}}$. Furthermore, $\hat{\mathbf{w}}_{avg} \in \mathcal{E}_{\hat{Y}}$ and $\mathrm{tr}(\hat{Y}) \leq \frac{6r^2}{\mu}$.*

Lemma 1.4 is a basic inequality used multiple times in the analysis. It bounds the loss via S. Since the algorithm is using terms instead of points, we are suffering an additional factor-t blow-up of the error compared to the original bound, which is carried through the lemmas in this section.

Lemma 1.4 *For any $\mathbf{w}$ and any $\mathbf{w}_{1:n}$ satisfying $\mathbf{w}_i{\mathbf{w}_i}^\top \preceq Y$ for all i, we have*

$$\Big|\sum_{i \in I_{good}} c_i|t_i|\langle \nabla f_i(\mathbf{w}) - \nabla \bar{f}(\mathbf{w}), \mathbf{w}_i\rangle\Big| \leq \mu N t S\sqrt{\mathrm{tr}(Y)}. \tag{3}$$

Proof of Lemma 1.4 Let F be the matrix whose i^{th} row is $\big(\nabla f_i(\mathbf{w}_0) - \nabla \bar{f}(\mathbf{w}_0)\big)$, and let W be the matrix whose i^{th} row is $\mathbf{w}_i$. We consider only the rows $i \in I_{good}$, so the dimension of each matrix is $t \times d$. We have

$$\Big|\sum_{i \in I_{good}} c_i|t_i|\langle \nabla f_i(\mathbf{w}_0) - \nabla \bar{f}(\mathbf{w}_0)\rangle\Big| = \mathrm{tr}(F^\top diag(|t_i|c_i)W)$$

$$\leq \|diag(|t_i|)diag(c_i)F\|_{op}\|W\|_*$$

by Hőlder's inequality. We can bound each part:

$$\|diag(t_i)\|_{op} \leq \max_{t_i \in I_{good}} |t_i| \leq N_{good} \leq \mu N$$

$$\|diag(c)\|_{op} \leq 1 \text{ since } c \in [0,1]$$

$$\|F\|_{op} \leq \sqrt{t}S, \text{ by the definition of S}$$

$$\|W\|_* \leq \sqrt{t\mathrm{tr}(Y)}, \text{ by Lemma 3.1 of Charikar et al. (2017)}$$

Combining these, we see that $\|diag(|t_i|c)F\|_{op}\|W\|_*$ is bounded by $\mu tNS\sqrt{\mathrm{tr}(Y)}$. ■

Lemma 1.5 bounds the difference between $f_i(\hat{\mathbf{w}}_i)$ and $f_i(\mathbf{w}^*)$, based on the optimality of our solution to SDP 1 in Algorithm 1. Its proof follows identically to Lemma 4.2 of Charikar et al. (2017).

Lemma 1.5 (c.f. Lemma 4.2 of Charikar et al. (2017)) *The solution* $\hat{\mathbf{w}}_{1:m}$ *to the SDP in Algorithm 1 satisfies:*

$$\sum_{i \in I_{good}} c_i |t_i| (f_i(\hat{\mathbf{w}}_i) - f_i(\mathbf{w}^*)) \leq \lambda \|\mathbf{w}^*\|_2^2. \tag{4}$$

Lemma 1.6 bounds the difference between $f_i(\hat{\mathbf{w}}_{avg})$ and $f_i(\hat{\mathbf{w}}_i)$. Its proof likewise identically follows Lemma 4.3 of Charikar et al. (2017):

Lemma 1.6 (c.f. Lemma 4.3 of Charikar et al. (2017)) *Let* $\hat{\mathbf{w}}_{avg} := (\sum_{i \in I_{good}} c_i |t_i| \hat{\mathbf{w}}_i)/(\sum_{i \in I_{good}} c_i |t_i|)$. *The solution* $\hat{\mathbf{w}}_{1:m}$, $\hat{Y}$ *to Algorithm 1 satisfies*

$$\sum_{i \in I_{good}} c_i |t_i| \Big(f_i(\hat{\mathbf{w}}_{avg}) - f_i(\hat{\mathbf{w}}_i) \Big) \leq \mu N t S \Big(\sqrt{\operatorname{tr}(\hat{Y})} + r \Big),$$

$$\sum_{i \in I_{good}} c_i |t_i| \Big(\bar{f}_i(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*) \Big) \leq \sum_{i \in I_{good}} c_i |t_i| \Big(f_i(\hat{\mathbf{w}}_{avg}) - f_i(\mathbf{w}^*) \Big) + 2\mu N t r S.$$

We next consider an analogue of Lemma 4.4 of Charikar et al. (2017). To deal with the different weights of terms, our Algorithm 2 considers the neighbors with their weights, and therefore uses different definitions of a and W (from those of Charikar et al. (2017)) in the analysis. Lemma 1.7 bounds $\operatorname{tr}(Y)$ and the difference between $f_i(\tilde{\mathbf{w}}_i)$ and $f_i(\hat{\mathbf{w}}_i)$.

Lemma 1.7 *For* $\tilde{\mathbf{w}}_i$ *as obtained in Algorithm 2,* $\tilde{Y} := \frac{2}{\mu N} \hat{W} \hat{W}^\top$, *and* $W := [\sqrt{|t_1|}\mathbf{w}_1, \ldots, \sqrt{|t_m|}\mathbf{w}_m]$. *we have*

$$\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^\top \preceq \tilde{Y}$$

for all i*, and also*

$$\operatorname{tr}(\tilde{Y}) \leq \frac{2r^2}{\mu}$$

In addition:

$$\operatorname{tr}(\hat{Y}) \leq \frac{2r^2}{\mu} + \frac{1}{\lambda} \Big(\sum^{m} c_i |t_i| \big(f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}}_i) \big) \Big)$$

Proof of Lemma 1.7 Let $\tilde{\mathbf{w}}_i = \sum_{j=1}^m a_{ij} \hat{\mathbf{w}}_j$ as defined in Algorithm 2. First, we want to show $\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^\top \preceq \tilde{Y}$:

$$\begin{aligned}
\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^\top &= \Big(\sum_{j=1}^n a_{ij} \hat{\mathbf{w}}_j \Big) \Big(\sum_{j=1}^n a_{ij} \hat{\mathbf{w}}_j \Big)^\top \\
&\preceq \sum_{j=1}^n a_{ij} \hat{\mathbf{w}}_j \hat{\mathbf{w}}_j^\top \\
&\preceq \sum_{j=1}^n \frac{2}{\mu N} |t_j| \hat{\mathbf{w}}_j \hat{\mathbf{w}}_j^\top \\
&= \frac{2}{\mu N} \sum_{j=1}^n |t_j| \hat{\mathbf{w}}_j \hat{\mathbf{w}}_j^\top \\
&= \tilde{Y}
\end{aligned}$$

and

$$\begin{aligned}\mathrm{tr}(\tilde{Y}) &= \frac{2}{\mu N}\mathrm{tr}(\hat{W}\hat{W}^\top) \\ &= \frac{2}{\mu N}\mathrm{tr}(diag([|t_i|]))\|\mathbf{w}\| \\ &\leq \frac{2}{\mu N}\sum_{i=1}^{m}|t_i|r^2 \\ &\leq \frac{2r^2}{\mu}.\end{aligned}$$

For the third claim, since $(\hat{\mathbf{w}}_{1:m}, \hat{Y})$ is the optimal solution of the SDP in Algorithm 1 and $(\tilde{\mathbf{w}}_{1:m}, \tilde{Y})$ is a feasible solution of that, we have

$$\sum_{i=1}^{m} c_i|t_i|f_i(\hat{\mathbf{w}}_i) + \lambda\mathrm{tr}(\hat{Y}) \leq \sum_{i=1}^{m} c_i|t_i|f_i(\tilde{\mathbf{w}}_i) + \lambda tr(\tilde{Y})$$

This gives us

$$\mathrm{tr}(\hat{Y}) \leq \frac{2r^2}{\mu} + \frac{1}{\lambda}\Big(\sum_{i=1}^{m} c_i|t_i|\big(f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}}_i)\big)\Big).$$

■

Then, analogous to Corollary 4.4 of Charikar et al. (2017), we show $\hat{\mathbf{w}}_{avg}$ can be viewed as a feasible solution to SDP in Algorithm 2, so we can bound $(f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}}))$ by $(f_i(\hat{\mathbf{w}}_{avg}) - f_i(\hat{\mathbf{w}}))$.

Corollary 1.8 *If* $\sum_{i \in I_{good}} c_i|t_i| \geq \frac{\mu N}{2}$, *then*

$$\sum_{i \in I_{good}} c_i|t_i|\big(f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}}_i)\big) \leq \mu N t\Big(\sqrt{\mathrm{tr}(\hat{Y})} + r\Big)$$

Proof of Corollary 1.8: First, we show that $\hat{\mathbf{w}}_{avg}$ is a feasible solution for the semidefinite program for $\tilde{\mathbf{w}}$ in Algorithm 2.
By taking $a_{ij} = \frac{c_j|t_j|}{\sum_{j' \in I_{good}} c_{j'}|t_{j'}|}$ for $j \in I_{good}$ and 0 otherwise, we get $a_{ij} \leq \frac{2|t_j|}{\mu N}$ since $\sum_{j' \in I_{good}} c_{j'}|t_{j'}| \geq \frac{\mu N}{2}$. We see

$$\hat{\mathbf{w}}_{avg} = \frac{\sum_{j \in I_{good}} c_j|t_j|\hat{\mathbf{w}}_j}{\sum_{j' \in I_{good}} c_{j'}|t_{j'}|} = \sum_{j=1}^{N} a_{ij}\hat{\mathbf{w}}_j$$

Then by optimality,

$$\sum_{i \in I_{good}} c_i|t_i|(f_i(\tilde{\mathbf{w}}_i) - f_i(\hat{\mathbf{w}})) \leq \sum_{i \in I_{good}} c_i|t_i|(f_i(\hat{\mathbf{w}}_{avg}) - f_i(\hat{\mathbf{w}}))$$

which is bounded by $\mu NtS\Big(\sqrt{\mathrm{tr}(\hat{Y})} + r\Big)$ by Lemma 1.6. ■

Lemma 1.9 shows $\sum_{I_{good}} c_i|t_i|$ is large enough. In other words, Algorithm 2 will not down weight good terms too much. Its proof follows identically to Lemma 4.5 of Charikar et al. (2017).

Lemma 1.9 (c.f. Lemma 4.5 of Charikar et al. (2017)) *Suppose that* $\frac{1}{N}\sum\limits_{i=1}^{m} c_i|t_i|(f_i(\tilde{\mathbf{w}}_i)-f_i(\hat{\mathbf{w}}_i)) \geq \frac{2}{\mu N}\sum_{i\in I_{good}} c_i|t_i|(f_i(\tilde{\mathbf{w}}_i)-f_i(\hat{\mathbf{w}}_i))$ *Then, the update step in Algorithm 2 satisfies*

$$\frac{1}{\mu N}\sum_{i\in I_{good}} |t_i|(c_i - c_i') \leq \frac{1}{2N}\sum_{i=1}^{m} |t_i|(c_i - c_i') \tag{5}$$

Moreover, the above supposition holds if $\lambda = \frac{\sqrt{8\mu}NtS}{r}$ *and* $\text{tr}(\hat{Y}) > \frac{6r^2}{2\mu}$.

Finally, we prove Theorem 1.3, which bounds the difference in the empirical loss of $\hat{\mathbf{w}}_{avg}$ and $\mathbf{w}^*$.

Proof of Theorem 1.3: First, show the the weights of I_{good} will never be too small. By Lemma 1.9, the invariant $\sum_{i\in I_{good}} c_i|t_j| \geq \frac{\mu N}{2} + \frac{\mu}{2}\sum_{i=1} c_i|t_i|$ holds throughout the algorithm. Therefore we get $\sum_{i\in I_{good}} c_i|t_j| \geq \frac{\mu N}{2}$. In particular, Algorithm 1 will terminate, since Algorithm 2 zeros out at least one outlier c_i each time, and this can happen at most $m-t$ times before $\sum_{i\in I_{good}} c_i|t_i|$ would drop below $\frac{\mu N}{2}$, which we showed impossible.

Now, let $(\hat{\mathbf{w}}_{1:m}, \hat{Y})$ be the value returned by Algorithm 1. By Lemma 1.6 we then have

$$\begin{aligned}\sum_{i\in I_{good}} c_i|t_i|(\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*)) &\leq \sum_{i\in I_{good}} c_i|t_i|(\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*)) + 2\mu NtSr \\ &\leq \sum_{i\in I_{good}} c_i|t_i|(\bar{f}(\hat{\mathbf{w}}_{i}) - \bar{f}(\mathbf{w}^*)) + 3\mu NtSr + \sqrt{6\mu}NtSr.\end{aligned}$$

By Lemma 1.5, we have $\sum_{i\in I_{good}} c_i|t_i|(f_i(\hat{\mathbf{w}}_i - f_i(\mathbf{w}^*)) \leq \lambda r^2$ and, by the assumption we have $\text{tr}(\hat{Y}) \leq \frac{6r^2}{\mu}$. Plugging in $\lambda = \frac{\sqrt{8\mu}NtS}{r}$, we get

$$\begin{aligned}\sum_{i\in I_{good}} c_i|t_i|(\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*)) &\leq \lambda r^2 + 3\mu NtSr + \sqrt{6\mu}NtSr \\ &= 3\mu NtSr + (\sqrt{6}+\sqrt{8})\sqrt{\mu}NtSr \\ &\leq 9\sqrt{\mu}NtSr\end{aligned}$$

Since $\sum_{i\in I_{good}} c_i|t_i| \geq \frac{\mu N}{2}$, dividing through by $\sum_{i\in I_{good}} c_i|t_i|$ yields $\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*) \leq 18\frac{tSr}{\sqrt{\mu}}$. ■

2 List-regression Algorithm

We finally introduce the main algorithm to cluster the terms. Again following Charikar et al. (2017), we initially use Algorithm 1 to assign a parameter $\hat{\mathbf{w}}_i$ for each term. In each iteration, we use Padded Decompositions to cluster the terms by their parameters, and then reuse Algorithm 1 on each cluster. After each iteration, we can decrease the radius of the ellipse containing I_{good} by half. Eventually, the algorithm will be able to constrain the parameters for all of the good terms in a very small ellipse, as illustrated in Figure 2. The algorithm will then output a list of candidate parameters, with one of them approximating $\mathbf{w}^*$.

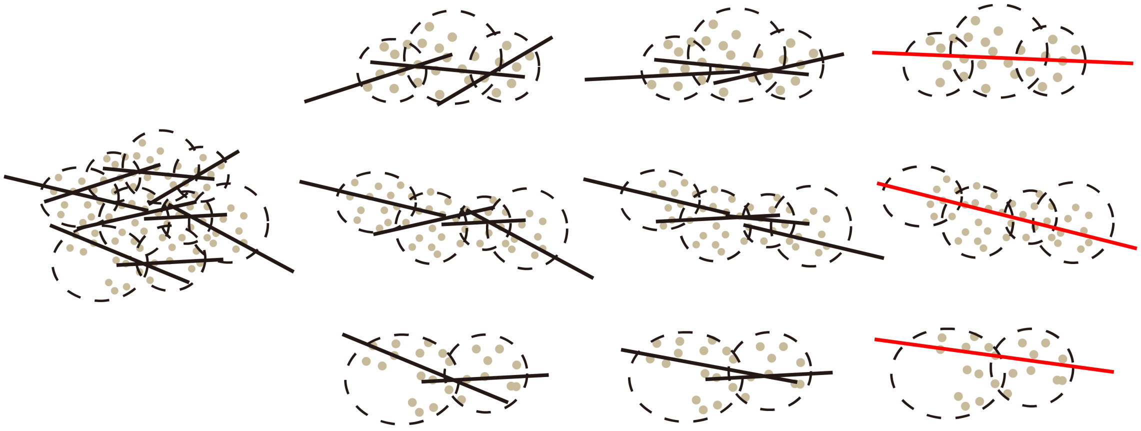

Figure 2: Algorithm 4

1: Run Algorithm 1. Get a $\hat{\mathbf{w}}_i$ for each term. **2:** Cluster the terms by their parameter $\hat{\mathbf{w}}_i$. **3:** Iteratively re-run Algorithm 1 on each cluster and re-cluster the terms, so that the $\hat{\mathbf{w}}_i$ of I_{good} gradually get closer. **4:** Finally terminate by picking a "good enough" cluster.

2.1 Padded Decomposition

Padded Decomposition is a randomized clustering technique developed by Fakcharoenphol et al. (2003). Given points $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ in a metric space, a padded decomposition with parameters (ρ, τ, δ) is a partitioning of the points $\mathcal{P} := \{P_i\}$ satisfying the following:

1. Each cluster P has diameter ρ,
2. For each point $\mathbf{w}_i$ and all $\mathbf{w}_j$ such that $\|\mathbf{w}_i - \mathbf{w}_j\| < \tau$, $\mathbf{w}_j$ will lie in the same cluster P as $\mathbf{w}_i$ with probability $1 - \delta$.

Fakcharoenphol et al. give a simple random clustering algorithm to produce padded decompositions, that uniformly samples balls with radius less than ρ from the space $\mathcal{W} = \hat{\mathbf{w}}_{1:m}$. Intuitively, if the radius of I_{good}, $\tau << \rho$, then we high probability, the ball with radius ρ will contain all of I_{good}.

Algorithm 3: Padded Decomposition

```
**Input:** $\hat{\mathbf{w}}_{1:m}, \rho, \tau$.
**Output:** Partition $\mathcal{P} = \{T\}$.
Initialize: let $\mathcal{P} = \emptyset$, $\mathcal{W} = \hat{\mathbf{w}}_{1:m}$. Sample $k \sim \text{Uniform}(2, \rho)$.
**while** $\mathcal{W} \neq \emptyset$ **do**
  Sample $i \sim \text{Uniform}(1, m)$.
  Let $T \leftarrow \text{Ball}(\hat{\mathbf{w}}_i, k\tau) \cap \mathcal{W}$.
  Update: $\mathcal{P} = \mathcal{P} \cup \{T\}$. $\mathcal{W} \leftarrow \mathcal{W} \backslash T$.
**end while**
**Return:** partition $\mathcal{P}$.
```

Lemma 2.1 (Padded Decomposition) *If all the elements of I have pairwise distance $d(\hat{\mathbf{w}}_i, \hat{\mathbf{w}}_j) \leq \tau$, and $\rho = \frac{1}{\delta}\tau\log(\frac{1}{\mu})$. then for the output partition $\mathcal{P}$ of Algorithm 3, with probability least $1 - \delta$, I will be contained in a single cluster $T \in \mathbb{P}$.*

The proof of this variant can be found in Appendix A of Charikar et al. (2017).

In Algorithm 4, we will generate multiple padded decompositions in each iteration, to ensure that with high probability most of the padded decompositions preserve all of I_{good} in a single cluster. At the end of each iteration, we will update $\hat{\mathbf{w}}_i$ by aggregating the padded decompositions.

Given a target radius r_{final}, we will check if the current radius $r < r_{final}$. If so, the algorithm will greedily find a list of candidate parameters $\mathbf{u}_1, ..., \mathbf{u}_s$, where the length of the list s is at most $\frac{2}{\mu}$. We can show that one of $\mathbf{u}_1, ..., \mathbf{u}_s$ must be close to $\mathbf{w}^*$. Finally, we will use a greedy covering algorithm following Juba et al. (2018) to find conditions on which the linear rules $\mathbf{u}_1, ..., \mathbf{u}_s$ have low loss, and return the pair $(\hat{\mathbf{w}}, \hat{\mathbf{c}})$ with at least a μ fraction of points and the smallest regression loss.

Algorithm 4: List-regression algorithm

```
Input: $m$ terms, target radius $r_{final}$.
Output: candidate solutions $\{\mathbf{u}_1, ..., \mathbf{u}_s\}$ and $\hat{\mathbf{w}}_{1:m}$.
Initialize $r^{(1)} \leftarrow r$,
$\hat{\mathbf{w}}^{(1)}_{1:m} \leftarrow$ Algorithm 1 with origin 0 and radius $r$ (all $i = 1, \ldots, m$ are "assigned" an output).
for $\ell = 1, 2, \ldots$ do
  $\mathcal{W} \leftarrow \{\hat{\mathbf{w}}^{(\ell)}_i | \ \hat{\mathbf{w}}^{(\ell)}_i$is assigned$\}$
  if $r^{(\ell)} < \frac{1}{2} r_{final}$ then
    Greedily find a maximal set of points $\mathbf{u}_1, ..., \mathbf{u}_s$ s.t.
      I: $|B(\mathbf{u}_j; 2r_{\text{final}}) \cap \mathcal{W}| \geq (1-\beta)\mu N, \quad \forall j.$
      II: $\|\mathbf{u}_j - \mathbf{u}_{j'}\|_2 > 4r_{\text{final}}, \quad \forall j \neq j'.$
    Return $\mathcal{U} = \{\mathbf{u}_1, ..., \mathbf{u}_s\}, \hat{\mathbf{w}}^{(\ell)}_{1:m}$.
  end if
  for $h = 1$ to $112 \log(\frac{\ell(\ell+1)}{\delta})$ do
    $\bar{\mathbf{w}}_{1:m}(h) \leftarrow$ unassigned
    Let $\mathcal{P}_h$ be a $(\rho, 2r^{(\ell)}, \frac{7}{8})$-padded decomposition of $\mathcal{W}$ with $\rho = \mathcal{O}(r^{(\ell)} log(\frac{2}{\mu}))$.
    for $T \in \mathcal{P}_h$ do
      Let $B(u, \rho)$ be a ball containing $T$. Run Algorithm 1 on $\mathcal{H} \cap B(u, \rho)$, with radius $r = \rho$
      and origin shifted to $u$.
      for each $\hat{\mathbf{w}}_i \in T$ assign $\bar{\mathbf{w}}_i(h)$ as the outputs of Algorithm 1.
    end for
  end for
  for $i = 1$ to $m$ do
    Find a $h_0$ such that $\|\bar{\mathbf{w}}_i(h_0) - \bar{\mathbf{w}}_i(h)\|_2 \leq \frac{1}{3} r^{(\ell)}$ for at least $\frac{1}{2}$ of the  $h$'s.
    $\hat{\mathbf{w}}^{(\ell+1)}_i \leftarrow \bar{\mathbf{w}}_i(h_0)$ (or "unassigned" if no such $h_0$ exists)
  end for
  $r^{(\ell+1)} \leftarrow \frac{1}{2} r^{(\ell)}$
end for
```

2.2 Analysis of List-regression, Algorithm 4

The analysis will require a "local" spectral norm bound that gives a tighter bound for any β fraction of points. This analysis largely follows the same outline as Charikar et al. (2017), but differs in some key details. By the local spectral bound, in each iteration, we get good estimates of $\mathbf{w}$ for any sufficiently large subset. A key observation is that in contrast to Charikar et al. (2017), we do not "lose" points from our clusters across iterations since our terms are all large enough that they are preserved. This enables a potentially arbitrarily-close approximation of $\mathbf{w}^*$ given enough data.

2.2.1 Local Spectral Norm Bound

For $\beta < 1$, we define a local spectral norm bound S_β on arbitrary subsets T in I_{good}, such that T takes up at least a β fraction of I_{good} ($N_T \geq \beta N$). Denote the number of points in T by N_T and the number of terms by m_T. We define

$$S_\beta := \max_{w \in \mathcal{H},} \max_{\substack{T \subset I_{good} \\ N_T \geq \beta N}} \frac{1}{\sqrt{m_T}} \|[\nabla f_j(\mathbf{w}) - \nabla \bar{f}(\mathbf{w})]_{j \in T}\|_{op}$$

Similar to the analysis of S, $S_\beta = \mathcal{O}(rS_{\beta_0})$, where recall, $S_{\beta_0} := \max_{T: N_T \geq \beta N} \Big\| \Big[\frac{1}{|t_j|} \sum_{i \in t_j} \mathbf{y}^{(i)} \mathbf{y}^{(i)\top} - \mathbb{E}[\mathbf{y}\mathbf{y}^\top \Big]_{j \in T} \Big\|_{op}$ is bounded by a constant. We denote the value of S_β in the ℓ^{th} iteration by $S_\beta^{(\ell)}$, where $S_\beta^{(\ell)} = \mathcal{O}(r^{(\ell)} S_{\beta_0})$.

With the local spectral norm bound for the gradients, we can obtain the local version of Lemma 1.6

Lemma 2.2 (c.f. Lemma 5.2 of Charikar et al. (2017)) *Let the weights $b_i \in [0, 1]$ satisfy $\sum_{i \in I_{good}} b_i |t_i| \geq \beta \mu N$, and define $\hat{\mathbf{w}}^b_{avg} := \sum_{i \in I_{good}} b_i |t_i| \hat{\mathbf{w}}_i / \sum_{i \in I_{good}} b_i |t_i|$. Then the output of Algorithm 1, $\hat{\mathbf{w}}_{1:m}, \hat{Y}$ satisfies*

$$\sum_{i \in I_{good}} b_i |t_i| \Big(f_i(\hat{\mathbf{w}}^b_{avg}) - f_i(\hat{\mathbf{w}}_i) \Big) \leq b_i |t_i| \langle \nabla f_i(\hat{\mathbf{w}}^b_{avg}), \hat{\mathbf{w}}^b_{avg} - \hat{\mathbf{w}}_i \rangle \leq \Big(\sum_{i \in I_{good}} b_i |t_i| \Big) t S_\beta \Big(\sqrt{tr(\hat{Y})} + r \Big)$$

Moreover, for any $\mathbf{w}, \mathbf{w}' \in \mathcal{H}$, we have

$$\Big| \sum_{i \in I_{good}} b_i |t_i| \Big(\bar{f}(\mathbf{w}) - \bar{f}(\mathbf{w}') \Big) - \sum_{i \in I_{good}} b_i |t_i| \Big(f_i(\mathbf{w}) - f_i(\mathbf{w}') \Big) \Big| \leq 2 \Big(\sum_{i \in I_{good}} b_i |t_i| \Big) tr S_\beta.$$

The proof is similar to Lemma 1.6.

2.2.2 Proof of the Main Theorem

We can now state and prove our main theorem for list regression. As noted at the outset, we will need to assume that the distribution over I_{good} is sufficiently similar relative to the degree of (strong) convexity of the loss.

Theorem 2.3 *Let any r_{final} and $\delta, \beta \leq \frac{1}{2}$ be given. Suppose that the loss functions f_i are κ-strongly convex and $S_{\beta_0} \leq \mathcal{O}(\frac{\kappa \sqrt{\mu}}{t \log(1/\mu)})$ for all $i \in I_{good}$. For $N = \mathcal{O}(\sigma_\epsilon^2 \log(m/\delta)/\beta \mu r^2_{final}))$ example points, let $\mathcal{U}, \hat{\mathbf{w}}_{1:m}$ be the output of Algorithm 4. Then with probability at least $1 - \delta$, $\mathcal{U}$ has size at most $\lfloor \frac{1}{(1-\beta)\mu} \rfloor$, and $\min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \mathbf{w}^*\|_2 \leq \mathcal{O}(r_{final})$. Moreover, $\|\hat{\mathbf{w}}_i - \mathbf{w}^*\|_2 \leq \mathcal{O}(r_{final})$ for every term $i \in I_{good}$.*

Towards proving Theorem 2.3, our main step is the following bound on the quality of a single iteration of Algorithm 4:

Theorem 2.4 *For some absolute constant C, the output $\hat{\mathbf{w}}_{1:m}$ of Algorithm 1 during Algorithm 4 satisfies*

$$\|\hat{\mathbf{w}}_i - \mathbf{w}^*\|_2^2 \leq C \cdot \frac{r^{(\ell)} t S_\beta^{(\ell)}}{\kappa\sqrt{\mu}}$$

for all terms $i \in I_{good}$.

The key to establishing Theorem 2.4 will be to use the bound on the statistical error from Lemma 2.2 and the strong convexity of f_i.

Lemma 2.5 *For any $b_i \in [0,1]$ satisfying $\sum_{i\in I_{good}} b_i|t_i| \geq \beta\mu N$, we have*

$$\frac{\sum_{i\in I_{good}} b_i|t_i| \|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2}{\sum_{i\in I_{good}} b_i|t_i|} \leq \frac{2}{\kappa}(\sqrt{\operatorname{tr}(\hat{Y})} + r)tS_\beta \tag{6}$$

Proof of Lemma 2.5 Recall that Lemma 2.2 says that for any $b_i \in [0,1]$ satisfying $\sum_{i\in I_{good}} b_i|t_i| \geq \beta\mu N$, we have

$$\sum_{i\in I_{good}} b_i|t_i| \langle \nabla f_i(\hat{\mathbf{w}}^b_{avg}), \hat{\mathbf{w}}^b_{avg} - \hat{\mathbf{w}}_i\rangle \leq \Big(\sum_{i\in I_{good}} b_i|t_i|\Big) tS_\beta\Big(\sqrt{\operatorname{tr}(\hat{Y})} + r\Big)$$

By strong convexity of f_i, we have

$$\begin{aligned}
0 &\leq \sum_{i\in I_{good}} b_i|t_i|\big(f_i(\hat{\mathbf{w}}^b_{avg}) - f_i(\hat{\mathbf{w}}_i)\big) \\
&\leq \sum_{i\in I_{good}} b_i|t_i|\Big(\langle \nabla f_i(\hat{\mathbf{w}}^b_{avg}), \hat{\mathbf{w}}^b_{avg} - \hat{\mathbf{w}}_i\rangle - \frac{\kappa}{2}\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}^b_{avg}\|_2^2\Big) \\
&\leq \left(\sum_{i\in I_{good}} b_i|t_i|\right) tS_\beta\Big(\sqrt{\operatorname{tr}(\hat{Y})} + r\Big) - \frac{\kappa}{2}\sum_{i\in I_{good}} b_i|t_i| \|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}^b_{avg}\|_2^2.
\end{aligned}$$

■

By applying Lemma 2.5 to $b'_i = \frac{1}{2}\Big(b_i + \frac{\sum_j b_j|t_j|}{\sum_j c_j|t_j|}c_i\Big)$, we obtain Lemma 2.6, which gives bounds in terms of $\hat{\mathbf{w}}_{avg}$ rather than $\hat{\mathbf{w}}^b_{avg}$:

Lemma 2.6 *For any $b_i \in [0,1]$ satisfying $\beta\mu N \leq \sum_{i\in I_{good}} b_i|t_i| \leq \sum_{i\in I_{good}} c_i|t_i|$, we have*

$$\frac{\sum_{i\in I_{good}} b_i|t_i| \|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2}{\sum_{i\in I_{good}} b_i|t_i|} \leq \frac{10}{\kappa}(\sqrt{\operatorname{tr}(\hat{Y})} + r)tS_\beta \tag{7}$$

Proof of Lemma 2.6 For convenience, let us define: $B = \sum_{i\in I_{good}} b_i|t_i|$, $C = \sum_{i\in I_{good}} c_i|t_i|$, $b'_i = \frac{1}{2}\Big(b_i + \frac{\sum_j b_j|t_j|}{\sum_j c_j|t_j|}c_i\Big) = \frac{1}{2}b_i + \frac{1}{2}\frac{B}{C}c_i$. Notice that $\sum_{I_{good}} b'_i|t_i| = B$ and $\hat{\mathbf{w}}^{b'}_{avg} = \frac{1}{2}\hat{\mathbf{w}}^b_{avg} + \frac{1}{2}\hat{\mathbf{w}}_{avg}$.

We invoke Lemma 2.5 twice, on b and b' respectively:

$$\begin{aligned}\frac{1}{B}\sum_{i\in I_{good}} b'_i|t_i|\|\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^b_{avg} - \frac{1}{2}\hat{\mathbf{w}}_{avg}\| = \frac{1}{B}\sum_{i\in I_{good}} b'_i|t_i|\|\frac{1}{2}\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^{b'}_{avg}\| &\le \frac{2}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta \\ \frac{1}{B}\sum_{i\in I_{good}} b_i|t_i|\|\frac{1}{2}\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^{b}_{avg}\| &\le \frac{1}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta.\end{aligned}$$

Since for any i, $b'_i \le \frac{1}{2}b_i$

$$\frac{1}{B}\sum_{i\in I_{good}} b_i|t_i|\|\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^b_{avg} - \frac{1}{2}\hat{\mathbf{w}}_{avg}\| \le \frac{2}{B}\sum_{i\in I_{good}} b'_i|t_i|\|\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^b_{avg} - \frac{1}{2}\hat{\mathbf{w}}_{avg}\| \le \frac{4}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta$$

Combining the two inequalities

$$\begin{aligned}\frac{1}{B}\sum_{i\in I_{good}} b'_i|t_i|\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}^b_{avg}\| &\le \frac{1}{B}\sum_{i\in I_{good}} b'_i|t_i|2\Big(\|\hat{\mathbf{w}}_i - \frac{1}{2}\hat{\mathbf{w}}^b_{avg} - \frac{1}{2}\hat{\mathbf{w}}_{avg}\| + \| - \frac{1}{2}\hat{\mathbf{w}}_i + \frac{1}{2}\hat{\mathbf{w}}^b_{avg}\|\Big) \\ &\le 2\Big(\frac{4}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta + \frac{1}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta\Big) \\ &= \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta\end{aligned}$$

■

Corollary 2.7 *In particular, no set of terms comprising more than a $\beta\mu$ fraction of the data (or probability weight) can have $\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 > \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta$.*

Proof Consider terms $t_1, \ldots, t_q$ with $\Pr[t_1 \vee \cdots \vee t_q] > \beta\mu$. Assume for contradiction that for all of these terms, $\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 > \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta$. We can assign b_i for each t_i such that $\sum b_i = \beta\mu$. Then

$$\begin{aligned}\frac{\sum_{i\in I_{good}} b_i|t_i|\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2}{\sum_{i\in I_{good}} b_i|t_i|} &\ge \frac{\sum_{i\in I_{good}} b_i|t_i|}{\sum_{i\in I_{good}} b_i|t_i|}\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 \\ &> \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta\end{aligned}$$

which contradicts Lemma 2.6. ■

Key Observation As we deleted all terms of size smaller than $\beta\mu N$, all the remaining terms have at least $\beta\mu$ probability-weight (or $\beta\mu N$ empirical size). Therefore, every term will satisfy

$$\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 \le \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})}+r)tS_\beta.$$

We can subsequently obtain Theorem 2.4 by thus invoking Corollary 2.7:

Proof of Theorem 2.4: By Corollary 2.7, we have $\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 \le \frac{10}{\kappa}(\sqrt{\mathrm{tr}(\hat{Y})} + r)tS_\beta$ for all $i \in I_{good}$. $\mathrm{tr}(\hat{Y}) \le \mathcal{O}(\frac{r^2}{\mu})$, so $\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 \le \frac{10}{\kappa}(\frac{r}{\sqrt{\mu}} + r)tS_\beta = \mathcal{O}(\frac{rtS_\beta}{\kappa\sqrt{\mu}})$. In addition, by Theorem 1.3, $\bar{f}(\hat{\mathbf{w}}_{avg}) - \bar{f}(\mathbf{w}^*) \le \mathcal{O}(\frac{rtS_\beta}{\sqrt{\mu}})$. By the strong convexity of $\bar{f}$, $\|\hat{\mathbf{w}}_{avg} - \mathbf{w}^*\|_2^2 \le \mathcal{O}(\frac{rtS_\beta}{\sqrt{\mu}})$. We combine the bounds to obtain $\|\hat{\mathbf{w}}_i - \mathbf{w}^*\|_2^2 \le 2(\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_{avg}\|_2^2 + \|\hat{\mathbf{w}}_{avg} - \mathbf{w}^*\|_2^2) \le \mathcal{O}\Big(\frac{rtS_\beta}{\sqrt{\mu}}\Big)$ ■

Finally, using Theorem 2.4, we show the radius $r^{(\ell)}$ (used in the ℓ^{th} iteration) can be decreased by half at each iteration.

Lemma 2.8 *In Algorithm 4, denote the set of parameters of good points of ℓ^{th} iteration by $I_{good}^{(\ell)} := \{\hat{\mathbf{w}}_i^{(\ell)} : i \in I_{good}\}$. If $\|\hat{\mathbf{w}}_i^{(\ell)} - \mathbf{w}^*\|_2 \le r^{(\ell)}$ and $S_{\beta_0} \le C' \cdot \frac{\kappa\sqrt{\mu}}{t\log(2/\mu)}$ for some constant C', then with probability $(1 - \frac{\delta}{\ell(\ell+1)})$ over the randomly chosen padded decompositions, $\|\hat{\mathbf{w}}_i^{(\ell+1)} - \mathbf{w}^*\|_2 \le \frac{1}{2}r^{(\ell)}$.*

Proof We call a padded decomposition partition $\mathcal{P}_h$ *good* if all of the terms of $I_{good}^{(\ell)}$ lie in a single cluster of $\mathcal{P}_h$. Denote the set of padded decompositions where $\mathcal{P}_h$ is good by H.

In the algorithm, we draw $q = 112\log\frac{\ell(\ell+1)}{\delta}$ random padded decompositions with parameters $(\rho, 2r^{(\ell)}, \frac{1}{8})$, where $\rho = \mathcal{O}(r^{(\ell)}\log\frac{2}{\mu})$, so that (i) each cluster P of $\mathcal{P}_h$ has diameter at most $\mathcal{O}(r^{(\ell)}\log\frac{2}{\mu})$, and (ii) for each padded decomposition and a parameter vector, all other parameter vectors within $2r^{(\ell)}$ will lie in the same cluster with probability 7/8.

Since we assume $\|\hat{\mathbf{w}}_i^{(\ell)} - \mathbf{w}^*\|_2 \le r^{(\ell)}$, with probability $\frac{7}{8}$, all of $I_{good}^{(\ell)}$ will lie in a single cluster, i.e. this padded decomposition $\mathcal{P}_h$ is good. Then, by a Chernoff Bound, the total number of good padded decompositions will be larger than $\frac{3}{4}q$ with probability $1 - \frac{\delta}{\ell(\ell+1)}$.

For a good padded decomposition (P is the cluster containing the terms of $I_{good}^{(\ell)}$), $\mathbf{w}^*$ is within distance $r^{(\ell)}$ of P. Therefore, if $P \subset B(u, \rho)$, then $\mathbf{w}^* \in B(u, \rho + r^{(\ell)})$. As we run Algorithm 1 on $B(u, \rho + r^{(\ell)})$, Theorem 2.4 will give us:

$$
\begin{aligned}
\|\bar{\mathbf{w}}_i(h) - \mathbf{w}^*\|_2^2 &\le C \cdot \frac{trS_\beta}{\kappa\sqrt{\mu}} \\
\|\bar{\mathbf{w}}_i(h) - \mathbf{w}^*\|_2 &\le \mathcal{O}\Bigg(\sqrt{\frac{t(\rho + r^{(\ell)})S_\beta^{(\ell)}}{\kappa\sqrt{\mu}}}\Bigg) \\
&= \mathcal{O}\Bigg(\sqrt{\frac{tr^{(\ell)}\log\frac{2}{\mu}\ S_\beta^{(\ell)}}{\kappa\sqrt{\mu}}}\Bigg)
\end{aligned}
$$

where $\bar{\mathbf{w}}_i(h)$ is the output $\hat{\mathbf{w}}_i$ of Algorithm 1.

We want to show $\|\bar{\mathbf{w}}_i(h) - \mathbf{w}^*\|_2 \le \frac{1}{6}r^{(\ell)}$. Recall that $S_\beta^{(\ell)} \le \mathcal{O}(S_{\beta_0}r^{(\ell)})$, so it suffices to have

$$
r^{(\ell)} \cdot \sqrt{\frac{t\log\frac{2}{\mu}\ S_{\beta_0}}{\kappa\sqrt{\mu}}} \le \mathcal{O}(r^{(\ell)}),
$$

i.e., $S_{\beta_0} \le \mathcal{O}\Big(\frac{\kappa\sqrt{\mu}}{t\log\frac{2}{\mu}}\Big)$, which is true by hypothesis (for some suitable C').

Since $\|\bar{\mathbf{w}}_i(h) - \mathbf{w}^*\|_2 \leq \frac{1}{6}r^{(\ell)}$ for any two good iterations h and h', and each iteration is only bad with probability $\leq \frac{1}{4}$, we can pick h_0 such that $\|\bar{\mathbf{w}}_i(h_0) - \bar{\mathbf{w}}_i(h)\|_2 \leq \frac{1}{3}r^{(\ell)}$ is true for half of the iterations. For any good h,

$$\begin{aligned}\|\bar{\mathbf{w}}_i(h_0) - \mathbf{w}^*\|_2 &\leq \|\bar{\mathbf{w}}_i(h_0) - \bar{\mathbf{w}}_i(h)\|_2 + \|\bar{\mathbf{w}}_i(h) - \mathbf{w}^*\|_2 \\ &\leq \frac{1}{3}r^{(\ell)} + \frac{1}{6}r^{(\ell)} \\ &\leq \frac{1}{2}r^{(\ell)}\end{aligned}$$

That is, $\hat{\mathbf{w}}_i^{(\ell+1)} := \bar{\mathbf{w}}_i(h_0)$ is within $\frac{1}{2}r^{(\ell)}$ of $\mathbf{w}^*$. ■

Proof of Theorem 2.3: Lemma 2.8 shows that on the ℓth iteration, $\mathbf{w}^*$ lies in a ball of radius $r^{(\ell)}$ around each $\hat{\mathbf{w}}_i^{(\ell)}$ for $i \in I_{good}$, where $r^{(\ell)}$ decreases by half on each iteration. In the final iteration, when $r^{(\ell)}$ reaches the target accuracy radius r_{final}, the algorithm greedily finds disjoint balls $B(\mathbf{u}_j; 2r_{\text{final}})$ on the parameter space $\mathcal{W}$, such that the corresponding terms for the covered parameters contain at least $(1-\beta)\mu N$ points, i.e for each ball B,

$$\sum_{\hat{\mathbf{w}}_i \in B} |t_i| \leq (1-\beta)\mu N$$

Since for all $i \in I_{good}$, $\|\hat{\mathbf{w}} - \mathbf{w}^*\| < r_{final}$, we now argue that all of the terms in I_{good} will lie in one ball. Indeed, if no term of I_{good} is contained in any ball, then any term of I_{good} gives a candidate for the greedy algorithm to add to the list. Therefore, at least one of the good terms $\hat{\mathbf{w}}_i, i \in I_{good}$ must be contained in some ball. Then for this ball $B(\mathbf{u}, r_{final})$,

$$\begin{aligned}\|\mathbf{u} - \mathbf{w}^*\| &\leq \|\mathbf{u} - \hat{\mathbf{w}}_i\| + \|\hat{\mathbf{w}}_i - \mathbf{w}^*\| \\ &\leq 2r_{final} + r_{final} \\ &= \mathcal{O}(r_{final})\end{aligned}$$

Since each ball contains at least $(1-\beta)\mu$ points, there can be at most $\lfloor \frac{1}{(1-\beta)\mu} \rfloor$ such balls, which completes the proof. ■

3 Obtaining a k-DNF Condition

Once we get outputs $\{\mathbf{u}_1, ..., \mathbf{u}_s\}$ from Algorithm 4, we switch from the parameter space $\{\mathbf{w}\}$ back to the Boolean data space $\{\mathbf{x}\}$, to search for corresponding clusters $\mathbf{c}$ for each candidate parameter $\mathbf{u}_i$. If we find a pair $(\mathbf{u}, \mathbf{c})$ such that $\mathbf{c}$ contains enough points and the loss $f_{\mathbf{c}}(\mathbf{u})$ is small, we return this pair as the final solution.

Suppose $\mathbf{u}$ is one of the candidates such that $\|\mathbf{u} - \mathbf{w}^*\| < \mathcal{O}(r_{final}) =: \gamma$, then $|\bar{f}(\mathbf{u}) - \bar{f}(\mathbf{w}^*)| \leq \gamma L = \mathcal{O}(\gamma)$, for some Lipschitz constant L (since f is just a regression loss on a bounded space, it is Lipschitz continuous). Recalling $\bar{f}$ is nonnegative, if $\bar{f}(\mathbf{w}^*) \leq \epsilon$, then $\bar{f}(\mathbf{u}) \leq \gamma + \epsilon$.

3.1 Bounding the Double-Counting Effect

We now address the effect of our double-counting of points. Recall that we introduced a copy of a point for each term it satisfied. Observe that on I_{good}, which contains t terms, this is at most t copies. We thus obtain

Lemma 3.1 *Let* $\mathbf{u}$ *be such that* $\|\mathbf{u} - \mathbf{w}^*\| < \gamma$*. Then* $|\bar{f}(\mathbf{u})| \leq t(\gamma + \epsilon)$ *(for the true distribution, without duplicated points).*

Proof Assume $\mathbf{w}^*_{true}$ is the true optimal linear fit: ignoring the common $1/|I_{good}|$ scaling,

$$\mathbf{w}^*_{true} := \underset{\mathbf{w}}{\operatorname{argmin}} \bar{f}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i \in I_{good}} f^{(i)}(\mathbf{w})$$

and $\mathbf{w}^*$ is the optimal linear fit for the double counted data: letting $a^{(i)} \in [1, t]$ denote the number of copies of each point after duplication and again ignoring the $\sum_{i \in I_{good}} a^{(i)}$ scaling factor,

$$\mathbf{w}^* := \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i \in I_{good}} a^{(i)} f^{(i)}(\mathbf{w}).$$

Now, for any $\mathbf{w}$, observe that $\sum_{i \in I_{good}} f^{(i)}(\mathbf{w}) \leq \sum_{i \in I_{good}} a^{(i)} f^{(i)}(\mathbf{w})$ and

$$\sum_{i \in I_{good}} a^{(i)} f^{(i)}(\mathbf{w}) \leq (\max_{i \in I_{good}} a^{(i)}) \sum_{i \in I_{good}} f^{(i)}(\mathbf{w}) \leq t \sum_{i \in I_{good}} f^{(i)}(\mathbf{w}).$$

Therefore,

$$|I_{good}|\bar{f}(\mathbf{w}^*) \leq |I_{good}| \sum_{i \in I_{good}} a^{(i)} f^{(i)}(\mathbf{w}^*) \leq |I_{good}| t \sum_{i \in I_{good}} a^{(i)} f^{(i)}(\mathbf{w}^*_{true}) \leq t|I_{good}|\bar{f}(\mathbf{w}^*_{true})$$

Therefore, if $|\bar{f}(\mathbf{w}^*_{true})| \leq \gamma + \epsilon$, then $|\bar{f}(\mathbf{w}^*)| \leq t(\gamma + \epsilon)$ ■

3.2 Greedy Set-Cover

We have obtained a parameter vector $\mathbf{u}$ such that the loss for each term $f_i(\mathbf{u})$ is close to $f_i(\mathbf{w}^*)$. We can now use a greedy set-cover algorithm to find the corresponding clusters $\mathbf{c}$, following the approach of Juba et al. (2018). At a high level, given regression parameters $\mathbf{u}$, we compute the loss $f(\mathbf{u})$ for each point, and then use the covering algorithm to find a collection of terms that cover enough points while minimizing the loss. Specifically, the algorithm greedily chooses terms t_j satisfying $\sum_{i \in t_j} f^{(i)}(\mathbf{u}) \leq (1 + \gamma)\mu\epsilon N$ to maximize the number of additional examples $(\mathbf{x}, \mathbf{y}, z)^{(i)}$ with $t_j(\mathbf{x}^{(i)}) = 1$ that did not satisfy a previously chosen term. It continues choosing terms this way until at least $(1 - \gamma/2)\mu N$ examples satisfy the collection of chosen terms. In other words, it associates with t_j the set of examples such that $t_j(\mathbf{x}^{(i)}) = 1$, and considers the collection of such sets corresponding to terms t_j with empirical loss $|t_j|f_j(\mathbf{u}) \leq (1 + \gamma)\mu\epsilon N$. It then follows the standard greedy algorithm for unweighted partial set cover on this instance.

Lemma 3.2 *Given a set of points* $\{\mathbf{x}^{(i)}\}_{i=1}^N$ *with weights* $f^{(i)}(\mathbf{u})$ *and terms* $\{t_j\}_{j=1}^m$*, if there exists an optimal k-DNF* $\mathbf{c}^*$ *that is satisfied by a* μ*-fraction of the points with total loss* ϵ*, then, the weighted greedy set cover algorithm can find a k-DNF* $\hat{\mathbf{c}}$*, that is satisfied by a* $(1 - \gamma)\mu$*-fraction of the points with total loss* $\mathcal{O}(t \log(\mu N)\epsilon)$

Proof Observe that since the loss is non-negative, each term t_j of $\mathbf{c}^*$ must satisfy

$$\mathbb{E}[f(\mathbf{u})|t_j]\Pr[t_j(\mathbf{x})=1] \leq \mathbb{E}[f(\mathbf{u})|\mathbf{c}]\Pr[\mathbf{c}(\mathbf{x})=1] \leq \mu(\epsilon+\gamma)$$

by Lemma 3.1. So, for $N = \mathcal{O}(\frac{\sigma^2 L^2}{\mu\epsilon\gamma^2}\log\frac{m}{\delta})$ examples, a union bound over the terms gives that *(i)* every term of $\mathbf{c}$ has empirical loss at most $(1+\gamma)\mu\epsilon N$ and conversely, *(ii)* every term t' with empirical loss $(1+\gamma)\mu\epsilon N$ has $\mathbb{E}[f(\mathbf{u})|t']\Pr[t'(\mathbf{x})] \leq \mathcal{O}(\mu\epsilon)$. Furthermore it follows from an analysis by Haussler 1988 that given $N = \mathcal{O}(\frac{t}{\mu\gamma^2}\log\frac{m}{\delta})$ examples, *(iii)* any t-term formula (out of the m possible terms) that empirically satisfies at least $(1-\gamma/2)\mu N$ examples must be satisfied with probability at least $(1-\gamma)\mu$ overall, and conversely, *(iv)* any formula that is true with probability at least μ will empirically satisfy at least $(1-\gamma/2)\mu N$ examples. The above will simultaneously hold with probability $1-\delta$ for suitable constants.

Now, by *(i)* the terms of $\mathbf{c}$ are available to the greedy algorithm, which can thus by *(iv)* cover $(1-\gamma)\mu N$ examples using at most t sets. Slavík 1997 has shown that the greedy algorithm obtains a $H((1-\gamma)\mu N)$-approximation to the minimum size set cover, where $H(\ell)$ denotes the ℓth harmonic number, which is $\leq \log(\mu N)+1$. Thus, the greedy algorithm finds a formula $\hat{\mathbf{c}}$ with at most $t(\log(\mu N)+1)$ terms. By *(ii)*, since $\hat{c}$ only contains terms with empirical loss $(1+\gamma)\mu\epsilon N$, $\mathbb{E}[f(\mathbf{u})|\hat{c}]\Pr[\hat{c}(\mathbf{x})=1] \leq \mathcal{O}(t\log(\mu N)\mu\epsilon)$. Furthermore, by *(iii)*, $\Pr[\hat{c}(\mathbf{x})=1] \geq (1-\gamma)\mu$, which completes the proof. ■

3.3 Generalization Bound

Lemma 3.2 gives us the sample complexity needed for a specific realization. However, there is still a gap between the data and true underlying distribution. In this section, we will bound the generalization error of linear regression on each possible k-DNF, and then take a union bound to achieve the main theorem. In short, the process will blow-up the complexity by d^3, where d is the dimension of the feature space.

We will use the Rademacher generalization bound for linear predictors. For a set a data, Lemma 3.3 bounds the expected loss $L_p(\cdot)$ and the empirical loss $\hat{L}_p(\cdot)$:

Lemma 3.3 (Bartlett & Mendelson (2002), Kakade et al. (2009)) *For $b>0$, $p\geq 1$, random variables $(\mathbf{Y},Z)$ distributed over $\{\mathbf{y}\in\mathbb{R}^d : \|y\|_2 \leq b\}\times[b,b]$, and any $\delta\in(0,1)$, let $L_p(\mathbf{w})$ denote $\mathbb{E}[|\langle\mathbf{w},\mathbf{Y}\rangle - Z|^p]$, and for an an i.i.d. sample of size N let $\hat{L}_p(\mathbf{w})$ be the empirical loss $\frac{1}{N}\sum_{j=1}^N |\langle\mathbf{w},\mathbf{y}^{(j)}\rangle - z^{(j)}|^p$. We then have that with probability $1-\delta$ for all $\mathbf{w}$ with $\|w\|_2\leq b$,*

$$|L_p(\mathbf{w}) - \hat{L}_p(\mathbf{w})| \leq \frac{2pb^{p+1}}{\sqrt{N}} + b^p\sqrt{\frac{2\ln(4/\delta)}{N}}.$$

In our case, we only consider squared error; in other words, $p=2$ for us. And notice, in our setting, we are given a bound B on the magnitude of the entries, so $b\leq\sqrt{s}B$. Equivalently, we get

$$|L_p(\mathbf{w}) - \hat{L}_p(\mathbf{w})| \leq \frac{4B^3 d^{\frac{3}{2}}}{\sqrt{N}} + o(B^2 d).$$

Therefore, to bound the gap of the expected loss $L_p(\cdot)$ and the empirical loss $\hat{L}_p(\cdot)$, it suffices for our sample complexity N to grow with $B^6 d^3$.

The above lemma bounds the gap of a specific set. To achieve the bound on any t term k-DNF, we can simply take a union bound on all k-DNFs. Since x has n Boolean attributes, there are $m = \binom{n}{k}$ possible terms, which is at most $m = n^k$. And there are $\binom{m}{t}$ t-term k-DNFs, i.e., n^{kt} in total, which means that if we replace δ with $\frac{\delta}{n^{kt}}$, we will obtain $1 - \delta$ confidence after the union bound. Overall we thus achieve a $\mathcal{O}(t \log(\mu N)(\gamma + \epsilon))$ approximation as claimed in the main theorem with $N = \mathcal{O}(\frac{B^6 d^3 \sigma^2 L^2 t}{\mu \gamma^3} \log(\frac{m}{\delta/m^t}))) = \mathcal{O}(\frac{B^6 d^3 \sigma^2 L^2 t^2}{\mu \gamma^3} \log(m/\delta)))$ examples.

4 Synthetic Data Experiment

The synthetic data experiment is designed to demonstrate our algorithm's ability to solve problems that cannot be handled by the sparse regression algorithms. We choose a t-term 2-DNF at random and uniformly generate Boolean attributes serving as $\mathbf{x}$, where μN of them satisfy the chosen DNF (good data). The $\mathbf{y}$ parts are all uniformly generated real attributes. We also generate a target optimal linear rule $\mathbf{w}^*$ with dimension equal to that of $\mathbf{y}$. For the good data, we set their labels $z^{(i)} = \langle \mathbf{y}^{(i)}, \mathbf{w}^* \rangle + noise$, where the noise is independently generated from zero-mean Gaussian distribution. For the bad data, $z^{(i)}$ are independently generated from a uniform distribution, similar to $\mathbf{y}$. We use our algorithm to generate a list of candidate parameters and their corresponding DNF. If there is one pair that is close to our planted $\mathbf{w}^*$ and DNF, or other output with even lower error, then we view the task as successful.

Specifically, we set the $dim(\mathbf{x}) = 7, dim(\mathbf{y}) = 10$, with $N = 100000$ points in total and $\mu = 0.5$. We generate $\mathbf{w}^*$ uniformly from $[-10, 10]$ and randomly choose a 4-term 2-DNF to define which points are good data. For the bad data, $\mathbf{y}^{(i)} \in [-1, 1], z^{(i)} \in [-10, 10]$. For good data, $\mathbf{y}^{(i)} \in [-1, 1]$ and $z^{(i)} = \langle \mathbf{y}^{(i)}, \mathbf{w}^* \rangle + noise$ with variance 100. We set $S = 0.1, \gamma = 0.1$, $r = 100$ and $r_{final} = 1$. We ran 5 trials and each time our algorithm can find several pairs of regression parameters and DNFs, with one of them to be the planted DNF. There are also other pairs of output with even lower error, meaning the algorithm finds even better conditions than our planted ones. Note that since the previous algorithms (Juba, 2017; Hainline et al., 2019) scale exponentially with d, such an instance would be infeasible to solve.

References

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Charikar, M., Steinhardt, J., and Valiant, G. Learning from untrusted data. In *Proc. 49th STOC*, pp. 47–60, 2017. Full version arXiv:1611.02315v2 [cs.LG].

Fakcharoenphol, J., Rao, S., and Talwar, K. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. 35th STOC*, pp. 448–455, 2003.

Hainline, J., Juba, B., Le, H. S., and Woodruff, D. P. Conditional sparse ℓ_p regression with optimal probability. In *Proc. 22nd AISTATS*, volume 89 of *PMLR*, pp. 369–382, 2019.

Haussler, D. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

Juba, B. Conditional sparse linear regression. In *Proc. 8th ITCS*, pp. 45:1–45:14, 2017.

Juba, B., Li, Z., and Miller, E. Learning abduction under partial observability. In *Proc. 32nd AAAI*, pp. 1888–1896, 2018.

Kakade, S. M., Sridharan, K., and Tewari, A. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems 21*, pp. 793–800, 2009.

Slavík, P. Improved performance of the greedy cover algorithm for partial cover. *Information Processing Letters*, 64(5):251–254, 1997.

Table 1: Table of notation

Data & DNFs	
$\mathbf{x}^{(i)} \in \{0,1\}^n$	The Boolean attributes (for defining conditions)
n	The dimension of $\mathbf{x}^{(i)}$
$\mathbf{y}^{(i)} \in \mathbb{R}^d$	The real attributes (factors for predictors)
d	The dimension of $\mathbf{y}^{(i)}$
$z^{(i)} \in \mathbb{R}$	The labels (dependent variable for regression)
$N \in \mathbb{N}$	The number of examples $\{(\mathbf{x}, \mathbf{y}, z)^{(i)}\}_{i=1}^N$
k	The maximum number of literals in the terms
t_i	A term defined by Boolean attributes with at most k literals.
$m \leq n^k$	The total number of terms
$\mathbf{w}_i \in \mathbb{R}^d$	The parameters of the linear predictor assigned to the term t_i
$\mathbf{c}$	A DNF formula defined as **or** of terms
	It is both a Boolean formula, and the subset of data satisfying that formula.
$\mathbf{c}^*$	A target optimal DNF in the data set
I_{good}	The subset of terms contained in $\mathbf{c}^*$
$\mathbf{w}^* \in \mathbb{R}^d$	The parameters of the linear predictor assigned to the DNF $\mathbf{c}^*$
t	The number of terms in $\mathbf{c}^*$
B	Bound on $\lVert\mathbf{y}\rVert_2$
r	Bound on $\lVert\mathbf{w}\rVert_2$
ϵ	The error of the optimal DNF $\mathbf{c}^*$
μ	The fraction of points of the dataset in $\mathbf{c}^*$
f	The loss function $f(\mathbf{w}) = (\langle \mathbf{w}, \mathbf{y} \rangle - z)^2$
κ	The convexity coefficient of f
L	The Lipschitz coefficient of f
σ	The standard error parameter of the subgaussian residuals $(\langle \mathbf{w}^*, \mathbf{y}_i \rangle - z_i)$
δ	The probability threshold of failure
γ	The coverage threshold, a portion of data we can lose.
$\beta \leq \gamma/t$	A portion of data we can lose after duplication.
S	A spectral bound on the change in covariances of terms in I_{good}, maximized over $\mathbf{w}$
S_0	A intrinsic bound of S, independent of $\mathbf{w}$
S_β	A local bound of S, maximized over subsets of terms of probability larger than β.
S_{β_0}	A local bound of S_0, maximized over subsets of terms of probability larger than β.
Algorithm 1	
r	The radius of the parameter space
Y	A positive semi-definite matrix defining an ellipsoid containing $\mathbf{w}$
λ	The regularization coefficient.
c_i	The weights (soft indicators) of each term t_i.
Algorithm 2	
ℓ	The iteration / time step.
r_{final}	An upper bound on the desired final radius, for termination
$\mathbf{u}$	The final output of the weight parameters
$\mathcal{P}_h$	A Padded Decomposition.
ρ	The cluster radius of the Padded Decompositions.