Conditional Sparse Linear Regression

Brendan Juba[*]

Washington University, St. Louis, USA
bjuba@wustl.edu

Abstract

Machine learning and statistics typically focus on building models that capture the vast majority of the data, possibly ignoring a small subset of data as “noise” or “outliers.” By contrast, here we consider the problem of *jointly* identifying a significant (but perhaps small) segment of a population in which there is a highly sparse linear regression fit, together with the coefficients for the linear fit. We contend that such tasks are of interest both because the models themselves may be able to achieve better predictions in such special cases, but also because they may aid our understanding of the data. We give algorithms for such problems under the sup norm, when this unknown segment of the population is described by a k-DNF condition and the regression fit is s-sparse for constant k and s. For the variants of this problem when the regression fit is *not* so sparse or using expected error, we also give a preliminary algorithm and highlight the question as a challenge for future work.

1998 ACM Subject Classification G.3 Probability and Statistics, F.2.0 Analysis of Algorithms and Problem Complexity: General

Keywords and phrases Linear regression, conditional regression, conditional distribution search

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.45

1 Introduction

Linear regression, the fitting of linear relationships among variables in a data set, is a standard tool in data analysis. In particular, for the sake of interpretability and utility in further analysis, we desire to find *highly sparse* linear relationships, i.e., involving only a few variables. Of course, such simple linear relationships often will not hold across an entire population. But, more frequently there will exist conditions – perhaps a range of parameters or a segment of a larger population – under which such sparse models fit the data quite well. For example, Rosenfeld et al. [22] used data mining heuristics to identify small segments of a population in which a few additional risk factors were highly predictive of certain kinds of cancer, whereas these same risk factors were not significant in the overall population. Simple rules for special cases may also hint at the more complex general rules. More generally, we need to develop new techniques to reason about populations in which most members are atypical in some way, which are colloquially (and somewhat abusively) referred to as *long-tailed* distributions. We are seeking computationally efficient, principled alternatives to ad-hoc approaches such as trying a variety of methods for clustering the data and hoping that the identified clusters can be modeled well.

[*] Supported by an AFOSR Young Investigator Award.

8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou; Article No. 45; pp. 45:1–45:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Our results

In this work we consider the design and analysis of efficient algorithms for the *joint* task of identifying significant segments of a population in which a sparse model provides a good fit. We are able to identify such segments when they are described by a k-DNF and there is a s-sparse regression fit for constant k and s. More specifically, we give algorithms when there is a linear relationship with respect to which the error is bounded by ϵ with probability 1 (i.e., ϵ sup norm). In this case, we find a condition in which the error is bounded by ϵ for a $1-\gamma$ fraction of the population (with probability $1-\delta$ over the sample of data).

▶ **Theorem 1** (Conditional sparse linear regression). *Suppose that D is a joint probability distribution over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$, and $z \in \mathbb{R}$ such that there is a k-DNF c for which for some s-sparse $\vec{a} \in \mathbb{R}^d$*

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}, \vec{y}\rangle - z| \leq \epsilon \middle| c(\vec{x}) = 1\right] = 1 \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu.$$

Then given ϵ, μ, and δ in $(0,1)$, $\gamma \in (0,1/2]$, and access to examples from D, for any constants s and k, there is an algorithm that runs in polynomial time in n, d, $1/\mu$, $1/\gamma$, and $\log 1/\delta$, and finds an s-sparse $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y}\rangle - z| \leq \epsilon \middle| c'(\vec{x}) = 1\right] \geq 1-\gamma \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq (1-\gamma)\mu.$$

Our algorithms make crucial use of the sought solution's sparsity. The key observation is that since the linear rule has constant sparsity, with respect to the relevant dimensions there are a constant number of "extremal examples" such that we can obtain low error on the unknown event by fitting these extremal examples. We can then use the linear rule we obtain from fitting such a set of examples to label the data according to whether or not that point has low error under the linear rule. Finally, this enables us to find an event on which the linear rule has low error. Thus, it suffices to simply perform a search over candidates for the extremal examples and return one for which the corresponding event captures enough of the data.

We also note a trivial (weak) approximation algorithm for an expected-error variant of the problem that does not rely on sparsity: when there is a k-DNF c and a linear rule a giving conditional expected error ϵ (and c is true with probability μ), we find a condition c' and a linear rule a' with conditional expected error $O(n^k\epsilon)$ and probability $\Omega(\mu/n^k)$. We pose the design of better algorithms for the dense regression and expected-error tasks as challenges for future work.

1.2 Related work

We are building on recent work by Juba [15] on identifying potentially rare events of interest in a distribution, which captures a family of data mining tasks similar to, e.g., association rule discovery [1] or "bump hunting" [10]. This work is closely related to theoretical work on *positive-reliable learning* [16, 17], which is in turn very closely related to the "heuristic learning" model introduced by Pitt and Valiant [21] and studied in depth by Bshouty and Burroughs [5]: these are models of classification in which one type of error is minimized subject to a hard bound on the other type of error. The key difference between heuristic or positive-reliable learning on the one hand, and the work by Juba or the work in data mining on the other, is that the latter works focus on bounding the error *conditioned on the identified event* (i.e., the *precision* rather than the raw *false-positive rate*). In the present work, we develop this perspective further, and seek to perform *supervised learning* in such a conditional distribution. In this context, these earlier works can be viewed as identifying a conditional distribution in which the class consisting solely of the constant 1 function fits the selected points with low error. We are generalizing this to the problem of fitting a (sparse) linear rule in the identified conditional distribution.

Our work also has some relationship to the enormous body of work on *robust statistics* [13, 23], in which *outliers* are identified and ignored or otherwise mitigated. The difference in what we consider here is two-fold. First, we are specifically interested in the case where *we may decline to fit the vast majority of the data,* thus treating most of the data as "outliers" in the model of robust statistics. Second, we are also interested in *finding a (simple) rule that identifies which subset of the data we are fitting* (and which subset we are ignoring). By contrast, in robust statistics, an arbitrary subset of the data may be considered "corrupted" and ignored. Note that without this extra structure, Hardt and Moitra [12] found that the problem of finding subspaces described by a single linear constraint is intractable (precisely, SSE-hard) when the subspace does not contain nearly all ($1 - 1/d$ fraction) of the data.

Similarly, our problem differs from linear mixed models [18, 14] in that linear mixed models seek several linear rules to try to explain (almost) all of the data. Again, in such models, the only description of the "clusters" are the linear models themselves (so points are taken to lie in the cluster of the linear fit with which they have the smallest residual).

Our problem is also very closely related to the problem solved by RANSAC [9] and its variants, that use sampling to find nontrivial linear relationships in data even when these are only of moderate density. The difference is principally that RANSAC is designed to find linear relationships in very low dimension (e.g., in $\mathbb{R}^2$), and does not scale to high dimensions since we need d points to determine a linear fit in $\mathbb{R}^d$, i.e., we need to hit the subspace d times when sampling. In the present work, by contrast, although the linear fit we are seeking is of constant sparsity, we wish to find linear relationships in asymptotically growing dimension d. Also, RANSAC-like algorithms, as in robust statistics (or linear mixed models), do not aim to provide a description of the data for which they find a linear relationship.

Finally, we note that our work has a connection to the list-learning model introduced independently (subsequent to the original posting of this work on arXiv) by Charikar et al. [6]. The connection is more technical, and we postpone discussing it to Section 5.

2 Problem definition and background

In this work, we primarily focus on the following task:

▶ **Definition 2** (Conditional linear regression)**.** The *conditional (realizable) linear regression* task is the following. We are given access to examples from an arbitrary distribution D over $\{0,1\}^n \times \mathbb{R}^d \times \mathbb{R}$ for which there exists a k-DNF c^* and $\vec{a}^* \in \mathbb{R}^d$ such that

1. $\Pr_{(x,y,z)\in D}[|\langle \vec{a}^*, \vec{y}\rangle - z| \leq \epsilon | c^*(\vec{x}) = 1] = 1$ and
2. $\Pr_{(x,y,z)\in D}[c^*(\vec{x}) = 1] \geq \mu$,

for some $\epsilon, \mu \in (0,1]$. Then with probability $1-\delta$, given ϵ, μ, δ, and γ as input, we are to find some $\vec{a}' \in \mathbb{R}^d$ and k-DNF c' such that

1. $\Pr_{(x,y,z)\in D}[|\langle \vec{a}', \vec{y}\rangle - z| \leq \epsilon | c'(\vec{x}) = 1] \geq 1-\gamma$ and
2. $\Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega\left(\left((1-\gamma)\frac{\mu}{nd}\right)^k\right)$ for some k

in time polynomial in $n, d, 1/\mu, 1/\epsilon, 1/\gamma$, and $1/\delta$. If $\vec{a}^*$ is assumed to have at most s nonzero entries and $\vec{a}'$ is likewise required to have at most s nonzero entries, then this is the *conditional sparse linear regression* task with *sparsity* s.

We will also briefly consider the following variant that in some contexts may be more natural.

▶ **Definition 3** (Conditional ℓ_2-linear regression). The *conditional ℓ_2-linear regression* task is the following. We are given access to examples from an arbitrary distribution D over $\{0,1\}^n \times \{\vec{y} \in \mathbb{R}^d : \|\vec{y}\|_2 \leq B\} \times [-B, B]$ for which there exists a k-DNF c^* and $\vec{a}^* \in \mathbb{R}^d$ with $\|\vec{a}^*\|_2 \leq B$ such that

1. $\mathbb{E}_{(x,y,z)\in D}\left[(\langle \vec{a}^*, \vec{y}\rangle - z)^2 | c^*(\vec{x}) = 1\right] \leq \epsilon$ and
2. $\Pr_{(x,y,z)\in D}[c^*(\vec{x}) = 1] \geq \mu$,

for some $B \in \mathbb{R}^+$, $\epsilon, \mu \in (0,1]$. Then with probability $1-\delta$, given B, ϵ, μ, δ, and γ as input, we are to find some $\vec{a}' \in \mathbb{R}^d$ and k-DNF c' such that

1. $\mathbb{E}_{(x,y,z)\in D}\left[(\langle \vec{a}', \vec{y}\rangle - z)^2 | c'(\vec{x}) = 1\right] \leq \text{poly}(B, d, n)\epsilon$ and
2. $\Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega\left(\left((1-\gamma)\frac{\mu}{Bdn}\right)^k\right)$ for some k

in time polynomial in $n, d, B, 1/\mu, 1/\epsilon, 1/\gamma$, and $1/\delta$.

One could further consider, for example, regression under the other ℓ_p norms, but we will not pursue this here.

In both variants of the problem, we have sought to only recover a condition c' that only comprises a polynomial fraction of the probability of the optimal condition μ. A controllable $1-\gamma$ factor loss is generally necessary when we are choosing among various candidate "clusters" based on sampling. Although in the earlier work on conditional distribution search [15] (see Definition 4, next), it was possible to find an event c' that actually captured the same probability mass as the target c (since there we are uniquely seeking a "cluster" that selects positive points), even in that setting, the reductions between related models generally incurred a controllable $1-\gamma$ loss. Nevertheless, the value of such a formal definition is usually in enabling us to formulate negative results, and in that case we seek the most liberal definition possible. Hence, here, we allow c' to only contain a polynomial fraction of μ depending on the main parameters, B, d, and n, that we might expect to encounter in an approximation guarantee, such as the one we show for Algorithm 2.

The restriction of c to be a k-DNF is not arbitrary. Although we could consider other classes of representations for c, it seems that essentially any of the other standard hypothesis classes that we might naturally consider here will lead to an intractable problem, even under the relatively liberal version of the problems defined above. This will follow since we can reduce the simpler problem of finding such conditions to our problem:

▶ **Definition 4** (Conditional distribution search). For a *representation class* $\mathcal{C}$ of $c : \{0,1\}^n \to \{0,1\}$, the *conditional distribution search problem* is as follows. Given access to i.i.d. examples $(\vec{x}^{(1)}, b^{(1)}), \ldots, (\vec{x}^{(m)}, b^{(m)})$ from an arbitrary distribution D over $\{0,1\}^n \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ such that $\Pr_{(x,b)\in D}[b = 1 | c^*(\vec{x}) = 1] = 1$ and $\Pr_{(x,b)\in D}[c^*(\vec{x}) = 1] \geq \mu$, with probability $1-\delta$, find some circuit c' such that

1. $\Pr_{(x,b)\in D}[b = 1 | c'(\vec{x}) = 1] \geq 1-\gamma$ and
2. $\Pr_{(x,b)\in D}[c'(\vec{x}) = 1] \geq \Omega(((1-\gamma)\mu/n)^k)$ for some k

in time polynomial in $n, 1/\mu, 1/\gamma$, and $1/\delta$.

▶ **Theorem 5** (Conditional distribution search reduces to conditional linear regression). *Suppose there is an algorithm that given access to examples from an arbitrary distribution D' over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ and $a^* \in \mathbb{R}$ such that*

$$\Pr_{(x,y,z)\in D'}\left[|a^* y - z| \leq \epsilon \middle| c^*(\vec{x}) = 1\right] = 1 \text{ and } \Pr_{(x,y,z)\in D'}[c^*(\vec{x}) = 1] \geq \mu,$$

with probability $1-\delta$, finds some $a' \in \mathbb{R}$ and circuit c' such that

$$\Pr_{(x,y,z)\in D'}\left[|a'y-z| \leq \epsilon \middle| c'(\vec{x})=1\right] \geq 1-\gamma \qquad \textit{and}$$

$$\Pr_{(x,y,z)\in D'}[c'(\vec{x})=1] \geq \Omega\left(\left(\frac{(1-\gamma)\mu}{n}\right)^k\right) \qquad \textit{for some } k$$

in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ and $1/\delta$. Then there is a randomized polynomial-time algorithm for conditional distribution search for $\mathcal{C}$.

Proof. Let D be a distribution satisfying the hypotheses of the conditional distribution search task for $\mathcal{C}$, that is, for some $c^* \in \mathcal{C}$,
1. $\Pr_{(x,b)\in D}[b=1|c^*(\vec{x})=1]=1$ and
2. $\Pr_{(x,b)\in D}[c^*(\vec{x})=1] \geq \mu$.

Let D' be the distribution over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ sampled as follows: given an example $(\vec{x}, b)$ from D, if $b=1$ we produce $(\vec{x},1,0)$ and otherwise we produce $(\vec{x},1,b')$ for b' uniformly distributed over $\{0,1\}$. Notice that for c^* and $a^*=0$, then whenever $c^*(\vec{x})=1$, $|a^*y-z|=0 \leq 1/3$ over the entire support of the distribution; and, by assumption, $\Pr_{(x,y,z)\in D'}[c^*(\vec{x})=1] = \Pr_{(x,b)\in D}[c^*(\vec{x})=1] \geq \mu$. So, the pair $a^*=0$ and c^* certainly satisfy the conditions for our task for $\epsilon=1/3$. Therefore, by hypothesis, an algorithm for our task given access to D' with $\epsilon=1/3$ and $\gamma'=\gamma/2$ must return a' and a circuit c' such that
1. $\Pr_{(x,y,z)\in D'}\left[|a'y-z| \leq 1/3|c'(\vec{x})=1\right] \geq 1-\gamma'$ and
2. $\Pr_{(x,y,z)\in D'}[c'(\vec{x})=1] \geq \Omega(((1-\gamma')\mu/n)^k)$ for some k.

But now, since the distribution we used is uniform over examples with $z=0$ and $z=1$ whenever $b=0$ (and $y\equiv 1$), it must be that whatever a' is returned, $|a'-z|>1/3$ with probability $1/2$ conditioned on $b=0$ in the underlying draw from D. We must therefore actually have that

$$\frac{1}{2}\Pr_{(x,b)\in D}[b=0|c'(\vec{x})=1] \leq \Pr_{(x,y,z)\in D'}\left[|a'y-z|>1/3\middle|c'(\vec{x})=1\right] \leq \frac{\gamma}{2}$$

so indeed, also $\Pr_{(x,b)\in D}[b=1|c'(\vec{x})=1] \geq 1-\gamma$. Thus c' is as needed for a solution to the conditional distribution search problem. Since it is trivial to implement the sampling oracle for D' given a sampling oracle for D, we obtain the desired algorithm. ◀

In turn now, algorithms for finding such conditions would yield algorithms for PAC-learning DNF [15], which is currently suspected to be intractable (c.f. in particular work by Daniely and Shalev-Shwartz [8] for some strong consequences of learning DNF).

▶ **Theorem 6** (Theorem 5 of [15]). *If there exists an algorithm for the conditional distribution search problem for conjunctions, then DNF is PAC-learnable in polynomial time.*

Informally, therefore, an algorithm for conditional realizable linear regression for conjunctions, or any class that can *express* conjunctions (instead of k-DNF), even under the relatively lax version of the problem formulated here, would yield a randomized polynomial time algorithm for PAC-learning DNF. This seems to rule out, in particular, the possibility of developing algorithms to perform regression under conditions described by halfspaces, decision trees, and so on.

For conditional ℓ_2-linear regression, a stronger conclusion holds: such algorithms would solve the *agnostic* variant of the conditional distribution search task, with a similar error bound:

▶ **Theorem 7** (Agnostic condition search reduces to conditional ℓ_2-linear regression)**.** *Suppose there is an algorithm that given access to examples from an arbitrary distribution D' over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ and $a^* \in [0,1]$ such that $\mathbb{E}_{(x,y,z)\in D'}\left[(a^*y-z)^2|c^*(\vec{x})=1\right] \leq \epsilon$ and $\Pr_{(x,y,z)\in D'}[c^*(\vec{x})=1] \geq \mu$, with probability $1-\delta$, finds some a' and circuit c' such that*

1. $\mathbb{E}_{(x,y,z)\in D'}\left[(a'y-z)^2|c'(\vec{x})=1\right] \leq p(n)\epsilon$ *for some polynomial p and*
2. $\Pr_{(x,y,z)\in D'}[c'(\vec{x})=1] \geq \Omega(((1-\gamma)\mu/n)^k)$ *for some k*

in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ and $1/\delta$. Then there is a randomized polynomial-time algorithm for agnostic conditional distribution search for $\mathcal{C}$: that is, if there exists $c \in \mathcal{C}$ achieving

1. $\Pr_{(x,b)\in D}[b=1|c(\vec{x})=1] \geq 1-\epsilon$ *and*
2. $\Pr_{(x,b)\in D}[c(\vec{x})=1] \geq \mu$

then the algorithm finds a circuit c'' achieving

1. $\Pr_{(x,b)\in D}[b=1|c''(\vec{x})=1] \geq 1-2p(n)\epsilon$ *and*
2. $\Pr_{(x,b)\in D}[c''(\vec{x})=1] \geq \Omega(((1-\gamma)\mu/n)^k)$ *for some k*

in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ and $1/\delta$.

Proof. For a given distribution D over (x,b) satisfying the promise for conditional distribution search, we use the same construction of D' and reduction as in the proof of Theorem 5. Here, we note that for $a^*=0$, given that $\Pr_{(x,b)\in D}[b=1|c(\vec{x})=1] \geq 1-\epsilon$ for the c assumed to exist for conditional distribution search

$$\mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[(0\cdot 1-z)^2|c(\vec{x})=1\right] \leq \frac{1}{2}\epsilon.$$

Therefore, an algorithm for conditional ℓ_2-linear regression must find some a' and circuit c' such that $\Pr_{(x,y,z)\in D'}[c'(\vec{x})=1] \geq \Omega(((1-\gamma)\mu/n)^k)$ for some k and

$$\mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[((a'-z)^2|c'(\vec{x})=1\right] \leq \frac{1}{2}p(n)\epsilon.$$

Now, again, since D' gives $z=0$ and $z=1$ equal probability whenever $b=0$, we note that for such examples the expected value of $(a'-z)^2$ is minimized by $a'=1/2$, where it achieves expected value $1/4$. Thus as $(a'-z)^2$ is surely nonnegative,

$$\frac{1}{4}\Pr_{(x,b)\in D}[b=0|c'(\vec{x})=1] \leq \mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[(a'-z)^2|c'(\vec{x})=1\right] \leq \frac{1}{2}p(n)\epsilon$$

so c' indeed also achieves $\Pr_{(x,b)\in D}[b=1|c'(\vec{x})=1] \geq 1-2p(n)\epsilon$. ◀

The restriction to constant sparsity is also key, as our problem contains as a special case (when $\mu=1$, that is, when the trivial condition that takes the entire population can be used) the standard sparse linear regression problem. Sparse linear regression for *constant* sparsity is easy, but when the sparsity is allowed to be large, the problem quickly becomes intractable: In general, finding sparse solutions to linear equations is known to be NP-hard [20], and Zhang, Wainwright, and Jordan [30] extend this to bounds on the quality of sparse linear regression that is achievable by polynomial-time algorithms, given that NP does not have polynomial-size circuits.

3 Algorithms for conditional sparse linear regression

We now turn to stating and proving our main theorem. In what follows, we use the following (standard) notation: $\Pi_{d_1,\ldots,d_s}$ denotes the projection (of $\mathbb{R}^d$) to the s coordinates $d_1, d_2, \ldots, d_s$

```
Algorithm 1: Find-and-eliminate.
input   : Examples $(\vec{x}^{(1)}, \vec{y}^{(1)}, z^{(1)}), \ldots, (\vec{x}^{(m)}, \vec{y}^{(m)}, z^{(m)})$, target fit $\epsilon$ and
          fraction $(1-\gamma/2)\mu$.
output  : A $k$-DNF over $x_1, \ldots, x_n$ and linear predictor over $y_1, \ldots, y_d$, or
          INFEASIBLE if none exist.
begin
    forall $(d_1, \ldots, d_s) \in \binom{[d]}{s}$, $(\sigma_1, \ldots, \sigma_{s+1}) \in \{\pm 1\}^{s+1}$ and $(j_1, \ldots, j_{s+1}) \in \binom{[m]}{s+1}$
     do
        Initialize $c$ to be the (trivial) $k$-DNF over all $\binom{2n}{k}$ terms of size $k$.
        Let $(\vec{a}, \epsilon')$ be a solution to the following linear system:

            $\langle \vec{a}, \Pi_{d_1,\ldots,d_s} \vec{y}^{(j_\ell)} \rangle - z^{(j_\ell)} = \sigma_\ell \epsilon'$ for $\ell = 1, \ldots, s+1$

         if $\epsilon' > \epsilon$ then continue to the next iteration.
        for $j = 1, \ldots, m$ do  if $|\langle \vec{a}, \Pi_{d_1,\ldots,d_s} \vec{y}^{(j)} \rangle - z^{(j)}| > \epsilon$ then
            forall $T \in c$ do  if $T(\vec{x}^{(j)}) = 1$ then  Remove $T$ from $c$.

        end

        if $\#\{j : c(\vec{x}^{(j)}) = 1\} > (1-\gamma/2)\mu m$ then  return $\vec{a}$ and $c$.
    end
    return INFEASIBLE.
end
```

from $[d]$ (which denotes the integers $1, \ldots d$). For a set S, we let $\binom{S}{k}$ denote the subsets of S of size exactly k.

At a high level, the algorithm (Algorithm 1, below) generates a *list* of possible coefficient vectors for the regression fit. For each such candidate, it generates labels for the points indicating whether or not the candidate linear fit achieves small error under that fit or not. It then solves the conditional distribution search problem given by these labels (by using the Elimination algorithm [15]), and estimates the fraction of the data captured this way. It returns the first linear fit that captures a sufficiently large fraction (or "INFEASIBLE" if none do).

▶ **Theorem 8** (Realizable sparse regression – full statement of Theorem 1)**.** *Suppose that D is a joint probability distribution over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$, and $z \in \mathbb{R}$ such that there is a k-DNF c for which for some s-sparse $\vec{a} \in \mathbb{R}^d$*

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}, \vec{y} \rangle - z| \leq \epsilon \middle| c(\vec{x}) = 1\right] = 1 \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu.$$

Then given ϵ, μ, and δ in $(0,1)$ and $\gamma \in (0, 1/2]$ and

$$m = O\left(\frac{1}{\mu\gamma}\left(s \log s + s \log d + n^k + \log \frac{1}{\delta}\right)\right)$$

examples from D, for any constants s and k, Algorithm 1 runs in polynomial time in n, d, and m $(= \mathrm{poly}(n, d, 1/\mu, 1/\gamma, \log 1/\delta))$ and finds an s-sparse $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y} \rangle - z| \leq \epsilon \middle| c'(\vec{x}) = 1\right] \geq 1-\gamma \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq (1-\gamma)\mu.$$

Proof. It is clear that the algorithm runs for $O(d^s m^{s+1})$ iterations, where each iteration (for constant s) runs in time polynomial in the bit length of our examples and $O(mn^k)$. Thus, for constant s and k, the algorithm runs in polynomial time overall, and it only remains to argue correctness.

We will first argue that the algorithm succeeds at returning some solution with probability $1-\delta/3$ over the examples. We will then argue that any solution returned by the algorithm is satisfactory with probability $1-2\delta/3$ over the examples, thus leading to a correct solution with probability $1-\delta$ overall.

Completeness part 1: Generating the linear rule

We first note that for $m \geq \frac{6}{\mu\gamma}\ln\frac{3}{\delta}$ examples, a Chernoff bound guarantees that with probability $1-\delta/3$, there are at least $(1-\gamma/2)\mu m$ examples satisfying the unknown condition c in the sample. Let S be the set of examples satisfying c. Given the set of s dimensions that are used in the sparse linear rule, we set up a linear program in $s+1$ dimensions to minimize ϵ' subject to the constraints

$$-\epsilon' \leq \langle \vec{a}, \vec{y}^{(j)}\rangle - z^{(j)} \leq \epsilon' \text{ for } j \in S.$$

It is well known (see, for example, Schrijver [24, Chapter 8]) that the optimum value for any feasible linear program may be obtained at a *basic feasible solution,* i.e., a vertex of the polytope, given by satisfying $s+1$ of the constraints with equality. Since each constraint corresponds to an example and sign (for the lower or upper inequality), this means that we can obtain $\vec{a}$ by solving for $\vec{a}$ and ϵ' in the following linear system

$$\langle \vec{a}, \vec{y}^{(j_\ell)}\rangle - z^{(j_\ell)} = \sigma_\ell \epsilon' \text{ for } \ell = 1, \ldots, s+1$$

for some set of $s+1$ examples, $j_1, \ldots, j_{s+1}$ and $s+1$ signs $\sigma_1, \ldots, \sigma_{s+1}$ corresponding to the tight constraints. Thus, when the algorithm uses the appropriate set of s dimensions, the appropriate $s+1$ examples, and the appropriate $s+1$ signs, we will recover an $\vec{a}^*$ and ϵ^* such that for all $j \in S$, $|\langle \vec{a}^*, \vec{y}^{(j)}\rangle - z^{(j)}| \leq \epsilon^* \leq \epsilon$.

Completeness part 2: Recovering a suitable condition given a rule

Now, given $\vec{a}^*$ such that for all $j \in S$, $|\langle \vec{a}^*, \vec{y}^{(j)}\rangle - z^{(j)}| \leq \epsilon$, we observe that the algorithm identifies a k-DNF h^* such that $h^*(\vec{x}^{(j)}) = 1$ for all $j \in S$. Indeed, the algorithm only eliminates a k-term T for examples j such that $|\langle \vec{a}^*, \vec{y}^{(j)}\rangle - z^{(j)}| > \epsilon$. Thus, it never eliminates any term appearing in c, and so in particular, $\Pr_{(x,y,z)\in D}[h^*(\vec{x}) = 1] \geq \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu$. Moreover, since (as noted above, with probability $1-\delta/3$) there are at least $(1-\gamma/2)\mu m$ examples satisfying c in the sample, there are at least $(1-\gamma/2)\mu m$ examples satisfying h^*. Thus, with probability $1-\delta/2$, when the algorithm considers the relevant s dimensions in the support of $\vec{a}$ and considers an appropriate choice of $s+1$ examples to obtain a suitable $\vec{a}^*$, it will furthermore obtain an h^* that will lead the algorithm to terminate and return $\vec{a}^*$ and h^*.

Soundness: Generalization bounds

Next, we argue that any $\vec{a}'$ and h' returned by the algorithm will suffice with probability $1-2\delta/3$ over the examples.

We will use the facts that

1. a union of k hypothesis classes of VC-dimension d has VC-dimension at most $O(d \log d + \log k)$ (for example, see [25, Exercise 6.11]),
2. linear threshold functions in $\mathbb{R}^s$ have VC-dimension $s+1$ (e.g., [25, Section 9.1.3]), and
3. the composition of classes of VC-dimension d_1 and d_2 has VC-dimension at most $d_1 + d_2$ (follows from [25, Exercise 20.4]).

We now consider the class of disjunctions of a k-CNF over $\{0,1\}^n$ and (intersections of) linear threshold functions $[|\langle(\vec{a}, -1), (\vec{y}, z)\rangle| \leq \varepsilon]$ for an s-sparse $\vec{a}$ over $\mathbb{R}^d$. By writing this latter class as a union over the $2^{\binom{n}{k}}$ k-CNFs and $\binom{d}{s}$ coordinate subsets of size s, we find that it has VC-dimension $O(s \log s + s \log(d/s) + n^k) = O(\log d + n^k)$ for constant s.[1]

An optimal bound for sample complexity in terms of VC-dimension was recently obtained by Hanneke [11] (superseding the earlier bounds, e.g., by Vapnik [28] and Blumer et al. [4], although these would suffice for us, too): in this case, given

$$m = O\left(\frac{1}{\mu\gamma}\left(s \log s + s \log d + n^k + \log \frac{1}{\delta}\right)\right)$$

examples, if $[|\langle(\vec{a}', -1), (\vec{y}, z)\rangle| \leq \varepsilon] \vee \neg h'(\vec{x})$ is consistent with all of the examples, then with probability $1 - \delta/3$ over the examples,

$$\Pr_{(x,y,z)\in D}[(|\langle(\vec{a}', -1), (\vec{y}, z)\rangle| \leq \varepsilon) \vee \neg h'(\vec{x})] \geq 1 - \mu\gamma/2$$

or, equivalently,

$$\Pr_{(x,y,z)\in D}[(|\langle(\vec{a}', -1), (\vec{y}, z)\rangle| > \varepsilon) \wedge h'(\vec{x})] \leq \mu\gamma/2.$$

Now, since for $m \geq \frac{4}{\mu\gamma} \ln \frac{3}{\delta}$, with probability $1 - \delta/3$,

$$\Pr_{(x,y,z)\in D}[h'(\vec{x})] \geq \frac{1-\gamma/2}{1+\gamma/2}\mu \geq (1-\gamma)\mu,$$

we find that for our choice of $\vec{a}'$ and h',

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y}\rangle - z| > \varepsilon \middle| h'(\vec{x})\right] \leq \frac{\gamma}{2}\frac{1+\gamma/2}{1-\gamma/2}$$

$$\text{and so,} \quad \Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y}\rangle - z| \leq \varepsilon \middle| h'(\vec{x})\right] \geq 1-\gamma \text{ since } \gamma \leq 1/2$$

as needed. ◀

Although it was not a focus of the analysis, we remark that the multiplicative Chernoff bound also guarantees that if *no* k-DNF event of probability greater than $(1-\gamma)\mu$ has a linear rule that is γ-close to having ϵ sup norm, then the algorithm is guaranteed to output INFEASIBLE with probability $1 - \delta$: the k-DNFs of probability less than $(1-\gamma)\mu$ fail the final test, and for the rest, the standard VC-dimension sample complexity analysis guarantees that we catch a point with error greater than the sup norm bound of ϵ in the sample (and so rule out the k-DNF during the Elimination algorithm). It follows that the algorithm is easily modified to solve a few natural variants of our problem. Suppose we return the list of all

[1] Exercises in Anthony and Biggs [2, Chapter 8, Exercise 6] and Mohri et al. [19, Exercise 3.15] on the growth function for ANDs/ORs of two distinct concept classes also yield this easily.

Algorithm 2: Dense Expected-error Regression Pigeonhole (DERP)

```
input   : Examples $(\vec{x}^{(1)}, \vec{y}^{(1)}, z^{(1)}), \dots, (\vec{x}^{(m)}, \vec{y}^{(m)}, z^{(m)})$, target fit $\epsilon$.
output  : A $k$-DNF over $x_1, \dots, x_n$ and linear predictor over $y_1, \dots, y_d$.
begin
    Initialize $c = \bot$, $\mu^* = 0$.
    forall Terms $T$ of size $k$ over $x_1, \dots, x_n$ do
        Put $S(T) = \{j : T(\vec{x}^{(j)}) = 1\}$.
        Let $\vec{a}$ minimize the squared-error on $(\vec{y}^{(j)}, z^{(j)})$ over $j \in S(T)$ subject to
         $\|\vec{a}\|_2 \leq B$.
        if $\frac{1}{m}\sum_{j \in S(T)}(\langle \vec{a}, \vec{y}^{(j)}\rangle - z^{(j)})^2 \leq 4\mu\epsilon$ and $|S(T)| \geq \mu^* m$ then
            Put $c = T$ and $\mu^* = |S(T)|/m$.
        end
    end
    return $c$ and $\vec{a}$
end
```

coefficient vectors and k-DNFs that would pass our termination condition. We then obtain a list that contains all events that have probability μ (and only those that have probability at least $(1-\gamma)\mu$) for which the conditional distribution is γ-close to one where the linear rule has ϵ sup norm. Or, suppose we return the pair for which the k-DNF empirically satisfies the most examples. We then return a k-DNF that is within a $1-\gamma$ factor of having the largest probability (provided that this is at least μ) among those with a suitable linear rule.

4 Towards conditional dense, expected-error linear regression

While sparsity is a highly desirable feature to have of a linear regression fit, it may be the case that solutions are often not so sparse that Algorithm 1 is truly efficient. Moreover, we may also wish for an algorithm that handles an *expected error* variant of the regression task; the sup norm is particularly sensitive to noise or outliers, and thus is usually not a particularly desirable norm to use on real data. Our technique certainly does not address either of these concerns. The simple Algorithm 2 illustrates the best technique we currently have for either dense regression or expected error regression.

▶ **Theorem 9.** *Algorithm 2 solves the conditional ℓ_2-linear regression task: given access to a joint distribution D over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$ with $\|\vec{y}\|_2 \leq B$, and $z \in [-B, B]$ such that there is a k-DNF c and $\vec{a} \in \mathbb{R}^d$ with $\|\vec{a}\|_2 \leq B$ such that*

$$\mathop{\mathbb{E}}_{(x,y,z)\in D}\left[(\langle \vec{a}, \vec{y}\rangle - z)^2 \middle| c(\vec{x}) = 1\right] \leq \epsilon \qquad \text{and} \qquad 2\mu \geq \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu$$

and given B, k, ϵ, μ, and $\delta \in (0,1)$, using

$$m = O\left(\frac{B^8 n^k}{\mu\epsilon}\left(k \log n + \log\frac{1}{\delta}\right)\right)$$

examples from D, for any constant k, Algorithm 2 runs in polynomial time and finds a $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\mathop{\mathbb{E}}_{(x,y,z)\in D}\left[(\langle \vec{a}', \vec{y}\rangle - z)^2 \middle| c'(\vec{x}) = 1\right] \leq O(n^k\epsilon) \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega(\mu/n^k).$$

Note that we can find such an estimate for μ by binary search.

Proof. We first observe that in particular, since for any T the objective function

$$\sum_{j \in S(T)} (\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)})^2$$

is convex, as is the set of $\vec{a}$ of ℓ_2-norm at most B, the main step of the algorithm is a convex optimization problem that can be solved in polynomial time, for example by gradient descent (see, e.g., [25, Chapter 14]). Thus, the algorithm can be implemented in polynomial time as claimed.

We next turn to correctness. Let c^* be the k-DNF promised by the theorem statement. By the pigeonhole principle, there must be some term T^* of c^* such that $\Pr[T^*(\vec{x}) = 1] \geq \mu / \binom{2n}{k}$. Observe that for the rule $\vec{a}^*$ promised to exist,

$$\begin{aligned}\mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \middle| T^*(\vec{x}) = 1\right] \Pr_D[T^*(\vec{x}) = 1] &\leq \mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \middle| c^*(\vec{x}) = 1\right] \Pr_D[c^*(\vec{x}) = 1] \\ &\leq \epsilon \cdot 2\mu.\end{aligned}$$

For a suitable choice of leading constant in the number of examples, a (multiplicative) Chernoff bound yields that with probability $1 - \delta/4$, at least $m \Pr[T^*(\vec{x}) = 1]/2$ examples satisfy T^* and noting that $(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \in [0, 2B^4]$, with probability $1 - \delta/4$,

$$\frac{1}{m} \sum_{j=1}^{m} (\langle \vec{a}^*, \vec{y} \rangle - z)^2 T^*(\vec{x}) \leq 4\mu\epsilon$$

Thus, the $\vec{a}'$ minimizing the squared error on the set of examples also achieves

$$\frac{1}{m} \sum_{j : T^*(\vec{x}^{(j)}) = 1} (\langle \vec{a}', \vec{y}^{(j)} \rangle - z^{(j)})^2 \leq 4\mu\epsilon$$

as needed, so with probability $1 - \delta/2$, at least T^* is considered for c and the algorithm produces some c and $\vec{a}$ as output.

To see that any such T and $\vec{a}$ is satisfactory, we first note that any T we produce as output must satisfy at least as many examples as T^* by construction, so T must satisfy at least

$$\Pr_D[T^*(\vec{x}) = 1] m/2 \geq \Omega\left(\frac{B^8}{\epsilon}\left(k \log n + \log \frac{1}{\delta}\right)\right)$$

examples. In particular, this is at least $\mu m / 2\binom{2n}{k}$ examples, and a Chernoff bound guarantees that for suitable constants, with probability $1 - \delta/4\binom{2n}{k}$, no T with $\Pr_D[T(x) = 1] < \mu/4\binom{2n}{k}$ satisfies so many examples. Next, simply note that if for the best a for T with $\|\vec{a}\|_2 \leq B$, $\mathbb{E}_D\left[(\langle \vec{a}, \vec{y} \rangle - z)^2 | T(\vec{x}) = 1\right] \Pr[T(x) = 1] > 8\mu\epsilon$, then since $\|\vec{y}\|_2 \leq B$, $z^2 \leq B^2$, and the loss function is B-Lipschitz on this domain, a Rademacher bound (see, for example, [25, Theorem 26.12]) guarantees that with probability $1 - \delta/4\binom{2n}{k}$, for any such $\vec{a}$,

$$\frac{1}{m} \sum_{j : T(\vec{x}^{(j)}) = 1} (\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)})^2 > 4\mu\epsilon$$

and T will not be considered. A union bound over both events for all such T establishes that any T that is returned has, with probability $1 - \delta/2$, both

$$\Pr_D[T(\vec{x}) = 1] \geq \frac{\mu}{4\binom{2n}{k}} \text{ and } \mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}, \vec{y} \rangle - z)^2 \middle| T(\vec{x}) = 1\right] \Pr_D[T(\vec{x}) = 1] \leq 8\mu\epsilon$$

and thus is as needed. Therefore, overall, with probability $1 - \delta$, the algorithm considers at least T^* as a candidate to output, and outputs a suitable term T and vector $\vec{a}$. ◀

5 Discussion and future directions

The main defect of Algorithm 2 is that in general it only recovers a condition with a $\Omega(1/n^k)$-fraction of the possible probability mass of the best k-DNF condition. This is in stark contrast to both Algorithm 1 and all of the earlier positive results for condition identification [15], in which we find a condition with probability at least a $(1-\gamma)$-fraction of that of the best condition, for any γ we choose. Indeed, we are most interested in the case where the probability of this event is relatively small and thus a $1/n^k$-fraction is extremely small. The main challenge here is to develop an algorithm for the dense and/or expected-error regression problem that similarly identifies a condition with probability that is a $(1-\gamma)$-fraction of that of the best condition.

Of course, the $O(n^k)$ blow-up in the expected error is also undesirable, but as indicated by Theorem 7, this is the same difficulty encountered in *agnostic learning*. Naturally, minimizing the amount by which constraints are violated is generally a harder problem than finding a solution to a system of constraints, and this is reflected in the quality of results that have been obtained. The results for such agnostic condition identification of k-DNFs in the previous work by Juba [15] suffers a similar blow-up in the error, which was recently improved to $\tilde{O}(\sqrt{n^k})$ by Zhang et al. [29]. The state-of-the-art algorithms for agnostic supervised learning for disjunctive classifiers by Awasthi et al. [3] suffer a similar blow-up of a $n^{k/3+o(1)}$-factor, and yet even for the harder problem of agnostic learning of linear threshold functions, only a sub-polynomial approximation factor is known to be necessary [7]. The question of what approximation factor is necessary is similarly wide open for our problem. We note briefly that a variant of Algorithm 2 in which we seek $\vec{a}$ satisfying $|\langle\vec{a},\vec{y}^{(j)}\rangle - z^{(j)}| \leq \epsilon$ for all j satisfying a candidate term T solves the sup norm variant of Definition 2 for dense regression, and *does not* suffer this increase of the error. Of course, as mentioned earlier, one typically wishes to solve ℓ_1 or ℓ_2-norm regression, as these are much better behaved.

It is also natural to ask if instead of constant sparsity (as used here), we could simply bound the ℓ_1-norm of the coefficient vector, as in LASSO [27]. It's well known that this tends to produce sparse solutions (with ℓ_2-regression), without necessarily demanding an a priori fixed constant bound on the sparsity. Again, our technique does not achieve this.

Looking towards developing a better algorithm and solving further, related tasks, we note that a common strategy seems to be emerging. Our first algorithm, for the sup norm (Algorithm 1) operated by first generating a list of possible coefficient vectors for the regression fit, and then learning a condition that captures the various candidates in the list. This strategy is similar to the list-learning model independently introduced by Charikar et al. [6] for solving a variety of statistical problems when seeking to capture only a minority fraction of the data. Of course, in our work we ultimately sought to find a condition to single out one member of the list, rather than producing the entire list as output. Also, at a technical level, while their work applies to a much, much broader variety of problems, their technique also suffers an increase in the losses that grows with $1/\sqrt{\mu}$. (The relatively trivial Algorithm 2, as discussed above, suffers a n^k-factor increase, but has no dependence on μ, which naturally may be more or less desirable depending on the setting.) These technical differences aside, we believe that both of these works suggest that a relatively broad family of problems involving statistics of minority sub-populations may be tackled by variants of the list-learning approach. Indeed, we observe that the problem is essentially similar to that tacked by list-decoding in coding theory (e.g., see Sudan [26] for an overview): although the amount of agreement with the data may be too small to uniquely determine a "best" hypothesis, it may be possible to output a small list containing all possible hypotheses. Although most of the work in coding theory is focused on finite characteristic (in contrast to most problems we would seek to solve in data analysis), it may be informative.

One immediate family of questions to be addressed is, which *conditional* variants of the standard supervised learning tasks can be solved efficiently? In particular, when can such tasks solved without suffering an increase in the loss that depends polynomially on $1/\mu$? For example, for which families of Boolean classification tasks do such algorithms exist? We know that the conditions (essentially) must be described by k-DNFs, but this seems to tell us nothing about which rules we can fit on such conditional distributions.

Acknowledgements. I thank Madhu Sudan for originally suggesting the joint problem of learning under conditional distributions. I also thank Ben Moseley for many helpful discussions about these problems. Finally, I thank the reviewers for their comments and suggestions.

References

1 Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. MIT Press, Cambridge, MA, 1996.

2 Martin Anthony and Norman Biggs. *Computational Learning Theory.* Number 30 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York, NY, 1992.

3 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Improved guarantees for agnostic learning of disjunctions. In *Proc. 23rd COLT*, pages 359–367, 2010.

4 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989. doi:10.1145/76359.76371.

5 Nader H. Bshouty and Lynn Burroughs. Maximizing agreements with one-sided error with applications to heuristic learning. *Machine Learning*, 59(1–2):99–123, 2005. doi:10.1007/s10994-005-0464-5.

6 Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning with untrusted data. arXiv:1611.02315, 2016.

7 Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proc. 48th STOC*, pages 105–117, 2016. doi:10.1145/2897518.2897520.

8 Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limtations on learning DNF's. In *Proc. 29th COLT*, volume 49 of *JMLR Workshops and Conference Proceedings*, pages 815–830, 2016.

9 Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi:10.1145/358669.358692.

10 Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999. doi:10.1023/A:1008894516817.

11 Steve Hanneke. The optimal sample complexity of PAC learning. *JMLR*, 17(38):1–15, 2016.

12 Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Proc. 26th COLT*, volume 30 of *JMLR Workshops and Conference Proceedings*, pages 354–375, 2013.

13 Peter J. Huber. *Robust Statistics.* John Wiley & Sons, New York, NY, 1981.

14 Jiming Jiang. *Linear and Generalized Linear Mixed Models and Their Applications.* Springer, Berlin, 2007.

15 Brendan Juba. Learning abductive reasoning using random examples. In *Proc. 30th AAAI*, pages 999–1007, 2016.

16 Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *JCSS*, 78:1481–1495, 2012. `doi:10.1016/j.jcss.2011.12.026`.

17 Varun Kanade and Justin Thaler. Distribution-independent reliable learning. In *Proc. 27th COLT*, volume 35 of *JMLR Workshops and Conference Proceedings*, pages 3–24, 2014.

18 Charles E. McCulloch and Shayle R. Searle. *Generalized, Linear, and Mixed Models.* John Wiley & Sons, New York, NY, 2001.

19 Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* MIT Press, Cambridge, MA, 2012.

20 B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995. `doi:10.1137/S0097539792240406`.

21 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988. `doi:10.1145/48014.63140`.

22 Avi Rosenfeld, David G. Graham, Rifat Hamoudi, Rommell Butawan, Victor Eneh, Saif Kahn, Haroon Miah, Mahesan Niranjan, and Laurence B. Lovat. MIAT: A novel attribute selection approach to better predict upper gastrointestinal cancer. In *Proc. IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7, 2015. `doi:10.1109/DSAA.2015.7344866`.

23 Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, New York, NY, 1987.

24 Alexander Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, 1986.

25 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, New York, NY, 2014.

26 Madhu Sudan. List decoding: Algorithms and applications. In J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, and T. Ito, editors, *IFIP International Converence on Theoretical Computer Science*, volume 1872 of *LNCS*, pages 25–41. Springer, 2000. `doi:10.1007/3-540-44929-9_3`.

27 Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc. B*, 58(1):267–288, 1996.

28 Vladimir Vapnik. *Estimation of Dependencies Based on Empirical Data.* Springer, New York, NY, 1982.

29 Mengxue Zhang, Tushar Mathew, and Brendan Juba. An improved algorithm for learning to perform exception-tolerant abduction. To appear in 31st AAAI, 2017.

30 Yuchen Zhang, Martin J. Wainwright, and Michael I. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *Proc. 27th COLT*, volume 35 of *JMLR Workshops and Conference Proceedings*, pages 921–948, 2014.