JMLR: Workshop and Conference Proceedings vol 40:1–14, 2015

Variable Selection is Hard

Dean P. Foster DEAN@FOSTER.NET
Microsoft Research New York City and University of Pennsylvania

Howard Karloff HOWARD@CC.GATECH.EDU

Justin Thaler JTHALER@FAS.HARVARD.EDU
Yahoo Labs New York

Abstract

Variable selection for sparse linear regression is the problem of finding, given an $m \times p$ matrix B and a target vector $\mathbf{y}$, a sparse vector $\mathbf{x}$ such that $B\mathbf{x}$ approximately equals $\mathbf{y}$. Assuming a standard complexity hypothesis, we show that no polynomial-time algorithm can find a k'-sparse $\mathbf{x}$ with $\|B\mathbf{x} - \mathbf{y}\|^2 \leq h(m,p)$, where $k' = k \cdot 2^{\log^{1-\delta} p}$ and $h(m,p) = p^{C_1} m^{1-C_2}$, where $\delta > 0, C_1 > 0, C_2 > 0$ are arbitrary. This is true even under the promise that there is an unknown k-sparse vector $\mathbf{x}^*$ satisfying $B\mathbf{x}^* = \mathbf{y}$. We prove a similar result for a statistical version of the problem in which the data are corrupted by noise.

To the authors' knowledge, these are the first hardness results for sparse regression that apply when the algorithm simultaneously has $k' > k$ and $h(m,p) > 0$.

1. Introduction

Consider a linear regression problem in which one is given an $m \times p$ matrix B and a target vector $\mathbf{y} \in \mathbb{R}^m$. The goal is to approximately represent $\mathbf{y}$ as a linear combination of as few columns of B as possible. If a polynomial-time algorithm $\mathcal{A}$ is presented with B and $\mathbf{y}$, and it is known that $\mathbf{y}$ is an *exact* linear combination of some k columns of B, and $\mathcal{A}$ is allowed to choose *more than* k columns of B, how many columns must $\mathcal{A}$ choose in order to generate a linear combination which is *close to* $\mathbf{y}$?

Note that we have allowed $\mathcal{A}$ to "cheat" both on the number of columns and on the accuracy of the resulting linear combination. In this paper, we show the problem is intractable despite the allowed cheating.

Formally, we define (g,h)-SPARSE REGRESSION as follows. Let $\mathbf{e}^{(m)}$ denote the m-dimensional vector of 1's, and for any vector $\mathbf{z}$, let $\|\mathbf{z}\|$ denote $\|\mathbf{z}\|_2$ and let $\|\mathbf{z}\|_0$ denote the number of nonzeros in $\mathbf{z}$. Let $g(\cdot)$ and $h(\cdot,\cdot)$ denote arbitrary functions. An algorithm for (g,h)-SPARSE REGRESSION satisfies the following.

- Given: An $m \times p$ Boolean matrix B and a positive integer k such that there is a real p-dimensional vector $\mathbf{x}^*$, $\|\mathbf{x}^*\|_0 \leq k$, such that $B\mathbf{x}^* = \mathbf{e}^{(m)}$. (Call such an input *valid*.)
- Goal: Output a (possibly random) p-dimensional vector $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq k \cdot g(p)$ such that $\|B\mathbf{x} - \mathbf{e}^{(m)}\|^2 \leq h(m,p)$ with high probability over the algorithm's internal randomness.

Since we are focused on hardness, restricting B to have entries in $\{0,1\}$ and the target vector to be $\mathbf{e}^{(m)}$ only makes the result stronger.

There is, of course, an obvious exponential-time deterministic algorithm for (g,h)-SPARSE REGRESSION, for $g(p) = 1$ for all p, and $h(m,p) = 0$: enumerate all subsets S of $\{1,2,...,p\}$ of size k. For each S, use Gaussian elimination to determine if there is an $\mathbf{x}$ with $\mathbf{x}_j = 0$ for all $j \notin S$ such that $B\mathbf{x} = \mathbf{e}^{(m)}$, and if there is one, to find it. Since $B\mathbf{x}^* = \mathbf{e}^{(m)}$ and $\|\mathbf{x}^*\|_0 \leq k$, for at least one S the algorithm must return an $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq k, B\mathbf{x} = \mathbf{e}^{(m)}$.

Before getting too technical, we warm up with a simple hardness proof for (g, h)-SPARSE REGRESSION. The short proof can be understood on its own. This theorem is just a warmup; later we will prove a much stronger result (relying, unfortunately, on a stronger complexity assumption). To the authors' knowledge, this simple proof was not known, though similar arguments had been used previously to show weaker hardness results for related problems (cf. (Arora et al., 1997, Proposition 6) and Çivril (2013)).

Theorem 1 *Let $0 < \delta < 1$. If there is a deterministic polynomial-time algorithm A for (g, h)-SPARSE REGRESSION, for which $g(p) = (1 - \delta) \ln p$ and $h(m, p) = m^{1-\delta}$, then* SAT $\in$ DTIME$(n^{O(\log \log n)})$.

Proof Feige (1998) gives a reduction from SAT, running in deterministic time $N^{O(\log \log N)}$ on SAT instances of size N, to SET COVER, in which the resulting, say, $m \times p$, incidence matrix B (whose rows are elements and columns are sets) has the following properties. There is a (known) k such that (1) if a formula $\phi \in$ SAT, then there is a collection of k *disjoint* sets which covers the universe (that is, there is a collection of k columns of B whose sum is $\mathbf{e}^{(m)}$), and (2) if $\phi \notin$ SAT, then no collection of at most $k \cdot [(1 - \delta) \ln p]$ sets covers the universe (in other words, no set of at most $k \cdot [(1 - \delta) \ln p]$ columns of B has a sum which is coordinate-wise at least $\mathbf{e}^{(m)}$). This is already enough to establish that any polynomial-time algorithm for $(g(p), 0)$-SPARSE REGRESSION implies an $N^{O(\log \log N)}$-time algorithm for SAT. The remainder of the proof is devoted to establishing the analogous statement for $(g(p), m^{1-\delta})$-SPARSE REGRESSION .

Build a matrix B' by stacking r copies of B atop one another, r to be determined later. Let $M = rm$. If $\phi \in$ SAT, then there is a collection of k columns summing to $\mathbf{e}^{(M)}$. This is a linear combination of sparsity at most k. If $\phi \notin$ SAT, then for any linear combination of at most $k \cdot [(1 - \delta) \ln p]$ column vectors, in each of the r copies of B, there is squared error of at least 1 (since the best one could hope for, in the absence of a set cover, is $m - 1$ 1's and one 0). This means that the squared error overall is at least r. We want $r > M^{1-\delta} = (rm)^{1-\delta}$, i.e., $r^{\delta} > m^{1-\delta}$, and hence we define $r = \lceil m^{1/\delta - 1} \rceil + 1$.

Construct an algorithm A' for SAT as follows. Run A on instance (B', k) of (g, h)-SPARSE REGRESSION for $T(N)$ time steps, where $T(N)$ is the time A would need on an $rm \times p$ matrix if (B', k) were a valid input for (g, h)-SPARSE REGRESSION (which it may not be); since in the valid case, A runs in time polynomial in the size of the input matrix B' (which is $N^{O(\log \log N)}$), $T(N)$ is also $N^{O(\log \log N)}$. If A outputs a vector $\mathbf{x}$ such that $\|B'\mathbf{x} - \mathbf{e}^{(M)}\|^2 \leq r$ and $\|\mathbf{x}\|_0 \leq k[(1 - \delta) \ln p]$ within this time bound, then A' outputs "Satisfiable." Otherwise (A doesn't terminate in the allotted time or it terminates and outputs an inappropriate vector), A' outputs "Unsatisfiable."

Clearly A' runs in time $N^{O(\log \log N)}$ on inputs ϕ of size N. It remains to show that A' is a correct algorithm for SAT.

To this end, suppose that $\phi \in$ SAT. In this case, there is a solution $\mathbf{x}^*$ of sparsity at most k with $B'\mathbf{x}^* = \mathbf{e}^{(M)}$, and since A is a correct algorithm for (g, h)-SPARSE REGRESSION, A will find such a solution, causing A' to output "Satisfiable" when run on ϕ. On the other hand, if $\phi \notin$ SAT, then there is no vector $\mathbf{x}^*$ with $\|\mathbf{x}^*\|_0 \leq k \cdot [(1 - \delta) \ln p]$ such that $\|B'\mathbf{x}^* - \mathbf{e}^{(M)}\|^2 \leq r$. Hence, A' must output "Unsatisfiable" when run on ϕ. We conclude that A' is a correct algorithm for SAT running in time $N^{O(\log \log N)}$ on instances of size N. ■

One can combine the PCP of Dinur and Steurer (2014) with Feige's construction, or the earlier construction of Lund and Yannakakis (1994), to strengthen the conclusion of Theorem 1 to SAT $\in$P.

Throughout, BPTIME(T) will denote the set of all languages decidable by randomized algorithms, with two-sided error, running in expected time T. Our main result is that unless $NP \subseteq$ BPTIME$(n^{\text{polylog}(n)})$, then even if $g(p)$ grows at a "nearly polynomial" rate, and $h(m, p) \leq p^{C_1} \cdot m^{1-C_2}$ for any positive constants C_1, C_2, there is no quasipolynomial-time (randomized) algorithm for (g, h)-SPARSE REGRESSION:

Theorem 2 *Assume that* NP $\not\subseteq$ BPTIME$(n^{polylog(n)})$. *For any positive constants* δ, C_1, C_2, *there exist a* $g(p)$ *in* $2^{\Omega(\lg^{1-\delta}(p))}$ *and an* $h(m,p)$ *in* $\Omega\left(p^{C_1} \cdot m^{1-C_2}\right)$ *such that there is no quasipolynomial-time randomized algorithm for* (g,h)-SPARSE REGRESSION.

Note that the "$-C_2$" cannot be removed from the condition $h(m,p) \leq p^{C_1} \cdot m^{1-C_2}$ in Theorem 2: the algorithm that always outputs the all-zeros vector solves (g,h)-SPARSE REGRESSION for $g(p) = 0$ and $h(m,p) = m$.

We also show, assuming a slight strengthening of a standard conjecture—known as the projection games conjecture (PGC) (cf. Moshkovitz (2012))—about the existence of probabilistically checkable proofs with small soundness error, that (g,h)-SPARSE REGRESSION is hard even if g grows as a constant power. We refer to our slight strengthening of the PGC as the "Biregular PGC," and state it formally in Section 2.3.

Theorem 3 *Assuming the Biregular PGC, the following holds: If* NP $\not\subseteq$ BPP, *then for any positive constants* C_1, C_2, *there exist a* $g(p)$ *in* $p^{\Omega(1)}$ *and an* $h(m,p)$ *in* $\Omega\left(p^{C_1} \cdot m^{1-C_2}\right)$ *such that there is no polynomial-time randomized algorithm for* (g,h)-SPARSE REGRESSION.

We might consider (g,h)-SPARSE REGRESSION to be a "computer science" version of Sparse Regression, in the sense that the data are deterministic and fully specified. In an alternative, "statistics" version of Sparse Regression, the data are corrupted by random noise unknown to the algorithm. Specifically we consider the following problem, which we call (g,h)-NOISY SPARSE REGRESSION.

- There are a positive integer k and an $m \times p$ Boolean matrix B, such that there exists an unknown p-dimensional vector $\mathbf{x}^*$ with $\|\mathbf{x}^*\|_0 \leq k$ such that $B\mathbf{x}^* = \mathbf{e}^{(m)}$. An m-dimensional vector $\boldsymbol{\epsilon}$ of i.i.d. $N(0,1)$ "noise" components ϵ_i is generated and $\mathbf{y}$ is set to $B\mathbf{x}^* + \boldsymbol{\epsilon} = \mathbf{e}^{(m)} + \boldsymbol{\epsilon}$. B, k, and $\mathbf{y}$ (but not $\boldsymbol{\epsilon}$ or $\mathbf{x}^*$) are revealed to the algorithm.
- Goal: Output a (possibly random) $\mathbf{x} \in \mathbb{R}^p$ such that $E[\|B(\mathbf{x}-\mathbf{x}^*)\|^2] \leq h(m,p)$ and $\|\mathbf{x}\|_0 \leq k \cdot g(p)$. Here, the expectation is taken over both the internal randomness of the algorithm and of the ϵ_i's.

We give a simple reduction from (g,h)-SPARSE REGRESSION to (g,h)-NOISY SPARSE REGRESSION that proves the following theorems.

Theorem 4 *Assume that* NP $\not\subseteq$ BPTIME$(n^{polylog(n))})$. *For any positive constants* δ, C_1, C_2, *there exist a* $g(p)$ *in* $2^{\Omega(\log^{1-\delta}(p))}$ *and an* $h(m,p)$ *in* $\Omega(p^{C_1} \cdot m^{1-C_2})$ *such that there is no quasipolynomial-time randomized algorithm for* (g,h)-NOISY SPARSE REGRESSION.

Theorem 5 *Assuming the Biregular PGC, the following holds. If* NP $\not\subseteq$ BPP, *then for any positive constants* C_1, C_2, *there exist a* $g(p)$ *in* $p^{\Omega(1)}$ *and an* $h(m,p)$ *in* $\Omega\left(p^{C_1} m^{1-C_2}\right)$ *such that there is no polynomial-time randomized algorithm for* (g,h)-NOISY SPARSE REGRESSION.

Importance and Prior Work. Variable selection is a crucial part of model design in statistics. People want a model with a small number of variables partially for simplicity and partially because models with fewer variables tend to have smaller generalization error; that is, they give better predictions on test data (data not used in generating the model). Standard greedy statistical algorithms to choose features for linear regression include forward stepwise selection (also known as stepwise regression or orthogonal least squares), backward elimination, and least angle regression. Standard non-greedy feature selection algorithms include LASSO and ridge regression.

There are algorithmic results for sparse linear regression that guarantee good performance under certain conditions on the matrix B. For example, equation (6) of Zhang et al. (2014) states that a "restricted eigenvalue condition" implies that the LASSO algorithm will give good performance for the statistical version of the problem. A paper by Natarajan (1995) presents an algorithm, also known as forward stepwise selection or orthogonal least squares (see also Blumensath and Davies (2007)), which achieves good performance provided that the L_2-norm of the pseudoinverse of the matrix obtained from B by normalizing each column is small (Natarajan, 1995). It appears that no upper bound for stepwise regression under reasonable assumptions on the matrix was known prior to the appearance of Natarajan's algorithm in 1995 (and the equivalence of Natarajan's algorithm with forward stepwise selection appears not to have been noticed until recently). For completeness, in the appendix to the full version (available at (Foster et al., 2014)), we include an example proving that Natarajan's algorithm can perform badly when the L_2-norm of that pseudoinverse is large, or in other words, that Natarajan's analysis of his algorithm is close to tight. A similar example had previously been given by Chen et al. (2001).

There have been several prior works establishing hardness results for variants of the sparse regression problem. Natarajan (1995) used a reduction from EXACT COVER BY 3-SETS to prove that, given an $m \times p$ matrix A, a vector $b \in \mathbb{R}^m$, and $\epsilon > 0$, it is NP-hard to compute a vector x satisfying $\|Ax - b\| < \epsilon$ if such an x exists, such that x has the fewest nonzero entries over all such vectors. Davis et al. (1997) proved a similar NP-hardness result. The hardness results of Natarajan (1995) and Davis et al. (1997) only establish hardness if the algorithm is not allowed to "cheat" simultaneously on both the sparsity and accuracy of the resulting linear combination.

Arora et al. (1997) showed that, for any $\delta > 0$, a problem called MIN-UNSATISFY does not have any polynomial-time algorithm achieving an approximation factor of $2^{\log^{1-\delta}(n)}$, assuming NP $\not\subseteq$ DTIME$(n^{\text{polylog}(n)})$. In this problem, the algorithm is given a system $Ax = b$ of linear equations over the rationals, and the cost of a solution x^* is the number of equalities that are violated by x^*. Amaldi and Kann (1998) built directly on the result of Arora et al. (1997) to show, in our terminology, that $(2^{\log^{1-\delta}(n)}, 0)$-SPARSE REGRESSION also has no polynomial-time algorithm under the same assumption.

Finally, Zhang et al. (2014) showed a hardness result for (g, h)-NOISY SPARSE REGRESSION. We defer a discussion of the result of Zhang et al. (2014) to Section 3.1. For now, we just note that their hardness only applies to algorithms which cannot "cheat" on the sparsity, that is, to algorithms that must generate a solution with at most k nonzeros.

In summary, to the best of our knowledge, our work is the first to establish that sparse linear regression is hard to approximate, even when the algorithm is allowed to "cheat" on *both* the sparsity of the solution output and the accuracy of the resulting linear combination.

2. Proofs of Theorems 2 and 3

2.1. Notation and Proof Outline

Throughout, we will use lower-case boldface letters to denote vectors. For any vector $\mathbf{y} \in \mathbb{R}^m$, $\|\mathbf{y}\|$ will denote the Euclidean norm of $\mathbf{y}$, while $\|\mathbf{y}\|_0$ will denote the sparsity (i.e., the number of nonzeros) of $\mathbf{y}$. For any vector $\mathbf{b}$, let $\mathbf{Ball}_\Delta(\mathbf{b}) = \{\mathbf{b}' : \|\mathbf{b} - \mathbf{b}'\|^2 \leq \Delta\}$ denote the ball of radius Δ around $\mathbf{b}$ in the *square* of the Euclidean norm. If $\mathbf{y} = \sum_i c_i \mathbf{w}_i$ represents a vector $\mathbf{y}$ as a linear combination of vectors $\mathbf{w}_i$, we say that $\mathbf{w}_i$ *participates* in the linear combination if $c_i \neq 0$. We will use the symbol N to denote the input size of SAT instances used in our reductions.

The first step in our proof of Theorem 2 is to establish Proposition 6 below.

Proposition 6 *If* SAT $\notin$ BPTIME$(N^{polylog(N)})$*, then for any constant* $\delta < 1$*, there are a polynomial* $m = m(p)$*, a* $k = k(p)$*, and a pair* $\sigma = \sigma(p), \Delta = \Delta(p)$ *of values both in* $2^{\Omega\left(\log^{1-\delta}(p)\right)}$*, such that no quasipolynomial-time randomized algorithm distinguishes the following two cases, given an* $m \times p$ *Boolean matrix* B*:*

1. *There is an* $\mathbf{x} \in \{0,1\}^p$ *such that* $B\mathbf{x} = \mathbf{e}^{(m)}$ *and* $\|\mathbf{x}\|_0 \leq k$.
2. *For all* $\mathbf{x} \in \mathbb{R}^p$ *such that* $B\mathbf{x} \in \mathbf{Ball}_\Delta(\mathbf{e}^{(m)})$, $\|\mathbf{x}\|_0 \geq k \cdot \sigma$.

(For the purpose of proving Theorem 2, having $\Delta = 1$ in Proposition 6 would actually suffice.)

The second step of the proof describes a simple transformation of any (g,h)-SPARSE REGRESSION algorithm for a "fast-growing" function h into a (g,h)-SPARSE REGRESSION algorithm for $h(m,p) = 1$. The proof appears in Section 2.4.

Proposition 7 *Let* C_1, C_2 *be any positive constants. Let* $\mathcal{A}$ *be an algorithm for* (g,h)-SPARSE REGRESSION *running in time* $T(m,p)$*, for some function* $g(\cdot) \geq 1$*, and for* $h(m,p) = p^{C_1} m^{1-C_2}$*. Then there is an algorithm* $\mathcal{A}'$ *for* $(g,1)$-SPARSE REGRESSION *that runs in time poly*$(T(poly(m,p),p))$.

Proof [Proof of Theorem 2 assuming Propositions 6 and 7] Suppose by way of contradiction that there are positive constants δ, C_1, C_2 such that there is a quasipolynomial-time randomized algorithm $\mathcal{A}$ for (g,h)-SPARSE REGRESSION where $g(p) = 2^{\Omega(\log^{1-\delta}(p))}$ and $h(m,p) = p^{C_1} m^{1-C_2}$. By Proposition 7, $\mathcal{A}$ can be transformed into a randomized quasipolynomial-time algorithm $\mathcal{A}'$ for $(g,1)$-SPARSE REGRESSION.

Clearly $\mathcal{A}'$ is capable of distinguishing the following two cases, for any $\Delta \geq 1$:

1. There is an $\mathbf{x} \in \{0,1\}^p$ such that $B\mathbf{x} = \mathbf{e}^{(m)}$ and $\|\mathbf{x}\|_0 \leq k$.
2. For all $\mathbf{x} \in \mathbb{R}^p$ such that $B\mathbf{x} \in \mathbf{Ball}_\Delta(\mathbf{e}^{(m)})$, $\|\mathbf{x}\|_0 \geq k \cdot g(p)$.

In particular, the above holds for $\Delta = \Delta(p)$ in $2^{\Omega(\log^{1-\delta}(p))} > 1$, which contradicts Proposition 6. ■

Proof Outline for Proposition 6. Lund and Yannakakis showed, assuming SAT cannot be solved by algorithms running in time $O(N^{\text{polylog}(N)})$, that SET-COVER cannot be approximated within a factor of $c \cdot \log_2 N$ for any constant $c < 1/4$ (Lund and Yannakakis, 1994). Here, an instance of SET-COVER consists of a set D of size N and a family $\{D_1, \ldots D_M\}$ of subsets of D, and the goal is to find a minimal collection of the D_i's whose union equals D.

Lund and Yannakakis' transformation from an instance ϕ of SAT to an instance of SET-COVER has a (known) remarkable property: if ϕ is satisfiable, then the generated instance of SET-COVER does not just have a small cover $\mathcal{C}$ of the base set D, it has a small *partition* of D. That is, if $\mathbf{c}_i$ denotes the indicator vector of set C_i, then $\sum_{C_i \in \mathcal{C}} \mathbf{c}_i = \mathbf{e}^{(|S|)}$. This is a stronger property than $\sum_{C_i \in \mathcal{C}} \mathbf{c}_i \geq \mathbf{e}^{(|S|)}$, which is the condition required to show hardness of SET-COVER. This observation naturally allows us to define a corresponding instance of the Linear Regression problem with a sparse solution; the columns of the matrix B in the regression problem are simply the indicator vectors $\mathbf{c}_i$.

A central ingredient in Lund and Yannakakis' transformation is a certain kind of set system $\{V_1, \ldots, V_M\}$ over a base set S. Their set system naturally requires that any union of fewer than ℓ V_i's or $\overline{V_i}$'s cannot *cover* S, unless there is some i such that both V_i and $\overline{V_i}$ participate in the union. As a result, they needed $|S|$ to be polynomial in M and exponential in ℓ. Since we are studying SPARSE REGRESSION rather than SET-COVER, we can impose a weaker condition on our set systems. Specifically, we need that:

any *linear combination* of ℓ indicator vectors of the sets or their complements is "far" from $\mathbf{e}^{(m)}$, unless the linear combination involves both the indicator vector of a set and the indicator vector of its complement.

(See Definition 8 below.) As a result, we are able to take $|S|$ to be much smaller relative to M and ℓ than can Lund and Yannakakis. Specifically, we can take $|S|$ to be *polynomial* in ℓ and *logarithmic* in M. This is the key to establishing hardness for super-logarithmic approximation ratios, and to obtaining hardness results even when we only require an approximate solution to the system of linear equations.

2.2. Proof of Proposition 6

2.2.1. PRELIMINARIES

A basic concept in our proofs is that of Δ-*useful set systems*, defined below.

Definition 8 *Let M and ℓ be positive integers, and let S be any finite set. A* set system $\mathcal{S}_{M,\ell} = \{V_1, \ldots, V_M\}$ *of size M over S is any collection of M distinct subsets of S. $\mathcal{S}_{M,\ell}$ is called* Δ-useful, *for $\Delta \geq 0$, if the following properties are satisfied.*

Let $\mathbf{v}_i \in \{0,1\}^{|S|}$ denote the indicator vector of V_i; that is, $\mathbf{v}_{i,j} = 1$ if $j \in V_i$, and 0 otherwise. Let $\bar{\mathbf{v}}_i$ denote the indicator vector of the complement of V_i. Then no ℓ-sparse linear combination of the vectors $\mathbf{v}_1, \bar{\mathbf{v}}_1, \ldots, \mathbf{v}_M, \bar{\mathbf{v}}_M$ is in $\mathbf{Ball}_\Delta(\mathbf{e}^{(|S|)})$, unless there is some i such that $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ both participate in the linear combination.

(Note that there is a 2-sparse linear combination involving $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ that exactly equals $\mathbf{e}^{(|S|)}$, namely, $\mathbf{v}_i + \bar{\mathbf{v}}_i = \mathbf{e}^{(|S|)}$.)

Lemma 9 *For any pair M, ℓ, $M \geq 2$, of positive integers, there exists a set $S = \{1, 2, ..., |S|\}$ of size $O(\ell^2 \cdot \log M)$ such that there is a Δ-useful set system $\mathcal{S}_{M,\ell}$ over S, for some $\Delta = \Omega(|S|)$. Moreover, there is a polynomial-time randomized algorithm that takes M and ℓ and generates a Δ-useful set system $\mathcal{S}_{M,\ell}$ over S with probability at least .99.*

The proof of Lemma 9 below shows that a random collection of sets of the right size works. It also seems likely that a deterministic construction that appears in Arora et al. (1997) could be modified to generate a Δ-useful set system.

Proof Throughout the proof, we set

$$|S| = \lceil 256\ell^2 \ln M \rceil. \tag{1}$$

To avoid notational clutter, we denote $\mathbf{e}^{(|S|)}$ simply as $\mathbf{e}$ throughout the proof. The core of our argument is the following technical lemma bounding the probability that $\mathbf{e}$ is "close" to the span of a "small" number of randomly chosen vectors from $\{0,1\}^{|S|}$.

Sublemma 10 *Given $M \geq 2$ and ℓ, define $|S|$ as above. Let $\mathbf{v}_1, \ldots, \mathbf{v}_\ell \in \{0,1\}^{|S|}$ be chosen independently and uniformly at random from $\{0,1\}^{|S|}$. Let E denote the event that there exist coefficients $c_1, \ldots, c_\ell \in \mathbb{R}$ such that $\|\mathbf{e} - \left(\sum_{i=1}^{\ell} c_i \mathbf{v}_i\right)\|^2 \leq |S|/32$. Then the probability of E is at most $M^{-32\ell}$.*

Proof Rather than reasoning directly about the Boolean vectors $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$ in the statement of the sublemma, it will be convenient to reason about vectors in $\{-1,1\}^\ell$. Accordingly, for any vector $\mathbf{v} \in \{0,1\}^{|S|}$, define $\mathbf{v}^* = 2\mathbf{v} - 1$. Notice that if $\mathbf{v}$ is a uniform random vector in $\{0,1\}^{|S|}$, then $\mathbf{v}^*$ is a uniform random vector in $\{-1,1\}^{|S|}$. The primary reason we choose to work with vectors over $\{-1,1\}^{|S|}$ rather than $\{0,1\}^{|S|}$ is

that vectors in $\{-1,1\}^{|S|}$ always have squared Euclidean norm exactly equal to $|S|$, in contrast to Boolean vectors, which can have squared Euclidean norm as low as 0 and as large as $|S|$. Throughout, given a set T of vectors in $\mathbb{R}^{|S|}$, and another vector $\mathbf{v}$ in $\mathbb{R}^{|S|}$, we let $\Pi_T(\mathbf{v})$ denote the projection of $\mathbf{v}$ onto the linear span of the vectors in T.

Note that for any Boolean vectors $\mathbf{w}, \mathbf{v} \in \{0,1\}^{|S|}$, $\|\mathbf{w}^* - \mathbf{v}^*\|^2 = 4\|\mathbf{w} - \mathbf{v}\|^2$. Hence, event E from the statement of Sublemma 10 occurs if and only if there exist coefficients $c_1, \ldots, c_\ell \in \mathbb{R}$ such that $\|\mathbf{e}^* - \left(\sum_{i=1}^{\ell} c_i \mathbf{v}_i^*\right)\|^2 \leq |S|/8$. We bound the probability of this occurrence as follows.

Let $T = \{\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*\}$. Note that

$$\min_{c_1,\ldots,c_\ell \in \mathbb{R}} \|\mathbf{e}^* - \left(\sum_{i=1}^{j} c_i \mathbf{v}_i^*\right)\|^2 = \|\mathbf{e}^*\|^2 - \|\Pi_T(\mathbf{e}^*)\|^2 = |S| - \|\Pi_T(\mathbf{e}^*)\|^2.$$

Combined with the previous paragraph, we see that event E occurs if and only if

$$\|\Pi_T(\mathbf{e}^*)\|^2 \geq 7|S|/8.$$

By equation (1), $7|S|/8 > 128\ell^2 \ln M$, so it suffices to bound from above the probability that

$$\|\Pi_T(\mathbf{e}^*)\|^2 > 128\ell^2 \ln M.$$

Bounding the probability that $\|\Pi_T(\mathbf{e}^*)\|^2 > 128\ell^2 \ln M$. Our analysis consists of two steps. In step 1, we bound the probability that $\|\Pi_T(\mathbf{w}^*)\|^2 > 128\ell^2 \ln M$, where $\mathbf{w}^*$ is a vector chosen uniformly at random from $\{-1,1\}^{|S|}$, independently of $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*$. In step 2, we argue that, even though $\mathbf{e}^*$ is a fixed vector (not chosen uniformly at random from $\{-1,1\}^{|S|}$), it still holds that $\|\Pi_T(\mathbf{e}^*)\|^2 > 128\ell^2 \ln M$ with exactly the same probability (where the probability is now only over the random choice of vectors $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^* \in \{-1,1\}^{|S|}$).

Step 1. Let $\ell' \leq \ell$ denote the dimension of the linear subspace $V := \text{span}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*)$. Let $\{\mathbf{z}_1^*, \ldots, \mathbf{z}_{\ell'}^*\}$ denote an arbitrary orthonormal basis for V. We have

$$\|\Pi_T(\mathbf{w}^*)\|^2 = \sum_{i=1}^{\ell'} \langle \mathbf{w}^*, \mathbf{z}_i \rangle^2 = \sum_{i=1}^{\ell'} \left(\sum_{j=1}^{|S|} \mathbf{w}_j^* \cdot \mathbf{z}_{i,j}\right)^2, \tag{2}$$

where $\mathbf{w}_j^*$ denotes the jth entry of $\mathbf{w}^*$, and $\mathbf{z}_{i,j}$ denotes the jth entry of $\mathbf{z}$. For any i, let w_i denote the value of the sum $w_i = \sum_{j=1}^{|S|} \mathbf{w}_j^* \cdot \mathbf{z}_{i,j}$. Since $\mathbf{z}_i$ is a vector in an orthonormal basis, we know that $\sum_{j=1}^{|S|} \mathbf{z}_{i,j}^2 = 1$.

Meanwhile, each $\mathbf{w}_j^*$ is a Rademacher random variable. Because we can view $\mathbf{v}_1^*, ..., \mathbf{v}_\ell^*$ and hence $\mathbf{z}_1^*, ..., \mathbf{z}_{\ell'}^*$ as fixed while $\mathbf{w}^*$ is random, and because $\sum_j \mathbf{z}_{i,j}^2 = 1$, standard concentration results for Rademacher sequences (Montgomery-Smith, 1990) imply that for $t > 0$ and for all i, $\Pr[w_i > t] = \Pr[\sum_{j=1}^{|S|} \mathbf{w}_j^* \cdot \mathbf{z}_{i,j} > t] \leq e^{-t^2/2}$, for all fixed $\mathbf{v}_1^*, ..., \mathbf{v}_\ell^*$, and hence even if $\mathbf{v}_1^*, ..., \mathbf{v}_\ell^*$ is random. In particular, $\Pr[w_i > \sqrt{128\ell \ln M}] \leq e^{-64\ell \ln M}$. A union bound over all i implies that $\Pr[w_i > \sqrt{128\ell \ln M}$ for all $i \in [\ell']] \leq \ell' \cdot e^{-64\ell \ln M} \leq e^{-32\ell \ln M} = M^{-32\ell}$. In this event, i.e., for all i $\sum_{j=1}^{|S|} \mathbf{w}_j^* \cdot \mathbf{z}_{i,j} \leq \sqrt{128\ell \ln M}$, we can bound the right-hand side of equation (2) by

$$\sum_{i=1}^{\ell'} \left(\sum_{j=1}^{|S|} \mathbf{w}_j^* \cdot \mathbf{z}_{i,j}\right)^2 \leq \ell' \cdot 128\ell \ln M \leq 128\ell^2 \ln M.$$

Step 2. For vectors $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*, \mathbf{w}^* \in \{-1,1\}^{|S|}$, let $\mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{w}^*)$ equal 1 if $\|\Pi_T(\mathbf{w}^*)\|^2 > 128\ell^2 \ln M$, and 0 otherwise, where $T = \{\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*\}$. We claim that $\mathbb{E}[\mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{w}^*)] = \mathbb{E}[\mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{e}^*)]$, where the first expectation is over random choice of $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*, \mathbf{w}^* \in \{-1,1\}^{|S|}$, and the second expectation is only over the random choice of $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^* \in \{-1,1\}^{|S|}$.

For two vectors $\mathbf{x}, \mathbf{y} \in \{-1,1\}^{|S|}$, let $\mathbf{x} \otimes \mathbf{y}$ denote the component-wise product of $\mathbf{x}$ and $\mathbf{y}$, i.e., $(\mathbf{x} \otimes \mathbf{y})_i = \mathbf{x}_i \cdot \mathbf{y}_i$. Then we can write:

$$\begin{aligned}
\mathbb{E}[\mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{w}^*)] &= 2^{-|S|\cdot(\ell+1)} \sum_{\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*, \mathbf{w}^* \in \{-1,1\}^{|S|}} \mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{w}^*) \\
&= 2^{-|S|\cdot(\ell+1)} \sum_{\mathbf{w}^* \in \{-1,1\}^{|S|}} \left[\sum_{\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^* \in \{-1,1\}^{|S|}} \mathbb{I}(\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*; \mathbf{w}^*) \right] \\
&= 2^{-|S|\cdot(\ell+1)} \sum_{\mathbf{w}^* \in \{-1,1\}^{|S|}} \left[\sum_{\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^* \in \{-1,1\}^{|S|}} \mathbb{I}(\mathbf{v}_1^* \otimes \mathbf{w}^*, \ldots, \mathbf{v}_\ell^* \otimes \mathbf{w}^*; \mathbf{w}^* \otimes \mathbf{w}^*) \right] \\
&= 2^{-|S|\cdot\ell} \sum_{\mathbf{v}_1', \ldots, \mathbf{v}_\ell' \in \{-1,1\}^{|S|}} \mathbb{I}(\mathbf{v}_1', \ldots, \mathbf{v}_\ell'; \mathbf{e}^*) \\
&= \mathbb{E}[\mathbb{I}(\mathbf{v}_1', \ldots, \mathbf{v}_\ell'; \mathbf{e}^*].
\end{aligned}$$

Here, the first equality is the definition of expectation; the second follows by rearranging the sum. The third equality holds because multiplying each of $\mathbf{v}_1^*, ..., \mathbf{v}_\ell^*$ and $\mathbf{w}^*$ component-wise by $\mathbf{w}^*$ is the same as premultiplying each vector by the *unitary* matrix $U = \text{diag}(\mathbf{w}^*)$, and premultiplying by a unitary matrix just rotates the space, which doesn't change lengths of projections. The fourth equality holds because, for any fixed $\mathbf{w}^*$, if the vectors $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*$ are uniformly distributed on $\{-1,1\}^{|S|}$, then so are the vectors $\mathbf{v}_1' := \mathbf{v}_1^* \otimes \mathbf{w}^*, \ldots, \mathbf{v}_\ell' := \mathbf{v}_\ell^* \otimes \mathbf{w}^*$. The final equality is the definition of expectation. ■

Let $V_1, \ldots, V_M$ be random subsets of S, and $\mathbf{v}_i$ be the indicator vector of V_i. Consider any subset Z of ℓ of the vectors $\mathbf{v}_1, \bar{\mathbf{v}}_1, \ldots, \mathbf{v}_M, \bar{\mathbf{v}}_M$ in which there is no i such that both $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ are in Z. (There are exactly $\binom{M}{\ell}2^\ell$ such subsets Z.) Then the vectors in Z are all uniformly random vectors in $\{0,1\}^{|S|}$ that are chosen independently of each other. Sublemma 10 implies that the probability that there exist coefficients $c_1, \ldots, c_\ell \in \mathbb{R}$ such that $\|\mathbf{e} - \left(\sum_{i=1}^\ell c_i \mathbf{v}_i\right)\|^2 \leq |S|/32$ is at most $M^{-32\ell}$.

We can take a union bound over all $\binom{M}{\ell} \cdot 2^\ell \leq M^{2\ell}$ possible choices of Z to conclude that $\|\mathbf{e} - \left(\sum_{i=1}^\ell c_i \mathbf{v}_i\right)\|^2 > |S|/32$ for *all* possible choices of the subset Z with probability at least $1 - M^{2\ell} \cdot M^{-32\ell} = 1 - M^{-30\ell} \geq 0.99$. In this event, there is no ℓ-sparse linear combination of the $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ vectors in $\text{Ball}_\Delta(\mathbf{e})$ for $\Delta = |S|/32$, unless $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ both participate in the linear combination for some i. That is, $\{V_1, \ldots, V_M\}$ is a Δ-useful set system, as desired. This completes the proof of Lemma 9. ■

Please see the remainder of the proof of Proposition 6 in the full version of this paper (Foster et al., 2014).

2.3. Proof of Theorem 3

Let $\text{PCP}_{1,\epsilon_{\text{sound}}}[b,k]_{|\Sigma|}$ denote the class of languages that have a PCP verifier with perfect completeness, soundness ϵ_{sound}, b bits of randomness used by the verifier, and k queries to a proof over alphabet Σ. The *sliding scale conjecture* of Bellare et al. (1994) postulates that for every $0 \leq \delta \leq 1$,

SAT $\in$ PCP$_{1,\epsilon}[O(\log(N)), 2]_{\text{poly}(1/\epsilon)}$, where $\epsilon = 2^{-\Omega(\log^{1-\delta}(N))}$. In words, the sliding scale conjecture asserts that SAT has a polynomial-length PCP in which the verifier makes two queries to the proof (note that each query reads an entire symbol from the alphabet Σ, and hence corresponds to approximately $\log|\Sigma|$ consecutive bits of the proof), and in which the soundness error falls exponentially with the number of bits of the proof that the verifier accesses.

Note that any k-query PCP can be transformed into a k-prover MIP by posing each of the PCP verifier's k queries to a different prover. Thus, the sliding scale conjecture (at $\delta = 0$) implies the existence of a 2-prover MIP for SAT in which $|R|, |Q_1|, |A_1|, |Q_2|, |A_2|$ are all at most poly(N), and whose soundness error is inversely polynomially small in N.

Moshkovitz (2012) formulates a slight strengthening of the sliding scale conjecture that she calls the *Projection Games Conjecture* (PGC). Here, the term *projection games* refers to a certain class of two-player one-round games defined over bipartite graphs. Projection games are equivalent, in a formal sense, to two-prover MIP's that satisfy functionality (the definition of which appears in the full version of this paper (Foster et al., 2014)). This equivalence is standard; we provide the details in the appendix of the full version for completeness.

Just like the sliding scale conjecture, the PGC postulates the existence of a 2-prover MIP for SAT in which $|R|, |Q_1|, |A_1|, |Q_2|, |A_2|$ are all at most poly(N), and in which the soundness error is polynomially small in N. But the PGC additionally postulates that there is such an MIP that satisfies the functionality property used in the proof of Proposition 6. Formally, the PGC states the following.[1]

Conjecture 11 (Reformulation of the projection games conjecture (PGC) (Moshkovitz, 2012)) *There exists $c > 0$ such that for all N and all $\epsilon \geq 1/N^c$, there is a 2-prover MIP for* SAT *instances of size N, with perfect completeness and soundness error at most ϵ, which satisfies the following two properties:*

- *$|R|, |Q_1|, |A_1|, |Q_2|, |A_2|$ are all at most poly$(N, 1/\epsilon)$.*
- *The MIP satisfies functionality.*

We will need a slight strengthening of the PGC that requires that SAT be efficiently reduced to a 2-prover MIP that satisfies uniformity (the definition of which appears in the full version of this paper (Foster et al., 2014)) in addition to functionality. We note that in the language of projection games, the uniformity condition is equivalent to requiring the bipartite graph underlying the projection game be biregular. Moreover, the most efficient known reductions from SAT to projection games do produce biregular graphs (and hence 2-prover MIP's satisfying uniformity) (Dinur and Steurer, 2014).

Conjecture 12 (Biregular PGC) *Conjecture 11 holds in which the MIP satisfies uniformity as well.*

Proposition 13 below describes a strengthening of Proposition 6 that holds under the Biregular PGC. Theorem 3 then follows by the argument of Section 2.1, using Proposition 7 in place of Proposition 13.

Proposition 13 *Assuming Conjecture 12, the following holds for some pair of values $\sigma = \sigma(p), \Delta = \Delta(p)$, both in $p^{\Omega(1)}$. If* SAT $\notin$ BPP, *then there are a polynomial $m = m(p)$ and $k = k(p)$ such that no* BPP *algorithm distinguishes the following two cases, given an $m \times p$ Boolean matrix B:*

1. Technically, Moshkovitz's PGC is slightly stronger than our Conjecture 11, in the full version of the paper, in that $|R|$, $|Q_1|$, and $|Q_2|$ are all required to be bounded by $N^{1+o(1)}\text{poly}(1/\epsilon)$, rather than poly$(N, 1/\epsilon)$. She states the weaker version in a footnote, and the weaker version suffices for our purposes.

1. *There is an* $\mathbf{x} \in \{0,1\}^p$ *such that* $B\mathbf{x} = \mathbf{e}^{(m)}$ *and* $\|\mathbf{x}\|_0 \leq k$.

2. *For all* $\mathbf{x} \in \mathbb{R}^p$ *such that* $B\mathbf{x} \in \mathbf{Ball}_\Delta(\mathbf{e}^{(m)})$, $\|\mathbf{x}\|_0 \geq k \cdot \sigma$.

Proof Conjecture 12 implies the existence of a perfectly complete two-prover MIP for SAT with soundness error $\epsilon_{\text{sound}} = 1/N^{\Omega(1)}$ that satisfies functionality and uniformity, with $|R|, |A_1|, |A_2|, |Q_1|, |Q_2|$ all bounded by $\text{poly}(N)$. Moreover, this MIP can be transformed into an equivalent MIP that satisfies both equality of question space sizes and disjointness of answer spaces (both of which are defined in the full version of this paper (Foster et al., 2014)), while keeping $|R|, |A_1|, |A_2|, |Q_1|, |Q_2|$ bounded by $\text{poly}(N)$. Disjointness of answer spaces can be ensured by simple padding (only one bit of padding is required). Equality of question space sizes can be ensured (Lund and Yannakakis, 1994), with at most a quadratic blowup in the size of Q_1 and Q_2. Namely, the verifier generates two independent queries (q_1, q_2) and (q_1', q_2') in $Q_1 \times Q_2$. Then, the verifier asks the first prover the query (q_1, q_2') and the second prover the query (q_2, q_1'). The provers ignore the second component, and answer the queries by answering the first component of the query according to the original MIP. The verifier accepts if and only if the verifier in the original MIP would have accepted the answers to (q_1, q_2).

Thus, we have established that Conjecture 12 implies the existence of a perfectly complete canonical MIP for SAT with soundness error $\epsilon_{\text{sound}} = 1/N^{\Omega(1)}$ and for which $|R|, |A_1|, |A_2|, |Q_1|, |Q_2|$ are all bounded by $\text{poly}(N)$. The remainder of the proof is now identical to that of Proposition 6; we provide the details for completeness.

The proof of Proposition 6 specified an algorithm $\mathcal{A}''$ that output a Boolean $m \times p$ matrix B of size polynomial in the parameters $(|R|, |A_1|, |A_2|, |Q_1|, |Q_2|)$, which are themselves polynomial in N (so that mp is polynomial in N), such that:

Property 1. If $\phi \in$ SAT, then there is a vector $\mathbf{x}^* \in \{0,1\}^p$ such that $B\mathbf{x}^* = \mathbf{e}^{(m)}$ and $\|\mathbf{x}^*\|_0 \leq |Q_1| + |Q_2|$.

Property 2. Abusing notation, let $\ell := \ell(N)$. If $\phi \notin$ SAT, then any $\mathbf{x} \in \mathbb{R}^p$ such that $B\mathbf{x} \in \mathbf{Ball}_\Delta(\mathbf{e}^{(m)})$ satisfies $\|\mathbf{x}\|_0 \geq (1 - \epsilon_{\text{sound}} \cdot \ell^2) \cdot \frac{\ell}{2} \cdot (|Q_1| + |Q_2|)$, for some $\Delta = \Omega\left(\ell^2 \cdot \log\left(|A_1| + |A_2|\right)\right)$.

Setting $\ell = (10 \cdot \epsilon_{\text{sound}})^{-1/2}$ and $k = |Q_1| + |Q_2|$, Properties 1 and 2 imply that

- If $\phi \in$ SAT, then there is a vector $\mathbf{x}^* \in \{0,1\}^p$ such that $B\mathbf{x}^* = \mathbf{e}^{(m)}$ and $\|\mathbf{x}^*\|_0 \leq k$.
- If $\phi \notin$ SAT, then any $\mathbf{x} \in \mathbb{R}^p$ such that $B\mathbf{x} \in \mathbf{Ball}_\Delta(\mathbf{e}^{(m)})$ satisfies $\|\mathbf{x}\|_0 \geq \sigma \cdot k$, where $\Delta(p) = \Omega\left(\ell^2 \cdot \log\left(|A_1| + |A_2|\right)\right) \geq \Omega(\ell) = p^{\Omega(1)}$, and $\sigma(p) = (1 - \epsilon_{\text{sound}} \cdot \ell^2) \cdot \frac{\ell}{2} \geq \Omega(\ell) = p^{\Omega(1)}$. (Recall that p is polynomial in N.)

Suppose there is a randomized algorithm $\mathcal{A}'$ that runs in $\text{BPTIME}(\text{poly}(m,p)) = \text{BPTIME}(\text{poly}(N))$ and distinguishes between the above two cases. By running $\mathcal{A}'$ on the matrix B output by $\mathcal{A}''$, we determine with high probability whether ϕ is satisfiable. Thus, if SAT $\notin \text{BPTIME}(\text{poly}(N))$, no such algorithm $\mathcal{A}'$ can exist. ■

2.4. Proof of Proposition 7

Proof Let $g(\cdot)$ be any function and let $h(m,p) = p^{C_1} \cdot m^{1-C_2}$ for some positive constants C_1, C_2. Let $\mathcal{A}$ be any algorithm for (g,h)-SPARSE REGRESSION that runs in randomized time $T(m,p)$. Then on any $m \times p$ input matrix B such that there exists a vector $\mathbf{x}^*$ with $\|\mathbf{x}^*\|_0 \leq k$, $\mathcal{A}$ outputs a vector $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq g(p) \cdot k$ satisfying $\|B\mathbf{x} - \mathbf{e}^{(m)}\|^2 \leq p^{C_1} \cdot m^{1-C_2}$.

We show how to transform $\mathcal{A}$ into an algorithm $\mathcal{A}'$ for the $(g,1)$-SPARSE REGRESSION problem, such that $\mathcal{A}'$ runs in time $T(p^{C_1/C_2} \cdot m^{1/C_2}, p)$.

Let $r = \lceil \left(p^{C_1} \cdot m^{1-C_2}\right)^{1/C_2} \rceil$. Let B' denote the $(mr) \times p$ matrix obtained by stacking r copies of B on top of each other. The algorithm $\mathcal{A}'$ simply runs $\mathcal{A}$ on B', to obtain an approximate solution $\mathbf{x}$ to the system of linear equations given by $B'\mathbf{x} = \mathbf{e}^{(rm)}$, and outputs $\mathbf{x}$. Notice that the running time of $\mathcal{A}'$ is indeed $O(T(p^{C_1/C_2} \cdot m^{1/C_2}, p)) = O\left(T\left(\text{poly}\left(m,p\right),p\right)\right)$.

Analysis of $\mathbf{x}$: Since $B\mathbf{x}^* = \mathbf{e}^{(m)}$, it also holds that $B'\mathbf{x}^* = \mathbf{e}^{(rm)}$. That is, the system $B'\mathbf{x} = \mathbf{e}^{(rm)}$ has an exact solution of sparsity k. Hence, $\mathcal{A}$ must output a vector $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq g(p) \cdot k$, satisfying $\|B'\mathbf{x} - \mathbf{e}^{(rm)}\|^2 \leq p^{C_1} \cdot (rm)^{1-C_2}$. Notice that $\|B'\mathbf{x} - \mathbf{e}^{(rm)}\|^2 = r \cdot \|B\mathbf{x} - \mathbf{e}^{(m)}\|^2$. Thus, $\|B\mathbf{x} - \mathbf{e}^{(m)}\|^2 = (1/r) \cdot \left[\|B'\mathbf{x} - \mathbf{e}^{(rm)}\|^2\right] \leq (1/r)\left[\cdot p^{C_1} \cdot (rm)^{1-C_2}\right] = p^{C_1} \cdot m^{1-C_2} \cdot r^{-C_2} \leq 1$. ■

3. The Statistical Problem

3.1. Motivation and Prior Work

In this section we motivate the (g,h)-NOISY SPARSE REGRESSION problem introduced in Section 1. The problem (g,h)-NOISY SPARSE REGRESSION is a sparse variant of a less challenging problem known as NOISY REGRESSION that is popular in the statistics literature. In NOISY REGRESSION, instead of being given a fixed matrix B for which we seek a sparse vector $\mathbf{x}$ such that $B\mathbf{x} \approx \mathbf{y}$, we assume corrupting random noise injected into the problem. Specifically, we assume that there are (1) a known $m \times p$ matrix X (which plays the role of B) and (2) an unknown real p-dimensional vector $\boldsymbol{\theta}$. An m-dimensional vector $\boldsymbol{\epsilon}$ of random noise is generated, with the ϵ_i's being i.i.d. $N(0,1)$ random variables. The vector $\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$ (but not $\boldsymbol{\epsilon}$ itself) is then revealed to the algorithm (along with X, which it already knew). On an instance specified by $(X, \boldsymbol{\theta})$ with $\mathbf{y}$ revealed to the algorithm, an algorithm will produce $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(X, \mathbf{y})$. We define $\hat{\mathbf{y}} = X\hat{\boldsymbol{\theta}}$. The instance's *prediction loss* is defined to be $\|\hat{\mathbf{y}} - E[\mathbf{y}]\|^2 = \|X(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\|^2$. (By "$E[\mathbf{y}]$" here, since $\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$, we mean the *vector* $X\boldsymbol{\theta}$.) Its expected value $E[\|X(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\|^2]$, over the random vector $\boldsymbol{\epsilon}$, is called the *risk*.

The goal is to make the risk as small as possible. However, as $\boldsymbol{\theta}$ is unknown, it is hard to measure the performance of an algorithm at a single $\boldsymbol{\theta}$. After all, an algorithm which luckily guesses $\boldsymbol{\theta}$ and then sets $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$ will have zero risk for that $\boldsymbol{\theta}$. For this reason, one usually seeks an algorithm that minimizes the supremum of the risk over all $\boldsymbol{\theta}$. In other words, this *minimax estimator* finds $\hat{\boldsymbol{\theta}}$ to minimize the supremum over $\boldsymbol{\theta}$ of $E[\|X(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\|^2]$. NOISY REGRESSION refers to the problem of minimizing this last supremum.

Note here that we are using random noise whose components have variance 1, regardless of the magnitudes of the entries of X. For our hard instances, the entries of X are 0-1.

It is well-known that ordinary least squares is a minimax estimator: least squares achieves risk p, *independently* of m (Foster and George, 1994, bottom of page 1950). Moreover, no estimator, polynomial-time or not, achieves smaller maximum risk. (It's not even *a priori* obvious that the risk can be made independent of m.) The problem (g,h)-NOISY SPARSE REGRESSION defined in Section 1 is a "sparse version" of NOISY REGRESSION which is identical to NOISY REGRESSION except that there is a positive integer k, known to the algorithm, such that $||\boldsymbol{\theta}||_0 \leq k$. The goal of the algorithm is to find an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ for which both the supremum over $\boldsymbol{\theta}$ of the sparsity $||\hat{\boldsymbol{\theta}}||_0$ and the supremum over $\boldsymbol{\theta}$ of the risk $E[\|X(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\|^2]$ are as small as possible.

As was the case for (g,h)-SPARSE REGRESSION, there is a naive exponential-time algorithm for SPARSE NOISY REGRESSION. Namely, try all subsets S of $\{1,2,...,p\}$ of size k. For each S, use or-

dinary least squares to find the $\hat{\boldsymbol{\theta}}_j$'s, except require that $\hat{\boldsymbol{\theta}}_j = 0$ for all $j \notin S$. An upper bound on the risk of this algorithm is known:

Theorem 14 *Foster and George (1994) For any $\boldsymbol{\theta}$, the risk of the naive exponential-time algorithm is bounded by $4k \ln p$.*

The question we are interested in is, how small can a *polynomial-time* algorithm make the risk, if it is allowed to "cheat" somewhat on the sparsity? More formally, if a polynomial-time algorithm is allowed to output a vector $\hat{\boldsymbol{\theta}}$ with $\|\hat{\boldsymbol{\theta}}\|_0 \leq k \cdot g(k)$ (where $g(k) \geq 1$ is an arbitrary function), how small can its risk be?

Theorem 4 (respectively, Theorem 5) from Section 1 asserts that the risk cannot be bounded above by $p^{C_1} \cdot m^{1-C_2}$ for *any* positive constants C_1, C_2, even if one is allowed to cheat by a nearly polynomial (respectively, polynomial) factor on the sparsity of the returned vector. This is in stark contrast to NOISY REGRESSION, where there is a polynomial-time algorithm (least-squares regression) that achieves risk p, independently of m.

Now we can describe the result of Zhang et al. (2014). There is a standard (non-greedy) algorithm known as LASSO which is known to achieve risk bounded by $O((1/\gamma^2(B))(k \log p))$, where $\gamma(B)$ is B's "restricted eigenvalue constant" (Zhang et al., 2014). Zhang, Wainwright, and Jordan prove that, unless NP⊆P/poly, for any $\delta > 0$ any polynomial-time algorithm has risk $\Omega((1/\gamma^2(B))(k^{1-\delta} \log p))$, for some matrices B (for which $\gamma(B)$ can be made arbitrarily small). To contrast their results with ours, Zhang et al. (2014) require a different complexity assumption and prove a different lower bound (involving $\gamma(B)$, which ours does not), but most importantly, their bound only applies to algorithms that return linear combinations of k columns, whereas ours allows linear combinations of $2^{\log^{(1-\delta)} p} \cdot k$ columns.

It is known that a matrix with i.i.d. $N(0,1)$ entries satisfies the restricted isometry property used in compressed sensing. From (van de Geer and Bhlmann, 2009, pages 1367-1368), which proves that the restricted isometry property implies the restricted eigenvalue condition, it is likely that the reciprocal of the restricted eigenvalue constant is polynomial. If true, this would imply that a thresholded version of Lasso (which retains the k largest components in absolute value and zeroes out the rest) has polynomial risk for random $N(0,1)$ matrices, which would imply that no proof of average-case hardness is possible; worst-case hardness, which is what our proofs provide, would be the strongest sort of hardness possible.

3.2. Proof of Theorems 4 and 5

Proposition 15 *For every pair $g(\cdot), h(\cdot,\cdot)$ of polynomials, if there is an algorithm for $(g, h/2)$-*NOISY SPARSE REGRESSION *that runs in time $T(m,p)$, then there is an algorithm for (g,h)-*SPARSE REGRESSION *that runs in time $O(T(m,p))$.*

Combining Proposition 15 with Theorem 2 proves Theorem 4 and combining Proposition 15 with Theorem 3 proves Theorem 5, respectively.

Proof Let $\mathcal{A}$ be an algorithm for $(g, h/2)$-NOISY SPARSE REGRESSION. The following algorithm $\mathcal{A}'$ solves (g,h)-SPARSE REGRESSION, with failure probability at most δ. $\mathcal{A}'$ executes the following procedure $\lceil \log(1/\delta) \rceil$ times: it generates an m-dimensional vector $\boldsymbol{\epsilon}$ whose components are (approximately) i.i.d. $N(0,1)$ random variables and sets $\mathbf{y} = \mathbf{e}^{(m)} + \boldsymbol{\epsilon}$. (We will assume that the algorithm generates $N(0,1)$ random variables exactly.) It then runs $\mathcal{A}$ on input $(B, k, \mathbf{y})$ to obtain a vector $\mathbf{x}$ of sparsity at most $k \cdot g(p)$, and checks whether $\|B\mathbf{x} - \mathbf{e}^{(m)}\|^2 \leq h(m,p)$. If so, it halts and outputs $\mathbf{x}$. If $\mathcal{A}'$ has not halted after $\lceil \lg(1/\delta) \rceil$ iterations of the above procedure, $\mathcal{A}'$ outputs a special failure symbol $\perp$. The rest of the proof appears in the full version of this paper (Foster et al., 2014). ■

References

Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(12):237 – 260, 1998. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/S0304-3975(97)00115-1. URL http://www.sciencedirect.com/science/article/pii/S0304397597001151.

Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi: 10.1006/jcss.1997.1472. URL http://dx.doi.org/10.1006/jcss.1997.1472.

Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistic checkable proofs and applications to approximation. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, page 820. ACM, 1994. ISBN 0-89791-663-8. doi: 10.1145/195058.195467. URL http://doi.acm.org/10.1145/195058.195467.

Thomas Blumensath and Mike E. Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. http://eprints.soton.ac.uk/142469/1/BDOMPvsOLS07.pdf, 2007.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, January 2001. ISSN 0036-1445. doi: 10.1137/S003614450037906X. URL http://dx.doi.org/10.1137/S003614450037906X.

Ali Çivril. A note on the hardness of sparse approximation. *Inf. Process. Lett.*, 113(14-16):543–545, 2013. doi: 10.1016/j.ipl.2013.04.014. URL http://dx.doi.org/10.1016/j.ipl.2013.04.014.

G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997. ISSN 0176-4276. doi: 10.1007/BF02678430. URL http://dx.doi.org/10.1007/BF02678430.

Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591884. URL http://doi.acm.org/10.1145/2591796.2591884.

Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi: 10.1145/285055.285059. URL http://doi.acm.org/10.1145/285055.285059.

Dean P. Foster and Edward I. George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, 22(4):1947–1975, 1994. doi: 10.1214/aos/1176325766. URL http://dx.doi.org/10.1214/aos/1176325766.

Dean P. Foster, Howard J. Karloff, and Justin Thaler. Variable selection is hard. *CoRR*, abs/1412.4832, 2014. URL http://arxiv.org/abs/1412.4832.

Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994. doi: 10.1145/185675.306789. URL http://doi.acm.org/10.1145/185675.306789.

S. J. Montgomery-Smith. The distribution of rademacher sums. *Proceedings of the American Mathematical Society*, 109(2):517–522, 06 1990. URL http://www.ams.org/journals/proc/1990-109-02/S0002-9939-1990-1013975-0/S0002-9939-1990-1013975-0.pdf.

Dana Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, volume 7408 of *Lecture Notes in Computer Science*, pages 276–287. Springer, 2012. ISBN 978-3-642-32511-3. doi: 10.1007/978-3-642-32512-0_24. URL http://dx.doi.org/10.1007/978-3-642-32512-0_24.

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

Sara A. van de Geer and Peter Bhlmann. On the conditions used to prove oracle results for the lasso. *Electron. J. Statist.*, 3:1360–1392, 2009. doi: 10.1214/09-EJS506. URL http://dx.doi.org/10.1214/09-EJS506.

Yuchen Zhang, Martin J. Wainwright, and Michael I. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In Maria-Florina Balcan and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014. Also see arXiv:1402.1918, last revised May 21, 2014.*, volume 35 of *JMLR Proceedings*, pages 921–948. JMLR.org, 2014. URL http://jmlr.org/proceedings/papers/v35/zhang14.html.