Embedding Hard Learning Problems Into Gaussian Space

Adam Klivans and Pravesh Kothari

The University of Texas at Austin, Austin, Texas, USA
{klivans,kothari}@cs.utexas.edu

Abstract

We give the first representation-independent hardness result for agnostically learning halfspaces with respect to the Gaussian distribution. We reduce from the problem of learning sparse parities with noise with respect to the uniform distribution on the hypercube (sparse LPN), a notoriously hard problem in theoretical computer science and show that any algorithm for agnostically learning halfspaces requires $n^{\Omega(\log(1/\epsilon))}$ time under the assumption that k-sparse LPN requires $n^{\Omega(k)}$ time, ruling out a polynomial time algorithm for the problem. As far as we are aware, this is the first representation-independent hardness result for supervised learning when the underlying distribution is restricted to be a Gaussian.

We also show that the problem of agnostically learning sparse polynomials with respect to the Gaussian distribution in polynomial time is as hard as PAC learning DNFs on the uniform distribution in polynomial time. This complements the surprising result of Andoni et. al. [1] who show that sparse polynomials are learnable under *random* Gaussian noise in polynomial time.

Taken together, these results show the inherent difficulty of designing supervised learning algorithms in Euclidean space even in the presence of strong distributional assumptions. Our results use a novel embedding of random labeled examples from the uniform distribution on the Boolean hypercube into random labeled examples from the Gaussian distribution that allows us to relate the hardness of learning problems on two different domains and distributions.

1998 ACM Subject Classification F.2.0. Analysis of Algorithms and Problem Complexity

Keywords and phrases distribution-specific hardness of learning, gaussian space, halfspace-learning, agnostic learning

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2014.793

1 Introduction

Proving lower bounds for learning Boolean functions is a fundamental area of study in learning theory ([3, 14, 12, 31, 8, 26, 30, 10]). In this paper, we focus on representation-independent hardness results, where the learner can output any hypothesis as long as it is polynomial-time computable. Almost all previous work on representation-independent hardness induces distributions that are specifically tailored to an underlying cryptographic primitive and only rule out learning algorithms that succeed on all distributions.

Given the ubiquity of learning algorithms that have been developed in the presence of distributional constraints (e.g., margin-based methods of [2, 4, 35] and Fourier-based methods of [22, 27]), an important question is whether functions that seem difficult to learn with respect to all distributions are in fact also difficult to learn even with respect to natural distributions. In this paper we give the first hardness result for a natural learning problem (agnostically learning halfspaces) with respect to perhaps the strongest possible distributional constraint, namely that the marginal distribution is a spherical multivariate Gaussian.

17th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14) / 18th Int'l Workshop on Randomization and Computation (RANDOM'14).
Editors: Klaus Jansen, José Rolim, Nikhil Devanur, and Cristopher Moore; pp. 793–809

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Learning Sparse Parities With Noise

Our main hardness result is based on the assumption of hardness of *learning sparse parities with noise*. Learning parities with noise (LPN) and its sparse variant are notoriously hard problems with connections to cryptography [25] in addition to several important problems in learning theory [13]. In this problem, the learner is given access to random examples drawn from the uniform distribution on the n dimensional hypercube (denoted by $\{-1,1\}^n$) that are labeled by an unknown parity function. Each label is flipped with a fixed probability η (*noise rate*), independently of others. The job of the learner is to recover the unknown parity. In the sparse variant, the learner is additionally promised that the unknown parity is on a subset of size at most a parameter k. It is easy to see that the exhaustive search algorithm for k-SLPN runs in time $n^{O(k)}$, and an outstanding open problem is to find algorithms that significantly improve upon this bound. The specific hardness assumption we take is as follows:

▶ **Assumption 1.** *Any algorithm for learning k-SLPN for any constant accuracy parameter ϵ must run in time $n^{\Omega(k)}$.*

The current best algorithm for SLPN is due to Greg Valiant [36] and runs in time $\Omega(n^{0.8k})$ for constant noise rates. Finding even an $O(n^{k/2})$-time algorithm for SLPN would be considered a breakthrough result. We note that the current best algorithm for LPN is due to Blum. et. al. [5] and runs in time $2^{O(n/\log n)}$.

Further evidence for the hardness of SLPN are the following surprising implications in learning theory: 1) an $n^{o(k)}$-time algorithm for SLPN would imply an $n^{o(k)}$-time algorithm for learning k-juntas and 2) a polynomial-time algorithm for $O(\log n)$-SLPN would imply a polynomial-time algorithm for PAC learning DNF formulas with respect to the uniform distribution on the cube *without queries* due to a reduction by Feldman et. al. [13]. The LPN and SLPN problems have also been used in previous work to show representation-independent hardness for agnostically learning halfspaces with respect to the uniform distribution on $\{-1,1\}^n$ [22] and for agnostically learning non-negative submodular functions [15].

1.2 Our results

We focus on giving hardness results for agnostically learning halfspaces and sparse polynomials. Learning halfspaces is one of the most well-studied problems in supervised learning. A halfspace (also known as a *linear classifier* or a *linear threshold function*) is a Boolean valued function (i.e. in $\{-1,1\}$) that can be represented as $\mathsf{sign}(\sum_{i=1}^{n} a_i \cdot x_i + c)$ for reals $a_1, a_2, \ldots, a_n$ and c with the input x being drawn from any fixed distribution on $\mathbb{R}^n$. Algorithms for learning halfspaces form the core of important machine learning tools such as the Perceptron [34], Artificial Neural Networks [38], Adaboost [18] and Support Vector Machines (SVMs) [38].

While halfspaces are efficiently learnable in the noiseless (PAC model of Valiant [37]) setting, the wide applicability of halfspace learning algorithms to labeled data that are not linearly separable has motivated the question of learning noisy halfspaces. Blum et. al. [6] gave an efficient algorithm to learn halfspaces under random classification noise. However, under adversarial noise (i.e.the *agnostic* setting), algorithmic progress has been possible only with distributional assumptions. Kalai et. al. [22] showed that halfspaces are agnostically learnable on the uniform distribution on the hypercube in time $n^{O(1/\epsilon^2)}$ and on the gaussian distribution in time $n^{O(1/\epsilon^4)}$. The latter running time was improved to $n^{O(1/\epsilon^2)}$ by Diakonikolas et. al. [9]. Shalev-Schwartz. et. al. [35] have given efficient agnostic algorithms for learning halfspaces in the presence of a large margin (their results

do not apply on spherical Gaussian distribution, as halfspaces with respect to Gaussian distributions may have exponentially small margins).

Kalai et. al. [22] showed that their agnostic learning algorithm on the uniform distribution on the hypercube is in fact optimal, assuming the hardness of the learning parity with noise (LPN) problem. No similar result, however, was known for the case of Gaussian distribution:

▶ **Question 1.** *Is there an algorithm running in time* $\mathsf{poly}(n, 1/\epsilon)$ *to agnostically learn halfspaces on the Gaussian distribution?*

There was some hope that perhaps agnostically learning halfspaces with respect to the Gaussian distribution would be easier than on the uniform distribution on the hypercube. We show that this is not the case and give a negative answer to the above question. In fact, we prove that any agnostic learning algorithm for the class of halfspaces must run in time $n^{\Omega(\log(1/\epsilon))}$.

▶ **Theorem 1** (See Theorem 8 for details)**.** *If Assumption 1 is true, any algorithm that agnostically learns halfspaces with respect to the Gaussian distribution to an error of* ϵ *runs in time* $n^{\Omega(\log(1/\epsilon))}$.

We next consider the problem of agnostically learning sparse (with respect to the number of monomials) polynomials. Since this is a real valued class of functions, we will work with the standard notion of ℓ_1 distance to measure errors. Thus, the distance between two functions f and g on the Gaussian distribution is given by $\mathbb{E}_{x\sim\gamma}[|f(x) - h(x)|]$. Note that ℓ_1 error reduces to the standard disagreement (or classification) distance in case of Boolean valued functions.

▶ **Question 2** (Agnostic Learning of Sparse Polynomials)**.** *For a function* $f : \mathbb{R}^n \to \mathbb{R}$*, normalized so that* $\mathbb{E}_{x\sim\gamma}[f(x)^2] = 1$*, suppose there is an* s*-sparse polynomial* p *such that* $\mathbb{E}_{x\sim\gamma}[|f(x) - p(x)|] \leq \delta \in [0, 1]$*. Is there an algorithm that uses random examples labeled by* f *to return a hypothesis* h *such that* $\mathbb{E}_{x\sim\gamma}[|h(x) - f(x)|] \leq \delta + \epsilon$*, in time* $\mathsf{poly}(s, n, 1/\epsilon)$*?*

On the uniform distribution on $\{-1, 1\}^n$, even the noiseless version (i.e. $\delta = 0$) of the question above is at least as hard as learning juntas. Indeed, a $\mathsf{poly}(s, n, 1/\epsilon)$ time algorithm for PAC learning s-sparse polynomials yields the optimal (up to polynomial factors) run time of $\mathsf{poly}(2^k, n, 1/\epsilon)$ for learning juntas. Agnostically learning sparse polynomials on the uniform distribution on $\{-1, 1\}^n$ is at least as hard as the problem of PAC learning DNFs with respect to the uniform distribution on $\{-1, 1\}^n$, a major open question in learning theory.

On the other hand, a surprising recent result by Andoni et. al.[1] shows that it is possible to learn sparse polynomials in the presence of *random* additive Gaussian noise with respect to the Gaussian distribution (as opposed to the agnostic setting where the noise is adversarial). Given the results of Andoni. etl. al. [1], a natural question is if the agnostic version of the question is any easier with respect to the Gaussian distribution. We give a negative answer to this question:

▶ **Theorem 2** (See Theorem 10 for details)**.** *If Assumption 1 is true, then, there is no algorithm running in time* $\mathsf{poly}(n, s, 2^d, 1/\epsilon)$ *to agnostically learn* s*-sparse degree* d *polynomials from random examples on the Gaussian distribution.*

A subroutine to find heavy Fourier coefficients of any function f on $\{-1, 1\}^n$ is an important primitive in learning algorithms and the problem happens to be just as hard as agnostic learning sparse polynomials described above. On the Gaussian distribution, Fourier-Transform based methods employ what is known as the *Hermite* transform [22, 28].

We show that the problem of finding heavy Hermite coefficients of a function on $\mathbb{R}^n$ from random examples is no easier than its analog on the cube. In particular, we give a reduction from the problem of PAC learning DNF formulas on the uniform distribution, to the problem of finding heavy Hermite coefficients of a function on $\mathbb{R}^n$ given random examples labeled by it. It is possible to derive this result by using the reduction of Feldman et. al.[13] who reduce PAC learning DNF formulas on the uniform distribution to sparse LPN by combining it with our reduction from sparse LPN to agnostic learning of sparse polynomials. However, we give a simple direct proof based on the properties of Fourier spectrum of DNF formulas due to [21].

To complement this negative result, we show that the problem becomes tractable if we are allowed the stronger value query access to the target function, in that, the learner can query any point of its choice and obtain the value of the target at the point from the oracle. On the uniform distribution on the hypercube, with query access to the target function, the task of agnostic learning s-sparse polynomials can in fact be performed in polynomial time in $s, n, 1/\epsilon$ using the well known Kushilevitz-Mansour (KM) algorithm [32]. The KM algorithm can be equivalently seen as a procedure to find the large Fourier coefficients of a function given query access to it. We show (in Appendix A) that it is possible to extend the KM algorithm to succeed in finding heavy Hermite coefficients.

▶ **Theorem 3.** *Given access to a queries from a function f such that $\mathbb{E}_{x\sim\gamma}[f(x)^2] = 1$, there is an algorithm that finds all the Hermite coefficients of f of degree d that are larger in magnitude than ϵ, in time $\mathsf{poly}(n, d, 1/\epsilon)$. Consequently, there exists an algorithm to agnostically learn s-sparse degree d polynomials on γ in time and queries $\mathsf{poly}(s, n, d, 1/\epsilon)$.*

1.3 Our Techniques

Our main result relates the hardness of agnostic learning of halfspaces on the Gaussian distribution to the hardness of learning sparse parities with noise on the uniform distribution on the hypercube (sparse LPN). The reduction involves embedding a set of labeled random examples on the hypercube into a set of labeled random examples on $\mathbb{R}^n$ such that the marginal distribution induced on $\mathbb{R}^n$ is the Gaussian distribution. To do this, we define an operation that we call as the *Gaussian lift* of a function, that takes an example label pair $(x, f(x))$ with $x \in \{-1, 1\}^n$ and produces $(z, f^\gamma(z))$ where z is distributed according to the Gaussian distribution if x is distributed according to the uniform distribution on $\{-1, 1\}^n$. We refer to the function f^γ as the *Gaussian lift* of f.

We show that given random examples labeled by f from the uniform distribution on $\{-1, 1\}^n$, one can generate random examples labeled by f^γ whose marginal distribution is the Gaussian. Further, we show how to recover a hypothesis close to f from a hypothesis close to f^γ. When f is a parity function, f^γ will be noticeably correlated with some halfspace. We show that the correlation is in fact exponentially small in n (but still enough to give us our hardness results) and requires a delicate computation which we accomplish looking at it as a limit of a quantity that can be estimated accurately. We then implement a similar idea for reducing sparse LPN to agnostically learning sparse polynomials on $\mathbb{R}^n$ under the Gaussian distribution by proving that the Gaussian lift of the parity function χ^γ is correlated with a monomial on $\mathbb{R}^n$ with respect to the Gaussian distribution.

We note that when allowed query access to the target function on $\{-1, 1\}^n$, one can extend the well known KM algorithm [32] to find heavy Hermite coefficients of any function on $\mathbb{R}^n$, given query access to it. The main difference in this setting is the presence of higher degree terms in Hermite expansion (as against only multilinear terms in the Fourier expansion).

1.4 Related Work

We survey some algorithms and lower bounds for the problem of agnostically learning halfspaces here. As mentioned before, [22], gave agnostic learning algorithms for halfspaces by assuming that the distribution is product Gaussian. They showed that their algorithm can be made to work under the more challenging log-concave distributions in polynomial time for any constant error. This result was recently improved by [23]. [35] gave a polynomial time algorithm for the problem under *large margin assumptions* on the underlying distribution. Following this, [4] gave a trade-off between time and accuracy in the large margin framework.

In addition to the representation-independent hardness results mentioned before, there is a line of work that shows *proper* hardness of agnostically learning halfspaces on arbitrary distributions via a reduction from hard problems in combinatorial optimization. [19] show that it is NP hard to properly (i.e. the hypothesis is restricted to be a halfspace) agnostically learn halfspaces on arbitrary distributions. Extending this result, [11] show that it is impossible to give an agnostic learning algorithm for halfspaces on arbitrary distribution that returns a polynomial threshold function of degree 2 as the hypothesis, unless P = NP.

2 Preliminaries

In this paper, we will work with functions that take both real and Boolean values (i.e. in $\{-1,1\}$) on the n-dimensional hypercube $\{-1,1\}^n$ and $\mathbb{R}^n$. For an element $x \in \{-1,1\}^n$, we will denote the coordinates of x by x_i. Let $\gamma = \gamma_n$ be the standard product Gaussian distribution on $\mathbb{R}^n$ with mean 0 and variance 1 in every direction and $\mathcal{U} = \mathcal{U}_n$, the uniform distribution on $\{-1,1\}^n$. We define the *sign* function on $\mathbb{R}$ as $\mathsf{sign}(x) = x/|x|$ for every $x \neq 0$. Set $\mathsf{sign}(0)$ to be 0. For $z \in \{-1,1\}^n$, the weight of z is the translated Hamming weight (to account for our bits being in $\{-1,1\}$) and is denoted by $|z| = \frac{1}{2}\sum_{i \in n} z_i + n/2$. For vectors $z \in \{-1,1\}^n$ and $y \in \mathbb{R}^n_+$, let $z \circ y$ denote the vector t such that $t_i = z_i \cdot y_i$.

A *half normal* random variable is distributed as the absolute value of a univariate gaussian random variable with mean zero and variance 1. We denote the distribution of a half normal random variable by $|\gamma|$. As is well known, $\mathbb{E}_{x \sim |\gamma|}[x] = \sqrt{2/\pi}$ and $Var[|\gamma|] = (1 - 2/\pi)$.

The *parity* function $\chi_S : \{-1,1\}^n \to \{-1,1\}$, for any $S \subseteq [n]$, is defined by $\chi_S(x) = \Pi_{i \in S} x_i$ for any $x \in \{-1,1\}^n$. For any $S \subseteq [n]$, the *majority* function MAJ_S is defined by $Maj_S(x) = \mathsf{sign}(\sum_{i \in S} x_i)$. The input x in the current context will come either from $\{-1,1\}^n$ or $\mathbb{R}^n$. When $S = [n]$, we will drop the subscript and write χ and MAJ for $\chi_{[n]}$ and $\mathsf{MAJ}_{[n]}$ respectively. The class of halfspaces is the class of all Boolean valued functions computed by expressions of the form $\mathsf{sign}(\sum_{i \in [n]} a_i \cdot x_i)$ for coefficients $a_i \in \mathbb{R}$ for each $1 \leq i \leq n$. The inputs to a halfspace can come from both $\{-1,1\}^n$ and $\mathbb{R}^n$.

For a probability distribution $\mathcal{D}$ on X ($\mathbb{R}^n$ or $\{-1,1\}^n$) and any functions $f, g : X \to \mathbb{R}$ such that $\mathbb{E}_{x \sim \mathcal{D}}[f(x)^2], \mathbb{E}_{x \sim \mathcal{D}}[g(x)^2] < \infty$, let $\langle f, g \rangle_{\mathcal{D}} = \mathbb{E}_{x \sim \mathcal{D}}[f(x) \cdot g(x)]$. The ℓ_1 and ℓ_2 norms of f w.r.t $\mathcal{D}$ are defined by $||f||_1 = \mathbb{E}_{x \sim \mathcal{D}}[|f(x)|]$ and $||f||_2 = \sqrt{\mathbb{E}_{x \sim \mathcal{D}}[f(x)^2]}$, respectively. We will drop the subscript in the notation for inner products when the underlying distribution is clear from the context.

Fourier Analysis on $\{-1,1\}^n$: Parity functions for each $\alpha \subseteq [n]$ form an orthonormal basis for the linear space of all real valued square summable functions on the uniform distribution on $\{-1,1\}^n$ (denoted by $L^2(\{-1,1\}^n, \mathcal{U})$). The (real) coefficients of the linear combination are referred to as the Fourier coefficients of f. For $f : \{-1,1\}^n \to \mathbb{R}$ and $\alpha \subseteq [n]$, the *Fourier coefficient* $\hat{f}(\alpha)$ is given by $\hat{f}(\alpha) = \langle f, \chi_\alpha \rangle = \mathbb{E}[f(x)\chi_\alpha(x)]$. The cardinality of the index set α is said to be the *degree* of the Fourier coefficient $\hat{f}(\alpha)$. The *Fourier*

expansion of f is given by $f(x) = \sum_{\alpha \subseteq [n]} \hat{f}(\alpha)\chi_\alpha(x)$. Finally, we have *Plancherel's theorem*: $\langle f, g\rangle_{\mathcal{U}} = \sum_{\alpha\subseteq[n]} \hat{f}(\alpha)\cdot\hat{g}(\alpha)$ and $||f||_2^2 = \sum_{\alpha\subseteq[n]}\hat{f}(\alpha)^2$ for any $f \in L^2(\{-1,1\}^n, \mathcal{U})$.

It is possible to exactly compute the Fourier coefficients of the Majority function $\mathsf{MAJ}_{[n]} = \mathsf{MAJ}$ on $\{-1,1\}^n$. We refer the reader to the online lecture notes of O'Donnell (Theorem 16, [33]).

▶ **Fact 1.** *Let* $\widehat{\mathsf{MAJ}}(\alpha)$ *be the Fourier coefficients of the majority function* $\mathsf{MAJ} = \mathsf{MAJ}_{[n]}$ *at index set* α *of cardinality* a *on* $\{-1,1\}^n$ *for an odd* n*. As* MAJ *is a symmetric function, the values of the coefficients depend only on the cardinality of the index set* a*. As* MAJ *is an odd function,* $\widehat{\mathsf{MAJ}}(\alpha) = 0$ *if* $|\alpha| = a$ *is even. For odd* a*:*

$$\widehat{\mathsf{MAJ}}(\alpha) = (-1)^{\frac{a-1}{2}} \cdot \frac{\binom{\frac{n-1}{2}}{\frac{a-1}{2}}}{\binom{n-1}{a-1}} \cdot \frac{2}{2^n} \cdot \binom{n-1}{\frac{n-1}{2}}.$$

In particular, $\widehat{\mathsf{MAJ}}(\alpha) = \sqrt{\frac{2}{n\pi}}$ *if* $a = 1$*.*

Hermite Analysis on $\mathbb{R}^n$: Analogous to the parity functions, the *Hermite polynomials* form an orthonormal and complete basis for $L^2(\mathbb{R}^n, \gamma_n)$, the linear space of all square integrable functions on $\mathbb{R}^n$ with respect to the spherical gaussian distribution $\gamma_n = \gamma$. These polynomials can be constructed in the univariate ($n = 1$) case by applying Gram Schmidt process to the family $\{1, x, x^2, \ldots\}$ giving the first few members as $h_0(x) = 1$, $h_1(x) = x$, $h_2(x) = \frac{x^2-1}{\sqrt{2}}$, $h_3(x) = \frac{x^3-3x}{\sqrt{6}}$, $\cdots$. The multivariate Hermite polynomials are obtained by taking products of univariate Hermite polynomials in each coordinate. Thus, for every n-tuple of non-negative integers $\Delta = (d_1, d_2, \ldots, d_n) \in \mathbb{Z}^n$, we have a polynomial $H_\Delta = \Pi_{i\in[n]} h_{d_i}(x_i)$. As γ_n is product and h_{d_i} are each orthonormal, H_Δ so constructed are clearly an orthonormal family of polynomials.

Analogous to the Fourier expansion, any function $f \in L^2(\mathbb{R}^n, \gamma_n)$ can be written uniquely as $\sum_{D\in\mathbb{Z}^n} \hat{f}(\Delta)\cdot H_\Delta$, where $\hat{f}(\Delta)$ is the *Hermite coefficient* of f at index Δ and is given by $\hat{f}(\Delta) = \mathbb{E}_{x\sim\gamma}[f(x)\cdot H_\Delta(x)]$. We have the Plancherel's theorem: $\langle f, g\rangle_\gamma = \sum_{\Delta\subseteq\mathbb{Z}^n} \hat{f}(\Delta)\cdot\hat{g}(\Delta)$ and $||f||_2^2 = \sum_{\Delta\subseteq\mathbb{Z}^n}\hat{f}(\Delta)^2$ for any $f \in L^2(\mathbb{R}^n, \gamma)$.

Agnostic Learning: The agnostic model of learning [20, 24] is a challenging generalization Valiant's PAC model of supervised learning that allows adversarial noise in the labeled examples. Given labeled examples from an arbitrary target function p, the job of an agnostic learner for a class $\mathcal{C}$ of real (or Boolean) valued functions is to produce a hypothesis h that has an error w.r.t p that is at most ϵ more than that of best fitting hypothesis from the class $\mathcal{C}$. Formally, we have:

▶ **Definition 4** (Agnostic learning with ℓ_1 error)**.** Let $\mathbb{F}$ be a class of real-valued functions with distribution $\mathcal{D}$ on X (either $\{-1,1\}^n$ or $\mathbb{R}^n$). For any real valued target function p on X, let $\mathrm{opt}(p, \mathbb{F}) = \inf_{f\in\mathbb{F}} \mathbb{E}_{x\sim\mathcal{D}}[|p(x) - f(x)|]$. An algorithm $\mathcal{A}$, is said to agnostically learn $\mathbb{F}$ on $\mathcal{D}$ if for every $\epsilon > 0$ and any target function p on X, given access to random examples drawn from $\mathcal{D}$ and labeled by p, with probability at least $\frac{2}{3}$, $\mathcal{A}$ outputs a hypothesis h such that $\mathbb{E}_{(x,y)\sim\mathsf{P}}[|h(x) - p(x)|] \leq \mathrm{opt}(p, \mathbb{F}) + \epsilon$.

The ℓ_1 error for real valued functions specializes to the disagreement (or Hamming) error for Boolean valued functions and thus the definition above is a generalization of agnostic learning a class of Boolean valued functions on a distribution. A general technique (due to [22]) for agnostic learning $\mathcal{C}$ on any distribution $\mathcal{D}$ is to show that every function in $\mathcal{C}$ is

approximated up to an ℓ_1 error of at most ϵ by a polynomial of low-degree $d(n, \epsilon)$, which can then be constructed using ℓ_1-polynomial regression. This approach to learning can equivalently be seen as learning based on Empirical Risk Minimization with absolute loss [38]. As observed (in [22]), since ℓ_1 error for Boolean valued functions is equivalent to the disagreement error, polynomial regression can also be used to agnostically learn Boolean valued function classes w.r.t disagreement error.

3 Hardness of Agnostically Learning Halfspaces on the Gaussian Distribution

In this section, we show that any algorithm that agnostically learns the class of halfspaces on the Gaussian distribution with an error of at most ϵ takes time $n^{\Omega(\log(1/\epsilon))}$. In particular, there is no fully polynomial time algorithm to agnostically learn halfspaces on the Gaussian distribution (subject to the hardness of sparse LPN). We reduce the problem of learning sparse parities with noise on the uniform distribution on the Boolean hypercube to the problem of agnostic learning halfspaces on the Gaussian distribution to obtain our hardness result.

Our approach is a generalization of the one adopted by [22] who used such a reduction to show the optimality of their agnostic learning algorithm for halfspaces on the uniform distribution on $\{-1, 1\}^n$. We begin by briefly recalling their idea here: Let χ_S be the unknown parity for some $S \subseteq [n]$. Observe that on the uniform distribution on $\{-1, 1\}^n$, χ_S is correlated with the majority function MAJ_S with a correlation of $\approx 1/\sqrt{|S|} \geq 1/\sqrt{n}$. Thus, the expected correlation between MAJ_S and the noisy labels is $\approx \eta/\sqrt{n}$ where η is the noise rate. In other words, MAJ_S predicts the value of the label at a uniformly random points from $\{-1, 1\}^n$ with probability $\approx 1/2 + \eta/\sqrt{n}$ (i.e. with an inverse polynomial advantage over random). The key idea here is to note that if we drop a coordinate, say $j \in S$ (i.e. a "relevant" variable for the unknown parity) from every example point to obtain labeled examples from $\{-1, 1\}^{n-1}$, then, the labels and example points are independent as random variables and thus no halfspace can predict the labels to an inverse polynomial advantage. On the other hand, if we drop a coordinate $j \notin S$, then, the labels are still correlated with the correct parity and thus, MAJ_S predicts the labels with an inverse polynomial advantage. Thus, drawing enough examples can allow us to distinguish between the two cases and construct S one variable at a time.

Such a strategy, however, cannot be directly applied to relate learning problems on *different* distributions. Instead, we show that given examples from $\{-1, 1\}^n$ labeled by some function f, we can simulate examples drawn according to the Gaussian distribution, labeled by some $f^\gamma : \mathbb{R}^n \to \{-1, 1\}$ (which we call as the *Gaussian lift* of f). Further, we show that when f is some parity χ_α, then, f^γ is noticeably correlated with some halfspace on the Gaussian distribution. Now, given examples drawn according to the Gaussian distribution, labeled by some f^γ, one can use the agnostic learner for halfspaces to recover α with high probability. We now proceed with the details of our proof. We first define the Gaussian lift of any function $f : \{-1, 1\}^n \to \mathbb{R}$. At any $x \in \mathbb{R}^n$, f^γ returns a value obtained by evaluating f at the point associated with the sign pattern of x.

▶ **Definition 5** (Gaussian Lift). The Gaussian lift of a function $f : \{-1, 1\}^n \to \mathbb{R}$ is a function $f^\gamma : \mathbb{R}^n \to \mathbb{R}$ such that for any $x \in \mathbb{R}^n$, $f^\gamma(x) = f(\mathsf{sign}(x_1), \mathsf{sign}(x_2), \ldots, \mathsf{sign}(x_n))$.

We begin with a general reduction from the problem of learning k-sparse parity with noise on the uniform distribution on $\{-1, 1\}^n$ to problem of learning any class $\mathcal{C}$ of functions

on $\mathbb{R}^n$ agnostically on the Gaussian distribution. This reduction works under the assumption that for every $S \subseteq [n]$, the Gaussian lift of the parity function on $\alpha \subseteq [n]$, denoted as χ_α^γ is noticeably correlated with some function from $c_\alpha \in \mathcal{C}$.

▶ **Lemma 6** (Correlation Lower Bound yields Reduction to SLPN). *Let $\mathcal{C}$ be a class of Boolean valued functions on $\mathbb{R}^n$ such that for every $\alpha \subseteq [n]$ for $|\alpha| \leq k$, there exists a function $c_\alpha \in \mathcal{C}$ such that $\langle c_\alpha, \chi_\alpha^\gamma \rangle \geq \theta(k)$and c_α depends on variables only in α.*

Suppose there exists an algorithm $\mathcal{A}$ (that may not be proper *and can output real valued hypotheses) to learn $\mathcal{C}$ agnostically over the Gaussian distribution to an ℓ_1 error of at most ϵ using time and samples $T(n, 1/\epsilon)$. Then, there exists an algorithm to solve SLPN that runs in time and examples $\tilde{O}(\frac{n}{(1-2\eta)\theta(k)}) + \tilde{O}(n) \cdot T(n, \frac{2}{(1-2\eta)\theta(k)})$ where η is the noise rate.*

Proof. We will assume that $\mathcal{C}$ is negation closed, that is, for every $c \in \mathcal{C}$, the function $-c \in \mathcal{C}$. This assumption can be easily removed by running the procedure described below twice, the second time with the labels of the examples negated. We skip the details of this easy adjustment here. Let χ_β be the target parity for some $\beta \subseteq [n]$ such that $|\beta| \leq k$. We claim that the following procedure determines if $j \in \beta$ for any $j \in [n]$ given noisy examples from χ_β with high probability.

1. For each example-label pair $(x, y) \in \{-1, 1\}^n \times \{-1, 1\}$, generate a new example label pair as follows.
   a. Draw independent half-normals $h_1, h_2, ..., h_n$.
   b. Let $z \in \mathbb{R}^{n-1}$ be defined so that $z_i = x_i \cdot h_i$ for each $i \in [n]$, $i \neq j$.
   c. Output (z, y) where $z = (z_1, z_2, \cdots z_{j-1}, z_{j+1}, \cdots, z_n) \in \mathbb{R}^{n-1}$. Denote the distribution of (z, y) by $\mathcal{D}_j$.
2. Set $\epsilon = (1 - 2\eta) \cdot \theta(k)$. Collect a set of $T(n, 1/\epsilon)$ examples, R, output by the procedure above.
3. Run $\mathcal{A}$ on R with ℓ_1 error parameter ϵ set to $(1 - 2\eta)\theta(k)/2$. Let h be the output of the algorithm.
4. Draw a set fresh set of $r = O(\log(1/\delta)/\epsilon^2)$, $\{(z^1, y^1), (z^2, y^2), \ldots, (z^r, y^r)\}$, again by the procedure above and estimate $err = \frac{1}{r} \cdot \sum_{i=1}^{r} [|h(z^i) - y^i|]$. Accept i as relevant if $err \leq 1 - \epsilon/4$. Else reject.

We now argue the correctness of this procedure. For D_j described above (obtained by dropping the j^{th} coordinate in the lifted examples), it is easy to see that the marginal distribution on the first $n - 1$ coordinates is γ_{n-1}, the spherical Gaussian distribution on $n - 1$ variables. Set $\epsilon = (1 - 2\eta) \cdot \theta(k)$.

Suppose $j \notin T$. In this case, for any example (z, y), $y = \chi_\beta^\gamma(z)$ with probability $1 - \eta$ independently of other examples. We know that there exists a $c_\beta \in \mathcal{C}$, depending only on coordinates in β such that $\langle c_\beta, \chi_\beta^\gamma \rangle \geq \theta(k)$. Thus, $\mathbb{E}_{(z,y)\sim\mathcal{D}_j}[c_\beta(z) \cdot y] = \epsilon$. In this case, thus, running $\mathcal{A}$ with the error parameter ϵ obtains $h : \{-1, 1\}^{n-1} \to \mathbb{R}$ with error at most

$$\begin{aligned}\mathbb{E}_{(z,y)\sim\mathcal{D}_j}[|c_\beta - y|] + \epsilon &= (1 - \eta) \cdot \mathbb{E}_{(z,y)\sim\mathcal{D}_j}[|c_\beta(z') - \chi_\beta^\gamma(z')|] + \\ &\qquad \eta \cdot \mathbb{E}_{(z,y)\sim\mathcal{D}_j}[|c_\beta(z') - \chi_\beta^\gamma(z')|]] + \epsilon \\ &= 1 - \epsilon/2.\end{aligned}$$

On the other hand, if $j \in T$, then, since the procedure drops the j^{th} coordinate of every example, the distribution of the labels y is uniformly random and independent of the distribution of the coordinates z_i, $i \neq j$. In this case, for any function in $h : \{-1, 1\}^{n-1} \to \mathbb{R}$, it can be easily checked that $\mathbb{E}[|c(z) - y|] \geq 1$ where the expectation is over the random variables (z, y).

We can estimate the ℓ_1 error $\mathbb{E}_{(z,y)\sim\mathcal{D}_j}[|y-h(z)|]$ of the hypothesis h produced by the algorithm, to an accuracy of $\epsilon/4$ with confidence $1-\delta$ using $r = O(\frac{\log(1/\delta)}{(1-2\eta)\theta(k)})$ examples. This is enough to distinguish between the two cases above. We can now repeat this procedure n times, once for every coordinate $j \in [n]$. Using a union bound, all random estimations and runs of $\mathcal{A}$ are successful with probability at least $2/3$ using an additional poly-logarithmic cost in n in time and samples required. Thus we obtain the stated running time and sample complexity. ◀

Next, we will show that χ_S^γ, the Gaussian lift of the parity function the subset $S \subseteq [n]$ is noticeably correlated with the majority function $\mathsf{MAJ}_S = \mathsf{sign}(\sum_{i\in S} x_i)$ with respect to the Gaussian distribution on $\mathbb{R}^n$. This correlation, while enough to yield the hardness result for agnostic learning of halfspaces when combined with Lemma 6, is an exponentially small quantity, in sharp contrast to the correlation between MAJ_S and χ_S on the uniform distribution on the hypercube (where it is $\approx 1/\sqrt{|S|}$). We thus need to adopt a more delicate method of estimating it as a limit of a quantity we can estimate accurately.

▶ **Lemma 7.** *Let m be an odd integer and consider $S \subseteq [n]$ such that $|S| = m$. Then,*

$$|\langle \mathsf{MAJ}_S, \chi_S^\gamma \rangle_\gamma| = 2^{-\Theta(m)}$$

Proof. Let $c = |\mathbb{E}_{x\sim\gamma_n}[\mathsf{MAJ}_S(x)\cdot\chi_S^\gamma(x)]|$. Each x_i above is independently distributed as $\mathcal{N}(0,1)$. Fix any odd integer t and define y_{ij} for each $1 \le i \le m$ and $1 \le j \le t$ to be uniform and independent random variables taking values in $\{-1,1\}$. The idea is to simulate each x_i by $\frac{1}{t}\sum_{j=1}^t y_{ij}$. In the limit as $t \to \infty$, the simulated random variable converges to in distribution to x_i. Call $f^t(x) = \mathsf{sign}(\sum_{i=1}^m \sum_{j=1}^t y_{ij})$ and $g^t(x) = \mathsf{sign}(\Pi_{i=1}^m \sum_{j=1}^t y_{ij})$, the functions obtained by applying the substitution above to MAJ_S and χ_S^γ respectively. Let $y = \{y_{ij} \in [m]\times[t]\}$ denote the inputs bits to f^t and g^t defined above. Thus,

$$c = \lim_{t\to\infty} \mathbb{E}[f^t \cdot g^t] = \lim_{t\to\infty} \mathbb{E}[\mathsf{sign}(\sum_{i=1}^m \sum_{j=1}^t y_{ij}) \cdot \Pi_{i=1}^m \mathsf{sign}(\sum_{j=1}^t y_{ij})] \tag{1}$$

Using Plancherel's Identity for the RHS above, we have:

$$c = \mathbb{E}[f^t \cdot g^t] = \sum_{\alpha\subseteq[m]\times[t]} \widehat{f^t}(\alpha)\cdot\widehat{g^t}(\alpha). \tag{2}$$

We now intend to estimate the RHS of the equation above. Towards this goal, we make some observations regarding the fourier coefficients $\widehat{f^t}(\alpha)$ and $\widehat{g^t}(\alpha)$.

▶ **Claim 1** (Fourier Coefficients of g^t). *For every $\alpha = \cup_{i=1}^m \alpha_i$ where $\alpha_i = \alpha \cap \{(i,j)|j\in[t]\}$ for each $1 \le i \le m$, $\widehat{g^t}(\alpha) = \Pi_{i=1}^m \widehat{\mathsf{MAJ}_{i\times[t]}}(\alpha_i)$.*

That is, the Fourier coefficient at α of g^t is the product of Fourier coefficients of majority functions at α_i, where the i^{th} majority function is on bits y_{ij} for $j \in [t]$.

Proof. $\widehat{g^t}(T) = \mathbb{E}[g^t(y)\cdot\chi_\alpha(y)] = \mathbb{E}[\Pi_{i=1}^m \mathsf{sign}(\sum_{j=1}^t y_{ij})\cdot\chi_\alpha(y) = \mathbb{E}[\Pi_{i=1}^m \chi_{\alpha_i}(y)\cdot\mathsf{sign}(\sum_{j=1}^t y_{ij})]$ $= \Pi_{i=1}^m \mathbb{E}[\mathsf{sign}(\sum_{j=1}^t y_{ij})\cdot\chi_{\alpha_i}(y)] = \Pi_{i=1}^m \widehat{\mathsf{MAJ}_{i\times[t]}}(\alpha_i)$, where for the third equality, we note that $\chi_\alpha = \Pi_{i=1}^m \chi_{\alpha_i}$ and for the last equality, the fact that y_{ij} are all independent and that α_i are disjoint. ◀

We now observe the term corresponding to each index α contributes a value with the same sign to the RHS of Equation (2).

▶ **Claim 2.** *Let $\alpha \subseteq [m] \times [t]$ and suppose $\widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha) \neq 0$. If $m = 4q+1$ for $q \in \mathbb{N}$, then, $\mathsf{sign}(\widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha)) = 1$. If $m = 4q+3$ for $q \in \mathbb{N}$ then, $\mathsf{sign}(\widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha)) = -1$.*

Proof of Claim. Set $m = 4q+1$, the other case is similar. Recall that t is odd. Let $|\alpha| = a$ for some odd a (otherwise at least one of α_i is even in which case $\hat{g^t}(\alpha) = 0$). From Fact 1: $\mathsf{sign}(\widehat{f^t}(\alpha)) = (-1)^{(a-1)/2}$.

Let $\alpha = \cup_{i=1}^{m} \alpha_i$ such that $\alpha_i \subseteq [m] \times [t]$ and let for each i, $|\alpha_i| = a_i$. Using the claim above, we have: $\mathsf{sign}(\widehat{g^t}(\alpha)) = \Pi_{i=1}^{m} \mathsf{sign}(\widehat{\mathsf{MAJ}_{i\times[t]}}(\alpha_i)) = \Pi_{i=1}^{m}(-1)^{(a_i-1)/2} = (-1)^{(a-m)/2}$. Thus, $\mathsf{sign}(\widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha)) = (-1)^{(2a-m-1)/2} = 1$. ◀

For the rest of the proof, assume that $m = 4q+1$. We are now in a position to analyze Equation (2). By Claim 2 above, we know that every term in the summation on the RHS of Equation (2) contributes a non-negative value. We group the Fourier coefficients of f and g based on the size of the index set and refer to the coefficients with index sets of size r by *layer* r. Observe that for any index set $\alpha \subseteq [m] \times [t] = \cup_{1 \leq i \leq m} \alpha_i$, if there is an i such that $\alpha_i = \emptyset$, then, $\widehat{g^t}(T) = 0$. Thus, the term corresponding to index α contributes 0 to the RHS of Equation (2). Thus, we can assume $|\alpha| \geq m$. We first estimate the contribution due to layer m:

▶ **Claim 3** (Contribution due to layer m)**.** *For large enough t,*

$$|\sum_{|\alpha|=m} \widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha)| = \Omega\left(1/\sqrt{m} \cdot (\frac{2}{\pi e})^{m/2}\right).$$

Proof of Claim. Recall that $\alpha = \cup_{i=1}^{m} \alpha_i$ with each $\alpha_i \subseteq i \times [t]$. By the discussion above, $|\alpha_i| = 1$. There are exactly t^m indices α that satisfy this condition.

Using Fact 1 we know that $\hat{f^t}(\alpha) = (-1)^{\frac{t-1}{2}} \cdot \frac{\binom{\frac{tm-1}{2}}{\frac{m-1}{2}}}{\binom{tm-1}{m-1}} \cdot \frac{2}{2^{tn}} \cdot \binom{tm-1}{\frac{tm-1}{2}}$. Using Fact 1 again, for g^t along with Claim 1, we have : $\hat{g^t}(T) = \left(\sqrt{\frac{2}{t\pi}}\right)^m$. Thus, each non-zero term in layer m of (2) contributes: $\widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha) = \frac{\binom{\frac{tm-1}{2}}{\frac{m-1}{2}}}{\binom{tm-1}{m-1}} \cdot \frac{2}{2^{tm}} \cdot \binom{tm-1}{\frac{tm-1}{2}} \cdot \left(\sqrt{\frac{2}{t\pi}}\right)^m$. Using asymptotically tight approximations for binomial coefficients, for large enough t:
$\frac{2}{2^{tm}} \cdot \binom{tm-1}{\frac{tm-1}{2}} = \Theta(\sqrt{\frac{1}{\pi \cdot (tm-1)}})$, and $\frac{\binom{\frac{tm-1}{2}}{\frac{m-1}{2}}}{\binom{tm-1}{m-1}} = \Omega((et)^{-\frac{m-1}{2}})$. Thus, the contribution to the RHS of Equation 2 by layer m asymptotically $\sum_{\alpha:|\alpha_i|=1} \widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha) = \Omega(t^m \cdot t^{-\frac{m-1}{2}} \cdot e^{-\frac{m-1}{2}} \cdot \sqrt{\frac{2}{\pi \cdot (tm-1)}} \cdot \left(\sqrt{\frac{2}{t\pi}}\right)^m = \Omega\left(\frac{1}{\sqrt{m}} \cdot (\frac{2}{\pi e})^{m/2}\right)$. ◀

The claim above is enough to give us a lower bound on c. Our aim in the following is to establish an inverse exponential upper bound on the correlation between MAJ_S and χ_S^γ. Together with the contribution due to layer m, we have that $c = 2^{-\Theta(m)}$. This will complete the proof.

▶ **Claim 4** (Contribution due to layers $r > m$)**.** *For large enough t, m,*

$$|\sum_{|\alpha|>m} \widehat{f^t}(\alpha) \cdot \widehat{g^t}(\alpha)| = 2^{-\Omega(m)}.$$

Proof of Claim. Let $r_i \geq 1$ for every $1 \leq i \leq m$ such that $\sum_{i \leq n} r_i = r$. Consider any $\alpha = \cup_{1 \leq i \leq m} \alpha_i$ such that $|\alpha_i| = r_i$. The number of indices α is $\Pi_{1 \leq i \leq m} \binom{t}{r_i} \leq t^r / \Pi_{i=1}^{n} r_i!$. If

any r_i is even, then, the coefficient $\widehat{g^t}(\alpha) = 0$. Thus, the only non-zero contribution to the correlation from layer r is due to the indices α such that all $|\alpha_i| = r_i$ are odd positive integers. Using Fact 1: $\widehat{f^t}(\alpha) = \frac{\binom{\frac{tm-1}{2}}{\frac{r-1}{2}}}{\binom{tm-1}{r-1}} \cdot \frac{1}{2^{tm-1}} \cdot \binom{tm-1}{\frac{tm-1}{2}}$, and $|\widehat{g^t}(\alpha)| \leq \Pi_{1\leq i\leq m} \frac{\binom{\frac{t-1}{2}}{\frac{r_i-1}{2}}}{\binom{t-1}{r_i-1}} \frac{1}{2^{t-1}} \cdot \binom{t-1}{\frac{t-1}{2}}$.

Let us estimate the sum squared of all coefficients $\widehat{g^t}(\alpha)$ such that $|\alpha_i| = r_i$ for each i. Recall that for the majority function on m bits, the sum squared of all coefficients of any layer q is $\approx (2/\pi)^{3/2} \cdot 1/q^{3/2}$. This can be derived directly using Fact 1 (see [33]).

$$\sum_{\alpha:|\alpha_i|=r_i} (\widehat{g^t}(\alpha))^2 \leq \sum_{\alpha:|\alpha_i|=r_i} \Pi_{i=1}^m (\sum_{|\alpha_i|=r_i} (\widehat{\mathsf{MAJ}_{i\times[t]}}(\alpha_i))^2) = \Pi_{i=1}^m (2/\pi)^{3/2} \cdot 1/r_i^{3/2}.$$

The maximum value over all $r_1, r_2, \ldots, r_m$ that give a non-zero $\widehat{g^t}(\alpha)$ (i.e. each r_i odd) of the expression on the RHS is: $(2/\pi)^{3m/2} \cdot (m/r)^{3/2} = 2^{-\Theta(m)} \cdot (m/r)^{3/2}$.

On the other hand, each coefficient of f^t of layer r is equal and the total sum squared of coefficients from layer r of f^t is at most $O(1/r^{3/2})$. Now, using Cauchy Schwarz inequality for the sum of product of fourier coefficients of f^t and g^t at indices corresponding to each valid partition, $r_1, r_2, \ldots, r_m$ of the integer $r > m$ and summing up over all valid partitions of r, the total contribution due to layer r to the correlation is at most: $2^{-\Theta(m)} \cdot 1/r^{3/2}$. Since $\sum_{r>m} 1/r^{3/2}$ converges, we have the claimed upper bound. ◀

◀

As an immediate corollary, we obtain the following hardness for the problem of agnostic learning of halfspaces on the Gaussian distribution.

▶ **Theorem 8** (Hardness of Agnostic Learning of Halfspaces). *Suppose there exists an algorithm $\mathcal{A}$ to learn the class of halfspaces agnostically over the Gaussian distribution to an error of at most ϵ that runs in time $T(n, 1/\epsilon)$. Then, there exists an algorithm to solve k-SLPN that runs in time $\tilde{O}(n \cdot T(n, \frac{2^{O(k)}}{(1-2\eta)})$ where η is the noise rate. In particular, if there is an algorithm that agnostically learns halfspaces on γ_n in time $n^{o(\log(1/\epsilon))}$ then there is an algorithm that solves SLPN for all parities of length $k = O(\log n)$ in time $n^{o(k)}$.*

For a proof, we use Lemma 6 with C as the class of all majorities of length k and note that $\theta(k) = 2^{-\Theta(k)}$.

3.1 Agnostically Learning Sparse Polynomials is Hard

We now reduce k-sparse LPN to agnostically learning degree k and 1-sparse polynomials on the Gaussian distribution and obtain that any algorithm to agnostically learn even a monomial of degree k up to any constant error on the Gaussian distribution runs in time $n^{\Omega(k)}$. We note that the polynomial regression algorithm [22] can be used to agnostically learns degree k polynomials to an accuracy of ϵ in time $n^{O(k)} \cdot \mathsf{poly}(1/\epsilon)$. Thus, our result shows that this running time cannot be improved (assuming that sparse LPN is hard). For a proof, we observe that the Gaussian lift of the parity function χ^γ is noticeably correlated with a sparse polynomial (in fact, just a monomial) on $\mathbb{R}^n$ under the Gaussian distribution. We then invoke Lemma 6 to complete the proof.

▶ **Lemma 9** (Correlation of χ_S^γ with monomials). *Let $M_S : \mathbb{R}^n \to \mathbb{R}$ be the monomial $M_S(x) = \Pi_{i\in S} x_i$. For $\chi_S^\gamma : \mathbb{R}^n \to \{-1, 1\}$, the Gaussian lift of the the parity on $S \subseteq [n]$, we have: $\mathbb{E}_{x\sim\gamma}[\chi_S^\gamma(x) \cdot M_S(x)] = (\frac{2}{\pi})^{|S|/2}$.*

Proof. $\mathbb{E}_{x\sim\gamma}[\chi_S^\gamma(x)\cdot M_S(x)] = (\mathbb{E}_{x_i\sim\gamma}[\mathsf{sign}(x_i)\cdot x_i])^{|}S| = (\mathbb{E}_{z\sim|\gamma|}[z])^{|S|} = (2/\pi)^{|S|/2}$. ◀
Using Lemma 6, we thus have:

▶ **Theorem 10** (Sparse Parity to Sparse Polynomials)**.** *If there is an algorithm to agnostically learn 1-sparse, degree k polynomials on the Gaussian distribution in time $T(n,k,1/\epsilon)$, then, there is an algorithm to solve k-SLPN in time $\tilde{O}(n)\cdot T(n,k,2^{O(k)}/(1-2\eta))$. In particular, if Assumption 1 is true, then any algorithm to agnostically learn degree k monomials up to any constant error runs in time $n^{\Omega(k)}$.*

4 Hardness of Finding Heavy Hermite Coefficients

In this section, we show that a polynomial time algorithm to find all large Hermite coefficients of any function f on the Gaussian distribution using random examples gives a PAC learning algorithm for DNF formulas on the uniform distribution on $\{-1,1\}^n$. The idea is to use the subroutine that recovers large *Hermite coefficients* of a function using random labeled examples to find heavy *Fourier coefficients* of functions on the uniform distribution (via the Gaussian lift) on the hypercube using random examples. Our reduction will then be completed using the properties of the Fourier spectrum of DNF formulas due to [21] (similar to the one used by [13]). Observe that given query access, finding heavy Fourier coefficients on the uniform distribution on $\{-1,1\}^n$ is easy and the reduction yields us a subroutine to find heavy Fourier coefficients by random examples alone.

▶ **Lemma 11.** *Suppose there is an algorithm $\mathcal{A}$, that, for $\epsilon > 0$, uses random examples drawn according to the spherical Gaussian distribution and labeled by an unknown $f : \mathbb{R}^n \to \{-1,1\}$ and returns (with probability at least $2/3$) the Hermite coefficients of f, that are at least ϵ in magnitude in time and samples $T(n,1/\epsilon)$.*

Then, there exists an algorithm, that uses random example access to a Boolean function $g : \{-1,1\}^n \to \{-1,1\}$ on the uniform distribution on $\{-1,1\}^n$, and returns (with probability at least $2/3$), every Fourier coefficient of g of total degree at most d and magnitude at least ϵ, in time and samples $T(n,(2/\pi)^{d/2}/\epsilon)$.

Proof. Given access to random labeled examples from the uniform distribution on $\{-1,1\}^n$ and labeled by a function g, we construct an algorithm $\mathcal{A}'$ which runs $\mathcal{A}$ on a examples labeled by the Gaussian lift, g^γ of g and recovers large Fourier coefficients of g from the set of large Hermite coefficients of g^γ. As before, to simulate a random examples from g^γ we do the following:

1. Draw a random example $(x, g(x))$ where $x \in \{-1,1\}^n$ is uniformly distributed.
2. Draw $y_1, y_2, \cdots, y_n$ as independent half-normals induced by unit variance, zero mean Gaussian.
3. Return $(x \circ y, g(x))$.

Notice that $x \circ y$ is distributed according to the Gaussian distribution. Further,

$$g^\gamma(x) = g(\mathsf{sign}(x_1), \mathsf{sign}(x_2), \cdots, \mathsf{sign}(x_n))$$

for each $x \in \mathbb{R}^n$. Let $\Delta \in \{0,1\}^n \subseteq \mathbb{Z}^n$ (i.e., Δ is an index of a *multilinear* Hermite coefficient). Thus, Δ corresponds to a subset $\beta \subseteq [n]$ such that $|\beta| = d$. We will now show that: $|\widehat{g^\gamma}(\Delta)| \geq (2/\pi)^{-d/2}\epsilon$. We can then run $\mathcal{A}$ to find all Hermite coefficients of g^γ of magnitude at least $(2/\pi)^{-d/2}$, collect all multilinear coefficients of degree at most d and return the corresponding index sets as the indices of the Fourier coefficients of f of magnitude at least ϵ and degree at most d. The Fourier coefficients of g at the indices returned can

then be efficiently computed by taking enough random samples and computing the empirical correlations. This will complete the proof.

For this purpose, note that, being a function on $\{-1,1\}^n$, $g = \sum_{\alpha\subseteq[n]} \hat{g}(\alpha)\cdot\Pi_{i\in\alpha}x_i$. Thus,

$$g^\gamma(x) = g(\mathsf{sign}(x_1),\mathsf{sign}(x_2),\ldots,\mathsf{sign}(x_n)) = \sum_{\alpha\subseteq[n]} \hat{g}(\alpha)\cdot\Pi_{i\in\alpha}\mathsf{sign}(\Pi_{i\in\alpha}x_i).$$

We now have: $\widehat{g^\gamma}_S = \mathbb{E}_{x\sim\gamma_n}[g^\gamma(x)\cdot H_S(x)] = \sum_{T\subseteq[n]}\hat{g}(T)\cdot\mathbb{E}_{x\sim\gamma_n}[\mathsf{sign}(\Pi_{i\in T}x_i)H_S(x)]$. For any $\alpha\neq\beta$ (the subset of $[n]$ corresponding to Δ), then, $\mathbb{E}_{x\sim\gamma_n}[\mathsf{sign}(\Pi_{i\in\alpha}x_i)H_\Delta(x)] = 0$. Thus, using independence of x_i for each $i\in[n]$ and that $\mathbb{E}[|x_i|] = \sqrt{2/\pi}$, we have:

$$\begin{aligned}\mathbb{E}_{x\sim\gamma_n}[g^\gamma(x)\cdot H_\Delta(x)] &= \hat{g}(\beta)\mathbb{E}_{x\sim\gamma_n}[\mathsf{sign}(\Pi_{i\in\beta}x_i)\cdot\Pi_{i\in\beta}x_i]\\ &= \hat{g}(\beta)\Pi_{i\in\beta}\mathbb{E}[|x_i|] = \hat{g}(\beta)(2/\pi)^{-|\beta|/2}.\end{aligned}$$ ◀

We first describe the main idea of the proof: We are given random examples drawn from $\{-1,1\}^n$ and labeled by some function f. We simulate the examples from the Gaussian lift f^γ by embedding the examples from $\{-1,1\}^n$ into $\mathbb{R}^n$ using half-normals as before. We then argue that if $\hat{f}(S)$ is large in magnitude, then so is the multilinear Hermite coefficient at S of f^γ. Thus finding heavy Hermite coefficients of f^γ gives us the indices of large Fourier coefficients of f, which can then be estimated by random sampling. We now provide the details, which are standard and based on [21]. We need the following lemma due to Jackson [21] (we actually state a slightly refined version due to Bshouty and Feldman [7]). In the following, we abuse the notation a little bit and use $\mathcal{D}$ to also refer to the PDF of the distribution denoted by $\mathcal{D}$.

▶ **Lemma 12.** *For any Boolean valued function $f:\{-1,1\}^n\to\{-1,1\}$ computed by a DNF formula of size s, and any distribution $\mathcal{D}$ over $\{-1,1\}^n$, there is $\alpha\subseteq[n]$ such that $|\alpha|\leq\log(2s+1)\cdot||2^n\cdot\mathcal{D}||_\infty$ and $|\hat{f}(\alpha)|\geq\frac{1}{s+1}$.*

On the uniform distribution, the lemma above directly yields a weak learner for DNF formulas. Jackson's key idea here is to observe that learning f on $\mathcal{D}$ is same as learning $2^n f.\mathcal{D}$ on the uniform distribution. Coupled with a boosting algorithm [16, 17, 29] that uses only the distributions for which $||2^n\mathcal{D}||_\infty$ is small ($\mathsf{poly}(1/\epsilon)$), one obtains the PAC learner for DNF formulas.

▶ **Theorem 13.** *If there is an algorithm to find Hermite coefficients of magnitude at least ϵ, of a function $f:\mathbb{R}^n\to\mathbb{R}$ on the Gaussian distribution from random labeled examples in time $\mathsf{poly}(n,1/\epsilon)$, then there is an algorithm to PAC learn DNF formulas on the uniform distribution in polynomial time.*

5 Conclusion and Open Problems

In this paper, we described a general method to embed hard learning problems on the discrete hypercube into the spherical Gaussian distribution on $\mathbb{R}^n$. Using this technique, we showed that any algorithm to agnostically learn the class of halfspaces on the Gaussian distribution runs in time $n^{\Omega(\log(1/\epsilon))}$. We also ruled out a fully polynomial algorithm to agnostically learn sparse polynomials on $\mathbb{R}^n$ complementing the result of Andoni et al. [1] who gave a polynomial time algorithm for learning the class with random additive Gaussian noise.

On the other hand, as described before, the fastest algorithm for agnostically learning halfspaces runs in time $n^{O(1/\epsilon^2)}$ [9]. Thus, an outstanding open problem is to close the gap between these two bounds. That is:

▶ **Question 3.** *What is the optimal time complexity for agnostically learning halfspaces on the Gaussian distribution? In particular, is there an algorithm that agnostically learns halfspaces on the Gaussian distribution in time* $n^{O(\log(1/\epsilon))}$*?*

Acknowledgement. We thank Chengang Wu for numerous discussions during the preliminary stages of this work. We thank the anonymous reviewers for pointing out the typos in a previous version of this paper.

References

1 Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning sparse polynomial functions. In *SODA*, 2014.
2 Shai Ben-David and Hans-Ulrich Simon. Efficient learning of linear perceptrons. In *NIPS*, pages 189–195, 2000.
3 Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *COLT*, pages 1046–1066, 2013.
4 Aharon Birnbaum and Shai Shalev-Shwartz. Learning halfspaces with the zero-one loss: Time-accuracy tradeoffs. In *NIPS*, pages 935–943, 2012.
5 A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.
6 Avrim Blum, Alan M. Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1998.
7 Nader H. Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2:359–395, 2002.
8 Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. *CoRR*, abs/1311.2272, 2013.
9 Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. *CoRR*, abs/0911.3389, 2009.
10 Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *SODA*, pages 1590–1606, 2011.
11 Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *SODA*, pages 1590–1606, 2011.
12 V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.
13 V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. On agnostic learning of parities, monomials and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.
14 Vitaly Feldman and Varun Kanade. Computational bounds on statistical query learning. In *COLT*, pages 16.1–16.22, 2012.
15 Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. In *COLT*, pages 711–740, 2013.
16 Yoav Freund. Boosting a weak learning algorithm by majority. In *COLT*, pages 202–216, 1990.
17 Yoav Freund. An improved boosting algorithm and its implications on learning complexity. In *COLT*, pages 391–398, 1992.
18 Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

19 Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.

20 D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

21 Jeffrey C. Jackson. An efficient membership-query algorithm for learning dnf with respect to the uniform distribution. *J. Comput. Syst. Sci.*, 55(3):414–440, 1997.

22 Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008.

23 Daniel M. Kane, Adam Klivans, and Raghu Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. In *COLT*, pages 522–545, 2013.

24 M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

25 Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In *EUROCRYPT*, pages 7–26, 2011.

26 Adam Klivans, Pravesh Kothari, and Igor Oliveira. Constructing hard functions from learning algorithms. *Conference on Computational Complexity, CCC*, 20:129, 2013.

27 Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *J. Comput. Syst. Sci.*, 68(4):808–840, 2004.

28 Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning geometric concepts via gaussian surface area. In *FOCS*, pages 541–550, 2008.

29 Adam R. Klivans and Rocco A. Servedio. Boosting and hard-core set construction. *Machine Learning*, 51(3):217–238, 2003.

30 Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, pages 553–562, 2006.

31 Adam R. Klivans and Alexander A. Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010.

32 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993.

33 Ryan O'Donnell. Fourier coefficients of majority. `http://www.contrib.andrew.cmu.edu/~ryanod/?p=877`, 2012.

34 Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

35 Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the zero-one loss. In *COLT*, pages 441–450, 2010.

36 Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *The 53rd Annual IEEE Symposium on the Foundations of Computer Science (FOCS)*, 2012.

37 L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

38 V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

A Finding Large Hermite Coefficients Using Queries

For $\Delta_1 \in \mathbb{Z}^k$ and $\Delta_2 \in \mathbb{Z}^{n-k}$, let $\Delta = \Delta_1 \circ \Delta_2$ denote the n-tuple obtained by concatenating Δ_1 and Δ_2. Similarly, for $s \in \mathbb{R}^k$ and $z \in \mathbb{R}^{n-k}$ let $t = s \circ z$ denote the element of $\mathbb{R}^n$ obtained by concatenating s and z. We are now ready to present the procedure to find heavy Hermite coefficients of a function given query access to it. Since heavy Hermite coefficients, in general, may not be multilinear, we adapt the idea of [32] to work in this setting. Our proof is based on that of [32] (see also the lecture notes by O'Donnell [33]).

▶ **Theorem 14.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be any function such that $||f||_2 = \mathbb{E}_{x\sim\gamma}[f(x)^2] = 1$. There exists an algorithm that uses query access to f and runs in time $\tilde{O}(nd/\epsilon^2)$ to return every index $\Delta \in \mathbb{Z}^k$ of degree d such that $|\hat{f}(\Delta)| \geq \epsilon$.*

Proof. We estimate every coefficient of f that is larger than ϵ within an error of $\epsilon/3$. Thus, for each $\Delta \in \mathbb{Z}^n$, we will obtain $\tilde{f}(\Delta)$ such that $|\tilde{f}(\Delta) - \hat{f}(\Delta)| \leq \epsilon/3$.

We first describe a subroutine which we will repeatedly use in the algorithm. For any $\Delta \in Z^k$, let

$$W_{\Delta_1} = \sum_{\Delta_2 \in \mathbb{Z}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2)^2.$$

▶ **Lemma 15.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function with query access. Given $\Delta_1 = \{i_1, i_2, \cdots, i_k\} \in \mathbb{Z}^k$ such that $\sum_{j\leq k} i_j \leq d$, there is an algorithm that returns a value v such that $|v - \sum_{T:T\in\mathbb{N}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2)^2| \leq \delta$ with probability at least $2/3$ in time and queries $\tilde{O}(\frac{nd}{\delta^2})$.*

Proof. Define $\hat{f}_{\Delta_1} : \mathbb{R}^{n-k} \to \mathbb{R}$ by

$$\hat{f}_{\Delta_1}(z) = \mathbb{E}_{x\sim\mathbb{R}^k}[f(x \circ z) \cdot H_{\Delta_1}(x)]. \tag{3}$$

For $W \in \mathbb{Z}^n$, let $W_k \in \mathbb{R}^k$ denote the first k coordinate values of W and W_{n-k} denote the last k. One then has $H_W(x \circ z) = H_{W_k}(x) \cdot W_{W_{n-k}}(z)$ for any $x \in \mathbb{R}^{W_k}$ and $z \in \mathbb{R}^{W_{n-k}}$. Then, we have:

$$\begin{aligned}
\hat{f}_{\Delta_1}(z) &= \mathbb{E}_{x\sim\mathbb{R}^k}[f(x \circ z) \cdot H_{\Delta_1}(x)] \\
&= \mathbb{E}_{x\sim\mathbb{R}^k}[\sum_{W\in\mathbb{Z}^n} \hat{f}(W) \cdot H_W(x \circ z) \cdot H_{\Delta_1}(x)] \\
&= \mathbb{E}_{x\sim\mathbb{R}^k}[\sum_{W\in\mathbb{Z}^n} \hat{f}(W) \cdot H_{W_k}(x) H_{W_{n-k}}(z) \cdot H_{\Delta_1}(x)] \\
&= \sum_{W\in\mathbb{Z}^n} \mathbb{E}_{x\sim\mathbb{R}^k}[\hat{f}(W) \cdot H_{W_k}(x) H_{W_{n-k}}(z) \cdot H_{\Delta_1}(x)]
\end{aligned}$$

For every W such that $W_k \neq S$, the term above evaluates to 0 due to the orthogonality of H_{Δ_1} and H_{W_k}

$$= \sum_{W=\Delta_1 \; circ \Delta_2} \hat{f}(\Delta_1 \circ \Delta_2) H_{\Delta_2}(z) \tag{4}$$

Now,

$$\begin{aligned}
\sum_{\Delta_2\in\mathbb{N}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2)^2 &= \mathbb{E}_{z\in\gamma^{n-k}}[(\sum_{\Delta_2\in\mathbb{N}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2) H_{\Delta_2}(z))^2] \\
&= \mathbb{E}_{z,z'\in\gamma^{n-k}}[(\sum_{\Delta_2\in\mathbb{N}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2) H_{\Delta_2}(z)) \cdot \\
&\qquad (\sum_{\Delta_2\in\mathbb{N}^{n-k}} \hat{f}(\Delta_1 \circ \Delta_2) H_{\Delta_2}(z'))] \\
&\text{Using Equation (4)} \\
&= E_{z,z'\in\gamma^{n-k}}[\hat{f}_{\Delta_2}(z) \cdot \hat{f}_{\Delta_2}(z')] \\
&\text{Using Equation (3)} \qquad (5) \\
&= \mathbb{E}_{x,x'\in\gamma^k,\; z,z'\in\gamma^{n-k}}[f(x \circ z) f(x' \circ z') H_{\Delta_1}(x) \cdot H_{\Delta_1}(x')] \qquad (6)
\end{aligned}$$

The quantity in the RHS of Equation (6) can be computed up to an additive error of at most $\delta > 0$ by drawing $\tilde{O}(1/\delta^2)$ random points from $\mathbb{R}^k$ and $\mathbb{R}^{n-k}$ and obtaining the values of f at the appropriate combinations using queries. Thus, we obtain the required result. ◀

We can now describe the algorithm:

1. Set $\delta = \epsilon^2/3$, $\beta = \frac{1}{3dn\epsilon^2}$.
2. $\mathbb{S} \leftarrow \emptyset$.
3. For $j = 1$ to n
   a. For $k = 1$ to d:
      i. For each $T \in \mathbb{S}$, if $Z = T \circ k$ is such that H_Z is of degree at most d:
         A. Estimate W_Z to an accuracy of δ to a confidence of $1 - \beta$.
         B. If $W_Z > \epsilon^2/2$, $\mathbb{S} \leftarrow \mathbb{S} \cup Z$.
4. Return $\mathbb{S}$.

The algorithm above is analogous to the Kushilevitz Mansour algorithm and it is easy the correctness based on the lemma above: We begin by noting the sum squared of all Hermite coefficients of f is 1 as the Hermite transformation preserves ℓ_2 norms. Thus, the number of coefficients that are larger than ϵ in magnitude are at most $1/\epsilon^2$. One can thus argue that with high probability, the size of $\mathbb{S}$ in the algorithm above is at most $O(1/\epsilon^2)$ at all times. In the j^{th} iteration, the algorithm tries to append any of the d powers of x_j to each of the indices in $\mathbb{S}$. For each such newly produced index Z, the algorithm estimates the Weight W_Z as in the lemma above. It adds Z to $\mathbb{S}$ whenever W_Z is estimated to be higher than $\epsilon^2/2$. Thus, each such iteration needs $O(nd)$ time to execute.

This completes the proof. ◀