Non-Gaussian Component Analysis via Lattice Basis Reduction

Ilias Diakonikolas ILIAS@CS.WISC.EDU
University of Wisconsin, Madison

Daniel M. Kane DAKANE@CS.UCSD.EDU
University of California, San Diego

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

Non-Gaussian Component Analysis (NGCA) is the following distribution learning problem: Given i.i.d. samples from a distribution on $\mathbb{R}^d$ that is non-gaussian in a hidden direction v and an independent standard Gaussian in the orthogonal directions, the goal is to approximate the hidden direction v. Prior work (Diakonikolas et al., 2017) provided formal evidence for the existence of an information-computation tradeoff for NGCA under appropriate moment-matching conditions on the univariate non-gaussian distribution A. The latter result does not apply when the distribution A is discrete. A natural question is whether information-computation tradeoffs persist in this setting. In this paper, we answer this question in the negative by obtaining a sample and computationally efficient algorithm for NGCA in the regime that A is discrete or nearly discrete, in a well-defined technical sense. The key tool leveraged in our algorithm is the LLL method (Lenstra et al., 1982) for lattice basis reduction.

Keywords: Non-Gaussian Component Analysis, Lattice Basis Reduction, SoS Lower Bounds

1. Introduction

1.1. Background and Motivation

Non-Gaussian Component Analysis. Non-gaussian component analysis (NGCA) is a distribution learning problem modeling the natural task of finding "interesting" directions in high-dimensional data. As the name suggests, the objective is to find a "non-gaussian" direction (or, more generally, low-dimensional subspace) in a high-dimensional dataset, under a natural generative model. NGCA was defined in Blanchard et al. (2006) and subsequently studied from an algorithmic standpoint in a number of works, see, e.g., Vempala and Xiao (2011); Tan and Vershynin (2018); Goyal and Shetty (2019) and references therein.

For concreteness, we start by defining the relevant family of high-dimensional distributions.

Definition 1 (High-Dimensional Hidden Direction Distribution) *For a distribution A on the real line with probability density function $A(x)$ and a unit vector $v \in \mathbb{R}^d$, consider the distribution over $\mathbb{R}^d$ with probability density function $\mathbf{P}_v^A(x) = A(v \cdot x) \exp\left(-\|x - (v \cdot x)v\|_2^2/2\right)/(2\pi)^{(d-1)/2}$. That is, $\mathbf{P}_v^A$ is the product distribution whose orthogonal projection onto the direction of v is A, and onto the subspace perpendicular to v is the standard $(d-1)$-dimensional normal distribution.*

The NGCA learning problem is the following: Given i.i.d. samples from a distribution $\mathbf{P}_v^A$ on $\mathbb{R}^d$, where the direction v is unknown, find (or approximate) v. The standard formulation assumes that the univariate distribution A is known to the algorithm, it matches its first k moments with $N(0,1)$, for some $k \in \mathbb{Z}_+$, and there is a non-trivial difference in the moment of order $(k+1)$.

Information-Computation Tradeoffs for NGCA. Since A has its $(k+1)^{th}$ moment differing from that of a standard Gaussian, a moment computation on $\mathbf{P}_v^A$ allows us to approximate v in roughly $O(d^{k+1})$ samples and time. Interestingly, ignoring computational considerations, the NGCA problem can usually be solved with $O(d)$ samples. Perhaps surprisingly, the aforementioned simple method (requiring $\Omega(d^{k+1})$ samples) is qualitatively the best known sample-polynomial time algorithm for the problem. Given this state of affairs, it is natural to ask whether this information-computation gap is inherent for the problem itself.

In prior work, Diakonikolas et al. (2017) provided formal evidence for the existence of an *information-computation tradeoff* for NGCA under appropriate assumptions on the univariate non-gaussian distribution A. The Diakonikolas et al. (2017) result holds for a restricted model of computation, known as the Statistical Query (SQ) model. Statistical Query (SQ) algorithms are the class of algorithms that are only allowed to query expectations of bounded functions of the underlying distribution rather than directly access samples. The SQ model was introduced by Kearns (1998) and has been extensively studied in learning theory. A recent line of work, see, e.g., Feldman et al. (2017a, 2015, 2017b), generalized the SQ framework for search problems over distributions. The reader is referred to Feldman (2016) for a survey.

In more detail, the SQ lower bound of Diakonikolas et al. (2017) applies even for the (easier) hypothesis testing version of NGCA, where the goal is to distinguish between the standard Gaussian $N(0, I)$ on $\mathbb{R}^d$ and a planted distribution $\mathbf{P}_v^A$, for a hidden direction v. (Hardness for hypothesis testing can easily be used to derive hardness for the corresponding search problem.) Roughly speaking, they established the following generic SQ-hardness result:

> **Informal Theorem (Diakonikolas et al., 2017):** Let A be a one-dimensional distribution that matches its first k moments with the standard Gaussian $G = N(0, 1)$ and its chi-squared norm with G, $\chi^2(A, G)$, is finite. Suppose we want to distinguish between $N(0, I)$ on $\mathbb{R}^d$ and the distribution $\mathbf{P}_v^A$ for a random direction v. Then any SQ algorithm for this testing task requires either at least $d^{\Omega(k)}/\chi^2(A, G)$ samples or at least $2^{d^{\Omega(1)}}$ time.

A concrete application of the above result, given in Diakonikolas et al. (2017), is an SQ lower bound for the classical problem of learning mixtures of high-dimensional Gaussians. To obtain the hard family of instances, we take the one-dimensional distribution A be a mixture of univariate Gaussians $\sum_{i=1}^k w_i N(\mu_i, \sigma^2)$ with pairwise separated and bounded means μ_i and common variance $\sigma^2 = 1/\mathrm{poly}(k)$ such that A matches $\Omega(k)$ moments with $N(0, 1)$. Moreover, A will have total total variation distance at least $1/2$ from $N(0, 1)$. Then, each distribution $\mathbf{P}_v^A$ will look like a collection of k "parallel pancakes", in which the means lie on a line (corresponding to the smallest eigenvalue of the identical covariance matrices of the components). The orthogonal directions will have an eigenvalue of one, which is much larger than the smallest eigenvalue.

More broadly, the aforementioned generic SQ lower bound (Diakonikolas et al., 2017) has been the basis for a host of new and near-optimal information-computation tradeoffs (in the SQ model) for high-dimensional estimation tasks, including robust mean and covariance estimation (Diakonikolas et al., 2017), robust sparse mean estimation (Diakonikolas et al., 2017), adversarially robust learning (Bubeck et al., 2018), robust linear regression (Diakonikolas et al., 2019), list-decodable estimation (Diakonikolas et al., 2018, 2021b), learning simple neural networks (Diakonikolas et al., 2020b), and robust supervised learning in a variety of noise models (Diakonikolas et al., 2020a,

2021c; Diakonikolas and Kane, 2020; Diakonikolas et al., 2021a). Interestingly, subsequent work has obtained additional evidence of hardness for some of these problems via reductions from lattice problems (Bruna et al., 2021) and variants of the planted clique problem (Brennan and Bresler, 2020).

Motivation for This Work. Interestingly, the generic SQ lower bound of Diakonikolas et al. (2017) is vacuous for the natural setting where the distribution A is discrete (in which case, we have $\chi^2(A, N(0,1)) = \infty$) or, more generally, when A has very large chi-squared norm with the standard Gaussian. More specifically, for the parallel pancakes distribution described above, one needs the "thickness parameter" (corresponding to the eigenvalue of the covariance in the hidden direction) to be at least inverse exponential in the dimension. A natural question, which served as one of the motivations for this work, is whether information-computation tradeoffs persist for the discrete case.

Consider for example the case where A is supported on a discrete domain of size k and matches its first $\Omega(k)$ moments with $N(0,1)$. This corresponds to the special case of the parallel pancakes distribution, where the component covariances are degenerate — having zero eigenvalue in the hidden-direction. Does any efficient algorithm for these instances require $d^{\omega_k(1)}$ samples?

We answer these questions in the negative by designing a sample and computationally efficient algorithm for NGCA when A is discrete or nearly discrete in a well-defined sense (Assumption 2). The key tool leveraged in our reuslt is the LLL algorithm (Lenstra et al., 1982) for lattice basis reduction. We note that prior work (Bruna et al., 2021; Song et al., 2021) had used the LLL algorithm to obtain efficient learners for related problems that could be viewed as special cases of NGCA.

Connection with Sum-of-Squares (SoS) and Low-Degree Tests. Before we proceed with a detailed description of our results, a final remark is in order. As already mentioned, the SQ lower bounds of Diakonikolas et al. (2017) are vacuous when A is a discrete distribution. On the other hand, recent work has established information-computation tradeoffs for NGCA when A is supported on $\{-1, 0, 1\}$, both for low-degree polynomial tests (Mao and Wein, 2021) and for SoS algorithms (Ghosh et al., 2020). At first sight, these hardness results combined with our algorithm might appear to cast doubt on the validity of the low-degree conjecture (Hopkins, 2018). We note, however, that the latter conjecture only posits that a *noisy* version of the corresponding problem is computationally hard (as opposed to the problem itself) — a statement that appears to hold true in our setting. Conceptually, we view our algorithmic contribution as a novel example of an efficient algorithm (beyond Gaussian elimination) not captured by the aforementioned restricted models of computation.

1.2. Our Contributions

We consider the NGCA learning problem under the following assumption:

Assumption 2 *The distribution A on $\mathbb{R}$ is such that:*

1. *There exist $r_j \in \mathbb{R}$ for $j \in [k]$ with $|r_j| = O(1)$, $B \in \mathbb{Z}_+$, and $\epsilon > 0$ such that a sample $y \sim A$ is deterministically within additive ϵ of some number of the form $\sum_{j=1}^{k} n_j r_j$, for $n_j \in \mathbb{Z}$ with $|n_j| \leq B$ for all $j \in [k]$.*

2. *The distribution A is anti-concentrated around 0, specifically $\Pr_{X\sim A}[|X| > 1/d] > 1/d$.*

3. *The distribution A is concentrated around 0, specifically $\Pr_{X\sim A}[|X| > \mathrm{poly}(d)] < 1/d$.*

Some comments are in order to interpret Assumption 2. Condition 1 above is the critical condition requiring that A is approximately supported on points that are (small) integer linear combinations of the r_j's. This is the key condition that underlies our main technique. Notice that this condition can be satisfied by any distribution A that has support size at most k, or even a distribution A that is supported on k intervals, each of length at most ϵ. In fact, it is sufficient for A to be $O(1/d)$-close in total variation distance to such a distribution, as there will be a constant probability that any $O(d)$ sample set drawn from it are supported on the appropriate intervals. This means that our algorithmic results applies, for example, to parallel pancake distributions, as long as the thickness of the pancakes is no more than $O(\epsilon/\sqrt{\log(d)})$.

Conditions 2 and 3 are technical conditions that are needed for our particular algorithm to work. However, note that if Condition 2 is not satisfied, then it is reasonably likely that $O(d)$ random samples from $\mathbf{P}_v^A$ will have much smaller variance in the v-direction than in any of the orthogonal directions. This provides a much easier method for approximating v. Condition 3 is essentially required to guarantee that we do not need to deal with unlimited precision to approximate points. However, it is easy to see that if this condition is violated, one can approximate v simply by normalizing any samples from $\mathbf{P}_v^A$ with ℓ_2-norm more than d.

We prove the following theorem:

Theorem 3 (Main Result) *Under Assumption 2, if $\epsilon < 2^{-\Omega(dk^2)} B^{-\Omega(k)}$ with sufficiently large implied universal constants in the big-Ω, there exists an algorithm that draws $m = 2d$ i.i.d. samples from $\mathbf{P}_v^A$ for an unknown unit vector $v \in \mathbb{R}^d$, runs in time $\mathrm{poly}(d, k, \log B)$, and outputs a vector v^* such that with constant probability either $\|v^* - v\|_2$ is small or $\|v^* + v\|_2$ is small.*

We note that Theorem 3 only guarantees an approximation of either v or $-v$. Such a guarantee may be inherent, as if A is a symmetric distribution we have that $\mathbf{P}_v^A = \mathbf{P}_{-v}^A$.

1.3. Overview of Techniques

We begin by considering the simple case where the univariate distribution A is supported *exactly* on integers. This special case provides a somewhat simpler version of our algorithm while capturing some of the key ideas. In this case, we draw $m = d + 1$ i.i.d. samples $x_i \in \mathbb{R}^d$ from $\mathbf{P}_v^A$ and note that (with high probability) they will satisfy a unique (up to scaling) linear relation $\sum_{i=1}^m c_i x_i = 0$, for some $c_i \in \mathbb{R}$ with at least one $c_i \neq 0$. In particular, we have that $\sum_{i=1}^m c_i (v \cdot x_i) = 0$. Since the quantities $v \cdot x_i$ for $i \in [m]$ are all integers, we hope to solve for them by finding the (with high probability unique, up to scaling) integer linear relation among the c_i's. It turns out that this can be achieved by leveraging the Lenstra-Lenstra-Lovasz (LLL) lattice basis reduction algorithm. Having found an integer solution $\sum_{i=1}^m c_i n_i = 0$ for $n_i \in \mathbb{Z}$, we can solve the system of linear equations $v \cdot x_i = n_i$, $i \in [m]$, for the hidden vector v.

We now proceed to deal with the case where A is no longer supported on integers, but is instead supported on elements that are *close* (within some additive error ϵ) to integers. In this case, we will similarly have $\sum_{i=1}^m c_i (v \cdot x_i) = 0$, which means that if n_i is the integer closest to $v \cdot x_i$, we will have that $\sum_{i=1}^m c_i n_i$ is close to 0. In order to solve for this near-integer linear-relation, we make essential use of basic lattice techniques. In particular, for $n = (n_i)_{i=1}^m \in \mathbb{Z}^m$, we define the quadratic form $Q(n) := \sum_{i=1}^m n_i^2 + N \left(\sum_{i=1}^m c_i n_i\right)^2$, for some appropriately large N. Note that

integer vectors with small norm under Q must have $|n_i|$ small for all $i \in [m]$ and have $|\sum_{i=1}^m c_i n_i|$ be very small. It is not hard to show that if ϵ is sufficiently small and N is chosen appropriately, with high probability over the samples x_i, taking n_i to be the integer closest to $v \cdot x_i$ for each $i \in [m]$ will give substantially the smallest non-zero norm under Q. Therefore, using the LLL algorithm to find an approximate smallest vector will return (some multiple of) this vector. Given the n_i's, we note that $v \cdot x_i \approx n_i$ for all $i \in [m]$, and we can then use least-squares regression to solve for an approximation to v.

Unfortunately, if the above approach is applied naively, it will work only if ϵ is assumed to be exponentially small in d^2, i.e., $2^{-\Omega(d^2)}$, rather than in d. This is because the LLL algorithm only guarantees a $2^{O(d)}$-approximation to the smallest vector. The $\sum_{i=1}^m n_i^2$-term in Q ensures that any such vector will have n_i at most exponential in d. But given that there are $2^{\Omega(d^2)}$ integer vectors with coefficients of this size, we can expect one to randomly have $|\sum_{i=1}^m c_i n_i|$ be only $2^{-\Omega(d^2)}$. This will be distinguishable from the vector we are looking for only if ϵ is smaller than this quantity.

In order to fix this issue, instead of taking only $m = d + 1$ samples from $\mathbf{P}_v^A$, we instead draw $m = 2d$ samples. These now have d linear relations and we note that the vector of n_i's should approximately satisfy all of them. In particular, letting V be the vector space of linear relations satisfied by the x_i's, we consider the quadratic form defined by $Q(n) := \|n\|_2^2 + N \|\mathrm{Proj}_V(n)\|_2^2$. This improves things because it is now much less likely that one of our $2^{O(d^2)}$ "small" n's will randomly have a small projection onto V. This allows us to operate even when ϵ is only $2^{-\Omega(d)}$. Note that we cannot hope to do much better than this because the LLL algorithm will still have an exponential gap between the shortest vector and the one that it finds.

Finally, we are also able to extend our algorithm to the setting where the distribution A is not supported on integers, but instead on numbers of the form $\sum_{j=1}^k a_j r_j$, where the a_j's are (not too large) integers and the r_j's are some k specific (known) real numbers. In this more general setting, instead of $v \cdot x_i \approx n_i$, we will have that $v \cdot x_i \approx \sum_{j=1}^k n_{i,j} r_j$, for some integers $n_{i,j}$, $i \in [m], j \in [k]$. We then set-up a quadratic form similar to the one before, namely $Q(n) = \|n\|_2^2 + N \|\mathrm{Proj}_V(t)\|_2^2$, where $t = (t_i)_{i \in [m]}$ is the vector with coordinates $t_i = \sum_{j=1}^k n_{ij} r_j$ for some integers n_{ij}. Once again, the correct integer vector n will be an unusually small vector with respect to this quadratic form; and if we can find it, we will be able to use it to approximate the hidden direction v.

A subtle issue in this case is that the correct vector (and multiples) need not be the *only* small vectors in this lattice. In particular, if the r_j's satisfy an approximate linear relation $\sum_{j=1}^k k_j r_j \approx 0$, then letting $n_{ij} = k_j \cdot \delta_{i,i_0}$, for some i_0, will also have $\mathrm{Proj}_V(t)$ small, because t will be small. To deal with this issue, we will need to apply the LLL algorithm and take not just the single smallest vector, but the smallest few vectors (in a carefully selected way). We can then show that the true vector n that we are looking for is in the subspace spanned by these vectors. By finding a lattice vector in this space such that t is large but $\mathrm{Proj}_V(t)$ is small, we can find a t where each t_i is approximately some multiple of $v \cdot x_i$ for all i. Using this t, we can solve for v as before.

Independent Work. Concurrent and independent work by Zadik et al. (2021) obtained a similar algorithm for NGCA under similar assumptions, by also leveraging the LLL algorthm. More concretely, the algorithm of Zadik et al. (2021) efficiently solves the NGCA problem when A is a discrete distribution on an integer lattice, roughly corresponding to the $k = 1$ and $\epsilon = 0$ case of our result.

2. Proof of Theorem 3

The pseudo-code for our algorithm is given below.

Algorithm `LLL-based-NGCA`
Input: $m = 2d$ i.i.d. samples from $\mathbf{P}_v^A$, where A satisfies Assumption 2 for given real numbers $r_1, r_2, \ldots, r_k$ and some parameters k, B, ϵ with $\epsilon < 2^{-C'dk^2} B^{-C'k}$, where $C' > 0$ is a sufficiently large constant.

1. Let $N = 2^{Cmk^2} B^{Ck}$ be a positive integer, for C a sufficiently large constant, such that $N < 1/\epsilon^2$.
2. Let $x_1, x_2, \ldots, x_m$ be m i.i.d. samples from $\mathbf{P}_v^A$ each rounded to the nearest multiple of $\delta = \epsilon/N^2$.
3. Let S be the $d \times m$ matrix with columns $x_1, x_2, \ldots, x_m$ and V be the right kernel of S.
4. Define the quadratic form Q on $\mathbb{Z}^{m \times k}$ such that for an input vector $n = \{n_{i,j}\}_{i \in [m], j \in [k]}$ we have that $Q(n) := \sum_{i=1}^m \sum_{j=1}^k n_{i,j}^2 + N \left\| \mathrm{Proj}_V \left(\left\{ \sum_{j=1}^k n_{i,j} r_j \right\}_{i \in [m]} \right) \right\|_2^2$.
5. Compute a δ-LLL reduced basis, for $\delta = 3/4$, $\{b_1, b_2, \ldots, b_{mk}\}$ for Q, where $b_i \in \mathbb{R}^{m \times k}$.
6. Apply the Gram-Schmidt orthogonalization process to the b_i's, using Q as our norm, to obtain an orthogonal basis $\{b_1^*, b_2^*, \ldots, b_{mk}^*\}$.
7. Let $\ell \in [mk]$ be the largest integer such that $Q(b_\ell^*) \leq mkB^2 + Nmk\epsilon^2$. Let W be the real span of $\{b_1, b_2, \ldots, b_\ell\}$.
8. Consider the quadratic form R on $\mathbb{R}^{m \times k}$ defined by $R(n) = \sum_{i=1}^m \left(\sum_{j=1}^k n_{i,j} r_j \right)^2$. For a sufficiently large universal constant $C > 0$, find a vector $w = \{w_{i,j}\}_{i \in [m], j \in [k]} \in W$ with $Q(w) = 2^{Cmk} B^2$ such that $R(w)/Q(w)$ is approximately maximized. Note that this can be done with an eigenvalue computation.
9. Write the vector w in the form $w = \sum_{i=1}^\ell c_i b_i$, for some $c_i \in \mathbb{R}$. Let $w' = \sum_{i=1}^\ell c_i' b_i$, where c_i' is the nearest integer to c_i.
10. Let v^* be the minimizer of $\sum_{i=1}^m \left(v^* \cdot x_i - \sum_{j=1}^k w_{i,j}' r_j \right)^2$. Note that this can be found using least squares regression. Return the normalization of v^*.

Background on Lattices and Lattice Basis Reduction For the sake of clarity, we review basic facts and definitions about lattices and lattice reduction algorithms. One can find references for the claims here for example in Cohen (2010), Chapters 2.5 and 2.6.

We define a *lattice* to be a finitely-generated (and thus discrete) subgroup of an inner product space. The lattice will inherit the inner product and norm from the underlying inner product space.

Given a lattice L, we note that as a group it must be isomorphic to $\mathbb{Z}^d$ for some d. We call d the *dimension* of L. We also call any set of d elements of L that generate L as a group a *basis* of L. Given such a basis $b_1, b_2, \ldots, b_d$, we let the associated *Gram Matrix* G be the $d \times d$ matrix with $G_{i,j} = \langle b_i, b_j \rangle$. We note that the Gram matrix can be thought of as an implicit way of defining the lattice L.

Given a basis $b_1, \ldots, b_d$ for a lattice using Gram-Schmidt orthogonalization, one can uniquely find vectors $b_1^*, \ldots, b_d^*$ in the underlying vector space so that $b_i^* = b_i + \sum_{j<i} c_{i,j} b_j$ for some real numbers $c_{i,j}$ and so that $\langle b_i^*, b_j^* \rangle = 0$ for $i \neq j$. We call this basis *reduced* if $\|b_i^*\|^2 \geq \|b_{i-1}^*\|^2/2$ for all $1 < i \leq d$ (we note that this is a slightly weaker condition than the standard condition for a reduced basis). The important result is that such a basis can be efficiently computed.

Theorem 4 (LLL) *There exists an algorithm that given the Gram matrix G for a lattice L (corresponding to some basis $b_1, \ldots, b_d$) so that G has rational entries, runs in time polynomial in the dimension of L and the bit complexity of the entries of G, and returns a reduced basis for L (by writing the elements of this basis as linear combinations of the b_i).*

Analysis of Our Algorithm To begin the analysis, we first analyze the infinite precision version of this problem, ignoring the rounding and instead simply letting the x_i's be i.i.d. samples from $\mathbf{P}_v^A$.

We begin by analyzing some of the basic properties of the above procedure. We start by showing that with high probability over our set of samples $x_1, \ldots, x_m$ the quadratic form $Q(n)$ is small if and only if the vector $y^{(n)} \in \mathbb{R}^m$ with coordinates $\sum_{j=1}^k n_{i,j} r_j$, $i \in [m]$, is approximately a multiple of the vector $y \in \mathbb{R}^m$ with coordinates $v \cdot x_i$, $i \in [m]$.

Specifically, we prove the following lemma:

Lemma 5 *Let $y \in \mathbb{R}^m$ be the vector with coordinates $y_i := v \cdot x_i$, $i \in [m]$. Consider a vector $n = (n_{i,j}) \in \mathbb{R}^{m \times k}$. Let $y^{(n)} \in \mathbb{R}^m$ be the vector with coordinates $(y^{(n)})_i := \sum_{j=1}^k n_{i,j} r_j$, $i \in [m]$, and let $(y^{(n)})' \in \mathbb{R}^m$ be the component of $y^{(n)}$ orthogonal to y. Then we have that*

$$Q(n) \leq \|n\|_2^2 + N \, \|(y^{(n)})'\|_2^2 \, . \tag{1}$$

Furthermore, for any positive integer M and with high probability over the choice of $S = [x_i]_{i=1}^m$, for all such n with $\|n\|_2 \leq M \, m \, k$, we have that:

$$Q(n) \geq N \, O(M)^{-mk/(m-d)} \, \|(y^{(n)})'\|_2^2 \, . \tag{2}$$

Proof We start by writing S using an orthonormal basis for $\mathbb{R}^d$ in which v is the first vector. (Note that changing the basis we use for $\mathbb{R}^d$ does not change V.) In this basis, observe that y is the first row of S. Moreover, all other entries of S in this basis are independent standard Gaussians. Thus, we can write S as $\begin{bmatrix} y \\ G \end{bmatrix}$, where G is an independent Gaussian matrix. Note that the kernel of G is a random subspace of $\mathbb{R}^m$ of dimension $m - d + 1$. Thus, after conditioning on y, V is a random $(m-d)$-dimensional subspace orthogonal to y. Also note that we can generate a subspace with the same distribution by taking the span of $m - d$ independent standard orthogonal-to-y Gaussians.

We next consider a vector $n = (n_{i,j}) \in \mathbb{R}^{m \times k}$. Let $y^{(n)} \in \mathbb{R}^m$ be the vector with coordinates $\sum_{j=1}^k n_{i,j} r_j$, and let $(y^{(n)})' \in \mathbb{R}^m$ be the component of $y^{(n)}$ orthogonal to y. To begin with, we note that $\|\mathrm{Proj}_V(y^{(n)})\|_2 \leq \|(y^{(n)})'\|_2$, and this implies Equation (1).

We prove Equation (2) by a union bound over the $O(M)^{mk}$ many vectors of appropriate norm. In particular, fix such an n. Recall that conditioning on y, the kernel of v is the span of $g_1, g_2, \ldots, g_{m-d}$, where the g_i are independent orthogonal-to-y Gaussians. Note that $\|\mathrm{Proj}_V(y^{(n)})\|_2 \geq \max_i |g_i \cdot y^{(n)}|/\|g_i\|_2$. Note that $g_i \cdot y^{(n)}$ is distributed like a Gaussian with standard deviation $\|(y^{(n)})'\|_2$. For $\delta > 0$, it is not hard to see that for each i we have that $|g_i \cdot y^{(n)}|/\|g_i\|_2 < \delta/\sqrt{mk}$ with probability $O(\delta)$ (for example because the probability that $|g_i \cdot y^{(n)}| < t\delta$ and $\|g_i\|_2 > (t-1)\sqrt{mk}$ is $O(\delta/t^2)$ for any positive integer t). Thus, the probability that $\|\mathrm{Proj}_V(y^{(n)})\|_2 < \delta/\sqrt{mk}$ is at most $O(\delta)^{m-d}$. Letting δ be equal to $(CM)^{-mk/(m-d)}$, for a sufficiently large constant C, yields the result. ■

We will henceforth assume that the high probability conclusion of Lemma 5 holds for the samples our algorithm has selected with $M := 2^{2Cmk}B$, for $C > 0$ a sufficiently large universal constant. Given this assumption, we next need to analyze which vectors give us small values of Q and what this means about the output of our call to the LLL algorithm. In particular, there is a particular vector n^* that would cause t to approximate y. We claim that $Q(n^*)$ is small and that this in turn implies that n^* is an integer linear combination of $b_1, b_2, \ldots, b_\ell$.

By assumption, each $y_i = v \cdot x_i$ is within additive ϵ of $\sum_{j=1}^k n^*_{i,j} r_j$, for some $n^*_{i,j} \in \mathbb{Z}$. Combining these $n^*_{i,j}$'s, we get a single vector $n^* = (n^*_{ij})_{i\in[m],j\in[k]} \in \mathbb{Z}^{m\times k}$ which has all entries with absolute value at most B, and by Lemma 5 satisfies $Q(n^*) \leq mkB^2 + Nmk\epsilon^2$. Note that n^* is a linear combination of the b_i's, namely $n^* = \sum_{i=1}^{mk} c_i b_i$. Let t be the largest i such that $c_i \neq 0$. Note that we can also write n^* as $\sum_{i=1}^{mk} c'_i b^*_i$, for some real c'_i, and that $c'_t = c_t$. Since the b^*_i are orthogonal with respect to the quadratic form Q, this implies that

$$Q(n^*) \geq Q(c'_t b^*_t) \geq Q(b^*_t) \, .$$

In particular, this means that $Q(b^*_t) \leq mkB^2 + Nmk\epsilon^2$. By our choice of ℓ, this implies that $t \leq \ell$, and in particular that n^* is a linear combination of $b_1, b_2, \ldots, b_\ell$.

Unfortunately, we cannot necessarily find n^* within this subspace. However, it will suffice for our purposes to find a vector z for which $\|z\|_2$ is large, but the part of z orthogonal to y is small. To do this, it will suffice to find an integer vector n for which $R(n)$ is large (implying that $\|y^{(n)}\|_2$ is large), but for which $Q(n)$ is small (implying that $y^{(n)}$ is nearly orthogonal to y). We know that n^* is such a vector and that it is somewhere in W. It now remains to find it.

Note that $n^* \in W$. Note that $R(n^*) = \|y^{(n^*)}\|_2^2 \geq \|y\|_2^2/2 + O(m\epsilon^2)$. By the anti-concentration Condition 2, with constant probability over the choice of y, this is $\Omega(1/d^2)$. On the other hand, we have that $Q(n^*) \leq mk(B^2 + N\epsilon^2)$. Therefore, we have that

$$R(n^*)/Q(n^*) \geq \Omega(1/(d^2 mkB^2)) \, .$$

Given our algorithm's choice of w, we have that $R(w)/Q(w) \geq \Omega(1/(d^2 mkB^2))$. On the other hand, we note that for $i \leq \ell$ we have that

$$Q(b_i) \leq 2^{mk} Q(b^*_\ell) \leq 2^{mk}(mkB^2) \, .$$

This in particular follows from the fact that $Q(b^*_{i+1}) \geq Q(b^*_i)/2$ for positive integers i. The latter statement can be derived, for example, from p. 86 of Cohen (2010)). This means that $\|b_i\|_2^2 \leq 2^{mk}(mkB^2)$, and thus $R(b_i) \leq 2^{O(mk)}B^2$. This implies that $R(w - w') \leq 2^{O(mk)}(B^2 + N\epsilon^2)$.

However, since $Q(b_i) \leq 2^{mk}(mkB^2)$, by similar reasoning, we obtain that $Q(w-w') \leq 2^{O(mk)}B^2$. Together, this implies that $\|w'\|_2^2 \leq Q(w') = \Theta(2^{Cmk}B^2)$, and that

$$R(w')/Q(w') = \Omega(1/(d^2mkB^2)) \; .$$

Assuming the high probability statement of Lemma 5 with $M := 2^{2Cmk}B$, we have that

$$Q(w') \geq N2^{-O(m^2k^2/(m-d))}B^{-mk/(m-d)}\|(y^{(w')})'\|_2^2 \; .$$

This implies that $\|(y^{(w')})'\|_2^2 \leq N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))}$. Note that this means that the vector with coordinates $\sum_{j=1}^k w'_{i,j}r_j$ is within $N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))}$ of some multiple of y. Thus, taking v^* to be an appropriate multiple of v yields an error of at most

$$N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))}$$

in the defining equation of v^*.

We next need to determine how close the above implies that v^* will be to a multiple of v. To analyze this, we consider the eigenvalues of the matrix $\sum_{i=1}^m x_i x_i^T$. By Condition 2, with large constant probability, the eigenvalue in the v-direction will be at least $\Omega(1/d^2)$. As the x_i's in orthogonal directions are independent standard Gaussians, it is not hard to see (for example via a cover argument) that with this large constant probability all eigenvalues of $\sum_{i=1}^m x_i x_i^T$ are at least $\Omega(1/d^2)$. The error in the least-squares regression problem equals

$$\sum_{i=1}^m \left(v^* \cdot x_i - \sum_{j=1}^k w'_{i,j}r_j \right)^2 = \sum_{i=1}^m \left(v^* \cdot x_i - \alpha y_i - ((y^{(w')})')_i \right)^2 \; ,$$

where α is some real multiple. Notice that $v^* \cdot x_i - \alpha y_i = (v^* - \alpha v) \cdot x_i$. Therefore the above is

$$(v^* - \alpha v)^T \sum_{i=1}^m x_i x_i^T (v^* - \alpha v) + O\left((m/d)\|(y^{(w')})'\|_2^2 + (m/d)\,\|(y^{(w')})'\|_2\,\|v^* - \alpha v\|_2 \right) \; .$$

In particular, noting that setting $v^* = \alpha v$ obtains a value of $O(m/d)\left(N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))}\right)^2$, the true v^* must satisfy

$$\|v^* - \alpha v\|_2 \leq N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))} \; .$$

On the other hand, since $R(w') > 1$, we have that $\|(y^{(w')})'\|_2^2 > 1$, which (assuming that all x_i's have norm $O(\sqrt{d})$ which holds with high probability) implies that $\|v^*\|_2 \gg 1/\sqrt{d}$. This means by the above that the normalization of v^* is within ℓ_2-error

$$N^{-1}2^{O(m^2k^2/(m-d))}B^{O(mk/(m-d))}$$

of $\pm v$.

Since we have selected $m = 2d$, $N = 2^{Cmk^2}B^{Ck}$, for C a sufficiently large constant, and $\epsilon < 2^{-C'dk^2}B^{-C'k}$, for some sufficiently large constant C', it follows that the normalization of v^* is exponentially close to $\pm v$.

Next we need to show that rounding the x_i's does not affect the correctness of our procedure. For this, we note that the above analysis only needed the following facts about the x_i:

1. Lemma 5 holds for $M = 2^{2Cmk}B$.

2. $\sum_{i=1}^{m} x_i x_i^T \succeq \Omega(I/d^2)$.

3. $v \cdot x_i$ is within $2^{-\Omega(dk^2)}B^{-\Omega(k)}$ (with sufficiently large constants in the big-Ω) of some integer linear combination of the r_i's with coefficients of absolute value at most B for all i.

We note that these hold with reasonable probability by the above. We claim that if they hold for the unrounded x_i's and if the x_i's have absolute value at most $\mathrm{poly}(d)$ (which happens with constant probability by Condition 3), then they hold for the rounded x_i's, perhaps with slightly worse implied constants in the big-O and big-Ω terms.

To show this, we begin with Condition 3. This still holds since rounding an x_i changes the value of $v \cdot x_i$ by at most $d\delta < \epsilon$.

For Condition 2, we note that changing each coordinate of x_i by δ changes $\sum_{i=1}^{m} x_i x_i^T$ by at most $dm\delta \max_i \|x_i\|_2$ in Frobenius norm. As this is much less than $1/d^2$, the minimum eigenvector of $\sum_{i=1}^{m} x_i x_i^T$ is still large enough after the rounding.

Finally, for Lemma 5, we note that the argument for Equation (1) still applies. For Equation (2), we note that for a vector z, $\mathrm{Proj}_V(z) = z - t$, where t is the unique vector in the range of S^T such that $St = Sz$. From this, we conclude that $t = S^T(SS^T)^{-1}Sz$. We claim that rounding this does not change the value of t (or, therefore, the value of $\|t - z\|_2^2$) by much. In particular, it is easy to see that rounding changes S and S^T by $O(md\delta)$ in Frobenius norm. The effect on $(SS^T)^{-1}$ is more complicated; but we know that $SS^T = \sum_i x_i x_i^T \succeq \Omega((1/d^2))\ I$. This and the fact that the rounding changes SS^T by relatively little in terms of Frobenius norm, suffices to imply that the rounding does not change much the value of $Q(n)$, for any vector n with coefficients of absolute value M.

Having established correctness, we need to bound the runtime. This is relatively straightforward, as we have to solve problems in dimension $\mathrm{poly}(md)$ with $\mathrm{poly}(md\log(B))$ bits of precision. In particular, Step 3 boils down to row-reduction; Step 4 requires computing a projection matrix; Step 5 uses the LLL algorithm; Step 8 can be done via an approximate eigenvalue computation; and Step 10 is least squares. Each of these operations can be performed in time that is polynomial in the dimension of the problem and in the number of bits of precision required.

This completes the proof of Theorem 3.

Remark 6 We remark that our algorithm works with any number $m > d$ samples, as long as ϵ is less than $2^{-\Omega(dk^2 m/(m-d))}B^{-\Omega(km/(m-d))}$ for sufficiently large constants in the big-Ω's. For example, one could take $m = d + 1$ samples, as long as $\epsilon < 2^{-C'd^2k^2}B^{-C'dk}$.

Acknowledgments

Ilias Diakonikolas was supported by NSF Medium Award CCF-2107079, NSF Award CCF-1652862 (CAREER), a Sloan Research Fellowship, and a DARPA Learning with Less Labels (LwLL) grant. Daniel M. Kane was supported by NSF Medium Award CCF-2107547, NSF Award CCF-1553288 (CAREER), a Sloan Research Fellowship, and a grant from CasperLabs.

References

G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, and K.-R. Müller. In search of non-gaussian components of a high-dimensional distribution. *Journal of Machine Learning Research*, 7(9):247–282, 2006. URL http://jmlr.org/papers/v7/blanchard06a.html.

M. S. Brennan and G. Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory, COLT 2020*, volume 125 of *Proceedings of Machine Learning Research*, pages 648–847. PMLR, 2020. URL http://proceedings.mlr.press/v125/brennan20a.html.

J. Bruna, O. Regev, M. J. Song, and Y. Tang. Continuous LWE. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021*, pages 694–707. ACM, 2021.

S. Bubeck, E. Price, and I. P. Razenshteyn. Adversarial examples from computational constraints. *CoRR*, abs/1805.10204, 2018. URL http://arxiv.org/abs/1805.10204.

H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer Publishing Company, Incorporated, 2010. ISBN 3642081428.

I. Diakonikolas and D. M. Kane. Near-optimal statistical query hardness of learning halfspaces with massart noise. *CoRR*, abs/2012.09720, 2020. URL https://arxiv.org/abs/2012.09720.

I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 73–84, 2017. Full version at http://arxiv.org/abs/1611.03473.

I. Diakonikolas, D. M. Kane, and A. Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1047–1060, 2018. Full version available at https://arxiv.org/abs/1711.07211.

I. Diakonikolas, W. Kong, and A. Stewart. Efficient algorithms and lower bounds for robust linear regression. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 2745–2754, 2019.

I. Diakonikolas, D. Kane, and N. Zarifis. Near-optimal SQ lower bounds for agnostically learning halfspaces and relus under gaussian marginals. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020a.

I. Diakonikolas, D. M. Kane, V. Kontonis, and N. Zarifis. Algorithms and SQ lower bounds for PAC learning one-hidden-layer relu networks. In *Conference on Learning Theory, COLT 2020*, volume 125 of *Proceedings of Machine Learning Research*, pages 1514–1539. PMLR, 2020b.

I. Diakonikolas, D. M. Kane, V. Kontonis, C. Tzamos, and N. Zarifis. Learning general halfspaces with general massart noise under the gaussian distribution. *CoRR*, abs/2108.08767, 2021a. URL https://arxiv.org/abs/2108.08767.

I. Diakonikolas, D. M. Kane, A. Pensia, T. Pittas, and A. Stewart. Statistical query lower bounds for list-decodable linear regression. *CoRR*, abs/2106.09689, 2021b. URL https://arxiv.org/abs/2106.09689.

I. Diakonikolas, D. M. Kane, T. Pittas, and N. Zarifis. The optimality of polynomial regression for agnostic learning under gaussian marginals in the SQ model. In *Conference on Learning Theory, COLT 2021*, volume 134 of *Proceedings of Machine Learning Research*, pages 1552–1584. PMLR, 2021c.

V. Feldman. Statistical query learning. In *Encyclopedia of Algorithms*, pages 2090–2095. Springer New York, 2016.

V. Feldman, W. Perkins, and S. Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC, 2015*, pages 77–86, 2015.

V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *J. ACM*, 64(2):8:1–8:37, 2017a.

V. Feldman, C. Guzman, and S. S. Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 1265–1277. SIAM, 2017b.

M. Ghosh, F. G. Jeronimo, C. Jones, A. Potechin, and G. Rajendran. Sum-of-squares lower bounds for sherrington-kirkpatrick via planted affine planes. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 954–965. IEEE, 2020.

N. Goyal and A. Shetty. Non-gaussian component analysis using entropy methods. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pages 840–851. ACM, 2019.

S. B. Hopkins. *Statistical inference and the sum of squares method*. PhD thesis, Cornell University, 2018.

M. J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6): 983–1006, 1998.

H.W. Lenstra, A.K. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

C. Mao and A. S. Wein. Optimal spectral recovery of a planted vector in a subspace. *CoRR*, abs/2105.15081, 2021. URL https://arxiv.org/abs/2105.15081.

M. J. Song, I. Zadik, and J. Bruna. On the cryptographic hardness of learning single periodic neurons. *CoRR*, abs/2106.10744, 2021. URL https://arxiv.org/abs/2106.10744.

Y. S. Tan and R. Vershynin. Polynomial time and sample complexity for non-gaussian component analysis: Spectral methods. In *Conference On Learning Theory, COLT 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 498–534. PMLR, 2018.

S. S. Vempala and Y. Xiao. Structure from local optima: Learning subspace juntas via higher order PCA. *CoRR*, abs/1108.3329, 2011. URL http://arxiv.org/abs/1108.3329.

I. Zadik, M. J. Song, A. S. Wein, and J. Bruna. Lattice-based methods surpass sum-of-squares in clustering, 2021.