Learning a mixture of two subspaces over finite fields

Aidao Chen[†] AIDAOCHEN2022@U.NORTHWESTERN.EDU
Northwestern University

Anindya De[*] ANINDYAD@SEAS.UPENN.EDU
University of Pennsylvania

Aravindan Vijayaraghavan[†] ARAVINDV@NORTHWESTERN.EDU
Northwestern University.

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We study the problem of learning a mixture of two subspaces over $\mathbb{F}_2^n$. The goal is to recover the individual subspaces A_0, A_1, given samples from a (weighted) mixture of samples drawn uniformly from the subspaces A_0 and A_1. This problem is computationally challenging, as it captures the notorious problem of "learning parities with noise" in the degenerate setting when $A_1 \subseteq A_0$. This is in contrast to the analogous problem over the reals that can be solved in polynomial time (Vidal'03). This leads to the following natural question: *is Learning Parities with Noise the only computational barrier in obtaining efficient algorithms for learning mixtures of subspaces over $\mathbb{F}_2^n$?*

The main result of this paper is an affirmative answer to the above question. Namely, we show the following results:

1. When the subspaces A_0 and A_1 are *incomparable*, i.e., $A_0 \not\subseteq A_1$ and $A_1 \not\subseteq A_0$, then there is a polynomial time algorithm to recover the subspaces A_0 and A_1.
2. In the case when $A_1 \subseteq A_0$ such that $\mathsf{dim}(A_1) \leq \alpha \cdot \mathsf{dim}(A_0)$ for $\alpha < 1$, there is a $n^{O(1/(1-\alpha))}$ time algorithm to recover the subspaces A_0 and A_1.

Thus, our algorithms imply computational tractability of the problem of learning mixtures of two subspaces, except in the degenerate setting captured by learning parities with noise.

Keywords: mixture models, subspaces, learning parities with noise

1. Introduction

Mixture models form an expressive class of probabilistic models that are widely used to find structure in unlabeled data from a heterogeneous population. Each of the k components in a mixture model represents one of the k sub-populations (assumed to be homogeneous) that constitute the overall heterogeneous population. A variety of mixture models ranging from Gaussian mixture models and mixtures of product distributions over continuous domains, to mixtures of ranking models, mixtures of subcubes over discrete domains are used to capture data in different domains. There is an extensive literature in statistics and computer science that gives efficient polynomial time algorithms for learning many mixture models with a constant number of mixture components (Feldman

[*] Supported by NSF grants CCF 1910534 and CCF 1926872. Part of the work was done while visiting the Simons Institute for Theory of Computing for the program "Probability, Geometry and Computation in High Dimensions".
[†] Supported by NSF grants CCF-1652491, CCF-1637585 and CCF-1934931.

et al., 2006; Kalai et al., 2010; Moitra and Valiant, 2010; Belkin and Sinha, 2010; Rabani et al., 2014; Li et al., 2015; Awasthi et al., 2010; Liu and Moitra, 2018; Chen and Moitra, 2019).

A common assumption in high-dimensional data analysis is to assume that the given data belong to a collection of lower dimensional subspaces. A prominent line of work in machine learning, computer vision and computational geometry (Vidal, 2003; Elhamifar and Vidal, 2013; Soltanolkotabi et al., 2014; Park et al., 2014) that formalizes this intuition is the problem of learning a mixture of subspaces (or subspace clustering). Given a set of points in n dimensions that belong to a union of $k \geq 2$ subspaces, the goal is to find the individual subspaces that contain all the points. When the points belong to $\mathbb{R}^n$, a beautiful result of Vidal (2003) shows that for any mixture of k subspaces, under some mild general-position assumption of the points in the subspaces,[1] there is an algorithm that runs in time $n^{O(k)}$ that recovers the k individual subspaces. Very recently, subspace clustering has also been studied with outlier noise, in the special case when the points in each cluster is drawn from a Gaussian supported on a subspace (Raghavendra and Yau, 2020; Bakshi and Kothari, 2020). However these guarantees are specific to the real domain. A natural question is whether such algorithmic guarantees also extend to other domains like $\mathbb{F}_2$.

Can we efficiently learn a mixture of subspaces over finite fields?

The algorithmic problem has a very different flavor over finite fields and becomes computationally challenging even in simple settings. In the simplest setting, we are given samples from a mixture of $k = 2$ unknown subspaces $A_0, A_1 \subseteq \mathbb{F}_2^n$ of dimension d_0, d_1 (respectively), with unknown mixing weights $w_0, w_1 \in [0, 1]$ that add up to 1. Each sample is drawn independently as follows: with probability w_0, the sample is drawn from U_{A_0}, the uniform distribution over subspace $A_0 \subseteq \mathbb{F}_2^n$, and with w_1 the sample is drawn from the uniform distribution U_{A_1} over $A_1 \subseteq \mathbb{F}_2^n$. The goal is to learn the individual subspaces A_0, A_1 from independent samples generated from this model. We refer the reader to Definition 4 for the formal definition of the model.

Learning mixtures of subspaces over $\mathbb{F}_2$ essentially generalizes the problem of learning mixtures of subcubes that was studied in (Chen and Moitra, 2019). In particular, subcubes correspond to (affine) subspaces where the constraints are given by standard unit vectors. On the other hand, in this work, we consider arbitrary subspaces of $\mathbb{F}_2^n$ (though we do not allow for affine subspaces). Our work can also be through the framework of *learning from positive examples* Denis et al. (2005); De et al. (2014); Canonne et al. (2020); Ernst et al. (2015) which studies the learnability of supervised concept classes (in this case subspaces) when the algorithm only gets positive samples.

More interestingly, the simple setting of $k = 2$ already captures the notorious problem of learning parities with noise (LPN) as a special case. One can encode LPN as learning a mixture of two subspaces A_0, A_1 where the subspaces $A_1 \subset A_0 \subseteq \mathbb{F}_2^n$ and $\dim(A_1) = \dim(A_0) - 1$ (see Proposition 21 and Proposition 20). The best known algorithm for LPN runs in time $\exp\left(O(n/\log n)\right)$ (Blum et al., 2003). Moreover LPN is also used as an average-case hardness assumption in learning theory and cryptography (Pietrzak, 2012). To avoid this computational barrier, we will assume that we are not in the degenerate setting when one subspace contains the other. We call the two subspaces A_0 and A_1 *incomparable* iff $A_0 \not\subseteq A_1$ and $A_1 \not\subseteq A_0$. This leads to the following natural question about the computational complexity of the problem:

Question. *Is LPN the only computational obstruction for learning a mixture of two subspaces? Can one design faster algorithms when the subspaces A_0, A_1 are incomparable?*

1. Such an assumption is necessary, to ensure that the individual subspaces are identifiable.

Our first result shows that one can indeed design a polynomial time algorithm when the two subspaces are incomparable.

Theorem 1 *There is an algorithm* INCOMPARABLE-SUBSPACE-RECOVERY *with the following guarantee: given oracle access to $\mathcal{O}(A_0, A_1, w_0, w_1)$ (for unknown A_0, A_1, w_0, w_1), $w_{min} > 0$ (such that $w_{min} \leq \min\{w_0, w_1\}$) and confidence parameter $\delta > 0$,*

1. INCOMPARABLE-SUBSPACE-RECOVERY *runs in sample and time complexity* $\mathsf{poly}(n/w_{min}) \cdot \log(1/\delta)$
2. *With probability $1-\delta$, the algorithm outputs the subspaces A_0, A_1, and estimates the weights w_0, w_1 up to any desired inverse polynomial accuracy.*

Hence the above result gives a significantly faster polynomial time algorithm if we are *not* in the degenerate *comparable* setting when one subspace contains the other. In contrast, when $A_1 \subset A_0$ and $dim(A_1) = dim(A_0) - 1$ (or vice versa), the best known algorithm takes $\exp(O(n/\log n))$ time. We remark that the algorithm succeeds in uniquely identifying and recovering the individual subspaces, as opposed to just finding a mixture of two subspaces that fits the data. In the parlance of statistics, our algorithm recovers the underlying model (sometimes referred to as *parameter estimation*) as opposed to just doing *density estimation*.

Next, observe that the (presumed) hardness of LPN only implies hardness of the subspace recovery problem when (i) $A_1 \subseteq A_0$ and (ii) $\mathsf{dim}(A_1) = \mathsf{dim}(A_0) - 1$. This naturally prompts the question whether subspace recovery remains hard if (say) $A_1 \subseteq A_0$ but $\mathsf{dim}(A_1) \ll \mathsf{dim}(A_0)$. In other words, we ask the following question:

Question. *Can we design fast algorithms for subspace recovery when $\mathsf{dim}(A_0)$ and $\mathsf{dim}(A_1)$ are substantially different? Note that we are not imposing any conditions on the comparability of the hidden subspaces A_0 and A_1.*

Our next result provides an affirmative answer to this question.

Theorem 2 *Let $w_{min} \geq 1/100$. Let $d_0 \geq d_1$ and suppose $\alpha := d_1/d_0 < 1 - \frac{\log d_0}{\sqrt{d_0}}$. There is an algorithm* SUBSPACE-RECOVER-LARGE-DIFF *with the following guarantee: given oracle access to $\mathcal{O}(A_0, A_1, w_0, w_1)$(for unknown A_0, A_1, w_0, w_1), $w_{min} > 0$ (such that $w_{min} \leq \min\{w_0, w_1\}$) and confidence parameter $\delta > 0$,*

1. SUBSPACE-RECOVER-LARGE-DIFF *runs in sample and time complexity* $\log(1/\delta)\mathsf{poly}(n) \cdot d_0^{O(1)/(1-\alpha)}$.
2. *With probability $1-\delta$, the algorithm outputs the subspaces A_0, A_1, and estimates the mixing weights up to any desired inverse polynomial accuracy.*

Informally speaking, if the ratio of dimensions α is bounded away from 1, the running time is polynomial. In general, the running time of the algorithm has a dependence of $O(1/(1-\alpha))$ in the exponent.

1.1. Overview of Techniques.

We now briefly describe the algorithmic ideas and techniques used to prove our results. The algorithms that establish Theorem 1 and Theorem 2 use very different ideas. We begin with an overview of Theorem 1.

Incomparable Setting (Theorem 1). The main component of the polynomial time algorithm in the incomparable setting is a careful procedure for *dimension reduction* that reduces the subspace clustering problem to $O(1)$ dimensions. We will construct a matrix $M \in \mathbb{F}_2^{r\times n}$ where $r = O(1)$ (in the actual proof, we set $r = 10$), and solve the clustering problem given samples of the form $y = Mx$ where x is drawn from the original mixture. Note that a subspace under any linear map M also gives a subspace; hence the samples in $\mathbb{R}^r$ are drawn from a mixture of subspaces MA_0 and MA_1. Any algorithm for learning a mixture of subspaces in $r = O(1)$ dimensions will allow us to cluster the points, and recover the individual subspaces A_0, A_1.

How do we choose the linear map M? A key property that we require of M is that if A_0 and A_1 are incomparable, then MA_0 and MA_1 should also remain incomparable. While it is not hard to see that such a M exists (even when $r = O(1)$), it is far from clear how to find it given that we do not have A_0 and A_1 explicitly. A natural choice for M is a random matrix, where every entry is chosen independently from $\mathbb{F}_2$. Random linear maps are often used for dimension reduction in the real domain to approximately preserve inner products and pairwise distances. However, a random map does not work in our setting, particularly when the target dimension $r \ll d_1$. This is because with high probability the subspaces collapse and $\mathbf{M}A_0 = \mathbf{M}A_1 = \mathbb{F}_2^r$, thereby making it impossible to recover the individual subspaces $\mathbf{M}A_0, \mathbf{M}A_1$.

Our approach instead proceeds in multiple rounds, where in each round, we reduce the dimension by one while preserving the property that the projected subspaces remain incomparable. More precisely, one can show that for a random linear map $\mathbf{M}_{n-1} \in \mathbb{F}_2^{(n-1)\times n}$, with *constant probability*, $\mathbf{M}_{n-1}A_0$ and $\mathbf{M}_{n-1}A_1$ are incomparable if A_0, A_1 are originally incomparable. However, this does not suffice per se, since we want to apply this for $\Omega(n)$ rounds (and thus, the probability of success becomes exponentially small). The crucial component of our algorithm is a testing procedure that runs in polynomial time, which given samples from a mixture of subspaces U, V, w.h.p. outputs whether U and V are *comparable* or *incomparable*. With such a procedure, in every phase we can reduce the dimension by 1, by sampling several random linear maps, running our testing procedure on each of them, and picking one that preserves incomparability of the subspaces. The guarantee of the testing procedure is given below.

Theorem 3 *There is an algorithm* TEST-COMPARABILITY *with the following guarantee: Given oracle access to $\mathcal{O}(U, V, w_U, w_V)$ (for unknown U, V, w_U, w_V), $w_{min} > 0$ (such that $\min\{w_U, w_V\} \geq w_{min}$) and confidence parameter $\delta > 0$,*

1. TEST-COMPARABILITY *runs in sample and time complexity $1/{w_{min}}^2 \cdot poly(n)\log(1/\delta)$.*
2. *With probability $1 - \delta$, the algorithm outputs* True *if U and V are comparable and* False *otherwise.*

The testing procedure uses the following main insight. Suppose for simplicity the span $\mathsf{span}(U \cup V) = \mathbb{F}_2^n$. We prove that the subspaces U and V are incomparable if and only if there exists a non-zero polynomial p of degree 2 that vanishes on $\mathcal{A} = U \cup V$. In fact, it will suffice to choose $\mathcal{A}$ to be a randomly chosen set of polynomial size sampled from the mixture of subspaces U and V. The set of feasible degree-2 polynomials can then be obtained by setting up a system of linear equations where the unknowns correspond to co-efficients of p.

Let us define $\mathbf{M} \in \mathbb{F}_2^{O(1)\times n}$ as $\mathbf{M} = \mathbf{M}_r \cdot \mathbf{M}_{r+1} \cdot \ldots \cdot \mathbf{M}_{n-1}$ – in other words, $\mathbf{M}$ is the linear map obtained by composing the dimension reduction maps over the $n - r$ rounds. Once the

dimension is reduced to $r = O(1)$, we use a brute-force algorithm to recover $\mathbf{M}A_0, \mathbf{M}A_1$. Finally, once we know $\mathbf{M}A_0, \mathbf{M}A_1$, we can draw uniform samples from $A_0 \backslash \{x \in A_0 : \mathbf{M}x \in \mathbf{M}A_1\}$ to recover A_0; we can recover A_1 similarly (see Lemma 16).

Significant dimension difference (Theorem 2). When the dimension of the subspaces are substantially different, we use algebraic ideas inspired from techniques in the real domain to recover the subspaces. The main algorithmic idea is by adapting ideas from related problem of subspace recovery over the reals (Hardt and Moitra, 2013; Bhaskara et al., 2019). To explain the idea, consider the setting with equal mixing weights of $1/2$, $d_0 \approx n$, and suppose $\alpha = 1 - \Omega(1)$. If we consider a random subsample of d_0 points from the data set, we expect to have roughly $d_0/2$ points from subspace A_0 and $d_0/2$ points from subspace A_1. Suppose $\alpha < 1/2$ (referred to as the "large gap case")i.e., $d_1 < d_0/2$, then with high probability there is a linear dependence in this sub-sample. Further, this linear dependence is (entirely) among points lying in the subspace A_1. This can be used to recover the subspace A_1 (and consequently, the subspace A_0 as well).

To see why this idea does not work in general, consider the case when the weights $w_0 = 0.9, w_1 = 0.1$ and $d_1 = 0.8d_0$. Then, to see a linear dependence among the points in A_1, we need to sample at least d_1 points from A_1. However, on an average, this will mean sampling around $(w_0/w_1) \cdot d_1 = 9d_1$ many points from A_0. As $9d_1$ is much larger than the ambient dimension and thus, we will find many *spurious* linear dependencies – i.e., dependencies which do not come from points belonging to A_1. Thus, this strategy will fail to identify A_1.

Instead, when $\alpha \geq 1/2$, we will adopt a *dimension gap amplification strategy*. In particular, we consider a non-linear map $\phi : \mathbb{F}_2^{d_0} \to \mathbb{F}_2^{d_0'}$ where $d_0' = \sum_{j=0}^{\ell} \binom{d_0}{j}$ for an appropriately chosen ℓ. Further, for a set B, let us define $\phi(B)$ as the set $\{\phi(x) : x \in B\}$. Roughly speaking, we want to choose an appropriate ℓ such that $\mathsf{dim}(\mathsf{span}(\phi(A_1)))/\mathsf{dim}(\mathsf{span}(\phi(A_0))) < 1/2$. For such an ℓ, we can now apply the strategy for the large gap case to recover A_1 and A_0. We note that the idea of such a dimension gap amplification was also applied in the related subspace recovery problem over reals (Bhaskara et al., 2019) – there, the goal was recover one subspace S of dimension $d \leq n$ containing $o(d/n)$ fraction of the points, while the rest of the points are drawn in general position from the whole of $\mathbb{R}^n$. While in spirit our idea is similar, it is challenging to get a handle on the dimensions of $\mathsf{span}(\phi(A_1))$ and $\mathsf{span}(\phi(A_0))$. In particular, the techniques of Bhaskara et al. (2019) which are meant for the reals, do not seem to be applicable in the finite field setting. Fortunately for us, some powerful results from additive combinatorics (Keevash and Sudakov, 2005; Ben-Eliezer et al., 2012) let us get precise estimates for $\mathsf{dim}(\mathsf{span}(\phi(A_0)))$ and $\mathsf{dim}(\mathsf{span}(\phi(A_1)))$. Roughly speaking, we show that for $\ell \approx 1/(1-\alpha)$, $\mathsf{dim}(\mathsf{span}(\phi(A_1)))/\mathsf{dim}(\mathsf{span}(\phi(A_0))) < 1/2$, thus reducing to the large gap case.

2. Preliminaries

We start by defining the subspace recovery problem formally.

Definition 4 *The Subspace-Recovery problem is instantiated by two subspaces of $\mathbb{F}_2^n$ - A_0 and A_1 of dimensions d_0 and d_1 respectively. In addition, we also have weights w_0 and w_1 such that $w_0 + w_1 = 1$.*

The subspaces A_0, A_1, dimensions d_0, d_1 as well as the weights w_0 and w_1 are unknown. For this instance, we define the sampling oracle $\mathcal{O}(A_0, A_1, w_0, w_1)$ is defined as follows: sample $\mathbf{b} \in \{0, 1\}$ where $\Pr[\mathbf{b} = 0] = w_0$ and $\Pr[\mathbf{b} = 1] = w_1$. If $\mathbf{b} = 0$, $\mathcal{O}(A_0, A_1, w_0, w_1)$ outputs a

uniformly random element from A_0 and if $\mathbf{b} = 1$, $\mathcal{O}(A_0, A_1, w_0, w_1)$ outputs a uniformly random element from A_1.

In the Subspace-Recovery problem, the algorithm is given access to the sampling oracle $\mathcal{O}(A_0, A_1, w_0, w_1)$, an error parameter $\epsilon > 0$ and a weight parameter $w_{min} > 0$ with the promise that $w_{min} \leq \min\{w_0, w_1\}$. The goal of the algorithm is to output subspaces A_0, A_1 and estimates $\hat{w}_0$, $\hat{w}_1$ such that $|w_0 - \hat{w}_0| + |w_1 - \hat{w}_1| \leq \epsilon$.

Without loss of generality, we will assume $d_0 \geq d_1$ from now on.

Remark 5 *Note that once A_0, A_1 is found, estimating w_0, w_1 is not hard, this is because $\mathbb{P}_{\mathbf{x}\sim\mathcal{O}(A_0,A_1,w_0,w_1)}[\mathbf{x} \in A_0 \setminus A_1] = w_0 \frac{|A_0 \setminus A_1|}{|A_0|}$. Formally, there is an algorithm with the following guarantee: given oracle access to $\mathcal{O}(A_0, A_1, w_0, w_1)$ (for unknown w_0, w_1), A_0, A_1 and confidence parameter $\delta > 0$,*

1. *this algorithm runs in sample and time complexity $\mathsf{poly}(n) \cdot 1/\epsilon^2 \cdot \log(1/\delta)$*
2. *With probability $1 - \delta$, the algorithm outputs $\hat{w}_0$, $\hat{w}_1$ such that $|w_0 - \hat{w}_0| + |w_1 - \hat{w}_1| \leq \epsilon$.*

By this observation, we can focus on finding A_0, A_1 from now on.

We next define the concept of incomparable subspaces.

Definition 6 *We define two subspaces A, B to be incomparable if and only if $A \nsubseteq B$ and $B \nsubseteq A$.*

2.0.1. SOME USEFUL NOTATION

1. For any $f : \mathbb{F}_2^n \to \mathbb{F}_2$, we use $\mathsf{zero}(f)$ to denote the set $\{x : f(x) = 0\}$.
2. For integers $n, d \in \mathbb{N}$, we use $\mathsf{RM}(n, d)$ to denote the set of polynomials of degree at most d over $\mathbb{F}_2^n$.
3. For integers $n, k \in \mathbb{N}$ with $n \geq k$, we use $\binom{n}{\leq k}$ to denote $\sum_{i=0}^{k} \binom{n}{i}$.
4. For a sample oracle $\mathcal{O}$ which return samples in $\mathbb{F}_2^n$, matrix $D \in \mathbb{F}_2^{k\times n}$, we use $D\mathcal{O}$ to denote a new sample oracle which each time returns $D\mathbf{x}$ where $\mathbf{x}$ is sampled from $\mathcal{O}$.
5. For an index set S, we use x_S to denote the set $\{x_i : i \in S\}$.
6. For a set S of vectors, we use $\mathsf{rank}(S)$ to denote $\dim(\mathsf{span}(S))$.

2.0.2. SOME USEFUL FACTS REGARDING POLYNOMIALS

We next list some useful facts regarding polynomials over the field $\mathbb{F}_2$. While most of these are easy and standard, we list them here for the sake of completeness.

Claim 7 *Let p be a polynomial over $\mathbb{F}_2^n$. If the polynomial p is not identically zero (as a formal expression) and its degree is at most c, then*

$$\mathbb{P}_{\mathbf{x}\sim\mathbb{F}_2^n}[p(\mathbf{x}) \neq 0] \geq 1/2^c.$$

Proof The proof is by induction on degree. If $c = 0$, then p is identically 1 and thus the claim follows trivially.

Now, as an inductive hypothesis, assume that the claim is true for all polynomials of degree at most $c-1$. Let p be a polynomial of degree c. Since p is not identically zero, there exists i such that p can be expressed as

$$p(x_1, \cdots, x_n) = q(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \cdot x_i + r(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), \quad (1)$$

where degree of q is at most $c-1$ and q is not identically zero. The above formulation uses the fact that polynomials over $\mathbb{F}_2$ are multilinear. Observe that any choice of $\mathbf{x}_{-i} = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n)$ such that $q(\mathbf{x}_{-i}) \neq 0$,

$$\Pr_{\mathbf{x}_i \sim \mathbb{F}_2}[p(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n) \neq 0] \geq \frac{1}{2}. \quad (2)$$

Now, applying the induction hypothesis on the polynomial $q(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, we have that

$$\Pr_{\mathbf{x} \sim \mathbb{F}_2^n}[q(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n) \neq 0] \geq \frac{1}{2^{c-1}}.$$

Combining this with (1) and (2), we get the claim. ■

Claim 8 *There is an efficient algorithm* SIZE-SYSTEM-POLYNOMIAL *which given a set of points as input $z_1, \ldots, z_R \in \mathbb{F}_2^n$, determines the size of the set $T = |\{p \in \mathsf{RM}(n,2) : p(z_1) = p(z_2) = \cdots = p(z_r) = 0\}|$.*

Proof Observe that p can be expressed as linear system of equations (i) where the unknowns are the coefficients of p and (ii) the equations are given by the constraints $\{p(z_i) = 0\}_{1 \leq i \leq R}$. Using Gaussian elimination, we can determine the rank r of this system. Observe that the size of T is just 2^r, thus proving the claim. ■

2.0.3. SOME USEFUL FACTS REGARDING SUBSPACES OF $\mathbb{F}_2^n$

We now list some useful facts about subspaces of $\mathbb{F}_2^n$.

Claim 9 *Let $k, d, n \in \mathbb{N}$ such that $k \geq 100d$. Let $V \subseteq \mathbb{F}_2^n$ be a subspace of dimension d. Let $\mathbf{x}_1, \cdots, \mathbf{x}_k$ be k vectors sampled uniformly at random from V. Then,*

$$\mathbb{P}_{\mathbf{x}_1, \cdots, \mathbf{x}_k}[\forall S \subseteq [k] \text{ such that } |S| \geq 0.9k, \text{ we have } \mathsf{span}(\mathbf{x}_S) = V] \geq 1 - 2^{0.4k}. \quad (3)$$

Proof We know that there always exist a linear bijection between V and $\mathbb{F}_2^d$. Without loss of generality, we assume $n = d, V = \mathbb{F}_2^d$. Without loss of generality, assume 0.9k is a integer. For a fixed S with $|S| = 0.9k$

$$\begin{aligned}
&\mathbb{P}[\mathsf{span}(\mathbf{x}_S) = \mathbb{F}_2^d] \\
&= \prod_{j=0}^{d-1} \left(1 - 2^{-0.9k+j}\right) && \text{See (Ferreira et al., 2012, Equation (2))} \\
&\geq 1 - \sum_{j=0}^{d-1} 2^{-0.9k+j} \geq 1 - 2^{-0.9k+d} \geq 1 - 2^{-0.89k}.
\end{aligned}$$

The number of choice of S is at most $\binom{k}{0.1k} \leq (10e)^{0.1k} \leq 2^{0.48k}$. Then the proof is completed by a union bound. ■

The next claim says that a union of two proper subspaces of $\mathbb{F}_2^n$ must differ substantially from any subspace of $\mathbb{F}_2^n$.

Claim 10 *Let S be a subspace of $\mathbb{F}_2^n$ and of dimension d. Let $U, V \subsetneq S$ be two proper subspaces. Then $|S \backslash (U \cup V)| \geq 2^{d-2}$.*

Proof Notice that the size of subspace in $\mathbb{F}_2$ is always a power of 2. There are two cases:
Case 1: $\dim(U) = \dim(V) = d - 1$.
Observe that $\dim(U \cap V) \geq d - 2$ and hence $|U \cup V| = |U| + |V| - |U \cap V| \leq 3 \cdot 2^{d-2}$.
Case 2: At least one of $\dim(U)$ or $\dim(V) \leq d - 2$.
In this case, $|U \cup V| \leq |U| + |V| \leq 2^{d-1} + 2^{d-2} \leq 3 \cdot 2^{d-2}$. Thus, in either case, $|U \cup V| \leq 3 \cdot 2^{d-2}$ which implies that $|S \backslash (U \cup V)| \geq 2^{d-2}$. ■

Claim 11 *Let $b_1, \cdots, b_t \in \mathbb{F}_2^n$ be linearly independent. Sample $\mathbf{M} \in \mathbb{F}_2^{m \times n}$ uniformly at random. Then $\mathbf{M}b_1, \cdots, \mathbf{M}b_t$ are independent and identically distributed. In other words, the joint distribution of $\mathbf{M}b_1, \cdots, \mathbf{M}b_t$ is the uniform distribution over $\mathbb{F}_2^{m \times t}$.*

Proof Let us first add vectors $b_{t+1}, \ldots, b_n$ such that $\{b_1, \ldots, b_n\}$ is a basis of $\mathbb{F}_2^n$. Let B be the matrix whose i^{th} column is b_i. Now, observe that the map $\Psi : \mathbb{F}_2^{m \times n} \to \mathbb{F}_2^{m \times n}$ defined as $\Psi : M \mapsto M \cdot B$ is a bijection. Thus, if the random variable $\mathbf{M}$ is uniform over $\mathbb{F}_2^{m \times n}$, then so is $\mathbf{M} \cdot B$. Consequently, the first t columns of $\mathbf{M} \cdot B$, namely, $\mathbf{M}b_1, \ldots, \mathbf{M}b_t$ are independent and identically distributed.

■

The following theorem gives a hypothesis testing routine for mixtures of subspaces over $\mathbb{F}_2^n$. The proof of this theorem is deferred to Appendix A.

Theorem 12 *Let $\mathbf{D}$ be a distribution of a mixture of two incomparable subspaces $A, B \subseteq \mathbb{F}_2^n$ with mixing weights $w_A, w_B \geq w_0$. Let $\{A_j, B_j\}_{j=1}^N$ be a collection of N sets of hypothesis with the property that there exists i such that $\{A_i, B_i\} = \{A, B\}$. There is an algorithm* CHOOSE-THE-RIGHT-HYPOTHESIS *which is given a confidence parameter δ, w_0, $\{A_j, B_j\}_{j=1}^N$ and a sampler for $\mathbf{D}$. Every subspace of $\{A_j, B_j\}_{j=1}^N$ will be represented by a basis of that subspace, and the algorithm will have the access to the basis. This algorithm has the following behavior,*

1. *It runs in $\mathsf{poly}(N, 1/w_0) \log(1/\delta)$ time.*
2. *With the probability $1 - \delta$ outputs the index i such that $\{A_i, B_i\} = \{A, B\}$.*

3. Testing Comparability of the Subspaces

In this section, the main goal is to prove Theorem 3 (restated below for the convenience of the reader). We recall that Theorem 3 gives an efficient algorithm which given samples from a mixture of two subspaces U, V, decides whether U and V are comparable. This result in turn is an important piece in our subspace recovery algorithm in the "incomparable" case. The algorithm TEST-COMPARABILITY is described in Figure 1.

Theorem 3 *There is an algorithm* TEST-COMPARABILITY *with the following guarantee: Given oracle access to $\mathcal{O}(U, V, w_U, w_V)$ (for unknown U, V, w_U, w_V), $w_{min} > 0$ (such that $\min\{w_U, w_V\} \geq w_{min}$) and confidence parameter $\delta > 0$,*

1. TEST-COMPARABILITY *runs in sample and time complexity $1/{w_{min}}^2 \cdot poly(n) \log(1/\delta)$.*
2. *With probability $1 - \delta$, the algorithm outputs* True *if U and V are comparable and* False *otherwise.*

The main idea of the algorithm is the following. First we take a few samples from the mixture to get $\mathsf{span}(U \cup V)$. By dimension reduction, it suffices to deal with the case $\mathsf{span}(U \cup V) = \mathbb{F}_2^n$. The crucial property we use is the following: If $\mathsf{span}(U \cup V) = \mathbb{F}_2^n$, U, V are incomparable iff there exists non-zero $p \in \mathsf{RM}(n, 2)$ such that p vanishes on the entire set $U \cup V$. The proof of Theorem 3 is deferred to the end of the section – to start, we prove some auxiliary lemmas.

Algorithm 1: TEST-COMPARABILITY

Input:
n – ambient dimension
$\mathcal{O}(U, V, w_U, w_V)$ – oracle for random samples from mixture of subspaces.
w_{min} – lower bound of two mixture weights.
Output: True (if comparable) or False (if incomparable)

Set $t = 16n/({w_{min}}^2)$;
Sample $\mathbf{x}_1, \cdots, \mathbf{x}_t$ from $\mathcal{O}(U, V, w_U, w_V)$;
Set $S = \mathsf{span}(\mathbf{x}_1, \cdots, \mathbf{x}_t)$, $v = dim(S)$;
Find $y_1, \cdots, y_v$ such that they form a basis of $S = \mathsf{span}(\mathbf{x}_1, \cdots, \mathbf{x}_t)$.;
Find a matrix $D \in \mathbb{F}_2^{v \times n}$ such that $Dy_i = e_i$ for all i, where e_i is the ith element of the standard basis of $\mathbb{F}_2^v$.;
Set $\mathcal{O}' = D\mathcal{O}(U, V, w_U, w_V) = \mathcal{O}(DU, DV, w_U, w_V)$;
Set $r = 8n^2/w_{min}$;
Sample $\mathbf{z}_1, \cdots, \mathbf{z}_r$ from $\mathcal{O}' = \mathcal{O}(DU, DV, w_U, w_V)$;
Use algorithm SIZE-SYSTEM-POLYNOMIAL to compute $T = |\{p \in \mathsf{RM}(v, 2) : p(\mathbf{z}_1) = p(\mathbf{z}_2) = \cdots = p(\mathbf{z}_r) = 0\}|$;
`// See Claim 8`
. **if** $T = 1$ **then**
  return True;
else
  return False;
end

Claim 13 *Assume $s \geq 8n/w_{min}$. Let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_s$ be sampled from a mixture of two subspaces $U, V \subseteq \mathbb{F}_2^n$(potentially comparable) of dimension at most d with mixing weights $w_U, w_V \geq w_{min}$. Then, with probability at least $1 - \exp(-s{w_{min}}^2/32)$, $\mathsf{span}(\mathbf{x}_1, \cdots, \mathbf{x}_s) = \mathsf{span}(U \cup V)$.*

Proof For fixed $x_1, \cdots, x_i$ such that $\mathsf{span}(x_1, \cdots, x_i) \subsetneq \mathsf{span}(U \cup V)$, we will show

$$\mathbb{P}_{\mathbf{x}_{i+1}}[\mathbf{x}_{i+1} \notin \mathsf{span}(x_1, \cdots, x_i)] \geq w_{min}/2. \tag{4}$$

Define $W = \mathsf{span}(x_1, \cdots, x_i)$. By our assumption, either $U \nsubseteq W$ or $V \nsubseteq W$. Let us assume that it is the former (the other case is symmetric). Under this assumption, $U \cap W$ is a proper subset of U. Since both are linear subspaces and the size of any linear space over $\mathbb{F}_2$ is always a power of 2, $|U \cap W| \leq 0.5|U|$. Hence

$$\mathbb{P}[\mathbf{x}_{i+1} \in U \backslash W] \geq w_U \frac{|U \backslash W|}{|U|} \geq w_{min} \cdot 0.5.$$

In other words, $rank(x_1, \cdots, \mathbf{x}_{i+1}) = rank(x_1, \cdots, x_i) + 1$ will hold with probability at least $w_{min}/2$, thus proving (4). Define $\mathbf{y}_i = \mathsf{rank}(\mathbf{x}_1, \cdots, \mathbf{x}_i) - \mathsf{rank}(\mathbf{x}_1, \cdots, \mathbf{x}_{i-1})$, then $\mathbf{y}_1, \cdots, \mathbf{y}_s$ satisfy the condition of Lemma 25 with $\gamma = w_{min}/2, d = \mathsf{rank}(U \cup V), k = s$. Claim 13 now follows by applying Lemma 25. ■

The next (easy) claim says that suppose the distribution $\mathbf{Z}$ (over $\mathbb{F}_2^d$) is *not too concentrated on any single element*. Then, a randomly chosen set of size roughly quadratic in d is a *hitting set* for quadratic polynomials over $\mathbb{F}_2^d$. In other words, any non-zero element of $\mathsf{RM}(d, 2)$ is non-zero on at least one element of this set.

Claim 14 *Let $\mathbf{Z}$ be a distribution over $\mathbb{F}_2^d$ such that the probability weight of every element is at least $w^*/2^d$. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$ be independent sampled from $\mathbf{Z}$. Then, we have*

$$\mathbb{P}\left[\forall q \in \mathsf{RM}(d,2) \setminus \{0\}, \exists j \in [t] \text{ s.t. } q(\mathbf{x}_j) \neq 0\right] \geq 1 - \exp\left(-tw^*/4 + \binom{d}{\leq 2} \log 2\right).$$

Proof Fix $q \in \mathsf{RM}(d, 2)$ such that $q \neq 0$. By Claim 7,

$$\mathbb{P}_{\mathbf{x} \sim_u \mathbb{F}_2^d}[q(\mathbf{x}) = 1] \geq 1/4.$$

As a consequence,

$$\mathbb{P}_{\mathbf{x} \sim Z}[q(\mathbf{x}) = 0] \leq 1 - \frac{w^*}{4}.$$

Hence

$$\mathbb{P}[q(\mathbf{x}_1) = \cdots = q(\mathbf{x}_t) = 0] \leq (1 - w^*/4)^t \leq \exp(-tw^*/4).$$

Notice that $|\mathsf{RM}(d, 2)| = 2^{\binom{d}{\leq 2}}$. Using the union bound, we get the claim. ■

We are now ready to finish the proof of Theorem 3.

Proof of Theorem 3. Without loss of generality, we assume $\delta = 0.1$, since we can always boost the probability at a multiplicative cost of $\log(1/\delta)$. By Claim 13, we know that $S = \mathsf{span}(U \cup V)$ (defined in Step 3 of the algorithm) with probability 0.999. Henceforth, we assume that $S = \mathsf{span}(U \cup V)$ holds.

By definition, D (defined in Step 5 of the algorithm) is a linear bijection between S and $\mathbb{F}_2^v$. Hence DU, DV are incomparable if and only if U, V are incomparable. Now observe that, $\mathcal{O}' = \mathcal{O}(DU, DV, w_U, w_V)$ will give samples from mixture of two subspaces DU, DV with mixing weights $w_U, w_V \geq w_{min}$. Notice that $\mathsf{span}(DU \cup DV) = \mathbb{F}_2^v$. We divide the rest of the analysis into two cases.

Case 1: DU, DV are comparable.
We have $DU = \mathbb{F}_2^v$ or $DV = \mathbb{F}_2^v$. By Claim 14, with probability 0.999, there will only be one polynomial (the zero polynomial) in the set $\{p \in \mathsf{RM}(v,2) : p(\mathbf{z}_1) = p(\mathbf{z}_2) = \cdots = p(\mathbf{z}_r) = 0\}$. In this case, $T = 1$. Thus, overall, with probability 0.998, algorithm returns the correct answer in this case.
Case 2: DU, DV are incomparable.
In this case, $dim(DU) \leq v - 1$ (and $dim(DV) \leq v - 1$). Thus, there exists non-zero vector b_U (resp. b_V) such that $\langle b_U, DU\rangle = \{0\}$ (resp. $\langle b_V, DV\rangle = \{0\}$). Now, consider the non-zero polynomial $p(x) = \langle b_U, x\rangle\langle b_V, x\rangle$. By definition it satisfies $p(DU \cup DV) = \{0\}$. Thus, in this case, the set $\{p \in \mathsf{RM}(v,2) : p(\mathbf{z}_1) = p(\mathbf{z}_2) = \cdots = p(\mathbf{z}_r) = 0\}$ has at least two elements. Thus, overall, with probability 0.999, the algorithm returns the correct answer in this case. ■

4. Learning Mixtures of Incomparable Subspaces

In this section, we give a polynomial time algorithm (Algorithm 2: INCOMPARABLE-SUBSPACE-RECOVERY) for recovering the subspaces A_0, A_1 when given access to samples from a mixture of two subspaces that are incomparable. We prove the following theorem.

Theorem 1 *There is an algorithm* INCOMPARABLE-SUBSPACE-RECOVERY *with the following guarantee: given oracle access to $\mathcal{O}(A_0, A_1, w_0, w_1)$ (for unknown A_0, A_1, w_0, w_1), $w_{min} > 0$ (such that $w_{min} \leq \min\{w_0, w_1\}$) and confidence parameter $\delta > 0$,*

1. INCOMPARABLE-SUBSPACE-RECOVERY *runs in sample and time complexity* $\mathsf{poly}(n/w_{min}) \cdot \log(1/\delta)$
2. *With probability $1 - \delta$, the algorithm outputs the subspaces A_0, A_1, and estimates the weights w_0, w_1 up to any desired inverse polynomial accuracy.*

The main idea is a new procedure for *dimension reduction* that reduces the subspace clustering problem to $O(1)$ dimensions. We will construct a linear map $M \in \mathbb{F}_2^{10\times n}$ such that after projecting using M, the subspaces obtained $MA_0 = \{Mx : x \in A_0\}$ and $MA_1 = \{Mx : x \in A_1\}$ are incomparable. The construction of M involves multiple rounds. In each round, we use Algorithm TEST-COMPARABILITY (and Theorem 3) as a black-box, and find a projection that brings down the dimension by one with high probability, while maintaining incomparability of the subspaces. Once we recover the subspaces MA_0, MA_1 in $O(1)$ dimensions (using a brute force algorithm: enumerate all possible pairs of subspace, then use Theorem 12), we can then recover the original subspaces A_0, A_1 by considering samples in $A_0 \cup A_1$ which are not mapped to $MA_0 \cap MA_1$ by M. We defer the proof of Theorem 1 to the end of section.

The following lemma is crucial in establishing Theorem 1. The lemma proves that with high probability, Algorithm FIND-A-GOOD-PROJECTOR (Algorithm 3) reduces the dimension to $r = 10$ while *preserving the incomparability* of the subspaces. If M is randomly chosen from $\mathbb{F}_2^{10\times n}$, then $MA_1 \subseteq MA_0$ since MA_0 collapses to $\mathbb{F}_2^{10}$ with high probability. Algorithm FIND-A-GOOD-PROJECTOR instead proceeds in multiple rounds, and reduces the dimension one per round. If the projector $\mathbf{M}'$ is chosen uniformly at random from $\mathbb{F}_2^{(n-1)\times n}$, with constant probability $\mathbf{M}'A_0, \mathbf{M}'A_1 \in \mathbb{F}_2^{n-1}$ remain incomparable. We can now use Algorithm TEST-COMPARABILITY (and Theorem 3)

Algorithm 2: INCOMPARABLE-SUBSPACE-RECOVERY

Input:
n – ambient dimension.
$\mathcal{O}(A_0, A_1, w_0, w_1)$ – oracle for random samples from mixture of subspaces.
w_{min} – lower bound of two mixture weights.
Output: two subspaces.

M=FIND-A-GOOD-PROJECTOR($n, \mathcal{O}(A_0, A_1, w_0, w_1), w_{min}$);
Use brute force to solve INCOMPARABLE-SUBSPACE-RECOVERY($10, M\mathcal{O}(A_0, A_1, w_0, w_1), w_{min}$), let U, V be the output ;
Set $t = 100n/w_{min}$;
Sample $\mathbf{x}_1, \cdots, \mathbf{x}_t$ from $\mathcal{O}(A_0, A_1, w_0, w_1)$;
return $\mathsf{span}(\{\mathbf{x}_i : M\mathbf{x}_i \notin V\}), \mathsf{span}(\{\mathbf{x}_i : M\mathbf{x}_i \notin U\})$;

to boost the success probability in each round by repeatedly sampling M' and rejecting it if the resulting subspaces are comparable.

Lemma 15 *Given samples from a mixture of two incomparable subspaces $A_0, A_1 \subseteq \mathbb{F}_2^n$ with mixing weights $w_0, w_1 \geq w_{min}$. There exists $M \in \mathbb{F}_2^{10\times n}$ such that MA_0, MA_1 are incomparable subspaces. Moreover, there is an algorithm* FIND-A-GOOD-PROJECTOR *that runs in time $1/w_{min} \cdot poly(n)$ and find such a M with probability at least 0.999.*

Algorithm 3: FIND-A-GOOD-PROJECTOR

Input:
n – ambient dimension
$\mathcal{O}(A_0, A_1, w_0, w_1)$ – oracle for random samples from mixture of subspaces.
w_{min} – lower bound of two mixture weights.
Output: a matrix $M \in \mathbb{F}_2^{10\times n}$.

Set $M = I_n$, where $I_n \in \mathbb{F}_2^{n\times n}$ is the identity matrix;
for $i = n; i > 10; i = i - 1$ **do**
  Sample $\mathbf{T} \in \mathbb{F}_2^{(i-1)\times i}$ uniformly at random;
  while TEST-COMPARABILITY($i, \mathbf{T}M\mathcal{O}(A_0, A_1, w_0, w_1), w_{min}, 1/n^2$) `// the last parameter is the failure probability we want.` **do**
    Sample $\mathbf{T} \in \mathbb{F}_2^{(i-1)\times i}$ uniformly at random;
  end
  $M = \mathbf{T}M$;
end
return M;

Proof We now show that Algorithm FIND-A-GOOD-PROJECTOR runs in polynomial time and finds a required projector M with high probability. Observe that from Theorem 3, every call of

TEST-COMPARABILITY (in step 4 of Algorithm 3) fails with probability at most $\delta = O(1/n^2)$. We will prove that at any iteration $i \in \{n, n-1, \ldots, 11\}$, a randomly chosen matrix $\mathbf{T} \in \mathbb{F}_2^{(i-1)\times i}$ (in step 3) succeeds with constant probability in preserving the incomparability of the subspaces. This ensures that it will suffice to sample $O(\log n)$ many random T per round before we succeed in that round (and hence $O(n \log n)$ overall).

Fix an iteration $i \in \{n, n-1, \ldots, 11\}$, and let $M \in \mathbb{F}_2^{i\times n}$ be the current projector. Let $U := MA_0, V := MA_1$, and assume U, V are incomparable. We show the following claim.

Claim: *For a random* $\mathbf{T} \in \mathbb{F}_2^{(i-1)\times i}$ *chosen in step 3,*

$$\mathbb{P}_{\mathbf{T}}[\mathbf{T}U, \mathbf{T}V \text{ are incomparable}] \geq 9/128. \tag{5}$$

We now prove the claim by considering two cases depending on the rank of $U \cup V$ i.e., the dimension of the span of $U \cup V$.

Case 1: $rank(U \cup V) \leq i-1$**.**
Let $v = rank(U \cup V)$ and $b_1, \cdots, b_v$ be a basis of $\mathsf{span}(U \cup V)$. By Claim 11, $\mathbf{T}b_1, \cdots, \mathbf{T}b_v$ can be viewed as being sampled independently from $\mathbb{F}_2^{i-1}$. A uniformly random matrix from $\mathbb{F}_2^{(i-1)\times(i-1)}$ is full-rank with probability at least $\prod_{j\geq 1}(1-2^{-j}) \geq 1/4$. Hence,

$$\mathbb{P}[\mathbf{T}b_1, \cdots, \mathbf{T}b_v \text{ are linearly independent}] \geq 1/4.$$

When $\mathbf{T}b_1, \cdots, \mathbf{T}b_v$ are linearly independent, $\mathbf{T}U, \mathbf{T}V$ are incomparable as required. This establishes (5) in Case 1.

Case 2: $rank(U \cup V) = i$**.**
Let $b_1, \ldots, b_{dim(U\cap V)}$ be a basis of $U \cap V$. We extend the basis such that $b_1, \ldots, b_{dim(U\cap V)}, c_1, \ldots, c_{dim(U)-dim(U\cap V)}$ is a basis of U, and similarly we extend the basis so that $b_1, \ldots, b_{dim(U\cap V)}, d_1, \ldots, d_{dim(V)-dim(U\cap V)}$ is a basis of V. Observe that $b_1, \ldots, b_{dim(U\cap V)}, c_1, \ldots, c_{dim(U)-dim(U\cap V)}, d_1, \ldots, d_{dim(V)-dim(U\cap V)}$ is a basis of $\mathsf{span}(U \cup V)$. Reorder this basis to get $a_1, \ldots, a_i$ such that $a_{i-1} = c_1, a_i = d_1$. Let $\mathbf{t}_j$ denote $\mathbf{T}a_j$. By Claim 11, $\mathbf{t}_1, \cdots, \mathbf{t}_i$ are independent and identically distributed. Let $\mathcal{E}$ be the event

$$\mathcal{E} = \begin{cases} \mathbf{t}_j \notin \mathsf{span}(\mathbf{t}_1, \cdots, \mathbf{t}_{j-1}) & \forall 1 \leq j \leq i-3 \\ \mathbf{t}_{i-2} \in \mathsf{span}(\mathbf{t}_1, \cdots, \mathbf{t}_{i-3}) & \\ \mathbf{t}_{i-1} \notin \mathsf{span}(\mathbf{t}_1, \cdots, \mathbf{t}_{i-2}) & \\ \mathbf{t}_i \notin \mathsf{span}(\mathbf{t}_1, \cdots, \mathbf{t}_{i-1}) & \end{cases}$$

Then,

$$\mathbb{P}_{\mathbf{T}}[\mathcal{E}] = (\prod_{j=1}^{i-3}(1 - 2^{j-1}/2^{i-1})) \cdot 1/4 \cdot 3/4 \cdot 1/2 \geq 3/4 \cdot 3/32 = 9/128.$$

Condition on $\mathcal{E}$. We now show that $\mathbf{T}U, \mathbf{T}V$ are incomparable as required. We will show $\mathbf{T}U \nsubseteq \mathbf{T}V$, the other direction is similar. By definition $\mathbf{t}_{i-1} = \mathbf{T}a_{i-1} = Tc_1 \in TU$, and $\mathbf{t}_{i-1} \notin \mathsf{span}(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_{i-2}, \mathbf{t}_i)$. However $\mathbf{T}V \subseteq \mathsf{span}(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_{i-2}, \mathbf{t}_i)$, hence $\mathbf{t}_{i-1} \notin \mathbf{T}V$, $\mathbf{T}U \nsubseteq \mathbf{T}V$. This establishes (5). Hence the lemma follows. ■

The following lemma shows that a few samples drawn uniformly from $S \setminus T$ suffice to recover S with high probability. This will allow us to recover A_0 and A_1 after clustering the points in $MA_0 \cup MA_1$.

Lemma 16 *Let S be a subspace of $\mathbb{F}_2^n$ and of dimension d. Let T be a proper subspace of S. Let $t \geq 8n$ be a integer. $\mathbf{x}_1, \cdots, \mathbf{x}_t$ are independently uniformly sampled from $S \backslash T$. Then,*

$$\mathbb{P}[\mathsf{span}(\mathbf{x}_1, \cdots, \mathbf{x}_t) = S] \geq 1 - e^{-t/128}.$$

Proof Let $V \subsetneq S$ be a fixed subspace. Then by Claim 10, $|S \backslash (T \cup V)| \geq 2^{d-2}$, which is at least $1/4$ of $|S|$. We have

$$\mathbb{P}_{\mathbf{x} \sim_u S \backslash T}[\mathbf{x} \notin V] \geq 1/4.$$

In other words, if $\mathsf{span}(\mathbf{x}_1, \cdots, \mathbf{x}_k) \neq S$, then $rank(\mathbf{x}_1, \cdots, \mathbf{x}_{k+1}) = rank(\mathbf{x}_1, \cdots \mathbf{x}_k) + 1$ will hold with probability at least $1/4$. Define the random variables $\mathbf{y}_i = \mathsf{rank}(\mathbf{x}_1, \cdots, \mathbf{x}_i) - \mathsf{rank}(\mathbf{x}_1, \cdots, \mathbf{x}_{i-1})$ for $i \in \{1, 2, \ldots, t\}$. Note that $\mathbf{y}_1, \cdots, \mathbf{y}_t$ are not quite independent (since the probability the rank increases at step i depends on the random choices of $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$ in previous iterations). But they satisfy the condition of Lemma 25 with $\gamma = 1/4, d = \dim(S), k = t$. The proof is completed after applying Lemma 25. ■

We are now ready to complete the proof of Theorem 1.

Proof of Theorem 1. Without loss of generality, we assume $\delta = 0.1$, since we can always boost the probability at a multiplicative cost of $\log(1/\delta)$. By Lemma 15, M satisfies the property that MA_0, MA_1 are incomparable with high probability (probability at least 0.999, say). Moreover assuming MA_0, MA_1 are incomparable, the brute force algorithm will return them with high probability.

Let $U = MA_0, V = MA_1$. We will show that $\mathsf{span}(\{\mathbf{x}_i : M\mathbf{x}_i \notin V\} = A_0$ with probability 0.998. Observe that $W = \{x \in A_0 : Mx \in MA_1\}$ is a proper subspace of A_0. Hence if $\mathbf{x}$ is drawn uniformly from A_0, $\mathbf{x}$ will not in W with probability at least $1/2$. By Chernoff bound, we expect to see at least $20n$ samples in $\{\mathbf{x}_i : M\mathbf{x}_i \notin V\}$ with probability 0.999 and all these samples can be viewed as uniformly drawn from $A_0 \backslash W$. By Lemma 16, $\mathsf{span}(\{\mathbf{x}_i : M\mathbf{x}_i \notin MA_1\} = A_0$ with probability 0.998. A similar argument shows that the algorithm also recovers A_1 with high probability. Finally, after recovering A_0, A_1 it is also easy to estimate the weights w_0, w_1 to inverse polynomial accuracy (see Remark 5). ■

5. Mixtures of two subspaces with signficant dimension difference

In this section, we prove Theorem 2 (restated below for convenience of the reader) which shows that there is a computationally efficient algorithm for learning a mixture of two subspaces with significantly different dimensions. Note that the following theorem does *not* assume that the two subspaces are incomparable.

Theorem 2 *Let $w_{min} \geq 1/100$. Let $d_0 \geq d_1$ and suppose $\alpha := d_1/d_0 < 1 - \frac{\log d_0}{\sqrt{d_0}}$. There is an algorithm* SUBSPACE-RECOVER-LARGE-DIFF *with the following guarantee: given oracle access to $\mathcal{O}(A_0, A_1, w_0, w_1)$(for unknown A_0, A_1, w_0, w_1), $w_{min} > 0$ (such that $w_{min} \leq \min\{w_0, w_1\}$) and confidence parameter $\delta > 0$,*

1. SUBSPACE-RECOVER-LARGE-DIFF *runs in sample and time complexity* $\log(1/\delta)\mathsf{poly}(n) \cdot d_0^{O(1)/(1-\alpha)}$.

2. *With probability $1-\delta$, the algorithm outputs the subspaces A_0, A_1, and estimates the mixing weights up to any desired inverse polynomial accuracy.*

The algorithm RECOVER-SUBSPACE-LARGE-DIFF is described in Figure 4. Before proving Theorem 2, we will make some simplifying assumptions (with their justifications given below) followed by some useful notation.

Remark 17 *Without loss of generality, we can assume*

1. $n = d_0$. *This is because we can first use Theorem 3 to test whether the underlying subspaces are incomparable. If they are incomparable, we can use Theorem 1 to recover the subspaces. If not, we can take $O(n/w_{min})$ samples from the mixture to get $\mathsf{span}(A_0 \cup A_1)$ with high probability (see Claim 13). We can then construct a linear bijection, say D, between $\mathsf{span}(A_0 \cup A_1)$ and $\mathbb{F}_2^{d_0}$. Applying the map D to every sample from the mixture, we can now assume that $n = d_0$.*

2. *The algorithm knows d_0, d_1. This is because we can enumerate all the possible values of d_0, d_1 and run the algorithm* SUBSPACE-RECOVER-LARGE-DIFF *to get a list of candidate hypothesis. We can then use the hypothesis testing algorithm in Theorem 12 to identify the correct one with high probability.*

3. *We set $\delta = 0.1$. This is because we can always boost the success probability of our algorithm at a multiplicative cost of $\log(1/\delta)$.*

4. *d_0 is at least a sufficiently large constant (which only depends on w_{min}). Otherwise, we can always apply a brute force algorithm to recover the subspaces.*

Notation.

1. We will use $\phi_\ell(x) \in \mathbb{F}_2^{\binom{n}{\leq \ell}}$ to represent the vector consisting of all the monomials of degree at most ℓ on x, including the constant term. As an example, when $\ell = 2$ and $n = 2$, we have $\phi_\ell(x) = (1, x_1, x_2, x_1x_2)$ – note that because the underlying field is $\mathbb{F}_2$, all the monomials are multilinear. We will use $\phi_\ell(A)$ to denote $\{\phi_\ell(x) : x \in A\}$. $\phi_\ell(A)$ is a set of vectors in $\mathbb{F}_2^{\binom{n}{\leq \ell}}$.

2. We define $t := d_0 - d_1 = (1-\alpha)d_0$ to denote the difference between the dimensions of the underlying subspaces A_0 and A_1.

3. For a sequence of vector $x_1, x_2, \cdots, x_k$, we define $x_{-i} := \{x_j : j \neq i\}$.

4. Let us denote by $y_i := \phi_\ell(x_i)$.

Finally, we note that for any subspace V of dimension d over $\mathbb{F}_2$, $rank(\phi_\ell(V)) = \binom{d}{\leq \ell}$.

We start with the following crucial lemma from Ben-Eliezer et al. (2012) (stated below). An equivalent version was also proven in (Keevash and Sudakov, 2005, Theorem 1.5).

Lemma 18 (Lemma 4, Ben-Eliezer et al. (2012)) *Let $x_1, x_2, \cdots, x_R$ be $R = 2^r$ distinct points in $\mathbb{F}_2^n$. Consider the linear space of degree d polynomials restricted to these points; that is, the space*

$$\{(p(x_1), \cdots, p(x_R)) : p \in \mathsf{RM}(n, d)\}.$$

The linear dimension of this space is at least $\binom{r}{\leq d}$.

Algorithm 4: SUBSPACE-RECOVER-LARGE-DIFF

Input:
d_0 – dimension of the larger subspace
$\alpha \leq 1$ – ratio of the dimensions of two subspaces
$\mathcal{O}(A_0, A_1, w_0, w_1)$ – oracle for random samples from mixture of subspaces.
w_{min} – minimum of two mixture weights.
Output: two subspaces U, V.

1 Set $\ell = \frac{2\log(100/w_{min})}{1-\alpha}$;
2 Use $\mathcal{O}(A_0, A_1, w_0, w_1)$ to sample $m = \binom{d_0}{\leq \ell}$ vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$;
3 Let S be the set of all $i \in [m]$ such that $\mathbf{y}_i := \phi_\ell(\mathbf{x}_i)$ can be expressed as linear combination of $\{\phi_\ell(\mathbf{x}_j) : j \neq i\}$;
4 **return** $U = \mathsf{span}(\{\mathbf{x}_i : i \in S\}), V = \mathsf{span}(\{\mathbf{x}_i : \mathbf{x}_i \notin U\})$;

As an easy corollary, we have the following claim.

Lemma 19 *Let $x_1, x_2, \cdots, x_R$ be distinct points in $\mathbb{F}_2^n$. If $R \geq 2^r$, then $rank(\{\phi_\ell(x_1), \cdots, \phi_\ell(x_R)\}) \geq \binom{r}{\leq \ell}$.*

Proof Without loss of generality, we can assume $R = 2^r$, since having more points can only increase the rank. Let $t = |\mathsf{RM}(n, \ell)|$. Say $\mathsf{RM}(n, \ell) = \{p_1, \cdots, p_t\}$. Let $A \in \mathbb{F}_2^{t \times R}$ be defined as $A_{i,j} = p_i(x_j)$. Applying Lemma 18 with $d = \ell$, we know the row-rank of A is at least $\binom{r}{\leq \ell}$. Let $B \in \mathbb{F}_2^{\binom{n}{\leq \ell} \times R}$ be the matrix whose ith column is $\phi_\ell(x_i)$. Since every polynomial is a linear combination of monomials, there exists $C \in \mathbb{F}_2^{t \times \binom{n}{\leq \ell}}$ such that $A = CB$, hence $\mathsf{rank}(B) \geq \mathsf{rank}(A) \geq \binom{r}{\leq \ell}$. ■

Proof of Theorem 2. Let I_0 (resp. I_1) be the set of all i such that $\mathbf{x}_i$ was sampled from A_0 (resp. A_1). We now define the events $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ and $\mathcal{E}_4$ as follows:

1. $\mathcal{E}_1$: $\forall i \in I_0, \mathbf{y}_i \notin \mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))$
2. $\mathcal{E}_2$: $|I_1| \geq 10\binom{\alpha d_0}{\leq \ell}$
3. $\mathcal{E}_3$: $\forall T \subseteq I_1$ such that $|T| \geq 0.9|I_1|$, we have $\mathsf{span}(\{\mathbf{x}_j\}_{j \in T}) = A_1$
4. $\mathcal{E}_4$: $\mathsf{span}(\{\mathbf{x}_j\}_{j \in I_0}) = A_0$

Assume $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ holds. Note that whenever $\mathcal{E}_1$ holds, it follows that S (defined in line 3 of SUBSPACE-RECOVER-LARGE-DIFF) is a subset of I_1. We now show that A_1 can be recovered from the span of the samples corresponding to S. Now, consider the set $\{\phi_\ell(\mathbf{x}_i) : i \in I_1 \setminus S\}$. By definition, the elements of this set are linearly independent (otherwise, they will belong in S). As $\dim(\mathsf{span}(\phi_\ell(A_1))) \leq \binom{\alpha d_0}{\leq \ell}$, it follows that $|\{\phi_\ell(\mathbf{x}_i) : i \in I_1 \setminus S\}| \leq \binom{\alpha d_0}{\leq \ell}$. As $i \mapsto \phi_\ell(\mathbf{x}_i)$ is a injection on $I_1 \setminus S$, it follows that $|\{i \in I_1 \setminus S\}| \leq \binom{\alpha d_0}{\leq \ell}$. Since $\mathcal{E}_2$ holds, $|I_1 \setminus S| \leq 0.1|I_1|$, hence $|S| \geq 0.9|I_1|$. Since $\mathcal{E}_3$ holds, $\mathsf{span}\,(\{\mathbf{x}_j\}_{j \in S}) = A_1$.

We now argue that the algorithm also recovers A_0. We claim $\{j \in [m] : \mathbf{x}_j \notin A_1\} = I_0$. Fix $j \in I_0$. Since $\mathcal{E}_1$ holds, $\phi_\ell(\mathbf{x}_j) = \mathbf{y}_j \notin \phi_\ell(A_1)$, then $\mathbf{x}_j \notin A_1$. Hence $I_0 \subseteq \{j : \mathbf{x}_j \notin A_1\}$. It

is not hard to see $\{j : \mathbf{x}_j \notin A_1\} \subseteq I_0$. Finally when $\mathcal{E}_4$ holds, we have $\mathsf{span}(\{\mathbf{x}_j : \mathbf{x}_j \notin A_1\}) = \mathsf{span}(\{\mathbf{x}_j : j \in I_0\}) = A_0$.

Thus, it remains to show that $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ and $\mathcal{E}_4$ hold simultaneously with probability 0.99.

Proof of $\mathbb{P}[\mathcal{E}_1] \geq 0.999$: First, observe that by definition, $\ell = \frac{2\log(100/w_{min})}{1-\alpha}$. Using the assumption on d_0 and w_{min}, it follows that

$$\ell = \frac{2\log(100/w_{min})}{1-\alpha} = O\left(\frac{\sqrt{d_0}}{\log d_0}\right); \quad d_0 \geq \frac{2\ell}{(1-\alpha)}. \tag{6}$$

From this, applying the constraints on d_0 and ℓ from (6), we get

$$\left(\frac{w_{min}}{100}\right)^{1/\ell} \geq 1 + \frac{1}{\ell} \cdot \log\left(\frac{w_{min}}{100}\right) \geq \frac{(1+\alpha)}{2} \geq \alpha + \frac{\ell}{d_0}. \tag{7}$$

Now, it is not difficult to see that $\binom{\alpha d_0}{\leq \ell} \leq \binom{\alpha d_0 + \ell}{\ell}$ – it easily follows from the combinatorial interpretation of binomial coefficients. Now, using this and (7), we get

$$\frac{\binom{\alpha d_0}{\leq \ell}}{\binom{d_0}{\leq \ell}} \leq \frac{\binom{\alpha d_0 + \ell}{\ell}}{\binom{d_0}{\ell}} \leq \left(\alpha + \frac{\ell}{d_0}\right)^{\ell} \leq \frac{w_{min}}{100}. \tag{8}$$

We now have,

$$\begin{aligned}
&\mathbb{P}\Big[\dim(\mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))) \leq (1 - 0.4w_{min})\binom{d_0}{\leq \ell}\Big] && (9)\\
\geq &\mathbb{P}\Big[\dim(\mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))) \leq (1 - 0.5w_{min})\binom{d_0}{\leq \ell} + \binom{\alpha d_0}{\leq \ell}\Big] \\
& && \text{using (8)}, \\
\geq &\mathbb{P}\Big[\dim(\mathsf{span}(\{\mathbf{y}_{-i}\})) \leq (1 - 0.5w_{min})\binom{d_0}{\leq \ell}\Big] \\
& && \text{using } \dim(\mathsf{span}(\phi_\ell(A_1))) = \binom{\alpha d_0}{\leq \ell}, \\
\geq &\mathbb{P}[|I_0| \leq (1 - 0.5w_{min})\binom{d_0}{\leq \ell}] \\
& && \text{using } |I_0| \geq |\{\mathbf{y}_{-i}\}| \geq \dim(\mathsf{span}(\{\mathbf{y}_{-i}\})), \\
\geq &1 - e^{-\frac{w_{min}^2}{24}\binom{d_0}{\leq \ell}} && (10)\\
& && \text{from a standard Chernoff bound.}
\end{aligned}$$

Let us now define the event $\mathcal{B}_i$ as the event that $i \in I_0$ and $\dim(\mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))) \leq (1 - 0.4w_{min})\binom{d_0}{\leq \ell}$. Let $r := \lceil (1 - 0.4w_{min}/\ell)d_0 + \ell \rceil$. Using reasoning similar to (8), we have

$$\frac{\binom{r}{\leq \ell}}{\binom{d_0}{\leq \ell}} \geq \frac{\binom{r}{\ell}}{\binom{d_0+\ell}{\ell}} \geq \left(\frac{r-\ell}{d_0}\right)^{\ell} \geq \left(1 - \frac{0.4w_{min}}{\ell}\right)^{\ell} \geq 1 - 0.4w_{min}.$$

Thus, it follows that if the event $\mathcal{B}_i$ holds, $\dim(\mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))) \leq \binom{r}{\leq \ell}$. Now, let us define the set $\mathcal{H}_i = \{x \in \mathbb{F}_2^{d_0} : \phi_\ell(x) \in \mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))\}$. By Lemma 19, we get that $|\mathcal{H}_i| \leq 2^{r+1}$.

Thus, we now have

$$\mathbb{P}[\mathbf{y}_i \in \mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))|\mathcal{B}_i] = \frac{|\mathcal{H}_i|}{2^{d_0}} \leq \frac{2^{r+1}}{2^{d_0}} \leq 2^{-\frac{0.35 w_{min} d_0}{\ell}}. \tag{11}$$

Applying the above inequality along with (10), we get

$$\mathbb{P}[\mathbf{y}_i \notin \mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))|i \in I_0] \geq 1 - 2^{\frac{-0.35 w_{min} d_0}{\ell}} - e^{-\frac{{w_{min}}^2}{24}\binom{d_0}{\leq \ell}} \geq 1 - 2^{\frac{-0.3 w_{min} d_0}{\ell}}. \tag{12}$$

By taking a union bound, it follows that

$$\mathbb{P}[\forall i \in I_0, \mathbf{y}_i \notin \mathsf{span}(\{\mathbf{y}_{-i}\} \cup \phi_\ell(A_1))] \geq 1 - \binom{d_0}{\leq \ell} 2^{\frac{-0.3 w_{min} d_0}{\ell}} \geq 1 - 2^{\frac{-0.2 w_{min} d_0}{\ell}}. \tag{13}$$

As we have chosen d_0 to be sufficiently large, the right hand side is at least 0.999 showing that $\mathbb{P}[\mathcal{E}_1] \geq 0.999$.

Proof of $\mathbb{P}[\mathcal{E}_2] \geq 0.999$: This follows from a straightforward Chernoff bound on the sampling process defining I_1.

Proof of $\mathbb{P}[\mathcal{E}_3] \geq 0.999$: This is a direct application of Claim 9.

Proof of $\mathbb{P}[\mathcal{E}_4] \geq 0.999$: This also follows from Claim 9.

■

6. Reduction from Learning Noisy Parities

In this section, we show how the problem of learning a mixture of two (comparable) subspaces captures the notorious hard problem of *learning parity with noise* (LPN).

Given $n \in \mathbb{N}$, the (n, ϵ)-LPN problem is instantiated by an (unknown) parity function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and a noise parameter $\epsilon \in (0, 1/2)$. The samples are generated i.i.d. by a sampling oracle $\mathcal{O} = \mathcal{O}(f, \epsilon)$ as follows. First, $\mathbf{x} \sim_u \mathbb{F}_2^n$ is sampled uniformly at random from $\mathbb{F}_2^n$. Then $\mathbf{b} \in \{0, 1\}$ is sampled such that $\mathbb{P}[\mathbf{b} = 0] = 1 - \epsilon$ and $\mathbb{P}[\mathbf{b} = 1] = \epsilon$. If $\mathbf{b} = 0$, $\mathcal{O}$ outputs $(\mathbf{x}, f(\mathbf{x}))$ and if $\mathbf{b} = 1$, outputs $(\mathbf{x}, 1 - f(\mathbf{x}))$. Given samples generated i.i.d. by the sampling oracle $\mathcal{O}(f, \epsilon)$, the goal is to learn the unknown parity function f.

The following simple proposition reduces LPN to learning mixtures of (comparable) subspaces in $\mathbb{F}_2^{n+1}$, where the subspaces have dimensions $n + 1$ and n respectively.

Proposition 20 *Suppose there exists an algorithm* ALG *that given samples from a mixture of two subspaces $A_0 = \mathbb{F}_2^{n+1}, A_1 \subseteq \mathbb{F}_2^{n+1}$ of dimensions $n + 1, n$ respectively, with mixing weights $2\epsilon, 1 - 2\epsilon$, runs in time $T = T(n, \delta)$ and solves this problem with probability $1 - \delta$. Then there is an algorithm that solves (n, ϵ)-*LPN *with probability $1 - \delta$ and running time $O(T) + \mathsf{poly}(n)$.*

Proof Consider a sample $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^{n+1}$ (with $\mathbf{x} \in \mathbb{F}_2^n$) drawn from a sampling oracle $\mathcal{O}(f, \epsilon)$ for the (n, ϵ)-LPN problem. We can view $(\mathbf{x}, \mathbf{y})$ as a sample from a mixture of two subspace $\mathbb{F}_2^{n+1}, A_1 \subseteq \mathbb{F}_2^{n+1}$ of dimension $n+1, n$ (respectively) with mixing weights $2\epsilon, (1 - 2\epsilon)$ as follows.

Let A_1 be the subspace of dimension n defined by the linear equation $f(\mathbf{x}) + \mathbf{y} = 0$ over $\mathbb{F}_2$. On the one hand, if $\mathbf{b} = 1$, then $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^{n+1}$ does not belong to A_1; it is drawn from $A_0 \setminus A_1$. On the other hand when $\mathbf{b} = 0$, $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^{n+1}$ lies in the subspace A_1. But this could correspond to a sample drawn from A_1 or to the portion of A_0 that overlaps with A_1 (recall that $A_1 \subset A_0$ and $|A_0 \cap A_1| = |A_0|/2$ in our case). Hence by setting the mixing weights of the subspaces $A_0 = \mathbb{F}_2^{n+1}, A_1$ to be $2\epsilon, 1 - 2\epsilon$ respectively, we can view a sample $(\mathbf{x}, \mathbf{y})$ drawn from the LPN problem as being drawn from the mixture of subspaces A_0, A_1.

Our goal is then to recover A_0, A_1 from i.i.d. samples of the form $(\mathbf{x}, \mathbf{y})$ drawn from the LPN problem. If the algorithm ALG succeeds in finding A_1, then this provides a parity function f (corresponding to the constraint defining A_1) that satisfies the LPN problem. ■

The next proposition shows that learning mixtures of two subspaces A_0, A_1 in $\mathbb{F}_2^{n+1}$ where $A_0 = \mathbb{F}_2^{n+1}$ and $\dim(A_1) = n$ is in fact equivalent to the LPN problem.

Proposition 21 *Suppose there is an algorithm ALG that solves (n, ϵ)-LPN with probability $1 - \delta$ and running time $T = T(n, \delta)$. Then, there is an algorithm that given samples from a mixture of two subspaces $\mathbb{F}_2^{n+1}, A_1 \subseteq \mathbb{F}_2^{n+1}$ of dimension $n + 1, n$ respectively with mixing weights $2\epsilon, 1 - 2\epsilon$, runs in time $O(nT) + \mathsf{poly}(n)$ and recovers A_1 with probability $1 - \delta - \exp(-n)$.*

Proof We start with a simple observation. Suppose (*) $x_{i_1} + x_{i_2} + \cdots + x_{i_k} = 0$ be the constraint defining subspace A_1, and suppose $j \in \{i_1, i_2, \cdots, i_k\}$. Consider the parity

$$f : \mathbb{F}_2^{\{1,2,\ldots,n+1\}\setminus\{j\}} \to \mathbb{F}_2, \text{ where } f(x) = \sum_{\ell \in \{i_1, i_2, \ldots, i_k\}\setminus\{j\}} x_\ell.$$

On one hand, if $(\mathbf{x}_1, \ldots, \mathbf{x}_{n+1})$ is drawn from A_1 (this is with probability $1 - 2\epsilon$), then the pair $(\mathbf{x}_{-j}, \mathbf{x}_j)$ satisfies the parity f by definition of A_1. On the other hand, if $(\mathbf{x}_1, \ldots, \mathbf{x}_{n+1})$ is drawn from A_0 (this is with probability 2ϵ), it satisfies parity f with probability $1/2$. In total, the parity f is satisfied with probability $1 - 2\epsilon + \frac{1}{2}(2\epsilon) = 1 - \epsilon$. Hence, a sample $(\mathbf{x}_1, \ldots, \mathbf{x}_{n+1})$ from the mixture of subspaces with weights $2\epsilon, 1 - \epsilon$, $(\mathbf{x}_{-j}, \mathbf{x}_j)$ can be viewed as a sample of (n, ϵ)-LPN with unknown parity f.

We do not know $\{i_1, i_2, \ldots, i_k\}$. However we can guess and try out $j = 1, \cdots, j = n + 1$ and get at most $n + 1$ candidate hypothesises. We can then use the well known hypothesis testing result from Proposition 22 to filter and find the correct subspace A_1 with high probability. ■

Acknowledgments

We thank Swastik Kopparty for telling us about the results in Ben-Eliezer et al. (2012).

References

Pranjal Awasthi, Avrim Blum, and Or Sheffet. Improved guarantees for agnostic learning of disjunctions. In Adam Tauman Kalai and Mehryar Mohri, editors, *COLT*, pages 359–367. Omnipress, 2010. ISBN 978-0-9822529-2-5. URL http://dblp.uni-trier.de/db/conf/colt/colt2010.html#AwasthiBS10.

Ainesh Bakshi and Pravesh Kothari. List-decodable subspace recovery via sum-of-squares. *ArXiv*, abs/2002.05139, 2020.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 103–112. IEEE, 2010.

Ido Ben-Eliezer, Rani Hod, and Shachar Lovett. Random low-degree polynomials are hard to approximate. *computational complexity*, 21(1):63–81, 2012.

Aditya Bhaskara, Aidao Chen, Aidan Perreault, and Aravindan Vijayaraghavan. Smoothed analysis in unsupervised learning via decoupling. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2019.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003. ISSN 0004-5411. doi: 10.1145/792538.792543. URL http://doi.acm.org/10.1145/792538.792543.

Clément L Canonne, Anindya De, and Rocco A Servedio. Learning from satisfying assignments under continuous distributions. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 82–101. SIAM, 2020.

Sitan Chen and Ankur Moitra. Beyond the low-degree algorithm: Mixtures of subcubes and their applications. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 869–880, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi: 10.1145/3313276.3316375. URL https://doi.org/10.1145/3313276.3316375.

Anindya De, Ilias Diakonikolas, and Rocco A Servedio. Learning from satisfying assignments. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 478–497. SIAM, 2014.

François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013. doi: 10.1109/TPAMI.2013.57. URL http://dx.doi.org/10.1109/TPAMI.2013.57.

Matthias Ernst, Maciej Liśkiewicz, and Rüdiger Reischuk. Algorithmic learning for steganography: proper learning of k-term dnf formulas from positive samples. In *International Symposium on Algorithms and Computation*, pages 151–162. Springer, 2015.

Jon Feldman, Rocco A. Servedio, and Ryan O'Donnell. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *Proceedings of the 19th annual conference on Learning Theory*, COLT'06, pages 20–34, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-35294-5, 978-3-540-35294-5. doi: 10.1007/11776420_5. URL http://dx.doi.org/10.1007/11776420_5.

Paulo JSG Ferreira, Bruno Jesus, Jose Vieira, and Armando J Pinho. The rank of random binary matrices and distributed storage applications. *IEEE communications letters*, 17(1):151–154, 2012.

Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Conference on Learning Theory*, pages 354–375, 2013.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two Gaussians. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 553–562. ACM, 2010.

Peter Keevash and Benny Sudakov. Set systems with restricted cross-intersections and the minimum rank ofinclusion matrices. *SIAM Journal on Discrete Mathematics*, 18(4):713–727, 2005.

Jian Li, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. Learning arbitrary statistical mixtures of discrete distributions. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 743–752, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335362. doi: 10.1145/2746539.2746584. URL https://doi.org/10.1145/2746539.2746584.

A. Liu and A. Moitra. Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638, 2018.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

Dohyung Park, Constantine Caramanis, and Sujay Sanghavi. Greedy subspace clustering. In *Neural Information Processing Systems*, December 2014.

Krzysztof Pietrzak. Cryptography from learning parity with noise. In *Proceedings of the 38th International Conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM'12, pages 99–114, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-27659-0. doi: 10.1007/978-3-642-27660-6_9. URL http://dx.doi.org/10.1007/978-3-642-27660-6_9.

Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. Learning mixtures of arbitrary distributions over large discrete domains. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 207–224, 2014.

Prasad Raghavendra and Morris Yau. List decodable subspace recovery. volume 125 of *Proceedings of Machine Learning Research*, pages 3206–3226. PMLR, 09–12 Jul 2020. URL http://proceedings.mlr.press/v125/raghavendra20a.html.

Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *Ann. Statist.*, 42(2):669–699, 04 2014. doi: 10.1214/13-AOS1199. URL http://dx.doi.org/10.1214/13-AOS1199.

René Esteban Vidal. Generalized principal component analysis (gpca): an algebraic geometric approach to subspace clustering and motion segmentation, 2003.

Appendix A. Hypothesis Test

In this section we will prove the following theorem.

Theorem 12 *Let $\mathbf{D}$ be a distribution of a mixture of two incomparable subspaces $A, B \subseteq \mathbb{F}_2^n$ with mixing weights $w_A, w_B \geq w_0$. Let $\{A_j, B_j\}_{j=1}^N$ be a collection of N sets of hypothesis with the property that there exists i such that $\{A_i, B_i\} = \{A, B\}$. There is an algorithm* CHOOSE-THE-RIGHT-HYPOTHESIS *which is given a confidence parameter δ, w_0, $\{A_j, B_j\}_{j=1}^N$ and a sampler for $\mathbf{D}$. Every subspace of $\{A_j, B_j\}_{j=1}^N$ will be represented by a basis of that subspace, and the algorithm will have the access to the basis. This algorithm has the following behavior,*

1. *It runs in $\mathsf{poly}(N, 1/w_0)\log(1/\delta)$ time.*
2. *With the probability $1-\delta$ outputs the index i such that $\{A_i, B_i\} = \{A, B\}$.*

We defer the proof to the end of this section.

In order to prove Theorem 12, we need a fundamental tool from statistics, namely "hypothesis testing for distributions". There are many equivalent forms of this algorithm — we use the following (convenient) version from De et al. (2014).

Proposition 22 (Simplified (De et al., 2014, Proposition 6)) *Let $\mathbf{D}$ be a distribution over W and $\mathbf{D}_\epsilon = \{\mathbf{D}_j\}_{j=1}^N$ be a collection of N distribution over W with the property that there exists $i \in [N]$ such that $d_{TV}(\mathbf{D}, \mathbf{D}_i) \leq \epsilon$. There is an algorithm T^D which is given an accuracy parameter ϵ, a confidence parameter δ, and is provided with access to (i) samplers for $\mathbf{D}$ and $\mathbf{D}_k$, for all $k \in [N]$ (ii) a evaluation oracle $EVAL_{\mathbf{D}_k}$, for all $k \in [N]$, which, on input $w \in W$, output the value $\mathbf{D}_k(w)$. This algorithm has the following behavior: It makes $m = O((1/\epsilon^2)(\log N + \log(1/\delta)))$ draws from $\mathbf{D}$ and each $\mathbf{D}_k$, $k \in [N]$, and $O(m)$ calls to each oracle $EVAL_{\mathbf{D}_k}$, $k \in [N]$, performs $O(mN^2)$ arithmetic operations, and with probability $1-\delta$ outputs an index $i^* \in [N]$ that satisfies $d_{TV}(\mathbf{D}, \mathbf{D}_{i^*}) \leq 6\epsilon$.*

Definition 23 *$\mathbf{D}(A, B, w_A, 1-w_A)$ is defined as the distribution induced by a mixture of two incomparable subspaces $A, B \subseteq \mathbb{F}_2^n$ of dimension at most d with mixing weights $w_A, 1-w_A$.*

Lemma 24 *Let A, B, C, D be 4 subspaces of $\mathbb{F}_2^n$. Suppose $\{A, B\} \neq \{C, D\}$. Let $\mathbf{D}_1 = \mathbf{D}(A, B, w_A, 1-w_A), \mathbf{D}_2 = \mathbf{D}(C, D, w_C, 1-w_C), w^* = min(w_A, 1-w_A, w_C, 1-w_C)$. Then $d_{TV}(\mathbf{D}_1, \mathbf{D}_2) \geq w^*/8$.*

Proof Without loss of generality, assume A has largest dimension among all 4 subspaces. We divide the rest of the analysis into a few cases.

- *Case* 1 : $A \neq C$ and $A \neq D$.
- $A = C$ or $A = D$. Assume A=C.[2]
  - *Case* 2 : $A = B$ or $A = D$.
  - $A \neq B$ and $A \neq D$.
    - *Case* 3 : A, B are incomparable.
    - *Case* 4 : A, D are incomparable.
    - *Case* 5 : $B \subsetneq A$ and $D \subsetneq A$.

Case 1:
In this case, $A \cap C$ and $A \cap D$ are two proper subspace of A. By Claim 10, $|A \backslash (C \cup D)| \geq |A|/4$, $d_{TV}(\mathbf{D}_1, \mathbf{D}_2) \geq w^*/4$.
Case 2:
Without loss of generality, assume $A = B$. We have $dim(A) \geq dim(D)$ and $D \neq A$. Hence $A \cap D$ is a proper subspace of A. $|(\mathbf{D}_1 - \mathbf{D}_2)(A \backslash D)| = (1 - w_C)|A \backslash D|/|A| \geq w^* \cdot 1/2$.
Case 3:
If $B \subseteq D$, we have $B \subsetneq D$. Since A, B are incomparable, A, D are incomparable. $|(\mathbf{D}_1 - \mathbf{D}_2)(D \backslash (A \cup B)| \geq w^*/4$. If $B \nsubseteq D$, $B \cap D$ is a proper subspace of B, $|(\mathbf{D}_1 - \mathbf{D}_2)(B \backslash (A \cup D)| \geq w^*/4$.
Case 4: similar to Cases 3.
Case 5:
If $|w_A - w_C| \geq w^*/2$, then $|(\mathbf{D}_1 - \mathbf{D}_2)(A \backslash (B \cup D))| = |w_A - w_C| \cdot |A \backslash (B \cup D))|/|A| \geq w^*/2 \cdot 1/4$.
If $|w_A - w_C| \leq w^*/2$, without loss of generality, assume $dim(B) \geq dim(D)$. Since $B \neq D$, $B \cap D$ is a proper subspace of B. $|(\mathbf{D}_1 - \mathbf{D}_2)(B \backslash D)| = |(w_A - w_C) \cdot |B \backslash D|/|A| + (1 - w_A)|B \backslash D|/|B|| \geq (1 - w_A)|B \backslash D|/|B| - |(w_A - w_C) \cdot |B \backslash D|/|A|| \geq w^*/2 - w^*/2 \cdot 1/2 = w^*/4$. ■

Proof [Proof of Theorem 12] Set $\epsilon = w_0/100, M = \lceil 1/\epsilon \rceil, \gamma = (1 - w_0)/M$. Let $\mathbf{D}_\epsilon = \{\mathbf{D}(A_j, B_j, w_0 + k * \gamma, 1 - w_0 - k * \gamma\}_{j \in [N], k \in [M] \cup \{0\}}$. It is not hard to see that there exist $\mathbf{D}^* \in \mathbf{D}_\epsilon$ such that $d_{TV}(\mathbf{D}^*, \mathbf{D}) \leq \epsilon$. By Proposition 22, we can find $\mathbf{D}' \in \mathbf{D}_\epsilon$ such that $d_{TV}(\mathbf{D}', \mathbf{D}) \leq 6\epsilon$ with probability $1 - \delta$. Say $\mathbf{D}' = \mathbf{D}(A', B', w', 1 - w')$. We claim $\{A', B'\} = \{A, B\}$. For a contradiction, suppose it is not true. Then by Lemma 24, $d_{TV}(\mathbf{D}', \mathbf{D}) \geq w_0/8 > 6\epsilon$, we derive a contradiction. ■

Appendix B. Generalized Chernoff Bound

Lemma 25 *Let $\gamma \in (0, 1), d, k \in \mathbb{N}$. Let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$ be a sequence of random variables such that for all $i \in [k]$*

$$\mathbb{P}[(\mathbf{x}_i = 1) \vee (\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_{i-1} \geq d) | \mathbf{x}_1, \cdots, \mathbf{x}_{i-1}] \geq \gamma.$$

Assume $k \geq 2d/\gamma$. Then

$$\mathbb{P}[\mathbf{x}_1 + \cdots + \mathbf{x}_k \geq d] \geq 1 - \exp\left(-k\gamma^2/8\right).$$

Proof We will use the coupling technique. Define

$$\mathbf{y}_i = \begin{cases} 1 & \text{if } \mathbf{x}_1 + \cdots + \mathbf{x}_{i-1} \geq d. \\ \mathbf{x}_i & \text{otherwise.} \end{cases}$$

Then

1. $\mathbf{x}_1 + \cdots + \mathbf{x}_k \geq d \iff \mathbf{y}_1 + \cdots + \mathbf{y}_k \geq d$.
2. For all $i \in [k]$,$\mathbb{P}[\mathbf{y}_i = 1 | \mathbf{y}_1, \cdots, \mathbf{y}_{i-1}] \geq \gamma$.

2. This is without loss of generality.

Define a submartingale $\mathbf{Z}_0, \cdots, \mathbf{Z}_k$ by $\mathbf{Z}_0 = 0$ and $\mathbf{Z}_j = \sum_{1 \le l \le j} \mathbf{y}_l - j\gamma$. Then,

$$
\begin{aligned}
&\mathbb{P}[\mathbf{x}_1 + \cdots + \mathbf{x}_k \ge d] \\
&= \mathbb{P}[\mathbf{y}_1 + \cdots + \mathbf{y}_k \ge d] \\
&= 1 - \mathbb{P}[\mathbf{y}_1 + \cdots + \mathbf{y}_k \le d - 1] \\
&\ge 1 - \mathbb{P}[\mathbf{Z}_k - \mathbf{Z}_0 \le d - 1 - k\gamma] \\
&\ge 1 - \exp\left(-\frac{(k\gamma - (d-1))^2}{2k}\right) && \text{by Azuma–Hoeffding inequality} \\
&\ge 1 - \exp\left(-k\gamma^2/8\right). && \text{by } k\gamma \ge 2d
\end{aligned}
$$

■