Algorithms for learning a mixture of linear classifiers

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: Sanjoy Dasgupta and Nika Haghtalab

Abstract

Linear classifiers are a basic model in supervised learning. We study the problem of learning a mixture of linear classifiers over Gaussian marginals. Despite significant interest in this problem, including in the context of neural networks, basic questions like efficient learnability and identifiability of the model remained open.

In this paper, we design algorithms for recovering the parameters of the mixture of k linear classifiers. We obtain two algorithms which both have polynomial dependence on the ambient dimension n, and incur an exponential dependence either on the number of the components k or a natural separation parameter $\Delta > 0$. These algorithmic results in particular settle the identifiability question under provably minimal assumptions.

Keywords: mixture models, linear classifier, method of moments

1. Introduction

Mixture models are a standard way to model data coming from a heterogeneous population. In particular, the population is assumed to consist of k subgroups (assumed to be homogeneous) and the data in each subgroup follows a parametric model. Given data from the overall population, the usual task is to recover the parameters of each of the components as well as their relative proportion in the 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 (Feldman 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; Chen et al., 2020).

The thrust of the study of mixture models, including nearly all the works cited above, has been in the unsupervised setting – i.e., where the data is unlabeled. However, another line of work, which has gained traction in recent years focuses on the supervised setting – i.e., the data is labeled (Viele and Tong, 2002; Chaganty and Liang, 2013; Sun et al., 2014; Gandikota et al., 2020; Chen et al., 2020; Diakonikolas and Kane, 2020).

In the current paper, we look at linear classifiers (aka halfspaces) – one of the most fundamental and well-studied classes of high-dimensional classifiers. In particular, we study the problem of learning mixtures of linear classifiers. We begin by describing our model. The parameters of the model are k (unknown) weights $w_1, \ldots, w_k$ which are positive and sum to 1, k (unknown) unit vectors $v_1, \ldots, v_k \in \mathbb{R}^n$. A sample is drawn as follows: the sample oracle select $i \in [k]$ with probability w_i, then we receive $(\mathbf{x}, \mathbb{1}_{\langle v_i, \mathbf{x} \rangle \geq 0})$ where $\mathbf{x} \sim \mathcal{N}(0, I_n)$. The goal is to (approximately) recover the weights $w_1, \ldots, w_k$ and the vectors $v_1, \ldots, v_k$.

Towards explaining our results, let us define $\Delta := \min_{j \neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$. At a high level, we give two algorithms for this problem. Both the algorithms have a polynomial dependence on n – the ambient dimension. The first algorithm (**??**) achieves a quasipolynomial dependence on k with an exponential dependence on $1/\Delta$. The second algorithm (Theorem 2) achieves a polynomial dependence on $1/\Delta$ but has an exponential dependence on k. A consequence of our result is that as long as $\Delta > 0$, our model is identifiable. Further, note that if $\Delta = 0$ (and say $k = 2$), the model is no longer identifiable. Thus, a dependence on $1/\Delta$ is qualitatively necessary for identifiability and *a fortiori*, for algorithmic results such as ours.

We note that both our model (Sun et al., 2014) as well as the broader question of *learning mixtures of (supervised) linear models* has been extensively studied in the literature (Chaganty and Liang, 2013; Gandikota et al., 2020; Chen et al., 2020), including in the context of neural networks (Jacobs et al., 1991; Jordan and Jacobs, 1994; Bishop, 1998). Despite this interest, basic statistical and algorithmic questions about this model remained open. As an example, until this work, the identifiability of this model (even when $k = 3$) was unresolved to the best of our knowledge.

1.1. Our Results

Our first result achieves a running time (and sample complexity) guarantee of the form $n^{O(\log k)/\Delta^2}$.

Theorem 1 *Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \leq \min\{w_1, \ldots, w_k\}$, there is an algorithm that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^{(\log k)/\Delta^2}, ((\log k)/\Delta^2)^{(\log k)/\Delta^2}, 1/w_{\min}).$$

2. *With probabilty $1 - \delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \leq \varepsilon,$$

where the min *is the minimum is over permutations π on $[k]$.*

The above running time guarantee is quasi-polynomial time as long as $\Delta = \Omega(1)$; moreover, it is polynomial time where $k = O(1)$ as well. The algorithm recovers all the unknown parameters within an error $\varepsilon > 0$, up to an ambiguity in relabeling the k components of the mixture (this is captured by the permutation π).

Our second result gives a $\mathrm{poly}((n/\Delta)^k)$ running time and sample complexity guarantee.

Theorem 2 *Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \leq \min\{w_1, \ldots, w_k\}$, there is an algorithm that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^k, k^k, \Delta^{-k}, 1/w_{\min})$$

2. *With probabilty $1-\delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\} \right) \le \varepsilon.$$

The above theorem gives polynomial time guarantees as long as k is a constant. The above two algorithmic results (Theorem 1 and Theorem 2) both have a polynomial dependence on the ambient dimension n, but trade off different exponential dependencies on k and the separation Δ. When $\Delta = \omega(\sqrt{\log k/k})$, Theorem 1 gives a faster algorithm that becomes quasi-polynomial time when $\Delta = \Omega(1)$. Meanwhile, Theorem 2 gives a faster algorithm when $\Delta = o(\sqrt{\log k/k})$; it remain polynomial time for $k = O(1)$ even when Δ has an inverse polynomial dependence on n.

Both of the above theorems are related to the following theorem, which achieves a running time of $(n/\varepsilon)^{O(\ell)}$ as long as the k vectors formed by the ℓ-th tensor power of the parameter vectors $v_1^{\otimes \ell}, v_2^{\otimes \ell}, \ldots, v_k^{\otimes \ell} \in \mathbb{R}^{n^\ell}$ are linearly independent (in a robust sense).

Theorem 3 *Let $\ell \in \mathbb{N}$. Let $U \in \mathbb{R}^{n^\ell \times k}$ be the matrix whose jth column is flattened $v_j^{\otimes \ell}$. Suppose $\sigma_{\min}(U) \ge 1/\tau$, where $\tau > 0$. Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \le \min\{w_1, \ldots, w_k\}$, there is an algorithm* ESTIMATE-PARAMETER *that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^\ell, \ell^\ell, \tau, 1/w_{\min}).$$

2. *With probabilty $1-\delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\} \right) \le \varepsilon.$$

Note that in the non-degenerate setting when the vectors $v_1, \ldots, v_k$ are linearly independent (in a robust sense), Theorem 3 already gives a polynomial time guarantee. Theorem 1 and Theorem 2 prove that even in the general case, one can choose an appropriate value of ℓ in Theorem 3 to recover the parameters.

Identifiability. The above algorithmic results succeed in uniquely identifying and recovering the individual parameters. This is as opposed to just finding a distribution 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*. The identifiability results hold as long as $\Delta > 0$. We remark that the model is not identifiable when $\Delta = 0$. This is because the distribution induced by an equal weight mixture of two linear classifiers $v_i = u$ and $v_j = -u$ is the same for every $u \in \mathbb{R}^n$! Moreover when $v_i = v_j$, there is non-identifiability by redistributing the weights of the two components arbitrarily. Hence our results prove identifiability under minimal assumptions.

1.2. Overview of techniques

We now briefly describe the algorithmic ideas and techniques that we will need to establish Theorem 1 and Theorem 2. Our algorithms are based on the method-of-moments framework and use tensor

decompositions to recover the parameters of the model. Our algorithmic results all use the same algorithmic framework, that consists of two main parts:
(i) *Extracting the low-rank tensor:* We first design a procedure that gives a good estimate for any $\ell \in \mathbb{N}$,

$$T = \sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}, \tag{1}$$

which is an order $2\ell+1$ tensor with a rank-k decomposition with one rank-1 term for each component.
(ii) *Parameter recovery through tensor decomposition:* We use an off-the-shelf algorithm for low-rank tensor decomposition, and show that they recover the parameters successfully.

(i) Extracting the low-rank tensor. Unlike latent variable models like mixtures of Gaussians (Moitra and Valiant, 2010; Janzamin et al., 2019), it is challenging to obtain a low-rank tensor by simply estimating the moments, for a linear threshold function (linear classifier). In order to estimate $\sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}$, we will instead use Hermite polynomials and consider coefficients of linear threshold functions in the Hermite basis. For a unit vector $v \in \mathbb{R}^n$, define $\mathbf{D}(v)$ be the distribution corresponding to $\mathcal{N}(0, I_n)$ conditioned on $\{x : \mathbb{1}_{\langle v,x\rangle \geq 0}\}$. The key observation is that:

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}(v)}[\mathsf{He}^{(2\ell+1)}(\mathbf{x})] \propto v^{\otimes(2\ell+1)},$$

where $\mathsf{He}^{(2\ell+1)}(\cdot)$ is the $(2\ell+1)$th order n-variable Hermite tensor (and the constant of proportionality is non-zero). See Definition 9 for a formal definition. We remark that Hermite polynomials have been used in a similar vein in the context of other learning problems like depth-2 neural networks (Janzamin et al., 2015; Ge et al., 2018; Awasthi et al., 2021).

Let $\mathbf{D}$ be the distribution of the positively-labeled samples. Note that $\mathbf{D}$ is just convex combination of $\mathbf{D}(v_1), \ldots, \mathbf{D}(v_k)$. Hence

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[\mathsf{He}^{(2\ell+1)}(\mathbf{x})] \propto \sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}.$$

The above relation naturally suggests the following meta-algorithm. We acquire few i.i.d. positive-labeled samples, the output will be the (rescaled) empirical mean of $(2\ell+1)$th order Hermite tensor evaluation. This is described in the Algorithm 1. We prove the following guarantee for estimating the tensor in Section 3.

Theorem 4 *There is an algorithm* EXTRACTING THE LOW-RANK TENSOR *that for a given k, ℓ, error tolerance parameters $\varepsilon, \delta > 0$ $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$, and access to samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity* $\log^2(1/\delta)/(\varepsilon^2) \cdot n^{O(\ell)}\ell^{O(\ell)}$.

2. *With probability $1-\delta$, the algorithm returns estimates $\mathbf{T} \in (\mathbb{R}^n)^{\otimes \ell}$ such that*

$$\|\mathbf{T} - \left(\sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}\right)\|_F \leq \varepsilon.$$

(ii) Recovering the parameters through tensor decompositions Once we have access to the tensor in (1), we use an off-the-shelf algorithm for efficient tensor decompositions (see Theorem 18). Polynomial algorithms exist for decomposing a rank-k tensor of the form in (1) as long as the flattened vectors given by $\{v_i^{\otimes \ell} : i \in [k]\}$ are linearly independent (in a robust sense). This is encapsulated in Theorem 3. To establish Theorem 1 and Theorem 2, we need to prove that the (robust) linear independence condition holds for a sufficiently large value of ℓ.

To prove Theorem 1, we show that $\ell = O(\log k/\Delta^2)$ suffices for $\{v_i^{\otimes \ell} : i \in [k]\}$ to be linearly independent. The key observation is that for $i \neq j$, $\langle v_i^{\otimes \ell}, v_j^{\otimes \ell} \rangle = (\langle v_i, v_j \rangle)^\ell$ decreases exponentially as ℓ grow. In fact, if the pairwise inner product of $v_1^{\otimes \ell}, \dots, v_k^{\otimes \ell}$ is at most $1/(2k)$, we can show that $v_1^{\otimes \ell}, \dots, v_k^{\otimes \ell}$ is robustly linear independent. This gives a running time of $n^{O(\log k)/\Delta^2}$.

To prove Theorem 2 we use a different approach to prove that $\ell = k$ suffices for $v_1^{\otimes \ell}, \dots, v_k^{\otimes \ell}$ to be linear independent in a robust sense. In particular, we use the notion of Kruskal rank to quantify the degree of linear independence with tensoring. The Kruskal rank (or Krank) of a matrix A is the largest k for which every set of k columns are linearly independent. The Khatri-Rao product of U and V which are size $m \times r$ and $n \times r$ respectively is an $mn \times r$ matrix $U \odot V$ whose i^{th} column is flattened $u_i \otimes v_i$. Let $A \in R^{n \times k}$ be the matrix whose jth column is v_j. Observe that pairwise linear independence implies $\text{Krank}(A) \geq 2$. Let $U \in \mathbb{R}^{n^\ell \times k}$ be the matrix whose jth column is flattened $v_j^{\otimes \ell}$. Observe that $U = A^{\odot k}$.

The idea is that Kruskal-rank increases with Khatri–Rao product. As a consequence, $\text{Krank}(U) \geq k$, i.e., $v_1^{\otimes k}, \dots, v_k^{\otimes k}$ are linear independent, thus establishing Theorem 2.

Comparison to Prior work Mixtures of supervised learning models like linear classifiers and other linear models have been extensively studied in machine learning literature (Jacobs et al., 1991; Chaganty and Liang, 2013; Gandikota et al., 2020; Chen et al., 2020), including in the context of neural networks as hierarchical mixtures of experts (Jacobs et al., 1991; Jordan and Jacobs, 1994; Bishop, 1998). The result that is most closely related to ours is that of Sun et al. (2014). Their model is the same as ours and in a nutshell, their main result shows that if the vectors $v_1, \dots, v_k$ are linearly independent, then there is a polynomial time algorithm that recovers the k-dimensional subspace spanned by the vectors $v_1, \dots, v_k$. However, the algorithm does not recover the parameters of the model. We do not know of algorithmic results on recovering the parameters of the mixture of k linear classifiers in any non-trivial setting. To be the best of our knowledge, even identifiability results for the model were not known for general k.

Our algorithms recover all the unknown parameters of the mixture (hence implying identifiability). The running time of the algorithms is either $n^{O(\log k)/\Delta^2}$ or $n^{O(k)}$. Moreover, when the vectors $v_1, \dots, v_k$ are linearly independent as in (Sun et al., 2014), Theorem 3 successfully recovers all the parameters in polynomial time.

2. Preliminaries and Notation

We start by defining the Mixture-of-Linear-Classifier problem formally.

Definition 5 *The Mixture-of-Linear-Classifier is instantiated by k unit vectors $v_1, \cdots, v_k$ in $\mathbb{R}^n$. In addition, we also have k corresponding weights $w_1, \cdots, w_k$ such that $w_1 + \cdots + w_k = 1$.*
The vectors $v_1, \cdots, v_k$ in $\mathbb{R}^n$ as well as the weights $w_1, \cdots, w_k$ are unknown. For this instance, the sampling oracle $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$ is defined as follows: sample $\mathbf{x} \sim \mathcal{N}(0, I_n)$, the standard spherical Gaussian in $\mathbb{R}^n$. Sample $\mathbf{z} \in [k]$ where $\mathbb{P}[\mathbf{z} = j] = w_j, \forall j \in [k]$.

$\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$ *outputs* $(\mathbf{x}, \mathbb{1}_{\langle \mathbf{x}, v_{\mathbf{z}} \rangle \geq 0}) \in \mathbb{R}^n \times \{0,1\}$.
In the Mixture-of-Linear-Classifier problem, the algorithm is given access to the number of component k*, the sample oracle* $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$*, an error parameter* ε *and a weight parameter* $w_{\min} \geq 0$ *with the promise that* $w_{\min} \leq \min\{w_1, \cdots, w_k\}$*. The goal of the algorithm is to output estimates* $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ *such that*

$$\min_{\pi} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\} \right) \leq \varepsilon,$$

where the min *is over all permutations on* $[k]$.

Notation. We use $\nabla_t^{(d)}$ to the denote the d-th order differential operator (with respect to t).

We next definite Hermite polynomials. We begin with univariate Hermite polynomials.

Definition 6 *The* d^{th} *univariate Hermite polynomial* $\mathsf{He}_d(x) : \mathbb{R} \to \mathbb{R}$ *is the formal polynomial*

$$\mathsf{He}_d(x) = \left(\left(\frac{\partial}{\partial t} \right)^d \exp(xt - t^2/2) \right) \Bigg|_{t=0}.$$

The following observation gives a recursive relation between He_d and He_{d+1}.

Observation 7 *For* $d \in \mathbb{Z}_{\geq 0}$*,*

$$\frac{d}{dx}\left(\mathsf{He}_d(x) e^{-x^2/2}\right) = -\mathsf{He}_{d+1}(x) e^{-x^2/2}$$

Proof We use Rodrigues formula for the Hermite polynomial (see e.g., equation (10) of Patarroyo, 2019), which is:

$$\mathsf{He}_d(x) = (-1)^d e^{\frac{x^2}{2}} \left(\frac{d}{dx}\right)^d \left(e^{\frac{-x^2}{2}}\right).$$

Or equivalently,

$$\mathsf{He}_d(x) e^{-\frac{x^2}{2}} = (-1)^d \left(\frac{d}{dx}\right)^d \left(e^{\frac{-x^2}{2}}\right).$$

The claim now follows easily. ■

The next observation gives an explicit formula for univariate Hermite polynomials.

Observation 8 *(see equation (3) of Patarroyo, 2019) For* $d \in \mathbb{Z}_{\geq 0}$*, we have*

$$\mathsf{He}_d(x) = d! \sum_{j=0}^{\lfloor \frac{d}{2} \rfloor} \frac{(-1)^j}{2^j (d-2j)! j!} x^{d-2j}$$

Next, we define multivariable Hermite polynomials.

Definition 9 *For* $n \in \mathbb{N}$*,*$d \in \mathbb{Z}_{\geq 0}$*. We use* $\mathsf{He}^{(d)}$ *to denote the* n*-variable* Hermite tensor *of order* d*.* $\mathsf{He}^{(d)} : \mathbb{R}^n \to (\mathbb{R}^n)^{\otimes d}$ *is defined as:*

$$\mathsf{He}^{(d)}(x) = \left(\nabla_t^{(d)} \exp(\langle x, t \rangle - \|t\|^2/2) \right) \Big|_{t=0}.$$

3. Extracting the low-rank tensor

In this section, the main goal is to prove Theorem 4. Theorem 4 gives an algorithm which given samples from a mixture of k linear classifiers, estimates the order-ℓ parameter moment. This result is an important piece in our parameter recovery algorithm. The algorithm EXTRACTING THE LOW-RANK TENSOR is described in Algorithm 1.

Theorem 4 *There is an algorithm* EXTRACTING THE LOW-RANK TENSOR *that for a given k, ℓ, error tolerance parameters $\varepsilon, \delta > 0$ $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$, and access to samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity* $\log^2(1/\delta)/(\varepsilon^2) \cdot n^{O(\ell)}\ell^{O(\ell)}$.
2. *With probability $1 - \delta$, the algorithm returns estimates $\mathbf{T} \in (\mathbb{R}^n)^{\otimes \ell}$ such that*

$$\|\mathbf{T} - \left(\sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}\right)\|_F \le \varepsilon.$$

Algorithm 1: EXTRACTING THE LOW-RANK TENSOR

```
Input:
$k$ – number of components
$\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$ – oracle for random samples from the mixture
$\ell$ – parameter for order of the tensor
$\varepsilon$ – error parameter
Output:
$\mathbf{T} \in (\mathbb{R}^n)^{\otimes(2\ell+1)}$ – estimate of $\sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}$
Set $t = (\varepsilon^{-2})n^{O(\ell)}\ell^{O(\ell)}$;
Use $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$ to sample $t$ independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_t$ from $\mathbf{D}$;
return $\mathbf{T} = 1/t \sum_{j\in[t]} 1/c(\ell)\mathsf{He}^{(2\ell+1)}(\mathbf{x}_j)$, where $c(\ell) = \sqrt{2/\pi}(-1)^{\ell}(2\ell-1)!!$;
```

The main idea behind the algorithm is to show that over the positive samples, the expectation of the $(2\ell + 1)$th-order Hermite tensor is proportional to $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$. Based on this, our algorithm is to just output an empirical estimator for the average $(2\ell + 1)$th-order Hermite tensor. The rest of this section is dedicated to proving the correctness of Algorithm 1 (Theorem 4) Towards this, we start with some definitions.

Definition 10 *Let $v \in \mathbb{R}^n$, $\mathbf{x} \sim \mathcal{N}(0, I_n)$. Define $\mathbf{D}(v)$ as the conditional distribution of $\mathbf{x}$ given $\langle v, \mathbf{x}\rangle \ge 0$. Or equivalently, the probability density function of $\mathbf{D}(v)$ is given by*

$$(2\pi)^{-n/2}e^{-\|z\|^2/2} \cdot 2\mathbb{1}_{\langle v,z\rangle \ge 0}.$$

Define $\mathbf{D}$ to be the distribution corresponding to positive samples from $\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$. Or equivalently, $\mathbf{D} = \sum_{j\in[k]} w_j \mathbf{D}(v_j)$.

The following observation says that the ℓ-th order derivative (with respect to t) of $f(\langle v, t\rangle)$ is proportional to $v^{\otimes \ell}$. It can be easily derived from the chain rule.

Observation 11 *Let $f : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, $v \in \mathbb{R}^n$, $\ell \in \mathbb{N}$. Then,*

$$\nabla_t^{(\ell)}(f(\langle v, t\rangle)) = f^{(\ell)}(\langle v, t\rangle) \cdot v^{\otimes \ell}$$

Next, we define a function $\Psi(t)$ (which is the mass that the Gaussian centered at t puts on $[0, \infty)$) and obtain an explicit formula for its derivatives of odd order.

Claim 12 *Define $\Psi : \mathbb{R} \to \mathbb{R}$ to be*

$$\Psi(t) = \int_{\mathbb{R}} \exp\left(-(x-t)^2/2\right) \mathbb{1}_{x \geq 0} dx.$$

For all $\ell \in \mathbb{Z}_{\geq 0}$, we have

$$\Psi^{(2\ell+1)}(0) = (-1)^\ell (2\ell - 1)!!$$

Proof To prove the above equality, we first swap the integration with differentiation and relate the resulting expression to Hermite polynomials.
We know that

$$\Psi(t) = \int_0^\infty \exp\left(-(x-t)^2/2\right) dx = \int_0^\infty \exp\left(tx - t^2/2\right) \exp\left(-x^2/2\right) dx.$$

$$\begin{aligned}
\text{Hence,}\quad \Psi^{(2\ell+1)}(0) &= \left(\left(\frac{\partial}{\partial t}\right)^{2\ell+1} f\right)\Bigg|_{t=0} \\
&= \left(\int_0^\infty \left(\frac{\partial}{\partial t}\right)^{2\ell+1} \exp\left(tx - t^2/2\right) \exp\left(-x^2/2\right) dx\right)\Bigg|_{t=0} && \text{by Leibniz integral rule} \\
&= \int_0^\infty \left(\left(\frac{\partial}{\partial t}\right)^{2\ell+1} \exp\left(tx - t^2/2\right)\right)\Bigg|_{t=0} \exp\left(-x^2/2\right) dx \\
&= \int_0^\infty \mathsf{He}_{2\ell+1}(x) \exp\left(-x^2/2\right) dx \\
&= \int_0^\infty d\left(-\mathsf{He}_{2\ell}(x) \exp\left(-x^2/2\right)\right) && \text{see Observation 7} \\
&= \mathsf{He}_{2\ell}(0) = (-1)^\ell (2\ell - 1)!! && \text{by Observation 8}
\end{aligned}$$

■

The following lemma is crucial in establishing Theorem 4. The lemma proves that the expectation of $(2\ell + 1)$th-order Hermite tensor over $\mathbf{D}$ is proportional to $\sum_{j \in [k]} w_j v_j^{\otimes(2\ell+1)}$.

Lemma 13 *For $\ell \in \mathbb{Z}_{\geq 0}$,*

$$\mathbb{E}_{\mathbf{x} \sim \mathbf{D}}[1/c(\ell) \mathsf{He}^{(2\ell+1)}(\mathbf{x})] = \sum_{j \in [k]} w_j v_j^{\otimes(2\ell+1)},$$

where $c(\ell) = \sqrt{2/\pi}(-1)^\ell (2\ell - 1)!!$.

Proof The high-level idea is to reduce the problem to the case $k = 1$. In particular, since $\mathbf{D} = \sum_{j\in[k]} w_j \mathbf{D}(v_j)$, it suffices to show that: for all unit vector $v \in \mathbb{R}^n$,

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}(v)}[\mathsf{He}^{(2\ell+1)}(\mathbf{x})] = \sqrt{\frac{2}{\pi}}(-1)^\ell(2\ell-1)!! \cdot v^{\otimes(2\ell+1)}.$$

Towards this, recall that,

$$\mathsf{He}^{(2\ell+1)}(x) = \left(\nabla_t^{(2\ell+1)} \exp(\langle x, t\rangle - \|t\|^2/2)\right)\Big|_{t=0}.$$

Then,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim\mathbf{D}(v)}[\mathsf{He}^{(2\ell+1)}(\mathbf{x})]\\
&= 2(\frac{1}{\sqrt{2\pi}})^n \int_{\mathbb{R}^n} \mathsf{He}^{(2\ell+1)}(x)\mathbb{1}_{\langle x,v\rangle\geq 0} \exp\left(-\|x\|^2/2\right)dx \quad \text{using Definition 10}\\
&= 2(\frac{1}{\sqrt{2\pi}})^n \int_{\mathbb{R}^n} \left(\nabla_t^{(2\ell+1)} \exp(\langle x, t\rangle - \|t\|^2/2)\right)\Big|_{t=0} \mathbb{1}_{\langle x,v\rangle\geq 0} \exp\left(-\|x\|^2/2\right)dx\\
&= 2(\frac{1}{\sqrt{2\pi}})^n \left(\nabla_t^{(2\ell+1)} \int_{\mathbb{R}^n} \exp\left(-\|x-t\|^2/2\right)\mathbb{1}_{\langle x,v\rangle\geq 0}dx\right)\Big|_{t=0} \quad \text{by Leibniz integral rule}
\end{aligned}$$

Let U be an orthonormal matrix such that the first row equals v^T. Thus, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim\mathbf{D}(v)}[\mathsf{He}^{(2\ell+1)}(\mathbf{x})]\\
&= 2(\frac{1}{\sqrt{2\pi}})^n \left(\nabla_t^{(2\ell+1)} \int_{\mathbb{R}^n} \exp\left(-\|y-Ut\|^2/2\right)\mathbb{1}_{y_1\geq 0}dy\right)\Big|_{t=0} \quad \text{change of variables, } y = Ux\\
&= \frac{2}{\sqrt{2\pi}} \left(\nabla_t^{(2\ell+1)} \int_{\mathbb{R}} \exp\left(-(y_1-\langle v,t\rangle)^2/2\right)\mathbb{1}_{y_1\geq 0}dy_1\right)\Big|_{t=0}\\
&= \frac{2}{\sqrt{2\pi}} \left(\nabla_t^{(2\ell+1)} \Psi(\langle v,t\rangle)\right)\Big|_{t=0} \quad \text{by Claim 12}\\
&= \frac{2}{\sqrt{2\pi}} \left(\Psi^{(2\ell+1)}(\langle v,t\rangle)\cdot v^{\otimes(2\ell+1)}\right)\Big|_{t=0} \quad \text{by Observation 11}\\
&= \sqrt{\frac{2}{\pi}}(-1)^\ell(2\ell-1)!!v^{\otimes(2\ell+1)} \quad \text{by Claim 12}
\end{aligned}$$

■

Our next goal is to bound the variance of our estimator in Algorithm 1. Towards this, we start with the following simple claim.

Claim 14 *Let $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(x) = f(-x), \forall x \in \mathbb{R}^n$. Then,*

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[f(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_n)}[f(\mathbf{x})]$$

Proof First, using the fact that the distribution $1/2(\mathbf{D}(v_j) + \mathbf{D}(-v_j))$ is $\mathcal{N}(0, I_n)$ and f is even, it follows that

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}(v_j)}[f(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_n)}[f(\mathbf{x})].$$

We now get the claim by noting that $\mathbf{D}$ is a convex combination of $\mathbf{D}(v_1), \ldots, \mathbf{D}(v_k)$. ■

Next, we upper bound the variance of a polynomial under the distribution $\mathbf{D}$.

Claim 15 *Let $s \in \mathbb{N}$. Let $m_1, \ldots, m_s : \mathbb{R}^n \to \mathbb{R}$ be monomials of degree at most t. Let $\alpha = (\alpha_1, \ldots, \alpha_s) \in \mathbb{R}^s$. Then,*

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[(\alpha_1 m_1(\mathbf{x}) + \ldots + \alpha_s m_s(\mathbf{x}))^2] \le s(2t-1)!!\|\alpha\|^2.$$

Proof The idea is to apply Claim 14. Claim 14 which lets us reduce the problem of computing the variance under $\mathbf{D}$ to that under the standard Gaussian.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[(\alpha_1 m_1(\mathbf{x}) + \ldots + \alpha_s m_s(\mathbf{x}))^2] \\
&\le \mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[s(\sum_{j\in[s]} \alpha_j^2 m_j(x)^2)] && \text{by Cauchy–Schwarz inequality} \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_n)}[s(\sum_{j\in[s]} \alpha_j^2 m_j(x)^2)] && \text{by Claim 14} \\
&\le s(2t-1)!!\|\alpha\|^2.
\end{aligned}$$

The last inequality follows from the fact that $m_j(x)^2$ is a monomial of degree at most $2t$ and thus its expectation under a Gaussian is at most $(2t-1)!!$. It is well-known that $\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,1)}[x^{2d}] = (2d-1)!!, \forall d \in \mathbb{N}$. A standard induction will give us the above fact. ■

The next proposition shows that $\mathrm{Var}_{\mathbf{x}\sim\mathbf{D}}[\mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})]$ is at most $\ell^{O(\ell)}$ for any fixed index α.

Proposition 16 *For $\ell \in \mathbb{N}$ with $\ell \ge 2$, fix $\alpha \in [n]^{2\ell+1}$,*

$$\underset{\mathbf{x}\sim\mathbf{D}}{\mathit{Var}}\left[\frac{1}{c(\ell)} \cdot \mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})\right] \le \ell^{c_1\ell},$$

where $c(\ell) = \sqrt{2/\pi}(-1)^\ell(2\ell-1)!!$ and c_1 is an absolute constant.

Proof We begin by noting that

$$\underset{\mathbf{x}\sim\mathbf{D}}{\mathrm{Var}}[\mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})] \le \mathbb{E}_{\mathbf{x}\sim\mathbf{D}}\left[\left(\mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})\right)^2\right]$$

By definition, $\mathsf{He}_\alpha^{(2\ell+1)}(x)$ can be expressed as $\prod_{j=1}^n \mathsf{He}_{s_j}(x_j)$ where:

1. $s_1, \ldots, s_n \in \mathbb{Z}_{\ge 0}$ depend on α.
2. $\sum_{j=1}^n s_j = 2\ell + 1$.

Hence $\mathsf{He}_\alpha^{(2\ell+1)}(x)$ is a polynomial of degree at most $2\ell+1$ and can be expanded as $\beta_1 m_1(x) + \ldots + \beta_z m_z(x)$ where:

1. $m_1, \ldots, m_z$ are monomials,

2. $z \leq (2\ell+1)^{(2\ell+1)}$,

3. for all $j \in [z]$, $|\beta_j| \leq (2\ell+1)!$.

The second item is true because $\mathsf{He}_s(\cdot)$ has at most $2s+1$ terms (see Observation 8) and $\mathsf{He}_\alpha^{(2\ell+1)}(x)$ can be expressed as $\prod_{j=1}^n \mathsf{He}_{s_j}(x_j)$. The last item is true because every coefficient of He_s is bounded by $s!$ (see Observation 8) and $\prod_{j=1}^n s_j! \leq (\sum_{j=1}^n s_j)! = (2\ell+1)!$.

Apply Claim 15, we have

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[\mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})^2] \leq z(4\ell+1)!!(z\left((2\ell+1)!)^2\right) \leq \ell^{c'\ell},$$

where c' is an absolute constant. ■

We are now ready to finish the proof of Theorem 4.

Proof of Theorem 4. Without loss of generality, we can assume $\delta = 0.1$, since we can always boost the success probability at a multiplicative cost of $O(\log(1/\delta)^2)$ via Claim 24. Define $T^* = \sum_{j=1}^k w_j v_j^{\otimes(2\ell+1)}$. We will show $\mathbf{T}$ is close to T^* with probability at least 0.9, where $\mathbf{T}$ is the empirical mean of $1/c(\ell)\mathsf{He}^{(2\ell+1)}(\mathbf{x})$. By Lemma 13, $T^* = \mathbb{E}_{\mathbf{x}\sim\mathbf{D}}[1/c(\ell)\mathsf{He}^{(2\ell+1)}(\mathbf{x})]$, hence $T^* = \mathbb{E}[\mathbf{T}]$. Fix $\alpha \in [n]^{2\ell+1}$, from Proposition 16 we know that

$$\begin{aligned}\mathsf{Var}[\mathbf{T}_\alpha] &= \frac{1}{t}\underset{\mathbf{x}\sim\mathbf{D}}{\mathsf{Var}}[1/c(\ell)\mathsf{He}_\alpha^{(2\ell+1)}(\mathbf{x})] \\ &\leq \ell^{c_1\ell}/t,\end{aligned}$$

where c_1 is an absolute constant. By Chebyshev's inequality,

$$\mathbb{P}[|\mathbf{T}_\alpha - T^*_\alpha| \geq \frac{\varepsilon}{\sqrt{n^{2\ell+1}}}] \leq \frac{\ell^{c_1\ell}/t}{\frac{\varepsilon^2}{n^{2\ell+1}}} = \frac{\ell^{c_1\ell}n^{2\ell+1}}{t\varepsilon^2}$$

Then,

$$\begin{aligned}\mathbb{P}[\vee_{\alpha\in[n]^{2\ell+1}}\{|\mathbf{T}_\alpha - T^*_\alpha| \geq \frac{\varepsilon}{\sqrt{n^{2\ell+1}}}\}] &\leq \frac{\ell^{c_1\ell}n^{4\ell+2}}{t\varepsilon^2} \\ &\leq 0.01 && \text{by the choice of } t.\end{aligned}$$

Hence,

$$\mathbb{P}[\wedge_{\alpha\in[n]^{2\ell+1}}\{|\mathbf{T}_\alpha - T^*_\alpha| < \frac{\varepsilon}{\sqrt{n^{2\ell+1}}}\}] \geq 0.99$$

As a result,

$$\mathbb{P}[\|\mathbf{T} - T^*\|_F \leq \varepsilon] \geq 0.99$$

■

4. Estimation algorithm for the Parameters of the Mixture of Linear Classifiers

In this section, we prove Theorem 1 and Theorem 2. Recall that $\Delta = \min_{j \neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$. Theorem 1 shows that there is an algorithm learns a mixture of k linear classifiers in time $\mathrm{poly}(n^{(\log k)/\Delta^2})$, where Δ is the minimum "separation" between each pair of linear classifiers. Meanwhile, Theorem 2 shows that there is an algorithm that does the same thing in time that is roughly $\mathrm{poly}((n/\Delta)^k)$. When $\Delta = \omega(\sqrt{\log k / k})$, Theorem 1 gives a faster algorithm. Meanwhile, Theorem 2 gives a faster algorithm when $\Delta = o(\sqrt{\log k / k})$.

Theorem 1 *Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \leq \min\{w_1, \ldots, w_k\}$, there is an algorithm that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^{(\log k)/\Delta^2}, ((\log k)/\Delta^2)^{(\log k)/\Delta^2}, 1/w_{\min}).$$

2. *With probabilty $1 - \delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \leq \varepsilon,$$

where the $\min$ is the minimum is over permutations π on $[k]$.

Theorem 2 *Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \leq \min\{w_1, \ldots, w_k\}$, there is an algorithm that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^k, k^k, \Delta^{-k}, 1/w_{\min})$$

2. *With probabilty $1 - \delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \leq \varepsilon.$$

The basic idea of the above two theorems are the same. Roughly speaking, Theorem 18 (from Bhaskara et al. (2014a)) says we can decompose a noisy third-order low-rank tensor efficiently under some mild non-degeneracy conditions. By Theorem 4, we can estimate $T^* = \sum_{j \in [k]} w_j v_j^{\otimes(2\ell+1)}$ accurately. Note that T^* can be viewed as a third-order low-rank tensor $\sum_{j \in [k]} (v_j^{\otimes \ell}) \otimes (v_j^{\otimes \ell}) \otimes (w_j v_j)$. Our approach will be combining Theorem 4 and Theorem 18.

In order to combine Theorem 4 and Theorem 18, $\sum_{j \in [k]} (v_j^{\otimes \ell}) \otimes (v_j^{\otimes \ell}) \otimes (w_j v_j)$ needs to satisfy the conditions of Theorem 18. The major challenge is to show that $v_1^{\otimes \ell}, \ldots, v_k^{\otimes \ell}$ are linear independent in a robust sense (that is measured in terms of the least singular value of the $n^\ell \times k$ matrix formed by the flattenings of these k tensored vectors as columns). Theorem 1 and Theorem 2 use different approaches to establish this condition. On the one hand, Claim 19 shows that $\ell = 10(\log k)/\Delta^2$ suffices. This leads to Theorem 1. On the other hand, Claim 22 shows that $\ell = k$ suffices (even when Δ can be a small inverse polynomial in n). This leads to Theorem 2.

We start by introducing the concept of Kruskal rank for convenience.

Definition 17 (Definition 1.2, Bhaskara et al. (2014a)) *The Kruskal rank (or Krank) of a matrix A is the largest k for which every set of k columns are linearly independent. Also the τ-*robust Krank *is denoted by Krank$_\tau(A)$, and is the largest k for which every $n \times k$ sub-matrix $A_{|S}$ of A has $\sigma_k(A_{|S}) \geq 1/\tau$.*

The following theorem from Bhaskara et al. (2014a) is crucial; see also (Janzamin et al., 2019; Goyal et al., 2014) for related tensor decomposition guarantees. Suppose $U \in \mathbb{R}^{m\times R}, V \in \mathbb{R}^{n\times R}, W \in \mathbb{R}^{p\times R}$. The theorem says that: if U, V are well-conditioned and columns of W are "pairwise well-conditioned", there is an algorithm that can recover all the rank-one terms from $T = \sum_{i=1}^{R} u_i \otimes v_i \otimes w_i$ efficiently. Moreover, the algorithm can also tolerate some inverse polynomial amount of noise.

Theorem 18 (Theorem 2.3, Bhaskara et al. (2014a)) *Suppose $U \in \mathbb{R}^{m\times R}, V \in \mathbb{R}^{n\times R}, W \in \mathbb{R}^{p\times R}$. u_i, v_i, w_i are the ith column of U, V, W, respectively. Suppose U, V, W satisfy that:*

1. *The condition numbers $\kappa(U), \kappa(V) \leq \kappa$,*
2. *The column vectors of W are not close to parallel: Krank$_{1/\delta}(W) \geq 2$,*
3. *The decompositions are bounded : for all i, $\|u_i\|_2, \|v_i\|_2, \|w_i\|_2 \leq C$.*

Suppose we are given tensor $T + E \in \mathbb{R}^{m\times n\times p}$ with the entries of E being bounded by $\varepsilon \cdot \mathrm{poly}(1/\kappa, 1/m, 1/n, 1/p, \delta, 1/C)$ and moreover T has a decomposition $T = \sum_{i=1}^{R} u_i \otimes v_i \otimes w_i$. There is an algorithm with the following guarantee:

1. *The algorithm runs in time complexity* $\mathrm{poly}(m, n, p)$.
2. *With probability* 0.99*, the algorithm returns* each rank one term *in the decomposition of T (up to renaming), within an additive error of ε.*

The next claim says the following. We can view $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$ as $\sum_{j\in[k]}(v_j^{\otimes \ell}) \otimes (v_j^{\otimes \ell}) \otimes (w_j v_j)$. If $\ell = 10(\log k)/\Delta^2$, the above tensor satisfy the condition of Theorem 18.

Claim 19 *Define $\Delta = \min_{j\neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$. Suppose $\Delta > 0$. Define $\ell = 10(\log k)/\Delta^2$. Consider $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)} = \sum_{j\in[k]}(v_j^{\otimes \ell}) \otimes (v_j^{\otimes \ell}) \otimes (w_j v_j)$. Let $U \in \mathbb{R}^{n^\ell \times k}$ be the matrix whose jth column is flattened $v_j^{\otimes \ell}$. Let $W \in \mathbb{R}^{n\times k}$ be the matrix whose jth column is $w_j v_j$. Then the following hold:*

1. *The condition numbers $\kappa(U) \leq \mathrm{poly}(k)$,*
2. *For all $i \neq j$, we have $w_i v_i, w_j v_j$ are not close to parallel: Krank$_{2/(w_{\min}\Delta)}(W) \geq 2$,*
3. *For all j, we have $\|v_j^{\otimes k}\|_F, \|w_j v_j\| \leq 1$.*

Proof *Proof of part (1):*

The main idea is the following. We have

$$\langle v_i^{\otimes \ell}, v_j^{\otimes \ell}\rangle = \begin{cases} 1 & i = j \\ \langle v_i, v_j\rangle^\ell & i \neq j \end{cases}$$

Since $\langle v_i, v_j\rangle^\ell \to 0$ as $\ell \to \infty$, we have that $U^T U \to I$ as $\ell \to \infty$. Hence we expect $\kappa(U)$ is small if ℓ is sufficiently large.

Using the variational characterization for singular values:

$$\sigma_{\min}(U) = \|\alpha_1 v_1^{\otimes \ell} + \ldots + \alpha_k v_k^{\otimes \ell}\|_F$$

for some unit vector $(\alpha_1, \ldots, \alpha_k)$.

Without loss of generality, we assume

1. $|\alpha_1|$ is the greatest one among $\{|\alpha_1|, \ldots, |\alpha_k|\}$,
2. $\alpha_1 \geq 0$.

From Cauchy–Schwarz, we have

$$\begin{aligned} \|\alpha_1 v_1^{\otimes \ell} + \ldots + \alpha_k v_k^{\otimes \ell}\|_F &\geq \langle \alpha_1 v_1^{\otimes \ell} + \ldots + \alpha_k v_k^{\otimes \ell}, v_1^{\otimes \ell}\rangle \\ &= \alpha_1 + \alpha_2 \langle v_2^{\otimes \ell}, v_1^{\otimes \ell}\rangle + \ldots + \alpha_k \langle v_k^{\otimes \ell}, v_1^{\otimes \ell}\rangle \\ &= \alpha_1 \left(1 + \alpha_2/\alpha_1 \langle v_2^{\otimes \ell}, v_1^{\otimes \ell}\rangle + \ldots + \alpha_k/\alpha_1 \langle v_k^{\otimes \ell}, v_1^{\otimes \ell}\rangle\right) \end{aligned} \tag{2}$$

For any $j \neq 1$,

$$\begin{aligned} |\alpha_j/\alpha_1 \langle v_j^{\otimes \ell}, v_1^{\otimes \ell}\rangle| &\leq |\langle v_j^{\otimes \ell}, v_1^{\otimes \ell}\rangle| = |\langle v_j, v_1\rangle|^\ell \\ &\leq (1 - \Delta^2/2)^\ell \qquad\qquad \text{since } \Delta = \min_{j \neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\} \\ &\leq \exp(-\Delta^2 \ell/2) \leq \frac{1}{2k}. \end{aligned}$$

Applying the above inequality along with (2), we get

$$\sigma_{\min}(U) \geq \alpha_1(1 - (k-1)/2k) \geq \frac{\alpha_1}{2} \geq \frac{1}{2\sqrt{k}}$$

Meanwhile

$$\|U\|^2 \leq \|U\|_F^2 = \sum_{j \in [k]} \|v_j^{\otimes k}\|_F^2 = k.$$

Therefore part (1) is true.

Proof of part (2): It follows by the definition $\Delta = \min_{j \neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$.

Proof of part (3): Recall that $\{v_j\}$ are unit vectors. ■

Next, we introduce the concept of Khatri-Rao product for convenience.

Definition 20 (Definition 1.3, Bhaskara et al. (2014a)) *The Khatri-Rao product of U and V which are size $m \times r$ and $n \times r$ respectively is an $mn \times r$ matrix $U \odot V$ whose i^{th} column is flattened $u_i \otimes v_i$.*

The next lemma is a robust analogue of the following fact: $\text{Krank}(A \odot B) \geq \min\{\text{Krank}(A) + \text{Krank}(B) - 1, R\}$, where A, B are matrix with R columns. Intuitively, it means that Krank will increase with Khatri-Rao product. It will be used to prove Claim 22.

Lemma 21 (Lemma A.4, Bhaskara et al. (2014b)) *A, B are matrix with R columns. Say Krank$_{\tau_1}(A) \geq k_A$, Krank$_{\tau_2}(B) \geq k_B$, where $k_A, k_B \in \mathbb{N}$. Let $t = \min\{k_A + k_B - 1, R\}$. Then Krank$_{(\tau_1 \tau_2 \sqrt{t})}(A \odot B) \geq t$.*

We will need the following claim, which shows that: $\sum_{j\in[k]} w_j v_j^{\otimes(2k+1)} = \sum_{j\in[k]} (v_j^{\otimes k}) \otimes (v_j^{\otimes k}) \otimes (w_j v_j)$ satisfies the condition of Theorem 18. This means we can recover the rank-one terms from $\sum_{j\in[k]} w_j v_j^{\otimes(2k+1)}$.

Claim 22 *Define $\Delta = \min_{j\neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$. Suppose $\Delta > 0$. Consider $\sum_{j\in[k]} w_j v_j^{\otimes(2k+1)} = \sum_{j\in[k]} (v_j^{\otimes k}) \otimes (v_j^{\otimes k}) \otimes (w_j v_j)$. Let $U \in \mathbb{R}^{n^k \times k}$ be the matrix whose jth column is flattened $v_j^{\otimes k}$. Let $W \in \mathbb{R}^{n \times k}$ be the matrix whose jth column is $w_j v_j$. Then the following hold:*

1. *The condition numbers $\kappa(U) \leq (1/\Delta)^{O(k)} k^{O(k)}$,*
2. *For all $i \neq j$, we have $w_i v_i, w_j v_j$ are not close to parallel: Krank$_{2/(w_{\min}\Delta)}(W) \geq 2$,*
3. *For all j, we have $\|v_j^{\otimes k}\|_F, \|w_j v_j\| \leq 1$.*

Proof *Proof of part (1):* The main idea is to apply Lemma 21. Let $A \in R^{n\times k}$ be the matrix whose jth column is v_j. Observe that $U = A^{\odot k}$. Roughly speaking, note that Krank$(A) \geq 2$, we have Krank$(A^{\odot(k-1)}) \geq k$. $A^{\odot(k-1)}$ has full column rank, so as U.

Let $A \in R^{n\times k}$ be the matrix whose jth column is v_j. Observe that $U = A^{\odot k}$. We know that Krank$_{2/\Delta}(A) \geq 2$ by the definition $\Delta = \min_{j\neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$. Apply Lemma 21 inductively, we have

$$\text{Krank}_{\left((2/\Delta)^k \sqrt{k!\cdot k}\right)} A^{\odot k} \geq k.$$

In other word,

$$\sigma_{\min}(A^{\odot k}) \geq (\Delta/2)^k / \sqrt{k! \cdot k} = \Delta^{O(k)} k^{-O(k)}.$$

Meanwhile

$$\|U\|^2 \leq \|U\|_F^2 = \sum_{j\in[k]} \|v_j^{\otimes k}\|_F^2 = k.$$

Therefore part (1) is true.
Proof of part (2): It follows by the definition $\Delta = \min_{j\neq j'} \min\{\|v_j - v_{j'}\|, \|v_j + v_{j'}\|\}$.
Proof of part (3): Recall that $\{v_j\}$ are unit vectors. ■

The next claim says that we can get an accurate estimate of v, w from a accurate estimate of $wv^{\otimes(2\ell+1)}$. This is useful since we get k rank-one tensors of the form $wv^{\otimes(2\ell+1)}$ after apply Theorem 18 (tensor decomposition) to $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$. While it is easy and standard, we include it here for the sake of completeness.

Claim 23 *Let $\ell \in \mathbb{N}, \varepsilon, w \in (0,1]$. Let $v \in \mathbb{R}^n$ be a unit vector. $\{e_\alpha : \alpha \in [n]\}$ is the standard basis of $\mathbb{R}^n$. Suppose $T \in (\mathbb{R}^n)^{\otimes(2\ell+1)}$ satisfies that*

$$\|wv^{\otimes(2\ell+1)} - T\|_F \le \frac{\varepsilon w}{4n^\ell}.$$

Let $\hat{w} \in \mathbb{R}, \hat{v} \in \mathbb{R}^n$ be such that $\hat{w} = \|T\|_F, \hat{v}_\alpha = \frac{\langle T, e_\alpha \otimes I_n^{\otimes \ell}\rangle}{\hat{w}}, \forall \alpha \in [n]$. Then

$$|w - \hat{w}| \le \varepsilon \tag{3}$$
$$\|v - \hat{v}\| \le \varepsilon$$

Proof Define

$$\delta = \frac{\varepsilon w}{4n^\ell}.$$

By triangle inequality, $|\|wv^{\otimes(2\ell+1)}\|_F - \|T\|_F| \le \delta \le \varepsilon$. Note $\|wv^{\otimes(2\ell+1)}\|_F = w$, hence $|w - \hat{w}| \le \delta \le \varepsilon$, i.e., (3) is true.

Fix $\alpha \in [n]$. By Cauchy-Schwarz inequality,

$$|\langle wv^{\otimes(2\ell+1)} - T, e_\alpha \otimes I_n^{\otimes \ell}\rangle| \le \delta \|e_\alpha \otimes I_n^{\otimes \ell}\|_F = \delta n^{\ell/2}.$$

As a consequence,

$$|wv_\alpha - \langle T, e_\alpha \otimes I_n^{\otimes \ell}\rangle| \le \delta n^{\ell/2}.$$

We know that $|\hat{w}v_\alpha - wv_\alpha| \le |\hat{w} - w| \le \delta$. Then,

$$|\hat{w}v_\alpha - \langle T, e_\alpha \otimes I_n^{\otimes \ell}\rangle| \le \delta(n^{\ell/2} + 1) \le 2\delta n^{\ell/2}.$$

Hence,

$$|v_\alpha - \frac{\langle T, e_\alpha \otimes I_n^{\otimes \ell}\rangle}{\hat{w}}| \le \frac{2\delta n^{\ell/2}}{\hat{w}} \le \frac{4\delta n^{\ell/2}}{w}.$$

The last inequality is due to $|w - \hat{w}| \le \delta \le w/2$.
Then we have

$$\|v - \hat{v}\| \le \frac{4\delta\sqrt{n}n^{\ell/2}}{w} \le \varepsilon$$

■

Next, we will prove Theorem 3. The main steps of the algorithm are:

1. Get a estimation of $T^* = \sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$ via Theorem 4.
2. Use Theorem 18 (tensor decomposition) to recover all the rank-one terms.
3. Use Claim 23 to recover the parameters from the rank-one terms.

Theorem 3 *Let $\ell \in \mathbb{N}$. Let $U \in \mathbb{R}^{n^\ell \times k}$ be the matrix whose jth column is flattened $v_j^{\otimes \ell}$. Suppose $\sigma_{\min}(U) \geq 1/\tau$, where $\tau > 0$. Given parameters $\varepsilon, \delta > 0$, $k \in \mathbb{N}$ and $w_{\min} > 0$ satisfying $w_{\min} \leq \min\{w_1, \ldots, w_k\}$, there is an algorithm* ESTIMATE-PARAMETER *that given samples from the model has the following guarantees:*

1. *The algorithm runs in sample complexity and time complexity*

$$\log^2(1/\delta)\varepsilon^{-2}\mathrm{poly}(n^\ell, \ell^\ell, \tau, 1/w_{\min}).$$

2. *With probabilty $1 - \delta$, the algorithm returns estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that*

$$\min_{\pi \in \mathrm{Perm}([k])} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \leq \varepsilon.$$

```
Algorithm 2: ESTIMATE-PARAMETER
Input:
$k$ – the number of component
$\ell$ – parameter for order of the tensor
$\tau$ – parameter for lower bound on least singular value of $U$
$\mathcal{O}(v_1, \cdots, v_k, w_1, \cdots, w_k)$ – the sample oracle
$\varepsilon$ – error parameter
$w_{\min}$ – weight lower bound
$\delta$ – failure probability
Output:
$\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ – estimate of $\{w_j, v_j : j \in [k]\}$
Apply Theorem 4, get $\mathbf{T}$ that is $\varepsilon\mathrm{poly}(n^{-\ell}, 1/\tau, w_{\min})$-close to $\sum_{j=1}^{k} w_j v_j^{\otimes(2\ell+1)}$ with
  probability at least 0.99;
View $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$ as $\sum_{j\in[k]} (v_j^{\otimes\ell}) \otimes (v_j^{\otimes\ell}) \otimes (w_j v_j)$. By Theorem 18, with probability at
  least 0.99, we can estimate each rank one term in the decomposition of $T^*$ (up to renaming),
  within additive error $\frac{\varepsilon w_{\min}}{8n^\ell}$;
By Claim 23, we can recover the parameters from the rank-one terms. This leads to estimates
  $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$.;
return $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$;
```

Proof The algorithm ESTIMATE-PARAMETER is described in Algorithm 2.

Without loss of generality, we can assume $\delta = 0.1$. This is because we can always boost the success probability of our algorithm at a multiplicative cost of $O(\log^2(1/\delta))$ via Claim 24.

Let $T^* = \sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$. By Theorem 4, there is an algorithm such that:

1. It runs with sample complexity and time complexity

$$\varepsilon^{-2}\mathrm{poly}(n^\ell, \ell^\ell, \tau, 1/w_{\min})$$

2. With probability 0.99, we can estimate T^* within an additive error of $\varepsilon\mathrm{poly}(n^{-\ell}, 1/\tau, w_{\min})$.

View $\sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$ as $\sum_{j\in[k]}(v_j^{\otimes\ell}) \otimes (v_j^{\otimes\ell}) \otimes (w_j v_j)$. We note that the following hold:

1. The condition numbers $\kappa(U) \le k\tau$,
2. For all $i \neq j$, we have $w_i v_i, w_j v_j$ are not close to parallel: $\text{Krank}_{\tau/w_{\min}}(W) \ge 2$,
3. For all j, we have $\|v_j^{\otimes k}\|_F, \|w_j v_j\| \le 1$.

Hence, T^* satisfies the condition of Theorem 18. By Theorem 18, with probability at least 0.99, we can estimate each rank one term in the decomposition of T^* (up to renaming), within additive error $\frac{\varepsilon w_{\min}}{8n^\ell}$. By Claim 23, we can recover the parameters from the rank-one terms. This leads to estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that

$$\min_{\pi}\left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \le \varepsilon,$$

where the min is over permutation π on $[k]$. ■

We are now ready to finish the proof of Theorem 1 and Theorem 2. The proofs has the same structure as Theorem 3.

Proof of Theorem 1. Without loss of generality, we can assume $\delta = 0.1$. This is because we can always boost the success probability of our algorithm at a multiplicative cost of $O(\log^2(1/\delta))$ via Claim 24. Define $\ell = 10(\log k)/\Delta^2$.

Let $T^* = \sum_{j\in[k]} w_j v_j^{\otimes(2\ell+1)}$. By Theorem 4, there is an algorithm such that

1. It runs in sample complexity and time complexity
$$\begin{aligned}&\varepsilon^{-2}\text{poly}(n^\ell, k, \ell^\ell, 1/w_{\min}, 1/\Delta)\\ &= \varepsilon^{-2}\text{poly}(n^{(\log k)/\Delta^2}, ((\log k)/\Delta^2)^{(\log k)/\Delta^2}, 1/w_{\min})\end{aligned}$$
2. With probability 0.99, we can estimate T^* within an additive error of $\varepsilon\text{poly}(1/n^\ell, 1/k, 1/w_{\min}, 1/\Delta)$.

By Claim 19, T^* satisfies the condition of Theorem 18. By Theorem 18, with probability at least 0.99, we can estimate each rank one term in the decomposition of T^* (up to renaming), within additive error $\frac{\varepsilon w_{\min}}{8n^\ell}$. By Claim 23, we can recover the parameters from the rank-one terms. This leads to estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that

$$\min_{\pi}\left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \le \varepsilon,$$

where the min is over permutation π on $[k]$. ■

Proof of Theorem 2. Without loss of generality, we can assume $\delta = 0.1$. This is because we can always boost the success probability of our algorithm at a multiplicative cost of $O(\log^2(1/\delta))$ via Claim 24. Let $T^* = \sum_{j\in[k]} w_j v_j^{\otimes(2k+1)}$. By Theorem 4, there is an algorithm such that:

1. It runs in sample complexity and time complexity $\varepsilon^{-2}\text{poly}(n^k, k^k, \Delta^{-k}, 1/w_{\min})$.
2. With probability 0.99, we can estimate T^* within an additive error of $\varepsilon\text{poly}(\Delta^k, k^{-k}, n^{-k}, w_{\min})$.

By Claim 22, T^* satisfies the condition of Theorem 18. By Theorem 18, with probability at least 0.99, we can estimate each rank one term in the decomposition of T^* (up to renaming), within additive error $\frac{\varepsilon w_{\min}}{8n^k}$. By Claim 23, we can recover the parameters from the rank-one terms. This leads to estimates $\{\hat{w}_j, \hat{v}_j : j \in [k]\}$ such that

$$\min_{\pi} \left(\max\{\|\hat{v}_j - v_{\pi(j)}\| : j \in [k]\} + \max\{\left|\hat{w}_j - w_{\pi(j)}\right| : j \in [k]\}\right) \leq \varepsilon,$$

where the $\min$ is over permutation π on $[k]$. ■

Acknowledgments

The first and last authors were supported by NSF grant CCF-1652491. In addition the last author was also supported by NSF grants CCF-1637585 and CCF-1934931, while the second author was supported by NSF grants CCF-1910534, CCF-1926872 and CCF-2045128.

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.

Pranjal Awasthi, Alex Tang, and Aravindan Vijayaraghavan. Efficient algorithms for learning depth-2 neural networks with general relu activations. *ArXiv 2107.10209*, 2021.

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.

Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*. ACM, 2014a.

Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. *Proceedings of the Conference on Learning Theory (COLT).*, 2014b.

Christopher M Bishop. Latent variable models. In *Learning in graphical models*, pages 371–403. Springer, 1998.

Arun Tejasvi Chaganty and Percy Liang. Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*, pages 1040–1048. PMLR, 2013.

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.

Sitan Chen, Jerry Li, and Zhao Song. Learning mixtures of linear regressions in subexponential time via fourier moments. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 587–600, 2020.

Ilias Diakonikolas and Daniel M Kane. Small covers for near-zero sets of polynomials and learning latent variable models. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–195. IEEE, 2020.

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.

Venkata Gandikota, Arya Mazumdar, and Soumyabrata Pal. Recovery of sparse linear classifiers from mixture of responses. In *Advances in Neural Information Processing Systems*, volume 33, pages 14688–14698, 2020.

Rong Ge, Jason D. Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkwHObbRZ.

Navin Goyal, Santosh Vempala, and Ying Xiao. Fourier PCA and robust tensor decomposition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 584–593, 2014. doi: 10.1145/2591796.2591875. URL http://doi.acm.org/10.1145/2591796.2591875.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Majid Janzamin, Rong Ge, Jean Kossaifi, and Animashree Anandkumar. Spectral learning on matrices and tensors. *Foundations and Trends in Machine Learning*, 12, 11 2019. doi: 10.1561/2200000057.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

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.

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.

Keith Y Patarroyo. A digression on hermite polynomials. *arXiv preprint arXiv:1901.01648*, 2019.

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.

Yuekai Sun, Stratis Ioannidis, and Andrea Montanari. Learning mixtures of linear classifiers. In *International Conference on Machine Learning*, pages 721–729. PMLR, 2014.

Kert Viele and Barbara Tong. Modeling with mixtures of linear regressions. *Statistics and Computing*, 12(4):315–330, 2002.

Appendix A. Auxiliary claim

The following claim is well-known, though we do not know a suitable reference for it. We include it here for the sake of completeness.

Claim 24 *Let X be a metric space with metric d. There is a fixed hidden element $x^* \in X$. $\varepsilon > 0$. Suppose there is a randomized algorithm* ALG *whose output $\mathbf{x}$ satisfies*

$$\mathbb{P}[d(\mathbf{x}, x^*) \leq \varepsilon] \geq 0.9.$$

Then, there is an algorithm SUCCESS-PROB-BOOSTER *with the following guarantee: given access to d, independent outputs from the algorithm* ALG *and a confidence parameter δ,*

1. *With probability $1 - \delta$, the algorithm returns a estimate $\hat{\mathbf{x}}$ such that*

$$d(\hat{\mathbf{x}}, x^*) \leq 3\varepsilon.$$

2. *The algorithm acquire $O(\log(1/\delta))$ independent outputs from the algorithm* ALG.
3. *The algorithm makes $O(\log^2(1/\delta))$ calls to d.*
4. *The algorithm runs in time complexity $O(\log^2(1/\delta))$.*

Proof of Claim 24. The algorithm is described in Algorithm 3.

Let $\mathcal{E}$ be the event $|\{i \in [t] : d(\mathbf{x}_i, x^*) \leq \varepsilon\}| \geq 0.8t$. Use standard Chernoff bound and the fact $t = 1000 \log(1/\delta)$, we have

$$\mathbb{P}[\mathcal{E}] \geq 1 - \delta.$$

Condition on $\mathcal{E}$. We now show that $\mathbf{x}_k$ (in the last line of the algorithm) satisfies $d(\mathbf{x}_k, x^*) \leq 3\varepsilon$.

```
Algorithm 3: SUCCESS-PROB-BOOSTER
Input:
$\delta$ – failure probability
Output:
$\hat{\mathbf{x}}$ – estimate of $x^*$
Set $t = 1000\log(1/\delta)$;
Acquire $t$ independent outputs $\mathbf{x}_1, \ldots, \mathbf{x}_t$ from the algorithm ALG;
Use BFPRT algorithm to select the $(0.3t^2)$th smallest element $\tau$ among
  $\{d(\mathbf{x}_i, \mathbf{x}_j) : 1 \le i < j \le t\}$;
Construct undirected graph $G = (V, E)$ where $V = [t]$ and $(i, j) \in E \iff d(\mathbf{x}_i, \mathbf{x}_j) \le \tau$;
Find $k \in [t]$ such that the degree of vertex $k$ is the highest in $G$;
return $\hat{\mathbf{x}} = \mathbf{x}_k$;
```

First we claim $\tau \le 2\varepsilon$. Since $\mathcal{E}$ holds, at least $\binom{0.8t}{2}$ pairs of vertices are 2ε-close to each other. Since $\binom{0.8t}{2} \ge 0.3t^2$, we know that $\tau \le 2\varepsilon$.

By the definition of τ, we know that $|E| \ge 0.3t^2$. Then the degree of the vertex k is at least $0.6t$. Since $|\{i \in [t] : d(\mathbf{x}_i, x^*) \le \varepsilon\}| \ge 0.8t$ holds, there exist $j \in [t]$ such that

1. $(k, j) \in E$.
2. $d(\mathbf{x}_j, x^*) \le \varepsilon$.

Then $d(\mathbf{x}_k, x^*) \le d(\mathbf{x}_k, \mathbf{x}_j) + d(\mathbf{x}_j, x^*) \le \tau + \varepsilon \le 3\varepsilon$. ■