On Learning Mixture Models with Sparse Parameters

Arya Mazumdar
University of California, San Diego

Soumyabrata Pal
University of Massachusetts, Amherst

Abstract

Mixture models are widely used to fit complex and multimodal datasets. In this paper we study mixtures with high dimensional sparse latent parameter vectors and consider the problem of support recovery of those vectors. While parameter learning in mixture models is well-studied, the sparsity constraint remains relatively unexplored. Sparsity of parameter vectors is a natural constraint in variety of settings, and support recovery is a major step towards parameter estimation. We provide efficient algorithms for support recovery that have a logarithmic sample complexity dependence on the dimensionality of the latent space. Our algorithms are quite general, namely they are applicable to 1) mixtures of many different canonical distributions including Uniform, Poisson, Laplace, Gaussians, etc. 2) Mixtures of linear regressions and linear classifiers with Gaussian covariates under different assumptions on the unknown parameters. In most of these settings, our results are the first guarantees on this problem while in the rest, we provide significant improvements on existing results in certain regimes.

1 INTRODUCTION

Mixture models are standard tools for probabilistic modeling of heterogeneous data, and have been studied theoretically for more than a century. Mixtures are used in practice for modeling data across different fields, such as, astronomy, genetics, medicine, psychiatry, economics, and marketing among many others [Moosman and Peel, 2000]. Mixtures with finite number of components are especially successful in modeling datasets having a group structure, or presence of a subpopulation within the overall population. Often, mixtures can handle situations where a single parametric family cannot provide a satisfactory model for local variations in the observed data [Titterington et al., 1985].

The literature on algorithmically learning mixture distributions is quite vast and comes in different flavors. Computational and statistical aspects of learning mixtures perhaps starts with [Dasgupta, 1999], and since have been the subject of intense investigation in both computer science and statistics [Achlioptas and McSherry, 2005, Kalai et al., 2010, Belkin and Sinha, 2010, Arora and Kannan, 2001, Moitra and Valiant, 2010, Feldman et al., 2008, Chan et al., 2014, Acharya et al., 2017, Hopkins and Li, 2018, Diakonikolas et al., 2018, Kothari et al., 2018, Hardt and Price, 2015]. A large portion of this literature is devoted to *density estimation* or PAC-learning, where the goal is simply to find a distribution that is close in some distance (e.g., TV distance) to the data-generating mechanism. The results on density estimation can be further subdivided into *proper* and *improper learning* approaches depending on whether the algorithm outputs a distribution from the given mixture family or not. These two guarantees turn out to be quite different.

A significant part of the literature on the other hand is devoted to *parameter estimation*, where the goal is to identify the mixing weights and the parameters of each component from samples. Apart from Gaussian mixtures, where all types of results exist, prior work for other mixture families largely focuses on density estimation, and very little is known for parameter estimation outside of Gaussian mixture models. In this paper, our focus is to facilitate parameter estimation in Gaussian mixtures and beyond. We consider the setting where the parameters of the mixture are themselves high dimensional, but sparse (i.e., have few nonzero entries). Sparsity is a natural regularizer in high dimensional parameter estimation problems and have been considered in the context of mixtures in [Verzelen and Arias-Castro, 2017, Arias-Castro and Pu, 2017, Azizyan et al., 2013], where it is assumed only few dimensions of the component means are relevant

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

for de-mixing. In this paper we consider a slightly different model where we assume the means themselves are sparse. The former problem can be reduced to our setting if one of the component means is known.

There are parameter estimation problems in other data subpopulation modeling, where functional relationships in data can be thought of as mixture of simple component models. Most prominent among these is the *mixed linear regression* problem [De Veaux, 1989]. In this setting, each sample is a tuple of (covariates,label). The label is stochastically generated by picking a linear relation uniformly from a set of two or more linear functions, evaluating this function on the covariates and possibly adding noise. The goal is to learn the set of unknown linear functions. The problem has been studied widely [Chaganty and Liang, 2013, Faria and Soromenho, 2010, Städler et al., 2010, Li and Liang, 2018, Kwon and Caramanis, 2018, Viele and Tong, 2002, Yi et al., 2014, 2016], with an emphasis on the EM algorithm and other alternating minimization (AM) techniques. It is interesting that [Städler et al., 2010] argued to impose sparsity on the solutions, implying that each linear function depends on only a small number of variables. In this paper we are concerned with exactly this same problem.

Similar to mixed linear regressions, there can be *mixed linear classifications.* In that setting, the labels are binary (or other categorical). The works in this domain is limited, with notable exceptions [Sun et al., 2014, Sedghi et al., 2016].

We consider the high dimensional parameter learning problem in a very general mixture model that covers all of the above settings. We assume the parameter vectors to be sparse, and focus on recovering the support of the vectors.

Note that, support recovery is an effective way to reduce the dimension of the latent space, and therefore can be considered as a key step towards parameter estimation. We study the support recovery problem in three different canonical mixture models as described above: mixtures of distributions (MD), mixtures of linear regressions (MLR), and mixtures of linear classifiers (MLC). The three models will differ somewhat in analysis as they pose different challenges; however there will be commonalities in the key techniques. We provide two flavors of results for support recovery namely, 1) *Exact support recovery:* where we recover the support of all unknown sparse latent parameters corresponding to all components of the mixture, 2) *Deduplicated support recovery:* where we recover the support of a crucial subset of latent parameters. To formally define the problems and state the results we need to define certain quantities.

It is worth mentioning that mixtures of sparse linear regressions and classifiers were also considered in some recent works that focus on a query-based model, i.e., where the covariates can be designed as queries [Yin et al., 2019, Krishnamurthy et al., 2019, Mazumdar and Pal, 2020, Gandikota et al., 2020, 2021, Polyanskii, 2021]. The query based setting is drastically different from our unsupervised setting, because in the former one can use the same covariates again and again to get potentially different labels, and thus identify the components. However, as we will see, some tools developed in [Gandikota et al., 2021] can still be relevant for support recovery in the current setting where we cannot dictate the covariates.

An interesting application of learning mixtures with sparse parameters is in high-dimensional clustering problems where cluster centers actually belong to a low-dimensional space. This is similar in spirit with sparse-PCA [Johnstone and Lu, 2009]; our objective is to identify a few important input features, so one can easily interpret its meaning. Our techniques can also be seen as a novel method for feature selection that can significantly speed up a learning algorithm.

Another practical application comes up naturally in recommendation systems where multiple users rate/purchase/evaluate items. User tastes can differ, and that can be modeled by a few unknown parameter vectors. It makes sense for the unknown vectors to be sparse, because most users have an affinity towards a few particular features of items among many possible. Sparse mixtures were motivated with such an application in the query based setting in [Gandikota et al., 2020, 2021].

1.1 Notations

We write $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We will use $\mathbf{1}_n, \mathbf{0}_n$ to denote an all one vector and all zero vector of dimension n respectively. We will use $\mathcal{Q}([n])$ to denote the power set of $[n]$ i.e. $\mathcal{Q}([n]) = \{\mathcal{C} \mid \mathcal{C} \subseteq [n]\}$.

For any vector $\mathbf{v} \in \mathbb{R}^n$, we use $\mathbf{v}_i$ to denote the i^{th} coordinate of $\mathbf{v}$ and for any ordered set $\mathcal{S} \subseteq [n]$, we will use the notation $\mathbf{v}_{|\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ to denote the vector $\mathbf{v}$ restricted to the indices in $\mathcal{S}$. Furthermore, we will use $\mathsf{supp}(\mathbf{v}) \triangleq \{i \in [n] : \mathbf{v}_i \neq 0\}$ to denote the support of $\mathbf{v}$ and $||\mathbf{v}||_0 \triangleq |\mathsf{supp}(\mathbf{v})|$ to denote the size of the support. Let $\mathsf{sign} : \mathbb{R} \to \{-1, +1\}$ be a function that returns the sign of a real number i.e. for any input $x \in \mathbb{R}$,

$$\mathsf{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Consider a multi-set of n-dimensional vectors $\mathcal{U} \equiv \{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(\ell)}\}$. We will write $\mathcal{S}_{\mathcal{U}}(i) \triangleq \{\mathbf{u} \in \mathcal{U} :$

$\mathbf{u}_i \neq 0\}$ to denote the multi-set of vectors in $\mathcal{U}$ that has a non-zero entry at the i^{th} index. Furthermore, for an ordered set $\mathcal{C} \subseteq [n]$ and vector $\mathbf{a} \in \{0,1\}^{|\mathcal{C}|}$, we will also write $\mathsf{occ}_{\mathcal{U}}(\mathcal{C}, \mathbf{a}) \triangleq \sum_{\mathbf{u}\in\mathcal{U}} \mathbf{1}[\mathbf{u}_{|\mathcal{C}} = \mathbf{a}]$ to denote the number of vectors in $\mathcal{U}$ that equal $\mathbf{a}$ when restricted to the indices in $\mathcal{C}$. For a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, we will use $\mathbf{M}_i$ to denote the i^{th} column of $\mathbf{M}$. Let $\mathbf{A}_{\mathcal{U}} \in \{0,1\}^{n\times \ell}$ denote the support matrix of $\mathcal{U}$ where each column vector $\mathbf{A}_i \in \{0,1\}^n$ represents the support of the vector $\mathbf{u}^{(i)} \in \mathcal{U}$. For ease of notation, we will omit the subscript $\mathcal{U}$ when the set of vectors is clear from the context.

We write $\mathcal{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2. We will denote the cumulative distribution function of a random variable Z by $\phi : \mathbb{R} \to [0,1]$ i.e. $\phi(a) = \int_{-\infty}^{a} p(z)dz$ where $p(\cdot)$ is the density function of Z. Also, we will denote $\mathsf{erf} : \mathbb{R} \to \mathbb{R}$ to be the error function defined by $\mathsf{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2)dt$. Since the error function erf is bijective, we define $\mathsf{erf}^{-1}(\cdot)$ to be the inverse of the $\mathsf{erf}(\cdot)$ function. Finally, for a fixed set $\mathcal{B}$ we will write $X \sim_{\mathsf{Unif}} \mathcal{B}$ to denote a random variable X that is uniformly sampled from the elements in $\mathcal{B}$.

1.2 Formal Problem Statements

Let $\mathcal{V}$ be a multi-set of ℓ unknown k-sparse vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(\ell)} \in \mathbb{R}^n$ such that $\left|\left|\mathbf{v}^{(i)}\right|\right|_0 \leq k$ for all $i \in [\ell]$. We consider the following problems described below:

Mixtures of Distributions with Sparse Latent Parameters (MD): Consider a class of distributions $\mathcal{P} \equiv \{\mathbf{P}(\theta)\}_{\theta\in\Theta}$ parameterized by some $\theta \in \Theta$ where $\Theta \subseteq \mathbb{R}$. We assume that all distributions in $\mathcal{P}$ satisfy the following property: $\mathbb{E}_{x\sim\mathbf{P}(\theta)} x^\ell$ can be written as a polynomial in θ of degree exactly ℓ, From Table 2 in [Belkin and Sinha, 2010], we know that many well-known distributions satisfy this property (further discussion later). A sample $\mathbf{x} \sim \mathcal{P}_d$ is generated as follows:

$$t \sim_{\mathsf{Unif}} [\ell] \text{ and } \mathbf{x}_i \mid t \sim \mathbf{P}(\mathbf{v}_i^{(t)}) \text{ independently } \forall i \in [n].$$

In other words, $\mathbf{x}$ is generated according to a uniform mixture of distributions each having a sparse unknown parameter vector. Consider $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^n$, m i.i.d. copies of $\mathbf{x}$, that we can use to recover $\mathcal{V}$.

Here are some examples of this setting:

1. $\mathbf{P}(\theta)$ can be a Gaussian distribution with mean θ and known variance σ^2. This setting corresponds to a mixture of high-dimensional isotropic Gaussian distributions with sparse means.
2. $\mathbf{P}(\theta)$ can be a uniform distribution with range $[\theta, b]$ for a fixed and known b.
3. $\mathbf{P}(\theta)$ can be a Poisson distribution with mean θ.

Mixtures of Sparse Linear Regressions (MLR). Consider m samples

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)}) \in \mathbb{R}^n \times \mathbb{R}$$

which are generated independently according to a distribution $\mathcal{P}_r$ defined as follows: for $(\mathbf{x}, y) \sim \mathcal{P}_r$, we have

$$\mathbf{x}_i \sim \mathcal{N}(0,1) \text{ independently for all } i \in [n]$$
$$\mathbf{v} \sim_{\mathsf{Unif}} \mathcal{V} \text{ and } y \mid \mathbf{x}, \mathbf{v} \sim \mathcal{N}(\langle \mathbf{v}, \mathbf{x}\rangle, \sigma^2).$$

In other words, each entry of $\mathbf{x}$ is sampled independently from $\mathcal{N}(0,1)$ and for a fixed $\mathbf{x}$, the conditional distribution of y given $\mathbf{x}$ is a Gaussian with mean $\langle \mathbf{v}, \mathbf{x}\rangle$ and known variance σ^2 where $\mathbf{v}$ is uniformly sampled from the multi-set $\mathcal{V}$.

Mixtures of Sparse Linear Classifiers (MLC). Consider m samples

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)}) \in \mathbb{R}^n \times \{-1, +1\}$$

which are generated independently according to a distribution $\mathcal{P}_c$ defined as follows: for $(\mathbf{x}, y) \sim \mathcal{P}_c$, we have

$$\mathbf{x}_i \sim \mathcal{N}(0,1) \text{ independently for all } i \in [n]$$
$$\mathbf{v} \sim_{\mathsf{Unif}} \mathcal{V} \text{ and } z \sim \mathcal{N}(0, \sigma^2) \text{ and } y = \mathsf{sign}(\langle \mathbf{v}, \mathbf{x}\rangle + z).$$

In other words, each entry of $\mathbf{x}$ is sampled independently from $\mathcal{N}(0,1)$ and for a fixed $\mathbf{x}$, the conditional distribution of y given $\mathbf{x}$ is $+1$ if $\langle \mathbf{v}, \mathbf{x}\rangle \geq -z$ and -1 otherwise; here, $\mathbf{v}$ is uniformly sampled from the multi-set of unknown vectors $\mathcal{V}$ and z denotes zero mean Gaussian noise with variance σ^2.

Our goal in all the three problems described above is to recover the support of unknown vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(\ell)} \in \mathcal{V}$ with minimum number of samples m. More formally, we look at two distinct notions of support recovery:

Definition 1 (Exact Support Recovery)**.** *We will say that an algorithm achieves Exact Support Recovery in the MLC/MLR/MD setting if it can recover the support of all the unknown vectors in $\mathcal{V}$ exactly.*

Definition 2 (Deduplicated set)**.** *A deduplicated set $\mathcal{V}'$ is a subset of $\mathcal{V}$ such that 1) $\mathsf{supp}(\mathbf{v}^{(1)}) \not\subseteq \mathsf{supp}(\mathbf{v}^{(2)})$ for any distinct $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathcal{V}'$ and 2) $\mathbf{v} \notin \mathcal{V}'$ if there exists $\mathbf{v}' \in \mathcal{V}$ satisfying $\mathsf{supp}(\mathbf{v}) \subseteq \mathsf{supp}(\mathbf{v}')$. Now,*

$$\mathsf{Trimmed}(\mathcal{V}) \triangleq \mathsf{argmax}_{\mathcal{V}'\subseteq\mathcal{V}} |\mathcal{V}'| \tag{1}$$

where the maximization is over all deduplicated sets.

We can show that the set $\mathsf{Trimmed}(\mathcal{V})$ is unique (see Lemma 13 in Appendix D).

Definition 3 (Deduplicated Support Recovery)**.** *We will say that an algorithm achieves Deduplicated support recovery in the MLR/MLC/MD setting if it can recover the support of all the unknown vectors in* $\mathsf{Trimmed}(\mathcal{V})$ *exactly.*

Note that in Definition 3, the objective is to recover supports of the largest set of vectors in $\mathcal{V}$, where no support is included completely in another support; this is easier than exact support recovery (Definition 1).

Remark 1. *If every unknown vector* $\mathbf{v} \in \mathcal{V}$ *had a unique non-zero index* $i \in [n]$ *i.e.* $\mathbf{v}_i \neq 0$ *and* $\mathbf{v}'_i = 0$ *for all* $\mathbf{v}' \in \mathcal{V} \setminus \{\mathbf{v}\}$*, then Deduplicated support recovery is equivalent to Exact Support Recovery. This condition, also known as the separability condition, has been commonly used in the literature for example in unique non-negative matrix factorization [Arora et al., 2016, Donoho and Stodden, 2004, Slawski et al., 2013] and approximate parameter recovery in MLC in the query-based setting [Gandikota et al., 2020].*

Note that a trivial approach to the support recovery problem is to first recover the union of support and then apply existing parameter estimation guarantees in the corresponding mixture setting. However, note that this approach crucially requires parameter estimation results for the corresponding family of mixtures which may be unavailable. We have provided a detailed discussion on our results and other relevant work including the alternate approach outlined above in Appendix C.

Main Technical Contribution beyond [Gandikota et al., 2021]. As discussed earlier, our unsupervised setting is different from the query-based setting of [Gandikota et al., 2021], where the focus is also support recovery. However, we crucially use a general technique introduced in [Gandikota et al., 2021] (see Lemma 1) for exact support recovery. Namely, support recovery is possible if we can estimate some subset statistics.

But computing estimates of these subset statistics to invoke the guarantees given in Lemma 1 is a difficult problem. For the three settings, namely MD/MLR/MLC, we provide distinct and novel techniques to compute these quantities. Our approach to compute the sufficient statistics in MD setting involve a two-step approach with polynomial identities : 1) first, using the method of moments, we compute estimates of the power sum polynomial of degree p in the variables $\{\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$ for all subsets $\mathcal{C} \subset [n]$ up to a certain size; 2) secondly, we use an elegant connection via Newton's identities to compute estimates on the elementary symmetric polynomial in the variables $\{\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$ which in turn allows us to compute the sufficient statistics. In MLR, for a set $\mathcal{C} \subseteq [n]$, we again analyze an interesting quantity namely $y^{|\mathcal{C}|} \cdot \left(\prod_{i\in\mathcal{C}} \mathbf{x}_i\right)$ that reveals the sufficient statistic for invoking Lemma 1. In MLC, our method is quite different and it involves conditioning on the event that certain coordinates of the covariate have large values. If this event is true, then analyzing the response variables reveals the sufficient statistics for invoking Lemma 1.

Organization: The rest of the paper is organized as follows: in Section 2, we provide the necessary preliminary lemmas for exact support recovery. In Section 3, we provide our main results on exact support recovery and discuss our core approaches in each of the settings namely MD/MLR/MLC at a high level. For example, see Corollary 2, Theorem 4, and Theorem 2 for representative results in the three settings respectively. In Appendix A.1 and A, we have provided additional results on Deduplicated support recovery. In Appendix B.1, B.2 and B.3, we provide the detailed proofs of all our results in the MD, MLC and MLR setting respectively. In Appendix C, we provide a detailed discussion on our Results and other related works. In Appendix D, we provide the missing proofs of lemmas in Section 2 and in Appendix F, we provide the proof of Lemma 1 proved in [Gandikota et al., 2021]. In Appendix E, we provide some technical lemmas that are used in the main proofs.

2 PRELIMINARIES

The missing proofs and algorithms of this section can be found in Appendix D.

To derive our support recovery results, we will crucially use the result of Lemma 1 below which has been proved in [Gandikota et al., 2021]. Recall the definition of $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ in Sec. 1.1. Lemma 1 states that if $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ is known for all sets $\mathcal{C} \subseteq [n]$ up to a cardinality of $\log \ell + 1$, then it is possible to recover the support of all the unknown vectors in $\mathcal{V}$. We restate the result according to our terminology.

Lemma 1. *[Corollary 1 in [Gandikota et al., 2021]] Let* $\mathcal{V}$ *be a set of* ℓ *unknown vectors in* $\mathbb{R}^n$*. Then, if* $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ *is provided as input for all sets* $\mathcal{C} \subset [n], |\mathcal{C}| \leq \log \ell + 1$ *and for all* $\mathbf{a} \in \{0,1\}^{|\mathcal{C}|}$*, then there exists an algorithm (see Algorithm 10) that can recover the support of the unknown vectors in* $\mathcal{V}$*.*

For the sake of completeness, we provided Algorithm 10 and Lemma 1 proof in Appendix F.

Remark 2. *Lemma 1 provides an unconditional guarantee for recovering the support of the unknown vectors in* $\mathcal{V}$*. In other words, in the worst case, we only need to*

know $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ *for all sets of size* $|\mathcal{C}| \leq \log \ell + 1$. *However, in [Gandikota et al., 2021][Theorems 1,2 and 4], significantly relaxed sufficient conditions for recovering the support of* $\mathcal{V}$ *under different structural assumptions were also provided. As noted in [Gandikota et al., 2021], these additional conditions are mild and in most cases, if* $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ *is known for all sets* $\mathcal{C} \subseteq [n]$ *up to a cardinality of* 3, *then it is possible to recover the support of all the unknown vectors in* $\mathcal{V}$.

Next, we describe another result, Lemma 2, proved in [Gandikota et al., 2021] that is also going to be useful for us. The main takeaway from Lemma 2 is that computing $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ (which represents the number of unknown vectors in $\mathcal{V}$ having non-zero values in at least one entry corresponding to $\mathcal{C}$) for all sets smaller than a fixed size (say t) is sufficient to compute $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ for all subsets $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq t$ and all vectors $\mathbf{a} \in \{0,1\}^{|\mathcal{C}|}$. In addition, we provide a result in Lemma 2 where we show that it is also possible to compute $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ if the quantities $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (which represents the number of unknown vectors in $\mathcal{V}$ having non-zero values in all entries corresponding to $\mathcal{C}$) are provided for all subsets $\mathcal{C} \subseteq [n]$ satisfying $|\mathcal{C}| \leq t$.

Lemma 2 (Partially proved in [Gandikota et al., 2021])**.** *Let* $\mathcal{V}$ *be a set of* ℓ *unknown vectors in* $\mathbb{R}^n$. *If* $|\bigcup_{i\in\mathcal{C}}\mathcal{S}(i)|$ *is provided as input for all sets* $\mathcal{C} \subset [n], |\mathcal{C}| \leq t$ *or alternatively* $|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)|$ *is provided as input for all sets* $\mathcal{C} \subset [n], |\mathcal{C}| \leq t$, *then we can compute* $\mathsf{occ}(\mathcal{C}, \mathbf{a})$ *for all sets* $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq t, \mathbf{a} \in \{0,1\}^{|\mathcal{C}|}$.

Corollary 1. *Let* $\mathcal{V}$ *be a set of* ℓ *unknown* k*-sparse vectors in* $\mathbb{R}^n$. *Suppose, for each* $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \log \ell + 1$, *we can compute* $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ *(or alternatively* $|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)|$*) with probability* $1-\gamma$ *using* $\mathsf{T}\log\gamma^{-1}$ *samples where* T *is independent of* γ. *Then, there exists an algorithm (see Algorithm 6) that can achieve Exact Support Recovery with probability at least* $1-\gamma$ *using* $O(\mathsf{T}\log(\gamma^{-1}(n + (\ell k)^{\log \ell + 1})))$ *samples.*

3 RESULTS AND TECHNIQUES

3.1 Mixtures of Distributions

In this section, we will present our main results and high level techniques in the MD setting. The detailed proofs of all results in this section can be found in Section B.1. We will start by introducing some additional notations specifically for this setting.

Additional Notations for MD: Recall that $\mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t$ can be written as a polynomial in θ of degree t. We will write

$$q_t(\theta) \triangleq \mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t = \sum_{i\in[t+1]}\beta_{t,i}\theta^{i-1}$$

Algorithm 1 RECOVER $|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)|$ IN MD SETTING

Require: Samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \sim \mathcal{P}_d$. Set $\mathcal{C} \subseteq [n]$.

1: For every $\mathbf{z} \leq 2\ell\mathbf{1}_{|\mathcal{C}|}$, compute estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ using Algorithm 9 on the set of samples $\{(\mathbf{x}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\}_{j=1}^m$.

2: For every $\mathbf{z} \leq 2\ell\mathbf{1}_{|\mathcal{C}|}$, compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ recursively using equation $\ell\widehat{U}^{\mathbf{z}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\widehat{V}^{\mathbf{u}} = \zeta_{\mathbf{z},\mathbf{z}}\cdot\widehat{V}^{\mathbf{z}}$.

3: For every $t \in [\ell]$, compute an estimate $\widehat{\mathsf{A}}_{\mathcal{C},t}$ of $\sum_{\substack{\mathcal{C}'\subseteq[\ell]\\|\mathcal{C}'|=t}}\prod_{\substack{i\in\mathcal{C}\\j\in\mathcal{C}'}}(\mathbf{v}_i^{(j)})^2$ recursively using Newton's identity $t\widehat{\mathsf{A}}_{\mathcal{C},t} = \sum_{p=1}^t(-1)^{p+1}\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}$.

4: Return $\max_{t\in[\ell]} t.\mathbf{1}[\widehat{\mathsf{A}}_{\mathcal{C},t} > 0]$.

to denote this aforementioned polynomial where we use $\{\beta_{t,i}\}_{i\in[t+1]}$ to denote its coefficients. For all sets $\mathcal{A} \subseteq [n]$, we will write $\mathcal{Q}_i(\mathcal{A})$ to denote all subsets of $\mathcal{A}$ of size at most i i.e. $\mathcal{Q}_i(\mathcal{A}) = \{\mathcal{C} \mid \mathcal{C} \subseteq \mathcal{A}, |\mathcal{C}| \leq i\}$. Let us define the function $\pi : \mathcal{Q}([n]) \times [n] \to [n]$ to denote a function that takes as input a set $\mathcal{C} \subseteq [n]$, an index $r \in \mathcal{C}$ and returns as output the position of r among all elements in $\mathcal{C}$ sorted in ascending order. In other words, for a fixed set $\mathcal{C}$ and all $j \in [|\mathcal{C}|]$, $\pi(\mathcal{C}, \cdot)$ maps the j^{th} smallest index in $\mathcal{C}$ to j; for example, if $\mathcal{C} = \{3, 5, 9\}$, then $\pi(\mathcal{C}, 3) = 1, \pi(\mathcal{C}, 5) = 2$ and $\pi(\mathcal{C}, 9) = 3$.

We will write $\mathbb{Z}^+$ to denote the set of non-negative integers and $(\mathbb{Z}^+)^n$ to denote the set of all n-dimensional vectors having entries which are non-negative integers. For two vectors $\mathbf{u}, \mathbf{t} \in (\mathbb{Z}^+)^n$, we will write $\mathbf{u} \leq \mathbf{t}$ if $\mathbf{u}_i \leq \mathbf{t}_i$ for all $i \in [n]$; similarly, we will write $\mathbf{u} < \mathbf{t}$ if $\mathbf{u}_i < \mathbf{t}_i$ for some $i \in [n]$. For any fixed subset $\mathcal{C} \subseteq [n]$ and vectors $\mathbf{u}, \mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we will write $\zeta_{\mathbf{t},\mathbf{u}}$ to denote the quantity $\zeta_{\mathbf{t},\mathbf{u}} \triangleq \prod_{i\in\mathcal{C}}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$. For any $\mathbf{u}, \mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} < \mathbf{z}$, we will define a path M to be a sequence of vectors $\mathbf{z}_1 > \mathbf{z}_2 > \cdots > \mathbf{z}_m$ such that $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m \in (\mathbb{Z}^+)^n$, $\mathbf{z}_1 = \mathbf{z}$ and $\mathbf{z}_m = \mathbf{u}$. Let $\mathcal{M}(\mathbf{z}, \mathbf{u})$ be the set of all paths starting from $\mathbf{z}$ and ending at $\mathbf{u}$. We will also write a path $\mathsf{M} \in \mathcal{M}(\mathbf{z}, \mathbf{u})$ uniquely as a set of $m-1$ ordered tuples $\{(\mathbf{z}_1, \mathbf{z}_2), (\mathbf{z}_2, \mathbf{z}_3), \ldots, (\mathbf{z}_{m-1}, \mathbf{z}_m)\}$ where each tuple consists of adjacent vectors in the path sequence. We will also write $\mathcal{T}(\mathsf{M}) \equiv \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ to denote the set of elements in the path.

We start with the following assumption which states that every unknown vector is bounded within an euclidean ball and furthermore, the magnitude of every non-zero co-ordinate of all unknown vectors is bounded from below:

Assumption 1. *We will assume that all unknown vectors in the set* $\mathcal{V}$ *are bounded within a ball of known*

radius R i.e. $\left|\left|\mathbf{v}^{(i)}\right|\right|_2 \leq R$ *for all* $i \in [\ell]$. *Furthermore, the magnitude of all non-zero entries of all unknown vectors in* $\mathcal{V}$ *is bounded from below by* δ *i.e.* $\min_{\mathbf{v}\in\mathcal{V}} \min_{i:\mathbf{v}_i\neq 0} |\mathbf{v}_i| \geq \delta$.

Now, we show our main lemma in this setting where we characterize the sufficient number of samples to compute $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right|$ for each set $\mathcal{C} \subseteq [n]$ with high probability in terms of the coefficients of the polynomials $\{q_t(\theta)\}_t$:

Lemma 3. *Suppose Assumption 1 is true. Let*

$$\Phi \triangleq \frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell!} \times \left(\max_{\mathbf{z}\leq 2\ell\mathbf{1}_{|\mathcal{C}|}} \frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}$$

$$g_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z}\leq 2\ell\mathbf{1}_{|\mathcal{C}|}} \mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $g_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 1) that can compute $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right|$ exactly for each set $\mathcal{C} \subseteq [n]$ with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|}) g_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_d$.

In order to prove Lemma 3, we first show that (see Lemma 10) for each fixed ordered set $\mathcal{C} \subseteq [n]$ and each vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we must have

$$\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}} = \frac{1}{\ell}\sum_{\mathbf{u}\leq\mathbf{t}} \zeta_{\mathbf{t},\mathbf{u}} \cdot \Big(\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big). \quad (2)$$

Note that each summand in equation 2 is a product of the powers of the co-ordinates of the same unknown vector. In Lemma 11, we show that for each set $\mathcal{C} \subseteq [n]$ and any vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we can compute $\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{t}_{\pi(\mathcal{C},i)}}$ via a recursive procedure provided for all $\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} \leq \mathbf{t}$, the quantity $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ is pre-computed. This implies that we can compute $\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{2p}$ for all $p \in [\ell]$ from the quantities $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u} \leq 2p\mathbf{1}_{|\mathcal{C}|}$. It is easy to recognize $\sum_{\mathbf{v}\in\mathcal{V}} \Big(\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\Big)^p$ as the power sum polynomial of degree p in the variables $\{\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. Now, let us define the quantity $\mathsf{A}_{\mathcal{C},t}$ for a fixed ordered set $\mathcal{C}$ and parameter $t \in [\ell]$ as follows:

$$\mathsf{A}_{\mathcal{C},t} \triangleq \sum_{\substack{\mathcal{C}'\subseteq[\ell]\\|\mathcal{C}'|=t}} \prod_{\substack{i\in\mathcal{C}\\ j\in\mathcal{C}'}} (\mathbf{v}_i^{(j)})^2$$

Notice that $\mathsf{A}_{\mathcal{C},t} > 0$ if and only if there exists a subset $\mathcal{C}' \subseteq [\ell], |\mathcal{C}'| = t$ such that $\mathbf{v}_i^{(j)} \neq 0$ for all $i \in \mathcal{C}, j \in \mathcal{C}'$. Hence, the maximum value of t such that $\mathsf{A}_{\mathcal{C},t} > 0$ is the number of unknown vectors in $\mathcal{V}$ having non-zero value in all the indices in $\mathcal{C}$. In other words, we have that

$$\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right| = \max_{t\in[\ell]} t \cdot \mathbf{1}[\mathsf{A}_{\mathcal{C},t} > 0].$$

Notice that $\mathsf{A}_{\mathcal{C},t}$ is the elementary symmetric polynomial of degree t in the variables $\{\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. We can use Newton's identities to state that for all $t \in [\ell]$,

$$t\mathsf{A}_{\mathcal{C},t} = \sum_{p=1}^{t} (-1)^{p+1} \mathsf{A}_{\mathcal{C},t-p} \Big(\sum_{\mathbf{v}\in\mathcal{V}} \Big(\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\Big)^p\Big)$$

using which, we can recursively compute $\mathsf{A}_{\mathcal{C},t}$ for all $t \in [\ell]$ ($\mathsf{A}_{\mathcal{C},0} = 1$) and hence $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right|$ if we were given $\sum_{\mathbf{v}\in\mathcal{V}} \Big(\prod_{i\in\mathcal{C}} \mathbf{v}_i^2\Big)^p$ as input for all $p \in [\ell]$ (see Lemma 12). Lemma 3 follows from making these set of computations robust. We next show Theorem 1 which follows from applying Lemma 3 and Corollary 1.

Theorem 1. *Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}} \mathsf{supp}(\mathbf{v}))$ and*

$$\Phi_m \triangleq \frac{\delta^{2\ell m}}{2\Big(3\ell\max(R^{2\ell m}, 2^\ell R^{\ell+m})\Big)^{(\ell-1)}\ell!} \times \left(\max_{\mathbf{z}\leq 2\ell\mathbf{1}_m} \frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}$$

$$f_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\leq 2\ell\mathbf{1}_{\log\ell+1}\\ \mathcal{C}\in\mathcal{F}_{\log\ell+1}}} \frac{\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2_{\log\ell+1}}.$$

Here $f_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ; furthermore, $f_{\ell,\mathcal{V}}$. Then, there exists an algorithm (see Algorithm 1 and 6) that achieves Exact Support Recovery with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{\log\ell+1}(n+(\ell k)^{\log\ell+1}))f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_d$.

Remark 3. *We can relax Assumption 1 in Theorem 1 without much further work. For our proofs to work out verbatim, it is sufficient to just have the following condition be true: given the latent variable t denoting the mixture component, coordinates of the random vector $\mathbf{x} \sim \mathcal{P}_d$ must be $(\log\ell+1)$-wise independent (any $\log\ell+1$ co-ordinates are independent). However, for the sake of simplicity, we have provided the setting where all co-ordinates of $\mathbf{x} \mid t$ are independent.*

Example: *Consider the setting when we obtain m i.i.d samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^n$ from a high dimensional Gaussian mixture $\mathcal{D} = \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}^{(1)}, \sigma^2\mathbf{I}) +$*

$\frac{1}{2}\mathcal{N}(\boldsymbol{\mu}^{(2)}, \sigma^2\mathbf{I})$ *with two components where* $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)} \in \mathbb{R}^n$ *satisfying* $\|\boldsymbol{\mu}^{(1)}\|_0, \|\boldsymbol{\mu}^{(2)}\|_0 \le k$ *are unknown and* $\sigma > 0$ *is known. Our goal is to recover the support of* $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}$ *while minimizing the number of samples* m*. For* $\mathbf{x} \sim \mathcal{D}$*, for all* $i \in [n]$*, we have that* $\mathbb{E}\mathbf{x}_i^2 = \sigma^2 + ((\boldsymbol{\mu}_i^{(1)})^2 + (\boldsymbol{\mu}_i^{(2)})^2)/2$*; for all* $i, j \in [n], i \neq j$*, we have*

$$\mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2 = \sigma^2(\mathbb{E}\mathbf{x}_i^2 + \mathbb{E}\mathbf{x}_j^2) - \sigma^4 + \Big(\frac{(\boldsymbol{\mu}_i^{(1)})^2(\boldsymbol{\mu}_j^{(1)})^2 + (\boldsymbol{\mu}_i^{(2)})^2(\boldsymbol{\mu}_j^{(2)})^2}{2}\Big)$$

Hence, in the first step, for all $i \in [n]$*, with probability* $1 - \gamma$ *we compute an estimate* u_i *of* $\mathbb{E}\mathbf{x}_i^2$ *(using Lemma 18) satisfying* $\left|u_i - \mathbb{E}\mathbf{x}_i^2\right| \le \delta^4/(64\sigma^2)$ *using* $O(\delta^{-8}\sigma^4 \max_i(\sigma^4, (\boldsymbol{\mu}_i^{(1)})^4, (\boldsymbol{\mu}_i^{(2)})^4)\log(n\gamma^{-1}))$ *samples. With this, we can infer the union of support correctly to be* $\mathcal{S} \equiv \{i \in [n] \mid u_i - \sigma^2 \ge \delta^2/4\}$*. This is because for any index* i *in the union of support, we must have* $\mathbb{E}\mathbf{x}_i^2 \ge \sigma^2 + \delta^2/2$ *while for any index* i *not in the union, we have* $\mathbb{E}\mathbf{x}_i^2 = \sigma^2$*. Next, in the second step, for all* $i, j \in \mathcal{S}; i \neq j$*, we compute an estimate* u'_{ij} *of* $\mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2$ *satisfying* $|u'_{ij} - \mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2| \le \delta^4/16$ *using* $O(\delta^{-8}\max_{i,j}(\sigma, \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\mu}_j^{(1)}, \boldsymbol{\mu}_i^{(2)}, \boldsymbol{\mu}_j^{(2)})^8 \log(n\gamma^{-1}))$ *samples with probability at least* $1 - \gamma$ *(see Lemma 18). In that case, if* i, j *belongs to the support of the same vector, then we will have* $|u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \ge 13\delta^4/32$ *while otherwise, we must have* $|u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \le 3\delta^4/32$*. Hence,* $\mathcal{T} = \{(i, j) \in \mathcal{S}, i \neq j \mid |u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \ge 13\delta^4/32\}$*. If there does not exist* $i, j \in \mathcal{S}, i \neq j$ *such that* $(i, j) \notin \mathcal{T}$*, then we return* $\mathsf{supp}(\boldsymbol{\mu}^{(1)}) = \mathsf{supp}(\boldsymbol{\mu}^{(2)}) = \mathcal{S}$ *implying that both supports are same. On the other hand, if there exists* $i, j \in \mathcal{S}, i \neq j$ *such that* $(i, j) \notin \mathcal{T}$ *then* i *belongs to the support of one vector while* j *belongs to the support of the other vector (both supports are not same). Let the support of one vector will be* $\{s \in \mathcal{S}, s \neq i \mid (i, s) \in \mathcal{T}\}$ *and the support of the other vector is* $\{s \in \mathcal{S}, s \neq j \mid (j, s) \in \mathcal{T}\}$*. Therefore, the sufficient sample complexity for recovering the support is* $m = O(\delta^{-8}\max_{i,j}(\sigma, \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\mu}_j^{(1)}, \boldsymbol{\mu}_i^{(2)}, \boldsymbol{\mu}_j^{(2)})^8 \log(n\gamma^{-1}))$*. Note that in this example, the algorithm is slightly different from the one presented in Algorithm 1; in, fact the algorithm follows that of deduplicated support recovery (see Section A) which is equivalent to exact support recovery for* $\ell = 2$ *(see Remark 1).*

Now, we provide a corollary of Theorem 1 specifically for mean-estimation in a mixture of distributions with constant number of components i.e. $\ell = O(1)$. In particular, consider the setting where

$t \sim_{\mathsf{Unif}} [\ell]$ and $\mathbf{x}_i \mid t \sim \mathbf{P}(\mathbf{v}_i^{(t)})$ independently $\forall i \in [n]$

$\mathbb{E}_{\mathbf{x}\sim\mathcal{P}_d}[\mathbf{x}_i \mid t = j] = \mathbf{v}_i^{(j)}$

i.e. the mean of the i^{th} co-ordinate of the random vector $\mathbf{x}$ distributed according to $\mathcal{P}_d$ is $\mathbf{v}_i^{(j)}$.

Corollary 2. *Consider the mean estimation problem described above. Let* $\mathcal{V}$ *be a set of* $\ell = O(1)$ *unknown vectors in* $\mathbb{R}^n$ *satisfying Assumption 1 and* $f_{\ell,\mathcal{V}}$ *be as defined in Theorem 1. Then, there exists an algorithm (see Algorithm 1 and 6) that with probability at least* $1 - \gamma$*, achieves Exact Support Recovery using* $O\Big(\log(n\gamma^{-1})\mathsf{poly}(\delta R^{-1})f_{\ell,\mathcal{V}}\Big)$ *samples generated according to* $\mathcal{P}_d$*.*

We can compare the sample complexity presented in Corollary 2 with the alternate approach for support recovery namely the two stage process of recovering the union of support followed by parameter estimation restricted to the union of support. As discussed in Section 1, most known results (other than [Moitra and Valiant, 2010]) for parameter estimation in Gaussian mixtures without separability assumptions hold for two mixtures and are therefore not applicable for $\ell > 2$. For general value of ℓ, the only known sample complexity guarantees for parameter estimation in mixture of Gaussians is provided in [Moitra and Valiant, 2010].

Note that computing the union of support is not difficult in the MD setting. In particular, in Lemma 3, the guarantees include the sample complexity of testing whether a particular index belongs to the union of support; this can be used to compute the union of support itself after taking a union bound over all indices leading to a multiplicative $\log n$ factor.

However, for one dimensional Gaussian mixture models (1D GMM), the parameter estimation guarantees in [Moitra and Valiant, 2010] (See Corollary 5) are polynomial in the inverse of the failure probability. Since parameter estimation in 1D GMM is used as a framework for solving the high dimensional problem, it can be extracted that the sample complexity in n dimensions must be polynomial in n with degree at least 1 to achieve a per coordinate error (error in ℓ_∞ norm). If restricted to the union of support of the unknown vectors in $\mathcal{V}$, then using the guarantees in [Moitra and Valiant, 2010] directly will lead to a polynomial dependence on ℓk. In essence, the sample complexity of the alternate approach has a logarithmic dependence on the latent space dimension and a polynomial dependence on sparsity k (for constant ℓ). Note that our sample complexity only has a logarithmic dependence on the dimension n (and is independent of k for constant ℓ) and is therefore essentially *dimension-free.*

For other distributions, to the best of our knowledge, the only known parameter estimation results that exist in literature are [Belkin and Sinha, 2010, Krishnamurthy et al., 2020]. In both of these works, the authors use the same assumption that $\mathbb{E}_{x\sim\mathbf{P}(\theta)}x^\ell$ can

be written as a polynomial in θ of degree exactly ℓ. While the guarantees in [Belkin and Sinha, 2010] are non-constructive, the results in [Krishnamurthy et al., 2020] need the restrictive assumption that the means must be multiple of some $\epsilon > 0$ and moreover, they have an exponential dependence on the noise variance and ϵ^{-1}. Our results do not have these limitations and are therefore widely applicable.

We also show additional results on Deduplicated Support Recovery in MD setting but due to space limitations, we have provided them in Appendix A.2.

3.2 Mixtures of Linear Classifiers

Algorithm 2 RECOVER $\left|\bigcup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ IN MLC SETTING

Require: Samples $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)}) \sim \mathcal{P}_c$. Set $\mathcal{C} \subseteq [n]$. Parameter $a > 0$.

1: Find the subset of samples $\mathcal{T} = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid \mathbf{x}_j^{(i)} > a \text{ for all } i \in [m]\}$.

2: Compute an estimate $\widehat{P}$ of $\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}})$ as $\widehat{P} = |\mathcal{T}|^{-1} \sum_{(\mathbf{x},y)\in\mathcal{T}} \mathbf{1}[y = 1]$.

3: Find $t \in [\ell]$ such that

$$\frac{1}{2}\Big(1 + \frac{t}{\ell}\Big) - \frac{t}{4\ell^2} \le \widehat{P} \le \frac{1}{2}\Big(1 + \frac{t}{\ell}\Big)$$

4: Return t

In this section, we will present our main results and high level techniques in the MLC setting. The detailed proofs of all results in this section can be found in Section B.2. We solve the sparse recovery problem when the observed samples are generated according to $\mathcal{P}_c$ under the following assumption which states that the unknown vectors in $\mathcal{V}$ either all have non-negative entries or they all have non-positive entries.

Assumption 2. *The non-zero entries of unknown vectors in $\mathcal{V}$ are all either positive ($\mathbf{v}_i \ge 0$ for all $i \in [n], \mathbf{v} \in \mathcal{V}$) or they are all negative ($\mathbf{v}_i \le 0$ for all $i \in [n], \mathbf{v} \in \mathcal{V}$).*

Although Assumption 2 looks restrictive, it can often be made in practice. As an example, in the recommendation system application motivated in the introduction (Section 1), the affinity of the users towards the different item features can be modeled by non-negative values; such a modeling assumption is similar to the motivation presented in the literature for non-negative matrix factorization [Ding et al., 2008].

Next, if Assumption 2 is satisfied, we show the sample complexity of computing $\left|\bigcup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ for each set $\mathcal{C} \subseteq [n]$.

Lemma 4. *Suppose Assumptions 1 and 2 are true. Let $a = \frac{\sqrt{2(R^2+\sigma^2)}}{\delta}\mathsf{erf}^{-1}\Big(1 - \frac{1}{2\ell}\Big)$. There exists an algorithm (see Algorithm 2) that can compute $\left|\bigcup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ for each set $\mathcal{C} \subseteq [n]$ with probability at least $1-\gamma$ using $O\Big((1-\phi(a))^{-|\mathcal{C}|}\ell^2 \log\gamma^{-1}\Big)$ i.i.d samples from $\mathcal{P}_c$.*

Let us present a high level proof of Lemma 4. Without loss of generality, let us assume that all unknown vectors in $\mathcal{V}$ have positive non-zero entries. For a fixed set $\mathcal{C} \subseteq [n]$, suppose we condition on the event $\mathcal{E}_{\mathcal{C}}$ which is true when for all $j \in \mathcal{C}$, $\mathbf{x}_j > a$ for some suitably chosen $a > 0$. Furthermore, let $\mathcal{E}_{\mathbf{v}}$ be the event that the particular vector $\mathbf{v} \in \mathcal{V}$ is used to generate the sample $(\mathbf{x}, y)$. Notice that if $\mathbf{v}_i = 0$ for all $i \in \mathcal{C}$, then conditioning on the event $\mathcal{E}_{\mathcal{C}}$ does not change the distribution of the response $y \mid \mathcal{E}_{\mathbf{v}}$; hence the probability of $y = 1$ is exactly $1/2$ in this case. On the other hand, if $\mathbf{v}_i \neq 0$ for some $i \in \mathcal{C}$, then conditioning on the event $\mathcal{E}_{\mathcal{C}}$ does change the distribution of the response $y \mid \mathcal{E}_{\mathbf{v}}$. In particular, if $\mathbf{v}_i \neq 0$, note that $\langle \mathbf{v}_{|\mathcal{C}}, \mathbf{x}_{|\mathcal{C}}\rangle \ge a\delta$ and therefore $\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}}, \mathcal{E}_{\mathbf{v}})$ must be larger than $1/2$ and is an increasing function of a. Of course, if a is chosen to $+\infty$, then $\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}}, \mathcal{E}_{\mathbf{v}}) = 1$ and therefore $2\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}}) = 1 + \ell^{-1}\left|\cup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$. Thus, if $a = +\infty$, we can use the fact that $\left|\cup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ is integral to compute $\left|\cup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ correctly from an estimate of $\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}})$ that is within an additive error of $1/4\ell$. Of course, we cannot choose $a = +\infty$ since no samples will satisfy the event $\mathcal{E}_{\mathcal{C}}$ in that case. However, we can choose a, $(a > 0)$ carefully so that it is small enough to make $\Pr(\mathcal{E}_{\mathcal{C}})$ reasonably large and at the same time, a is large enough to allow us to correctly compute $\left|\cup_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ from a reasonably good estimate of $\Pr(y = 1 \mid \mathcal{E}_{\mathcal{C}})$. Next, we can again use Lemma 4 and Corollary 1 to arrive at the main theorem for Mixtures of Linear Classifiers:

Theorem 2. *Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumptions 1 and 2. Let $a = \frac{\sqrt{2(R^2+\sigma^2)}}{\delta}\mathsf{erf}^{-1}\Big(1 - \frac{1}{2\ell}\Big)$. Then, there exists an algorithm (see Algorithm 2 and 6) that achieves Exact Support Recovery with probability at least $1-\gamma$ using $O\Big((1-\phi(a))^{-(\log\ell+1)}\ell^2 \log(\gamma^{-1}(n + (\ell k)^{\log\ell+1}))\Big)$ samples generated according to $\mathcal{P}_c$.*

The only comparable result is provided in [Sedghi et al., 2016] who provide parameter estimation guarantees in the MLC setting. However, since it is not evident how to recover the union of support in the sparse MLC setting (unlike MD/MLR); directly applying the result in [Sedghi et al., 2016] will lead to polynomial dependence on n which is undesirable. Moreover, the guarantees in [Sedghi et al., 2016] also require the latent parameter vectors to be linearly independent. In contrast, our sample complexity guarantees for support

recovery scale logarithmically with n and also does not need the latent parameter vectors to be linearly independent (in fact they are not even required to be distinct).

3.3 Mixtures of Linear Regression

Finally, we move on to the mixtures of linear regression or MLR setting. Note that the sample complexity guarantees for MLC (Theorem 2) is also valid in the MLR setting as we can simulate MLR responses by simply taking the sign of the response in the MLR dataset. However, note that the sample complexity presented in Theorem 2 has a poor dependence on R, δ and ℓ. Here we solve the support recovery problem provided the unknown vectors in $\mathcal{V}$ are all binary and demonstrate significantly better sample complexity guarantees under this assumption. The detailed proofs of all results in this section can be found in Section B.3. As usual, we start with a lemma where we characterize the sample complexity of estimating $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ correctly:

Lemma 5. *If the unknown vectors in the set $\mathcal{V}$ are all binary i.e. $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(\ell)} \in \{0,1\}^n$, then, with probability at least $1-\gamma$, for each set $\mathcal{C} \subseteq [n]$, there exists an algorithm (see Algorithm 4) that can compute $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ using $O(\ell^2(k+\sigma^2)^{|C|/2}(\log n)^{2|C|}\log\gamma^{-1})$ i.i.d samples from $\mathcal{P}_r$.*

We provide a high level proof of Lemma 5 here. We consider the random variable $y^{|\mathcal{C}|}\cdot\left(\prod_{i\in\mathcal{C}}\mathbf{x}_i\right)$ where $(\mathbf{x}, y)\sim\mathcal{P}_r$. Clearly, we can write $y = \langle\mathbf{v},\mathbf{x}\rangle + \zeta$ where $\zeta\sim\mathcal{N}(0,\sigma^2)$ and $\mathbf{v}$ is uniformly sampled from the set of unknown vectors $\mathcal{V}$. We can show that

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}_r} y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)$$
$$= \mathbb{E}_{\mathbf{x},\zeta}\ell^{-1}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)\cdot\Big(\langle\mathbf{v},\mathbf{x}\rangle+\zeta\Big)^{|\mathcal{C}|}$$
$$\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big) = \frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbb{E}_{\mathbf{x}}\mathbf{x}_i^2\cdot\mathbf{v}_i\Big) = \frac{\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|}{\ell}.$$

Hence, by using the fact that $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ is integral, we can estimate the quantity correctly from a reasonably good estimate of $\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)$. Again, by an application of Corollary 1, we arrive at the following theorem:

Theorem 3. *Let $\mathcal{V}$ be a set of ℓ unknown binary vectors in $\{0,1\}^n$. Then, with probability at least $1-\gamma$, there exists an algorithm (see Algorithm 4 and 6) that achieves Exact Support Recovery with*

$$O\left(\ell^2(\log^4 n(k+\sigma^2))^{\frac{\log\ell+1}{2}}\log((n+(\ell k)^{\log\ell+1})\gamma^{-1})\right)$$

samples generated according to $\mathcal{P}_r$.

As in mixtures of distributions, it is possible to recover the union of support of the unknown vectors in $\mathcal{V}$ in the MLR setting with a small number of samples (see Lemma 17 in Appendix E). Therefore an alternate approach that can be used for support recovery is to recover the union of support followed by parameter estimation with the features being restricted to the union of the support. Note that if the set of unknown vectors satisfy Assumption 1, then estimating each vector up to an ℓ_2 norm of δ will suffice for support recovery. Hence, by using Lemma 17 followed by Theorem 1 in [Li and Liang, 2018], we arrive at the following result for support recovery:

Theorem 4. *Let $\mathcal{V}$ be a set of ℓ unknown vectors satisfying Assumption 1. Further, assume that any two distinct vectors $\mathbf{v},\mathbf{v}'\in\mathcal{V}$ satisfies $\|\mathbf{v}-\mathbf{v}'\|_2\geq\Delta$. Then, with high probability, there exists an algorithm that achieves Exact Support Recovery with*

$$O\Big(\ell k\log\Big(\frac{\ell k}{\delta}\Big)\mathsf{poly}\Big(\frac{\ell\sigma}{\Delta}\Big)+\Big(\frac{\sigma\ell}{\Delta}\Big)^{O(\ell^2)}+$$
$$\ell^2(R^2+\sigma^2)(\log n)^3/\delta^2\Big)$$

samples generated according to $\mathcal{P}_r$.

If the unknown vectors in $\mathcal{V}$ are restricted to being binary, then the sample complexity in Theorem 4 has a linear dependence on the sparsity but on the other hand, its dependence on σ, ℓ is very poor; note that Theorem 4 uses parameter estimation framework in mixtures of Gaussians ([Moitra and Valiant, 2010]) as a black-box leading to the polynomial in ℓ, σ with a possibly high degree. Moreover, the sample complexity in Theorem 4 has an $\exp(\ell^2)$ dependence on the number of unknown vectors which is undesirable when the number of unknown vectors ℓ is large. In contrast, the sample complexity of Theorem 3 has a polynomial dependence on ℓ, k, σ whose degree can be precisely extracted from the expression. In particular, in the regime where σ or ℓ is large, Theorem 3 provides significant improvements over the guarantees in Theorem 4. Finally, although not mentioned explicitly in Theorem 1 in [Li and Liang, 2018], it can be extracted that the sample complexity is polynomial in γ^{-1} where γ is the failure probability; this leads to a similar dependence on the failure probability in Theorem 4. On the other hand, the sample complexity in Theorem 3 depends logarithmically on γ^{-1}.

We also show additional results on Deduplicated Support Recovery in MLR setting but due to space limitations, we have provided them in Appendix A.3.

Remark 4 (Computational complexity)**.** *All our algorithms described in the MD/MLR/MLC settings are efficient namely their computational complexities are polynomial in the dimension n and sparsity k.*

Acknowledgement: This research is supported in part by NSF awards CCF 2133484 and CCF 1934846.

References

Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Sample-optimal density estimation in nearly-linear time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1278–1289. SIAM, 2017.

Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *Conference on Learning Theory*, 2005.

Ery Arias-Castro and Xiao Pu. A simple approach to sparse clustering. *Computational Statistics & Data Analysis*, 105:217–228, 2017.

Sanjeev Arora and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Symposium on Theory of Computing*, 2001.

Sanjeev Arora, Rong Ge, Ravi Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization—provably. *SIAM Journal on Computing*, 45(4):1582–1611, 2016.

Martin Azizyan, Aarti Singh, and Larry Wasserman. Minimax theory for high-dimensional gaussian mixtures with sparse mean separation. *Advances in Neural Information Processing Systems*, 26:2139–2147, 2013.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science*, 2010.

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.

Siu-On Chan, Ilias Diakonikolas, Rocco A Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 2014.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science*, pages 634–644, 1999.

Richard D De Veaux. Mixtures of linear regressions. *Computational Statistics & Data Analysis*, 8(3):227–245, 1989.

Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1060. ACM, 2018.

Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2008.

David Donoho and Victoria Stodden. When does nonnegative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, pages 1141–1148, 2004.

Susana Faria and Gilda Soromenho. Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):201–225, 2010.

Jon Feldman, Ryan O'Donnell, and Rocco A Servedio. Learning mixtures of product distributions over discrete domains. *SIAM Journal on Computing*, 2008.

Avi Feller, Evan Greif, Nhat Ho, Luke Miratrix, and Natesh Pillai. Weak separation in mixture models and implications for principal stratification. *arXiv preprint arXiv:1602.06595*, 2016.

Venkata Gandikota, Arya Mazumdar, and Soumyabrata Pal. Recovery of sparse linear classifiers from mixture of responses. In *Advances in Neural Information Processing Systems 33: NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Venkata Gandikota, Arya Mazumdar, and Soumyabrata Pal. Support recovery of sparse signals from a mixture of linear measurements. *arXiv preprint arXiv:2106.05951*, 2021.

Moritz Hardt and Eric Price. Tight bounds for learning a mixture of two gaussians. In *Symposium on Theory of Computing*, 2015.

Philippe Heinrich and Jonas Kahn. Strong identifiability and optimal minimax rates for finite mixture estimation. *The Annals of Statistics*, 46(6A):2844–2870, 2018.

Nhat Ho and XuanLong Nguyen. Convergence rates of parameter estimation for some weakly identifiable finite mixtures. *The Annals of Statistics*, 44(6):2726–2755, 2016.

Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Symposium on Theory of Computing*, 2018.

Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Innovations in Theoretical Computer Science*, 2013.

Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaus-

sians. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 553–562. ACM, 2010.

Pravesh K Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046. ACM, 2018.

Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Sample complexity of learning mixture of sparse linear regressions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Algebraic and analytic approaches for parameter learning in mixture models. In *Proc. 31st International Conference on Algorithmic Learning Theory (ALT)*, volume 117, pages 468–489, 2020.

Jeongyeol Kwon and Constantine Caramanis. Global convergence of em algorithm for mixtures of two component linear regression. *arXiv preprint arXiv:1810.05752*, 2018.

Yuanzhi Li and Yingyu Liang. Learning mixtures of linear regressions with nearly optimal complexity. In *Conference On Learning Theory*, pages 1125–1144. PMLR, 2018.

Tudor Manole and Nhat Ho. Uniform convergence rates for maximum likelihood estimation under two-component gaussian mixture models. *arXiv preprint arXiv:2006.00704*, 2020.

Arya Mazumdar and Soumyabrata Pal. Recovery of sparse signals from a mixture of linear samples. In *International Conference on Machine Learning (ICML)*, 2020.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science*, 2010.

F Moosman and D Peel. Finite mixture models. *Wiley*, 3:4, 2000.

Nikita Polyanskii. On learning sparse vectors from mixture of responses. *Advances in Neural Information Processing Systems*, 34, 2021.

Hanie Sedghi, Majid Janzamin, and Anima Anandkumar. Provable tensor methods for learning mixtures of generalized linear models. In *Artificial Intelligence and Statistics*, pages 1223–1231. PMLR, 2016.

Martin Slawski, Matthias Hein, and Pavlo Lutsik. Matrix factorization with binary components. In *Advances in Neural Information Processing Systems*, pages 3210–3218, 2013.

Nicolas Städler, Peter Bühlmann, and Sara Van De Geer. l1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010.

Yuekai Sun, Stratis Ioannidis, and Andrea Montanari. Learning mixtures of linear classifiers. In *ICML*, pages 721–729, 2014.

D Michael Titterington, Adrian FM Smith, and Udi E Makov. *Statistical analysis of finite mixture distributions*. Wiley, 1985.

Nicolas Verzelen and Ery Arias-Castro. Detection and feature selection in sparse mixture models. *The Annals of Statistics*, 45(5):1920–1950, 2017.

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

Xinyang Yi, Constantine Caramanis, and Sujay Sanghavi. Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*, pages 613–621, 2014.

Xinyang Yi, Constantine Caramanis, and Sujay Sanghavi. Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. *arXiv preprint arXiv:1608.05749*, 2016.

Dong Yin, Ramtin Pedarsani, Yudong Chen, and Kannan Ramchandran. Learning mixtures of sparse linear regressions using sparse graph codes. *IEEE Transactions on Information Theory*, 65(3):1430–1451, 2019.

A Results for Deduplicated Support Recovery

A.1 Preliminary Results

In the next few lemmas, we characterize the set $\mathsf{Trimmed}(\mathcal{V})$ and show some useful properties. We start with the following definition:

Definition 4 (t-good)**.** *A binary matrix $\mathbf{A} \in \{0,1\}^{n\times \ell}$ with all distinct columns is called t-good if for every column $\mathbf{A}_i$, there exists a set $S \subset [n]$ of at most t-indices such that $\mathbf{A}_i|_S = \mathbf{1}_t$, and $\mathbf{A}_j|_S \neq \mathbf{1}_t$ for all $j \neq i$.*

Let $\mathcal{V}$ be set of ℓ unknown vectors in $\mathbb{R}^n$, and $\mathbf{A} \in \{0,1\}^{n\times \ell}$ be its support matrix. Let $\mathbf{B}$ be the sub-matrix obtained by deleting duplicate columns of $\mathbf{A}$. The set $\mathcal{V}$ is called t-good if $\mathbf{B}$ is t-good.

Notice that if any set $\mathcal{V}$ is t-good then it must be r-good for all $r \geq t$. In Lemma 6, we show that $\mathsf{Trimmed}(\mathcal{V})$ is $(\ell-1)$-good and in Lemma 8, we provide sufficient conditions for deduplicated support recovery of the set of unknown vectors $\mathcal{V}$.

Lemma 6. *For all sets of ℓ unknown vectors $\mathcal{V}$, $\mathsf{Trimmed}(\mathcal{V})$ must be $(\ell-1)$-good.*

Lemma 7. *If it is known whether $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq s+1$, then there exists an algorithm that achieves Deduplicated support recovery of the set of unknown vectors $\mathcal{V}$ provided $\mathsf{Trimmed}(\mathcal{V})$ is known to be s-good for $s \leq \ell-1$ and $|\mathsf{Trimmed}(\mathcal{V})| \geq 2$.*

Lemma 8. *If it is known whether $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n], |\mathcal{C}| = \ell$, then there exists an algorithm (see Algorithm 7) that achieves Deduplicated support recovery of the set of unknown vectors $\mathcal{V}$.*

Corollary 3. *Let $\mathcal{V}$ be a set of ℓ unknown k-sparse vectors in $\mathbb{R}^n$. Suppose with probability $1-\gamma$, for each $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell$, we can compute if $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ correctly with $\mathsf{T}\log\gamma^{-1}$ samples where T is independent of γ. Then, there exists an algorithm (see Algorithm 8) that can achieve Deduplicated support recovery with probability at least $1-\gamma$ using $O(\mathsf{T}\log(\gamma^{-1}(n+(\ell k)^\ell)))$ samples.*

Remark 5. *Corollary 3 describes the sample complexity for deduplicated support recovery using Lemma 8 which provides the worst-case guarantees as $\mathsf{Trimmed}(\mathcal{V})$ is $(\ell-1)-$good for all sets $\mathcal{V}$. We can also provide improved guarantees for deduplicated support recovery provided $\mathsf{Trimmed}(\mathcal{V})$ is known to be $s-$good by using Lemma 7. However, for the sake of simplicity of exposition, we have only provided results for deduplicated support recovery in mixture models using Corollary 3.*

A.2 Mixtures of Distributions (MD)

Now, we provide results on deduplicated support recovery in the MD setting. Note that from Lemma 8, for partial recovery, we only need to estimate correctly if $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ for ordered sets $\mathcal{C} \subseteq [n]$. Notice that $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ if and only if $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2 > 0$. From our previous arguments, $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2$ can be computed if the quantities $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u} \leq 2\mathbf{1}_{|\mathcal{C}|}$ are pre-computed. The following lemma stems from making this computation robust to the randomness in the dataset:

Lemma 9. *Suppose Assumption 1 is true. Let*

$$\Phi \triangleq \max_{\mathbf{z}\leq 2\mathbf{1}_{|\mathcal{C}|}} \frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z}\leq 2\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $h_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 3) that can compute if $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ correctly for each set $\mathcal{C} \subseteq [n]$ with probability at least $1-\gamma$ using $O(h_{\ell,\mathcal{V}}\log\gamma^{-1})$ samples generated according to $\mathcal{P}_d$.

The subsequent theorem follows from Lemma 9 and Corollary 3. Note that, compared to exact support recovery (Theorem 5) the sample complexity for deduplicated support recovery has significantly improved dependency on δ and furthermore, it is also independent of R.

Theorem 5. *Let $\mathcal{V}$ be a set of unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and*

$$\Phi_m = \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}} \frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h'_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\le 2\mathbf{1}_\ell \\ \mathcal{C}\in\mathcal{F}_\ell}} \frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi_\ell^2}$$

where $h'_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. Accordingly, there exists an algorithm (see Algorithm 3 and 8) that achieves Deduplicated support recovery with probability at least $1-\gamma$ using $O\Big(h'_{\ell,\mathcal{V}}\log(\gamma^{-1}(n+(\ell k)^\ell))\Big)$ samples generated from $\mathcal{P}_d$.

A.3 Mixtures of Linear Regression (MLR)

Our final results are for deduplicated support recovery in the MLR setting under different assumptions. Below, we state Assumption 3 which is a generic condition and if satisfied by the set of unknown vectors $\mathcal{V}$ allows for deduplicated support recovery of $\mathcal{V}$.

Assumption 3. *We assume that there exists positive numbers $\alpha_1, \alpha_2, \ldots, \alpha_\ell > 0$ such that for all sets $\mathcal{C} \subseteq [n], |C| \le \ell$ the following condition is satisfied by the set of ℓ unknown vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(\ell)} \in \mathcal{V}$:*

$$\textit{If there exists } \mathbf{v}\in\mathcal{V} \textit{ such that } \prod_{j\in\mathcal{C}}\mathbf{v}_j \neq 0$$

$$\textit{then } \left|\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{j\in\mathcal{C}}\mathbf{v}_j\Big)\right| \ge \alpha_{|\mathcal{C}|}.$$

Theorem 6. *Suppose the following conditions are satisfied:*

1. *All unknown vectors in $\mathcal{V}$ are bounded within a ball of radius R i.e. $\left|\left|\mathbf{v}^{(i)}\right|\right|_2 \le R$ for all $i \in [\ell]$.*
2. *Assumption 3 is satisfied by the set of unknown vectors $\mathcal{V}$.*

Accordingly, there exists an algorithm (see Algorithms 5 and 8) that achieves Deduplicated support recovery with probability at least $1-\gamma$ using

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1})/\alpha_\ell^2)$$

samples from $\mathcal{P}_r$.

Next, using Theorem 6, we provide deduplicated support recovery guarantees in two cases: 1) The set of unknown vectors in $\mathcal{V}$ satisfies Assumptions 1 and all unknown parameters are non-negative 2) The non-zero entries in the unknown vectors in $\mathcal{V}$ are distributed according to a zero mean Gaussian $\mathcal{N}(0,\nu^2)$.

Corollary 4. *Consider a set of ℓ unknown vectors $\mathcal{V}$ that satisfies Assumptions 1 and furthermore, every non-zero entry in all the unknown vectors is positive ($\mathbf{v}_i \ge 0$ for all $i \in [n], \mathbf{v}\in\mathcal{V}$). In that case, Assumption 3 is satisfied with $\alpha_{|\mathcal{C}|} \ge \delta^{|\mathcal{C}|}$. Accordingly, there exists an algorithm that achieves Deduplicated support recovery with probability at least $1-\gamma$ using*

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n/\delta)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1}))$$

samples from $\mathcal{P}_r$.

Corollary 5. *If all non-zero entries in the set of unknown vectors $\mathcal{V}$ are sampled i.i.d according to $\mathcal{N}(0,\nu^2)$, then with probability $1-\eta$, Assumption 3 is satisfied with $\alpha_{|\mathcal{C}|} \ge \delta_{|\mathcal{C}|}^{|\mathcal{C}|}$ where*

$$\delta_{|\mathcal{C}|} = \Big(\sqrt{\frac{\pi}{8}}\frac{\nu\eta}{\ell|\mathcal{C}|(\ell k)^{|\mathcal{C}|}}\Big).$$

Conditioned on this event, there exists an Algorithm that achieves Deduplicated support recovery with probability at least $1-\gamma$ *using*

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1})/\delta_\ell^2)$$

samples from $\mathcal{P}_r$.

B Detailed Algorithms and Results

B.1 Mixtures of Distributions (MD)

Lemma 10. *For each fixed set* $\mathcal{C}\subseteq[n]$ *and each vector* $\mathbf{t}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$*, we must have*

$$\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}}=\frac{1}{\ell}\sum_{\mathbf{u}\leq\mathbf{t}}\zeta_{\mathbf{t},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big).$$

Proof. We will have

$$\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}}=\frac{1}{\ell}\sum_{j\in[\ell]}\Big(\prod_{i\in\mathcal{C}}q_{\mathbf{t}_{\pi(\mathcal{C},i)}}(\mathbf{v}_i^{(j)})\Big).$$

From the above equations, note that each summand is a product of polynomials in $\mathbf{v}_i^{(j)}$ for a fixed j. Expanding the polynomial and using the fact that $\zeta_{\mathbf{t},\mathbf{u}}=\prod_{i\in\mathcal{C}}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$ is the coefficient of the monomial $\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $j\in[\ell]$, we obtain the proof of the lemma.

□

Lemma 11. *For each fixed set* $\mathcal{C}\subseteq[n]$ *and each vector* $\mathbf{t}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$*, we can compute* $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{t}_{\pi(\mathcal{C},i)}}$ *provided for all* $\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ *satisfying* $\mathbf{u}\leq\mathbf{t}$*, the quantities* $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ *are pre-computed.*

Proof. We will prove this lemma by induction. For the base case, we have from Lemma 10 that $\ell\mathbb{E}\mathbf{x}_i=\beta_{1,2}\sum_{j\in[\ell]}\mathbf{v}_i^{(j)}+\beta_{1,1}$. Hence $\sum_{j\in[\ell]}\mathbf{v}_i^{(j)}$ can be computed from $\mathbb{E}\mathbf{x}_i$ by using the following equation:

$$\sum_{j\in[\ell]}\mathbf{v}_i^{(j)}=\frac{1}{\beta_{1,2}}\Big(\ell\mathbb{E}\mathbf{x}_i-\beta_{1,1}\Big).$$

Now suppose for all vectors $\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u}\leq\mathbf{t}$, the lemma statement is true. Consider another vector $\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ such that there exists an index $j\in|\mathcal{C}|$ for which $\mathbf{z}_j=\mathbf{t}_j+1$ and $\mathbf{z}_i=\mathbf{t}_i$ for all $i\neq j$. From the statement of Lemma 10, we know that

$$\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}=\frac{1}{\ell}\sum_{\mathbf{u}\leq\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big)$$

where $\zeta_{\mathbf{z},\mathbf{u}}=\prod_{i\in\mathcal{C}}\beta_{\mathbf{z}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$. From our induction hypothesis, we have already computed $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u}<\mathbf{z}$ (the set $\{\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}\mid\mathbf{u}<\mathbf{z}\}$ is equivalent to the set $\{\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}\mid\mathbf{u}\leq\mathbf{t}\}$). Since $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ is already pre-computed, we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ as follows:

$$\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}-\sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big)=\zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\Big).$$

This completes the proof of the lemma. □

Lemma 12. *For each fixed set* $\mathcal{C}\subseteq[n]$*, we can compute* $\big|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\big|$ *provided for all* $p\in[\ell]$*, the quantity* $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ *is pre-computed.*

Proof. Let us fix a particular subset $\mathcal{C} \subseteq [n]$. Now, let us define the quantity

$$\mathsf{A}_{\mathcal{C},t} = \sum_{\substack{\mathcal{C}' \subseteq [\ell] \\ |\mathcal{C}'|=t}} \prod_{\substack{i \in \mathcal{C} \\ j \in \mathcal{C}'}} (\mathbf{v}_i^{(j)})^2$$

Notice that $\mathsf{A}_{\mathcal{C},t} > 0$ if and only if there exists a subset $\mathcal{C}' \subseteq [\ell], |\mathcal{C}'| = t$ such that $\mathbf{v}_i^{(j)} \neq 0$ for all $i \in \mathcal{C}, j \in \mathcal{C}'$. Hence, the maximum value of t such that $\mathsf{A}_{\mathcal{C},t} > 0$ is the number of unknown vectors in $\mathcal{V}$ having non-zero value in all the indices in $\mathcal{C}$. In other words, we have that

$$\left| \bigcap_{i \in \mathcal{C}} \mathcal{S}(i) \right| = \max_{t \in [\ell]} t \cdot \mathbf{1}[\mathsf{A}_{\mathcal{C},t} > 0].$$

Let $t^\star$ be the maximum value of t for which $\mathsf{A}_{\mathcal{C},t} > 0$. We will have $\mathsf{A}_{\mathcal{C},t^\star} \geq \delta^{2\ell|\mathcal{C}|}$. It is easy to recognize $\sum_{\mathbf{v} \in \mathcal{V}} \Big(\prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^p$ as the power sum polynomial of degree p in the variables $\{\prod_{i \in \mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v} \in \mathcal{V}}$. On the other hand, $\mathsf{A}_{\mathcal{C},t}$ is the elementary symmetric polynomial of degree t in the variables $\{\prod_{i \in \mathcal{C}} \mathbf{v}_i^2\}_{\mathbf{v} \in \mathcal{V}}$. We can use Newton's identities to state that for all $t \in [\ell]$,

$$t\mathsf{A}_{\mathcal{C},t} = \sum_{p=1}^{t} (-1)^{p+1} \mathsf{A}_{\mathcal{C},t-p} \Big(\sum_{\mathbf{v} \in \mathcal{V}} \Big(\prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^p \Big)$$

using which, we can recursively compute $\mathsf{A}_{\mathcal{C},t}$ for all $t \in [\ell]$ if we were given $\sum_{\mathbf{v} \in \mathcal{V}} \Big(\prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^p$ as input for all $p \in [\ell]$. We can also express $\mathsf{A}_{\mathcal{C},t}$ as a complete exponential Bell polynomial B_t

$$\mathsf{A}_{\mathcal{C},t} = \frac{(-1)^n}{n!} \mathsf{B}_t \Big(- \sum_{\mathbf{v} \in \mathcal{V}} \prod_{i \in \mathcal{C}} \mathbf{v}_i^2, -1! \Big(\sum_{\mathbf{v} \in \mathcal{V}} \prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^2, -2! \Big(\sum_{\mathbf{v} \in \mathcal{V}} \prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^3, \ldots, -(t-1)! \Big(\sum_{\mathbf{v} \in \mathcal{V}} \prod_{i \in \mathcal{C}} \mathbf{v}_i^2 \Big)^t \Big).$$

□

We are now ready to prove Lemma 3.

Lemma (Restatement of Lemma 3)**.** *Suppose Assumption 1 is true. Let*

$$\Phi \triangleq \frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)} \ell!} \left(\max_{\mathbf{z} \leq 2\ell \mathbf{1}_{|\mathcal{C}|}} \frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u} < \mathbf{z}} \sum_{\mathsf{M} \in \mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s}) \in \mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r} \in \mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}} \right)^{-1}$$

$$g_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z} \leq 2\ell \mathbf{1}_{|\mathcal{C}|}} \mathbb{E} \prod_{i \in \mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $g_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 1) that can compute $\left| \bigcap_{i \in \mathcal{C}} \mathcal{S}(i) \right|$ exactly for each set $\mathcal{C} \subseteq [n]$ with probability at least $1 - \gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|}) f_{\ell,\mathcal{V}} \Big)$ samples generated according to $\mathcal{P}_d$.

Proof. Suppose, for every vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \leq 2\ell \mathbf{1}_{|\mathcal{C}|}$, we compute an estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E} \prod_{i \in \mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left| \widehat{U}^{\mathbf{z}} - \mathbb{E} \prod_{i \in \mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}} \right| \leq \Phi_{\mathbf{z}}$ where $\Phi_{\mathbf{z}}$ is going to be determined later. Recall that in Lemma 12, we showed

$$\ell \mathbb{E} \prod_{i \in \mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}} - \sum_{\mathbf{u} < \mathbf{z}} \zeta_{\mathbf{z},\mathbf{u}} \cdot \Big(\sum_{j \in [\ell]} \prod_{i \in \mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}} \Big) = \zeta_{\mathbf{z},\mathbf{z}} \cdot \Big(\sum_{j \in [\ell]} \prod_{i \in \mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}} \Big). \tag{3}$$

Using the computed $\widehat{U}^{\mathbf{z}}$'s , we can compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j \in [\ell]} \prod_{i \in \mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \leq 2\ell \mathbf{1}_{|\mathcal{C}|}$. Let us denote the error in estimation by $\epsilon_{\mathbf{z}}$ i.e. we have $\left| \widehat{V}^{\mathbf{z}} - \sum_{j \in [\ell]} \prod_{i \in \mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}} \right| \leq \epsilon_{\mathbf{z}}$. Now, we prove the following claim.

Claim 1. *We must have*

$$\epsilon_{\mathbf{z}} \leq \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell\Phi_{\mathbf{u}} \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}$$

Proof. We will prove this lemma by induction. Let $\mathbf{e}_i$ be the standard basis vector having a non-zero entry at the i^{th} index and is zero everywhere else. For the base case, we have from Lemma 10 that $\ell\mathbb{E}\mathbf{x}_i = \beta_{1,2}\sum_{j\in[\ell]}\mathbf{v}_i^j + \beta_{1,1}$. Therefore, we must have

$$\begin{aligned}\ell\mathbb{E}\mathbf{x}_i - \ell\widehat{U}^{\mathbf{e}_i} &= \beta_{1,2}\Big(\sum_{j\in[\ell]}\mathbf{v}_i^j - \widehat{U}^{\mathbf{e}_i}\Big)\\ \implies \ell\Phi_{\mathbf{e}_i} &= \beta_{1,2}\epsilon_{\mathbf{e}_i}.\end{aligned}$$

From definition, (recall that $\zeta_{\mathbf{z},\mathbf{u}} = \prod_{i\in\mathcal{C}}\beta_{\mathbf{z}_{\pi(i)},\mathbf{u}_{\pi(i)}+1}$), we have $\zeta_{\mathbf{e}_i,\mathbf{e}_i} = \beta_{1,2}$ which completes the proof of the base case. Now suppose for all vectors $\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u}\leq\mathbf{t}$, the lemma statement is true. Consider another vector $\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ such that there exists an index $j\in|\mathcal{C}|$ for which $\mathbf{z}_j = \mathbf{t}_j + 1$ and $\mathbf{z}_i = \mathbf{t}_i$ for all $i\neq j$. From the statement of Lemma 10, we know that

$$\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(i)}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{u}_{\pi(i)}}\Big) = \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{z}_{\pi(i)}}\Big).$$

Hence, we must have

$$\begin{aligned}&\Big(\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(i)}} - \ell\widehat{U}^{\mathbf{z}}\Big) - \Big(\sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{u}_{\pi(i)}} - \widehat{V}^{\mathbf{u}}\Big)\Big) = \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{z}_{\pi(i)}} - \widehat{V}^{\mathbf{z}}\Big)\\ &\implies \zeta_{\mathbf{z},\mathbf{z}}\epsilon_{\mathbf{z}} \leq \ell\Phi_{\mathbf{z}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\epsilon_{\mathbf{u}}.\end{aligned}$$

Now, by using our induction hypothesis, we must have

$$\begin{aligned}\zeta_{\mathbf{z},\mathbf{z}}\epsilon_{\mathbf{z}} &\leq \ell\Phi_{\mathbf{z}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\left(\frac{\ell\Phi_{\mathbf{u}}}{\zeta_{\mathbf{u},\mathbf{u}}} + \sum_{\mathbf{v}<\mathbf{u}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{u},\mathbf{v})}\frac{\ell\Phi_{\mathbf{v}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)\\ &\implies \epsilon_{\mathbf{z}} \leq \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\left(\frac{\ell\Phi_{\mathbf{u}}}{\zeta_{\mathbf{z},\mathbf{z}}\zeta_{\mathbf{u},\mathbf{u}}} + \sum_{\mathbf{v}<\mathbf{u}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{u},\mathbf{v})}\frac{\ell\Phi_{\mathbf{v}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\zeta_{\mathbf{z},\mathbf{z}}\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)\\ &\implies \epsilon_{\mathbf{z}} \leq \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\Phi_{\mathbf{u}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}.\end{aligned}$$

This completes the proof of the claim. □

Hence, for fixed $\Phi_{\mathbf{z}} = \Phi$ for all $\mathbf{z}\leq 2\ell\mathbf{1}_{|\mathcal{C}|}$, we get

$$\epsilon_{\mathbf{z}} \leq \Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big).$$

For a fixed Φ, let us write ϵ to denote the following quantity:

$$\epsilon \triangleq \max_{\mathbf{z}\leq 2\ell\mathbf{1}_{|\mathcal{C}|}}\Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{Q\in\mathcal{Q}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in Q}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(Q)}\zeta_{\mathbf{r},\mathbf{r}}}\Big)$$

Consider a fixed subset of indices $\mathcal{C} \subseteq [n]$ and a fixed vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$. Using the fact $\max_{\mathbf{v}\in\mathcal{V}, i\in[n]} \mathbf{v}_i^2 \le R^2$, we have that

$$\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p \le R^{2p|\mathcal{C}|} \quad \text{and} \quad \mathsf{A}_{\mathcal{C},t} = \sum_{\substack{\mathcal{C}'\subseteq[\ell]\\|\mathcal{C}'|=t}}\prod_{\substack{i\in\mathcal{C}\\ j\in\mathcal{C}'}}(\mathbf{v}_i^{(j)})^2 \le \binom{\ell}{t}R^{2(t+|\mathcal{C}|)} \le 2^\ell R^{2(t+|\mathcal{C}|)}.$$

We can compute an estimate $\widehat{\mathsf{A}}_{\mathcal{C},t}$ of $\mathsf{A}_{\mathcal{C},t}$ by using $\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}$ in the following set of recursive equations

$$t\widehat{\mathsf{A}}_{\mathcal{C},t} = \sum_{p=1}^{t}(-1)^{p+1}\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}.$$

Claim 2.

$$\left|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\right| \le \epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(t-1)}t! \text{ for all } t\in[\ell].$$

Proof. We will prove this claim by induction. For the base case i.e. $t = 1$, notice that

$$\left|\widehat{\mathsf{A}}_{\mathcal{C},1} - \mathsf{A}_{\mathcal{C},1}\right| \le \left|\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} - \sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\right| \le \epsilon.$$

Now, suppose for all $t \le k$, the following holds true:

$$\left|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\right| \le \epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{t-1}t!.$$

For ease of notation, let us denote $a = 3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})$. In that case, for $t = k+1$, we must have

$$\begin{aligned}
t\left|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\right| &\le \sum_{p\le t}\left|\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}} - \mathsf{A}_{\mathcal{C},t-p}\cdot\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p\right| \\
&\le \left|\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} - \sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^{(k+1)}\right| \\
&\quad + \sum_{p\le t-1}\left|\epsilon a^{t-2}(t-1)!\cdot\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p + \epsilon\cdot\mathsf{A}_{\mathcal{C},t-p} + \epsilon^2 a^{t-2}(t-1)!\right| \\
&\le \epsilon + \sum_{p\le t-1}\left|\epsilon a^{t-2}(t-1)!\ell R^{2\ell|\mathcal{C}|} + \epsilon\cdot 2^\ell R^{2(\ell+|\mathcal{C}|)} + \epsilon^2 a^{t-2}(t-1)!\right| \\
&\le \epsilon + \sum_{p\le t-1}\epsilon a^{t-1}(t-1)! \le \epsilon a^{(t-1)}t!.
\end{aligned}$$

Hence, $\left|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\right| \le \epsilon a^{t-1}t!$ thus proving our claim. □

Hence, to identify $t^\star$ correctly, we must have

$$\begin{aligned}
&\epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell! \le \frac{\delta^{2\ell|\mathcal{C}|}}{2} \\
&\implies \Phi \le \frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell!}\left(\max_{\mathbf{z}\le 2p\mathbf{1}_{|\mathcal{C}|}}\frac{1}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}
\end{aligned}$$

where we inserted the definition of Φ. Therefore, for every vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \le 2\ell \mathbf{1}_{|\mathcal{C}|}$, in order to compute $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left|\widehat{U}^{\mathbf{z}} - \mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \Phi$, the number of samples that is sufficient with probability $1-\gamma$ is going to be

$$O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|})\frac{\max_{\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}} \mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}\Big).$$

□

Theorem (Restatement of Theorem 1)**.** *Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and*

$$\Phi_m = \frac{\delta^{2\ell m}}{2\Big(3\ell\max(R^{2\ell m}, 2^\ell R^{\ell+m})\Big)^{(\ell-1)}\ell!}\left(\max_{\mathbf{z}\le 2\ell\mathbf{1}_m}\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}$$

$$f_{\ell,\mathcal{V}} = \max_{\substack{\mathbf{z}\le 2\ell\mathbf{1}_{\log\ell+1}\\ \mathcal{C}\in\mathcal{F}_{\log\ell+1}}}\frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2_{\log\ell+1}}$$

where $f_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ. Then, there exists an algorithm (see Algorithm 1 and 6) that achieves Exact Support Recovery with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{\log\ell+1}(n+(\ell k)^{\log\ell+1}))f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_d$.

Proof. The proof follows directly from Corollary 1 and Lemma 3. □

Corollary (Restatement of Corollary 2)**.** *Consider the mean estimation problem where $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}_d}[\mathbf{x}_i \mid t=j] = \mathbf{v}_i^{(j)}$. Let $\mathcal{V}$ be a set of $\ell = O(1)$ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1 and $f_{\ell,\mathcal{V}}$ be as defined in Theorem 5. Then, there exists an algorithm (see Algorithm 1 and 6) that with probability at least $1-\gamma$, achieves Exact Support Recovery using $O\Big(\log(n\gamma^{-1})\mathsf{poly}(\delta R^{-1})f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_d$.*

Proof. We can re-scale the samples (dividing them by R) so that Assumption 1 will be satisfied with $\delta' = \delta/R$ and $R' \le 1$. Since ℓ is a constant, $\Phi_{\log\ell} = O(\mathsf{poly}(\delta R^{-1}))$. Therefore, the corollary follows from Theorem 1. □

Algorithm 3 ESTIMATE IF $|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ IN MD SETTING

Require: Samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)} \sim \mathcal{P}_d$. Set $\mathcal{C} \subseteq [n]$.

1: For every $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$, compute estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ using Algorithm 9 on the set of samples $\{(\mathbf{x}_i^j)^{\mathbf{z}_{\pi(\mathcal{C},i)}}\}_{j=1}^m$.

2: For every $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$, compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi\mathcal{C},i}}$ recursively using the following equation:

$$\ell\widehat{U}^{\mathbf{z}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\widehat{V}^{\mathbf{u}} = \zeta_{\mathbf{z},\mathbf{z}}\cdot\widehat{V}^{\mathbf{z}}.$$

3: If $\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} \ge \delta^{2|\mathcal{C}|}/2$, return True and otherwise return False.

Lemma (Restatement of Lemma 9)**.** *Suppose Assumption 1 is true. Let*

$$\Phi \triangleq \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}}\frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $h_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 3) that can compute if $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ correctly for each set $\mathcal{C}\subseteq[n]$ with probability at least $1-\gamma$ using $O(h_{\ell,\mathcal{V}}\log\gamma^{-1})$ samples generated according to $\mathcal{P}_d$.

Proof. For a fixed ordered set $\mathcal{C} \subseteq [n]$, consider the statistic $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2$. If $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2 > 0$, then $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ and otherwise, if $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2 = 0$, then $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$. Hence it suffices to estimate correctly if $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2 > 0$ or not. From Lemma 11, we know that for each set $\mathcal{C} \subseteq [n]$, we can compute $\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^2$ provided for all $\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} \le 2\mathbf{1}_{|\mathcal{C}|}$, the quantity $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ is pre-computed.

Suppose, for every vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$, we compute an estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left|\widehat{U}^{\mathbf{z}} - \mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \Phi$ where Φ is going to be determined later. Using the computed $\widehat{U}^{\mathbf{z}}$'s , we can compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$. As before, let us denote the error in estimation by $\epsilon_{\mathbf{z}}$ i.e. we have $\left|\widehat{V}^{\mathbf{z}} - \sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \epsilon_{\mathbf{z}}$. Note that we showed in Lemma 12 that for fixed Φ, we get for all $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$,

$$\epsilon_{\mathbf{z}} \le \Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\Big).$$

Note that the minimum value of $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2$ is at least $\delta^{2|\mathcal{C}|}$ and therefore, it suffices $\epsilon_{\mathbf{z}}$ to be less than $\delta^{2|\mathcal{C}|}/2$. Hence, it is sufficient if

$$\Phi \le \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}} \frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}.$$

Now, we use Lemma 18 to complete the proof of the lemma (similar to Lemma 12)

□

Theorem (Restatement of Theorem 5)**.** *Let $\mathcal{V}$ be a set of unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and*

$$\Phi_m = \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}} \frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell \prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h'_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\le 2\mathbf{1}_\ell \\ \mathcal{C}\in\mathcal{F}_\ell}} \frac{\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi_\ell^2}$$

where $h'_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. Accordingly, there exists an algorithm (see Algorithm 3 and 8) that achieves Deduplicated support recovery with probability at least $1-\gamma$ using $O\Big(h'_{\ell,\mathcal{V}} \log(\gamma^{-1}(n + (\ell k)^\ell))\Big)$ samples generated from $\mathcal{P}_d$.

Proof. The proof follows from Lemma 9 and Corollary 3. □

B.2 Mixtures of Linear Classifiers (MLC)

Recall that in this section, we solve the sparse recovery problem when the observed samples are generated according to $\mathcal{P}_c$ under Assumption 2.

Lemma (Restatement of Lemma 4)**.** *Suppose Assumptions 1 and 2 are true. Let $a = \frac{\sqrt{2(R^2+\sigma^2)}}{\delta}\mathsf{erf}^{-1}\Big(1-\frac{1}{2\ell}\Big)$. There exists an algorithm (see Algorithm 2) that can compute $\Big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\Big|$ for each set $\mathcal{C} \subseteq [n]$ with probability at least $1-\gamma$ using $O\Big((1-\phi(a))^{-|\mathcal{C}|}\ell^2 \log\gamma^{-1}\Big)$ i.i.d samples from $\mathcal{P}_c$.*

Proof. Without loss of generality, let us assume that all unknown vectors in $\mathcal{V}$ have positive non-zero entries. for each fixed set $\mathcal{C} \subseteq [n]$, we will condition on event $\mathcal{E}_{\mathcal{C}}$ defined as follows: for all $j \in \mathcal{C}$, the data-point $\mathbf{x}$ satisfies $\mathbf{x}_j > a$ for some suitably chosen $a > 0$. Recall that the minimum magnitude of any non-zero entry in an unknown vector in $\mathcal{V}$ is at least δ. Further condition on the event $\mathcal{E}_{\mathbf{v}}$ which is true when a particular unknown vector $\mathbf{v}$ is being sampled from $\mathcal{V}$. In that case, we show the following claim:

Claim 3.

$$\Pr(y=1 \mid \mathcal{E}_{\mathbf{v}}, \mathcal{E}_{\mathcal{C}}) = \frac{1}{2} \text{ if } \mathbf{v}_{|\mathcal{C}} = \mathbf{0}$$
$$1 \geq \Pr(y=1 \mid \mathcal{E}_{\mathbf{v}}, \mathcal{E}_{\mathcal{C}}) \geq \frac{1}{2} + \frac{1}{2} \cdot \mathsf{erf}\Big(\frac{a\delta}{\sqrt{2(R^2+\sigma^2)}}\Big) \text{ if } \mathbf{v}_{|\mathcal{C}} \neq \mathbf{0}.$$

Proof. In order to see the above equation, note that if $\mathbf{v}_{|\mathcal{C}} = \mathbf{0}$, then $\langle \mathbf{v}, \mathbf{x} \rangle + z \sim \mathcal{N}(0, ||\mathbf{v}||_2^2 + \sigma^2)$ or in other words, conditioning on the event $\mathcal{E}_{\mathcal{C}}$ has no effect on the distribution of y. On the other hand, if $\mathbf{v}_{|\mathcal{C}} \neq 0$, conditioning on the event $\mathcal{E}_{\mathcal{C}}$ modifies the distribution of y. Consider an index $j \in \mathsf{supp}(\mathbf{v}) \cap \mathcal{C}$. Since $\mathbf{v}_j \mathbf{x}_j \geq a\delta$, we must have $\langle \mathbf{v}_{|\mathcal{C}}, \mathbf{x}_{|\mathcal{C}} \rangle \geq a\delta$ using Assumption 2. Therefore, the probability that $y = 1$ must be at least $\Pr(\langle \mathbf{v}_{|[n]\setminus\mathcal{C}}, \mathbf{x}_{|[n]\setminus\mathcal{C}} \rangle + z \geq -a\delta)$. Using the fact that $\langle \mathbf{v}_{|[n]\setminus\mathcal{C}}, \mathbf{x}_{|[n]\setminus\mathcal{C}} \rangle + z \sim \mathcal{N}(0, \nu^2 + \sigma^2)$ (where $\nu \leq R$) and the property of error function ($\Pr_{u\sim\mathcal{N}(0,\sigma^2)}(|u| \leq a) = \mathsf{erf}(a/\sqrt{2}\sigma)$), we prove the claim. □

Hence we must have

$$\frac{1}{2} + \frac{|\cup_{i\in\mathcal{C}} \mathcal{S}(i)|}{2\ell} \geq \Pr(y=1 \mid \mathcal{E}_{\mathcal{C}}) \geq \frac{1}{2} + \frac{|\cup_{i\in\mathcal{C}} \mathcal{S}(i)|}{2\ell} \mathsf{erf}\Big(\frac{a\delta}{\sqrt{2(R^2+\sigma^2)}}\Big)$$

We choose a such that $\mathsf{erf}\Big(\frac{a\delta}{\sqrt{2(R^2+\sigma^2)}}\Big) \geq 1 - \frac{1}{2\ell}$ in which case, we must have

$$\frac{1}{2}\Big(1 + \frac{1}{\ell}\Big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\Big|\Big) - \frac{1}{4\ell^2} \cdot \Big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\Big| \leq \Pr(y=1 \mid \mathcal{E}_{\mathcal{C}}) \leq \frac{1}{2}\Big(1 + \frac{1}{\ell}\Big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\Big|\Big)$$

Clearly, for each value of $\big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\big| \in \{0, 1, \ldots, \ell\}$, the interval in which $\Pr(y=1 \mid \mathcal{E}_{\mathcal{C}})$ lies are disjoint and each interval is separated by at least $1/4\ell$. Hence, if we are able to estimate $\Pr(y=1 \mid \mathcal{E}_{\mathcal{C}})$ up to an additive factor of $1/8\ell$, then we can uniquely (and correctly) decode the value of $\big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\big|$. By using Chernoff bound, with $O(\ell^2 \log \gamma^{-1})$ samples satisfying the event $\mathcal{E}_{\mathcal{C}}$, we can estimate $\Pr(y=1 \mid \mathcal{E}_{\mathcal{C}})$ (See Step 2 in Algorithm 2 for the estimator) with probability at least $1 - \gamma/2$. From our previous analysis, we chose $a = \frac{\sqrt{2(R^2+\sigma^2)}}{\delta} \mathsf{erf}^{-1}\Big(1 - \frac{1}{2\ell}\Big)$. The probability that for a sample $(\mathbf{x}, y) \sim \mathcal{P}_c$, the event $\mathcal{E}_{\mathcal{C}}$ is true is exactly $O\Big((1-\phi(a))^{|\mathcal{C}|}\Big)$. Therefore, with $(1-\phi(a))^{-|\mathcal{C}|}\ell^2 \log \gamma^{-1}\Big)$ samples, we will have $O(\ell^2 \log \gamma^{-1})$ samples satisfying the event $\mathcal{E}_{\mathcal{C}}$ with probability at least $1 - \gamma/2$. Hence, this allows us to recover $\big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\big|$ with probability at least $1 - \gamma$. □

Theorem (Restatement of Theorem 2)**.** *Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumptions 1 and 2. Let $a = \frac{\sqrt{2(R^2+\sigma^2)}}{\delta} \mathsf{erf}^{-1}\Big(1 - \frac{1}{2\ell}\Big)$. Then, there exists an algorithm (see Algorithm 2 and 6) that achieves Exact Support Recovery with probability at least $1 - \gamma$ using $O\Big((1-\phi(a))^{-(\log \ell + 1)} \ell^2 \log(\gamma^{-1}(n + (\ell k)^{\log \ell + 1}))\Big)$ samples generated according to $\mathcal{P}_c$.*

Proof. The proof follows directly from Lemma 4 and Corollary 1. □

B.3 Mixtures of Linear Regression (MLR)

B.3.1 Unknown binary Vectors

Lemma (Restatement of Lemma 5)**.** *If the unknown vectors in the set $\mathcal{V}$ are all binary i.e. $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(\ell)} \in \{0,1\}^n$, then, with probability at least $1 - \gamma$, for each set $\mathcal{C} \subseteq [n]$, there exists an algorithm (see Algorithm 4) that can compute $\big|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\big|$ using $O(\ell^2(k+\sigma^2)^{|C|/2}(\log n)^{2|C|} \log \gamma^{-1})$ i.i.d samples from $\mathcal{P}_r$.*

Algorithm 4 RECOVER $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ IN MLR SETTING

Require: Samples $(\mathbf{x}^{(1)},y^{(1)}),(\mathbf{x}^{(2)},y^{(2)}),\ldots,(\mathbf{x}^{(m)},y^{(m)})\sim\mathcal{P}_r$. Set $\mathcal{C}\subseteq[n]$.

1: Return $\mathsf{round}\Big(\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)\Big)$

Proof. Consider the random variable $y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)$ where $(\mathbf{x},y)\sim\mathcal{P}_r$. Clearly, we can write $y=\langle\mathbf{v},\mathbf{x}\rangle+\zeta$ where $\zeta\sim\mathcal{N}(0,\sigma^2)$ and $\mathbf{v}$ is uniformly sampled from the set of unknown vectors $\mathcal{V}$. Therefore, we must have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}_r}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)=\mathbb{E}_{\mathbf{x},\zeta}\ell^{-1}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)\cdot\Big(\langle\mathbf{v},\mathbf{x}\rangle+\zeta\Big)^{|\mathcal{C}|}$$

$$\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)=\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbb{E}_{\mathbf{x}}\mathbf{x}_i^2\cdot\mathbf{v}_i\Big)=\frac{\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|}{\ell}.$$

This is because in the expansion of $(\langle\mathbf{v},\mathbf{x}\rangle+\zeta)^{|\mathcal{C}|}$, the only monomial containing $\mathbf{x}_i$ for all $i\in\mathcal{C}$ is $\prod_{i\in\mathcal{C}}\mathbf{v}_i\mathbf{x}_i$. For any other monomial, the product with $\prod_{i\in\mathcal{C}}\mathbf{x}_i$ will contain some $\mathbf{x}_j, j\in\mathcal{C}$ such that the degree of $\mathbf{x}_j$ in the monomial is 1; the expectation of this monomial goes to zero as all the $\mathbf{x}_i$'s are independent. Since $\mathbb{E}\mathbf{x}_i^2=1$ for all $i\in[n]$ and $\prod_{i\in\mathcal{C}}\mathbf{v}_i$ is $\mathtt{1}$ iff $\mathbf{v}_i=1$ for all $i\in\mathcal{C}$ (and $\mathtt{0}$ otherwise), we obtain the desired equations. We estimate $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ by computing the following sample average

$$\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big).$$

From definition for $(\mathbf{x},y)\sim\mathcal{P}_r$, we must have $y\sim\ell^{-1}\sum_{\mathbf{v}\in\mathcal{V}}\mathcal{N}(0,||\mathbf{v}||_0^2+\sigma^2)$. Therefore, we must have $\mathbb{E}y^2\le k+\sigma^2$ since $\mathbf{v}\in\{0,1\}^n, ||\mathbf{v}||_0\le k$ for all $\mathbf{v}\in\mathcal{V}$. By using Gaussian concentration inequalities, we must have $\Pr(|y|>t)\le\exp(-t^2/2(k+\sigma^2))$. Therefore, with probability $1-n^{-10}$, we have $|y|<20\sqrt{k+\sigma^2}\log n$. Similarly, with probability $1-n^{-10}$, $|\mathbf{x}_i|$ is bounded from above by $20\log n$. We take a union bound over all $|C|+1$ random variables and all m samples to infer that $\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)$ is bounded within a ball of radius $O((k+\sigma^2)^{|C|/2}(\log n)^{2|C|})$ with probability at least $1-O(m|C|n^{-10})$. Subsequently, we use Hoeffding's inequality (see Lemma 14) to say that

$$\Pr\Big(\left|\frac{1}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)-\frac{\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|}{\ell}\right|\ge\frac{1}{2\ell}\Big)\le\exp\Big(-\Omega\Big(\frac{m}{\ell^2(k+\sigma^2)^{|C|/2}(\log n)^{2|C|}}\Big)\Big).$$

Hence, with $m=O(\ell^2(k+\sigma^2)^{|C|/2}(\log n)^{2|C|}\log\gamma^{-1})$ samples, we can estimate $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ exactly with probability at least $1-\gamma$. □

We can now show the following result:

Theorem (Restatement of Theorem 3)**.** *Let $\mathcal{V}$ be a set of ℓ unknown binary vectors in $\{0,1\}^n$. Then, with probability at least $1-\gamma$, there exists an algorithm (see Algorithms 4 and 8) that achieves Exact Support Recovery with*

$$O\Big(\ell^2(k+\sigma^2)^{(\log\ell+1)/2}(\log n)^{2(\log\ell+1)}\log((n+(\ell k)^{\log\ell})\gamma^{-1})\Big)$$

samples generated according to $\mathcal{P}_r$.

Proof. The proof follows directly from Lemma 5 and Corollary 1. □

B.3.2 Separability Assumption on Parameters

Below, we show that if Assumption 3 is satisfied, then we can recover the support of the unknown vectors. We start with the following theorem:

Algorithm 5 ESTIMATE IF $|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)>0|$ IN MLR SETTING

Require: Samples $(\mathbf{x}^{(1)},y^{(1)}),(\mathbf{x}^{(2)},y^{(2)}),\ldots,(\mathbf{x}^{(m)},y^{(m)})\sim\mathcal{P}_r$. Set $\mathcal{C}\subseteq[n]$.

1: If $\frac{2\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)\geq\alpha_{|\mathcal{C}|}$, return True else return False.

Theorem (Restatement of Theorem 6)**.** *Suppose the following conditions are satisfied:*

1. *All unknown vectors in $\mathcal{V}$ are bounded within a ball of radius R i.e. $\big|\big|\mathbf{v}^{(i)}\big|\big|_2\leq R$ for all $i\in[\ell]$.*
2. *Assumption 3 is satisfied by the set of unknown vectors $\mathcal{V}$.*

Accordingly, with probability at least $1-\gamma$, there exists an algorithm (see Algorithms 5 and 8) that achieves Deduplicated support recovery using

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1})/\alpha_\ell^2)$$

samples from $\mathcal{P}_r$.

Proof. Again, we look at the random variable $y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)$ where $(\mathbf{x},y)\sim\mathcal{P}_r$ and therefore, we must have

$$y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)=\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)\cdot\Big(\langle\mathbf{v},\mathbf{x}\rangle+\zeta\Big)^{|\mathcal{C}|}$$
$$\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)=\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbb{E}\mathbf{x}_i^2\cdot\mathbf{v}_i\Big)=\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{j\in\mathcal{C}}\mathbf{v}_j\Big).$$

Notice that $\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)=0$ if $\big|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\big|=0$ and $\Big|\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)\Big|\geq\alpha_{|\mathcal{C}|}/\ell$ otherwise (by using Assumption 3). We estimate $\mathbb{E}y^{|\mathcal{C}|}\cdot\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i\Big)$ by computing the following sample average

$$\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big).$$

From the definition of $\mathcal{P}_r$, we must have $y\sim\ell^{-1}\sum_{\mathbf{v}\in\mathcal{V}}\mathcal{N}(0,||\mathbf{v}||_2^2+\sigma^2)$. Therefore, we have that $\mathbb{E}y^2\leq R^2+\sigma^2$ since $||\mathbf{v}||_2\leq R$ for all $\mathbf{v}\in\mathcal{V}$ from the statement of the Theorem. By using Gaussian concentration inequalities, we must have $\Pr(|y|>t)\leq\exp(-t^2/2(R^2+\sigma^2))$. Therefore, with probability $1-n^{-10}$, we have $|y|<20\sqrt{R^2+\sigma^2}\log n$. Similarly, with probability $1-n^{-10}$, $|\mathbf{x}_i|$ is bounded from above by $20\log n$. We take a union bound over all $|C|+1$ random variables and all m samples to infer that $\Big(y^{(j)}\Big)^{|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)$ is bounded within a ball of radius $O((R^2+\sigma^2)^{|C|/2}(\log n)^{2|C|})$ with probability at least $1-O(m|C|n^{-10})$. Subsequently, we use Hoeffding's inequality (see Lemma 14) to say that

$$\Pr\Big(\Big|\frac{1}{m}\cdot\sum_{j=1}^{m}y^{(j)\,|\mathcal{C}|}\Big(\prod_{i\in\mathcal{C}}\mathbf{x}_i^{(j)}\Big)-\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{j\in\mathcal{C}}\mathbf{v}_j\Big)\Big|\geq\frac{\alpha_{|\mathcal{C}|}}{2\ell}\Big)\leq\exp\Big(-\Omega\Big(\frac{m\alpha_{|\mathcal{C}|}^2}{\ell^2(R^2+\sigma^2)^{|C|/2}(\log n)^{2|C|}}\Big)\Big).$$

Hence, with $m=O(\ell^2(R^2+\sigma^2)^{|C|/2}(\log n)^{2|C|}\log\gamma^{-1}/\alpha_{|\mathcal{C}|}^2)$ samples, we can estimate if $\big|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\big|>0$ or not correctly with probability at least $1-\gamma$. The proof now follows directly from using Corollary 3. □

Corollary (Restatement of Corollary 4)**.** *Consider a set of ℓ unknown vectors $\mathcal{V}$ that satisfies Assumptions 1 and furthermore, every non-zero entry in all the unknown vectors is positive ($\mathbf{v}_i\geq0$ for all $i\in[n],\mathbf{v}\in\mathcal{V}$). In that case, Assumption 3 is satisfied with $\alpha_{|\mathcal{C}|}\geq\delta^{|\mathcal{C}|}$. Accordingly, there exists an algorithm that achieves Deduplicated support recovery with probability at least $1-\gamma$ using*

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n/\delta)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1}))$$

samples from $\mathcal{P}_r$.

Proof. Note that when all the unknown vectors in set $\mathcal{V}$ are non-negative, it must happen that for each set $\mathcal{C} \subseteq [n]$, $\left|\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)\right| \geq \alpha_{|\mathcal{C}|}$ is a sum of positive terms (provided is it non-zero) each of which is at least $\delta^{|\mathcal{C}|}$. Therefore, it must happen that $\alpha_{|\mathcal{C}|} \geq \delta^{|\mathcal{C}|}$. The above argument also holds true when all the unknown vectors in set $\mathcal{V}$ are non-positive. We can directly use Theorem 6 to arrive at the statement of the corollary. □

Corollary (Restatement of Corollary 5)**.** *If all non-zero entries in the set of unknown vectors $\mathcal{V}$ are sampled i.i.d according to $\mathcal{N}(0,\nu^2)$, then with probability $1-\eta$, Assumption 3 is satisfied with $\alpha_{|\mathcal{C}|} \geq \delta_{|\mathcal{C}|}^{|\mathcal{C}|}$ where*

$$\delta_{|\mathcal{C}|} = \left(\sqrt{\frac{\pi}{8}}\frac{\nu\eta}{\ell|\mathcal{C}|(\ell k)^{|\mathcal{C}|}}\right).$$

Conditioned on this event, there exists an Algorithm that achieves Deduplicated support recovery with probability at least $1-\gamma$ using

$$O(\ell^2(R^2+\sigma^2)^{\ell/2}(\log n)^{2\ell}\log((n+(\ell k)^\ell)\gamma^{-1})/\delta_\ell^2)$$

samples from $\mathcal{P}_r$.

Proof. For a fixed set $\mathcal{C} \subseteq [n]$, consider the random variable $\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)$. For each vector $\mathbf{v} \in \mathcal{V}$ such that $\prod_{j\in\mathcal{C}} \mathbf{v}_j \neq 0$, we denote the minimum index $i \in \mathcal{C}$ such that $\mathbf{v}_i \neq 0$ by $i^\star$ and therefore $\mathbf{v}_{i^\star} \sim \mathcal{N}(0,\nu^2)$. Now, for each $\mathbf{v} \in \mathcal{V}$, let us condition on a fixed realization of non-zero indices of $\mathbf{v}$ in $\mathcal{C}$ other than $i^\star$. Let $\mathcal{V}_\mathcal{C} \subseteq \mathcal{V}$ be the set of vectors such that $\prod_{j\in\mathcal{C}} \mathbf{v}_j \neq 0$. Therefore, we must have

$$\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right) \mid \mathbf{v}_j \text{ for all } j \in \mathcal{C}\setminus i^\star, \mathbf{v} \in \mathcal{V}_\mathcal{C} \sim \mathcal{N}\Big(0, \nu^2 \sum_{\mathbf{v}\in\mathcal{V}_\mathcal{C}} \prod_{j\in\mathcal{C}\setminus i^\star} \mathbf{v}_j^2\Big). \tag{4}$$

Therefore, conditioned on $\mathbf{v}_j$ for all $j \in \mathcal{C}\setminus i^\star, \mathbf{v} \in \mathcal{V}_\mathcal{C}$, by standard Gaussian anti-concentration inequality (see Lemma 16), we must have with probability $1-\rho$,

$$\left|\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)\right| \geq \sqrt{\frac{\pi}{8}}\rho\nu\sqrt{\sum_{\mathbf{v}\in\mathcal{V}_\mathcal{C}} \prod_{j\in\mathcal{C}\setminus i^\star} \mathbf{v}_j^2}. \tag{5}$$

for each vector $\mathbf{v} \in \mathcal{V}_\mathcal{C}$, we must have with probability at least $1-(|\mathcal{C}|-1)\rho$ that

$$\left|\prod_{j\in\mathcal{C}\setminus i^\star} \mathbf{v}_j\right| \geq \left(\sqrt{\frac{\pi}{8}}\rho\nu\right)^{(|\mathcal{C}|-1)}. \tag{6}$$

By taking a union bound, we can conclude that with probability at least $1-\ell\rho$, we must have

$$\left|\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)\right| \geq \left(\sqrt{\frac{\pi}{8}}\rho\nu\right)^{|\mathcal{C}|}$$

since there exists at least one vector $\mathbf{v} \in \mathcal{V}_\mathcal{C}$ such that equation 6 holds true for $\mathbf{v}$. Next, after taking another union bound over all subsets of size $|\mathcal{C}|$ restricted to the union of support (at most $(\ell k)^{|\mathcal{C}|}$ of them), we have that with probability $1-|\mathcal{C}|(\ell k)^{|\mathcal{C}|}\rho$,

$$\left|\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)\right| \geq \left(\sqrt{\frac{\pi}{8}}\rho\nu\right)^{|\mathcal{C}|}.$$

Subsequently, we have with probability at least $1-\eta/\ell$

$$\left|\sum_{\mathbf{v}\in\mathcal{V}} \left(\prod_{j\in\mathcal{C}} \mathbf{v}_j\right)\right| \geq \left(\sqrt{\frac{\pi}{8}}\frac{\nu\eta}{\ell|\mathcal{C}|(\ell k)^{|\mathcal{C}|}}\right)^{|\mathcal{C}|}.$$

After taking a final union bound over $|\mathcal{C}| \leq \ell$ and subsequently using Theorem 6, we complete the proof of the corollary. □

C Discussion on Our Results and Other Related Works

Mixtures of Distributions: Our technique of learning the supports of the latent parameter vectors in mixture of simple distributions is based on the *method of moments* [Hsu and Kakade, 2013, Hardt and Price, 2015]. This method works in general, as long as moments of the distribution of each coordinate can be described as a polynomial in the component parameters. The authors in [Belkin and Sinha, 2010] showed (see Table 2 in [Belkin and Sinha, 2010]) that most common distributions, including Gaussian, Uniform, Poisson, and Laplace distributions, satisfy this assumption. Our results in this part that include sample complexity guarantees for both exact support recovery (see Theorem 1) and Deduplicated support recovery (see Theorem 5) are not only applicable to many canonical distributions but also makes progress towards quantifying the sufficient number of moments in the general problem defined in Sec. 1.2.

An alternate approach to the support recovery problem is to first recover the *union* of supports of the unknown parameters and then apply known parameter estimation guarantees to identify the support of each of the unknown vectors after reducing the dimension of the problem. Note that this approach crucially requires parameter estimation results for the corresponding family of mixtures which may be unavailable. To the best of our knowledge, most constructive sample complexity guarantees for parameter estimation in mixture models without separability assumptions correspond to mixtures of Gaussians [Kalai et al., 2010, Belkin and Sinha, 2010, Moitra and Valiant, 2010, Hardt and Price, 2015, Feller et al., 2016, Ho and Nguyen, 2016, Manole and Ho, 2020, Heinrich and Kahn, 2018]. Moreover, most known results correspond to mixtures of Gaussians with two components. The only known results for parameter estimation in mixtures of Gaussians with more than 2 components is [Moitra and Valiant, 2010] but as we describe later, using the alternate approach with the guarantees in [Moitra and Valiant, 2010] results in a polynomial dependence on the sparsity. On the contrary, our sample complexity guarantees scales logarithmically with the sparsity or dimension (for constant ℓ), see Corollary 2, which is a significant improvement over the alternate approach.

For other than Gaussian distributions, [Belkin and Sinha, 2010, Krishnamurthy et al., 2020] studied parameter estimation under the same moment-based assumption that we use. However, [Belkin and Sinha, 2010] uses non-constructive arguments from algebraic geometry because of which, their results did not include bounds on the sufficient number of moments for learning the parameters in a mixture model. In [Krishnamurthy et al., 2020], the authors resolve this question to a certain extent for these aforementioned families of mixture models as they quantify the sufficient number of moments for parameter estimation under the restrictive assumption that the latent parameters lie on an integer lattice. Therefore, our results for these distributions form the first guarantees for support recovery.

Mixtures of Linear Regression For the support recovery problem in the sparse mixtures of linear regressions (MLR) setting, we provide a suite of results under different assumptions. In particular, we study the exact support recovery problem when the unknown sparse parameters are binary (see Theorem 3) and the deduplicated support recovery problem when 1) the unknown sparse parameters have non-negative values (see Corollary 4), or 2) the unknown sparse parameters are distributed according to a Gaussian (see Corollary 5). As in the MD setting, an alternate approach for the support recovery problem is to first find the union of support of the unknown parameters and then apply existing parameter estimation guarantees to recovery the support of each of the unknown linear functions. The state of the art guarantees in MLR for parameter estimation is given by [Li and Liang, 2018] providing a sample complexity guarantee which is linear in the dimension (linear in sparsity when restricted to the union of support). Our results for support recovery are polynomial in sparsity and are therefore worse than the parameter estimation guarantees of [Li and Liang, 2018] applied to our sparse setting (see Theorem 4) when the sparsity is large. On the other hand, the sample complexity guarantees of [Li and Liang, 2018] scales exponentially with ℓ^2 and polynomially with the inverse of the failure probability. In contrast, our sample complexity guarantees are polynomial in ℓ and logarithmic in the inverse of the failure probability.

Mixtures of Linear Classifiers Unlike the MLR and MD setting, mixture of linear classifiers (MLC) is far less studied. It is understandably more difficult to analyze than MLR since only the sign of the linear function of the covariates is retained. We study the exact support recovery problem in sparse MLC (see Theorem 2) under the setting that all the parameters of the unknown vectors are either nonnegative or they are all nonpositive. Although this assumption might seem restrictive, note that theoretical work in the MLC setting is extremely limited. To the best of our knowledge, there are only two relevant papers [Sun et al., 2014, Sedghi et al., 2016]

that have studied this problem. In [Sun et al., 2014], the authors do not make any assumptions on sparsity and provide an algorithm for recovering the subspace in which the parameter vectors corresponding to the unknown linear functions lie. In contrast, support recovery is a different objective and hence is incomparable to the subspace recovery guarantees. The second work, [Sedghi et al., 2016] uses tensor decomposition based methods to provide sample complexity guarantees for learning the parameter vectors; but their sample complexity is inversely proportional to the square of the minimum eigenvalue of the matrix comprising the unknown parameter vectors as columns. This is an unwanted dependence as it implies that if the parameter vectors are linearly dependent, then the algorithm will require infinite samples to recover the parameter vectors. On the other hand, our support recovery guarantees do not have any such assumption on the parameters. Moreover, unlike the MD setting, it is not evident in MLC how to recover the union of support of the unknown sparse vectors. Hence the sample complexity obtained by applying the results in [Sedghi et al., 2016] directly will lead to a polynomial dependence on the dimension of the latent space which is undesirable (ideally, we require a logarithmic dependence on the latent space dimension). Our results exhibit such dependence on the dimension and also does not assume linear independence of the parameter vectors. We believe this to be an important progress towards further understanding of theoretical properties of mixtures where the response is a mixture of nonlinear functions of the covariates.

D Missing Proofs and Algorithms from Sections 2 and A.1

Proof of Lemma 2 when $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ is provided. Suppose we are given $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all sets $\mathcal{C}\subseteq[n]$ satisfying $|\mathcal{C}|\leq s$. Notice that the set $\cap_{i\in\mathcal{C}}\mathcal{S}(i)$ is equivalent to the set $\mathsf{occ}(C,\mathbf{1}_{|C|})$ or the number of unknown vectors in $\mathcal{V}$ whose restriction to the indices in $\mathcal{C}$ is the all one vector and in particular, $\mathsf{occ}((i),1)=\mathcal{S}(i)$. Note that for each family of t sets $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_t$, we must have

$$\left|\bigcup_{i=1}^{t}\mathcal{A}_i\right|=\sum_{u=1}^{t}(-1)^{u+1}\sum_{1\leq i_1<i_2<\cdots<i_u\leq t}\left|\bigcap_{b=1}^{u}\mathcal{A}_{i_b}\right|.$$

We now show using induction on s that the quantities $\left\{\left|\bigcup_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\leq s\right\}$ are sufficient to compute $|\mathsf{occ}(C,\mathbf{a})|$ for all subsets C of indices of size at most s, and any binary vector $\mathbf{a}\in\{0,1\}^{\leq s}$.

Base case ($t=1$):

The base case follows since we can infer $|\mathsf{occ}((i),0)|=\ell-|\mathsf{occ}((i),1)|$ from $|\mathsf{occ}((i),1)|$ for all $i\in[n]$.

Inductive Step: Let us assume that the statement is true for $r<s$ i.e., we can compute $|\mathsf{occ}(\mathcal{C},\mathbf{a})|$ for all subsets $\mathcal{C}$ satisfying $|\mathcal{C}|\leq r$ and any binary vector $\mathbf{a}\in\{0,1\}^{\leq r}$ from the quantities $\left\{\left|\bigcup_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\leq r\right\}$ provided as input. Now, we prove that the statement is true for $r+1$ under the induction hypothesis. Note that we can also rewrite $\mathsf{occ}(\mathcal{C},\mathbf{a})$ for each set $\mathcal{C}\subseteq[n],\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ as

$$\mathsf{occ}(\mathcal{C},\mathbf{a})=\bigcap_{j\in\mathcal{C}'}\mathcal{S}(j)\bigcap_{j\in\mathcal{C}\setminus\mathcal{C}'}\mathcal{S}(j)^c$$

where $\mathcal{C}'\subseteq\mathcal{C}$ corresponds to the indices in $\mathcal{C}$ for which the entries in $\mathbf{a}$ is $\mathtt{1}$. Fix any set $i_1,i_2,\ldots,i_{r+1}\in[n]$. Then we can compute $\left|\bigcap_{b=1}^{r+1}\mathcal{S}(i_b)\right|$ using the following equation:

$$(-1)^{r+3}\left|\bigcap_{b=1}^{r+1}\mathcal{S}(i_b)\right|=\sum_{u=1}^{r}(-1)^{u+1}\sum_{\substack{j_1,j_2,\ldots,j_u\in\{i_1,i_2,\ldots,i_{r+1}\}\\ j_1<j_2<\cdots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{S}(j_b)\right|-\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right|.$$

Finally for each proper subset $\mathcal{Y}\subset\{i_1,i_2,\ldots,i_{r+1}\}$, we can compute $\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|$ using the following set of equations:

$$\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right| = \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)^c\right|$$
$$= \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right| - \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)\right|$$
$$= \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right| - \left|\bigcup_{i_b\in\mathcal{Y}}\Big(\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(i_b)\Big)\right|.$$

The first term is already pre-computed and the second term is again a union of intersection of sets. for each $j_b \in \mathcal{Y}$, let us define $\mathcal{H}(j_b) := \bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(j_b)$. Therefore we have

$$\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right| = \sum_{u=1}^{|\mathcal{Y}|}(-1)^{u+1}\sum_{\substack{j_1,j_2,\ldots,j_u\in\mathcal{Y}\\ j_1<j_2<\cdots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{H}(j_b)\right|.$$

We can compute $\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right|$ because the quantities on the right hand side of the equation have already been pre-computed (using our induction hypothesis). Therefore, the lemma is proved.

□

Proof of Lemma 2 when $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ is provided. Suppose we are given $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all sets $\mathcal{V}\subseteq[n]$ satisfying $|\mathcal{V}|\leq s$. We will omit the subscript $\mathcal{V}$ from hereon for simplicity. As in Lemma 2, the set $\cap_{i\in\mathcal{C}}\mathcal{S}(i)$ is equivalent to the set $\mathsf{occ}(C,\mathbf{1}_{|C|})$ or the number of unknown vectors in $\mathcal{V}$ whose restriction to the indices in $\mathcal{C}$ is the all one vector and in particular, $\mathsf{occ}((i),1)=\mathcal{S}(i)$. We will re-use the equation that for t sets $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_t$, we must have

$$\left|\bigcup_{i=1}^{t}\mathcal{A}_i\right| = \sum_{u=1}^{t}(-1)^{u+1}\sum_{1\leq i_1<i_2<\cdots<i_u\leq t}\left|\bigcap_{b=1}^{u}\mathcal{A}_{i_b}\right|.$$

We now show using induction on s that the quantities $\left\{\left|\bigcap_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\leq s\right\}$ are sufficient to compute $|\mathsf{occ}(C,\mathbf{a})|$ for all subsets C of indices of size at most s, and any binary vector $\mathbf{a}\in\{0,1\}^{\leq s}$.

Base case ($t=1$):

The base case follows since we can infer $|\mathsf{occ}((i),0)| = \ell - |\mathsf{occ}((i),1)|$ from $|\mathsf{occ}((i),1)|$ for all $i\in[n]$.

Inductive Step: Let us assume that the statement is true for $r<s$ i.e., we can compute $|\mathsf{occ}(\mathcal{C},\mathbf{a})|$ for all subsets $\mathcal{C}$ satisfying $|\mathcal{C}|\leq r$ and any binary vector $\mathbf{a}\in\{0,1\}^{\leq r}$ from the quantities $\left\{\left|\bigcap_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\leq r\right\}$ provided as input. Now, we prove that the statement is true for $r+1$ under the induction hypothesis. Note that we can also rewrite $\mathsf{occ}(\mathcal{C},\mathbf{a})$ for any set $\mathcal{C}\subseteq[n],\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ as

$$\mathsf{occ}(\mathcal{C},\mathbf{a}) = \bigcap_{j\in\mathcal{C}'}\mathcal{S}(j)\bigcap_{j\in\mathcal{C}\setminus\mathcal{C}'}\mathcal{S}(j)^c$$

where $\mathcal{C}'\subseteq\mathcal{C}$ corresponds to the indices in $\mathcal{C}$ for which the entries in $\mathbf{a}$ is $\mathtt{1}$. Fix any set $i_1,i_2,\ldots,i_{r+1}\in[n]$. Then we can compute $\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right|$ using the following equation:

$$\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right| = \sum_{u=1}^{r+1}(-1)^{u+1}\sum_{\substack{j_1,j_2,\ldots,j_u\in\{i_1,i_2,\ldots,i_{r+1}\}\\ j_1<j_2<\cdots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{S}(j_b)\right|.$$

Finally for any proper subset $\mathcal{Y}\subset\{i_1,i_2,\ldots,i_{r+1}\}$, we can compute $\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|$ using the following set of equations:

$$\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right| = \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)^c\right|$$
$$= \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right| - \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)\right|$$
$$= \left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right| - \left|\bigcup_{i_b\in\mathcal{Y}}\Big(\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(i_b)\Big)\right|.$$

The first term is already pre-computed and the second term is again a union of intersection of sets. For any $i_b \in \mathcal{Y}$, let us define $\mathcal{H}(j_b) := \bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(j_b)$. Therefore we have

$$\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right| = \sum_{u=1}^{|\mathcal{Y}|}(-1)^{u+1}\sum_{\substack{j_1,j_2,\ldots,j_u\in\mathcal{Y}\\ j_1<j_2<\cdots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{H}(j_b)\right|.$$

We can compute $\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right|$ because the quantities on the right hand side of the equation have already been pre-computed (using our induction hypothesis). Therefore, the lemma is proved.

□

Proof of Corollary 1. We know that all vectors $\mathbf{v}\in\mathcal{V}$ satisfy $\|\mathbf{v}\|_0 \le k$ as they are k-sparse. Therefore, in the first stage, by computing $|\mathcal{S}(i)|$ for all $i\in[n]$, we can recover the union of support of all the unknown vectors $\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v})$ by computing $\mathcal{T} = \{i\in[n] \mid \mathcal{S}(i) > 0\}$. The probability of failure in finding the union of support exactly is at most $n\gamma$. Once we recover $\mathcal{T}$, we compute $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C}\subseteq\mathcal{T}, |\mathcal{C}| \le \log\ell + 1$ (or alternatively $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C}\subseteq\mathcal{T}, |\mathcal{C}| \le \log\ell + 1$). The probability of failure for this this event $(\ell k)^{\log\ell+1}\gamma$. From Lemma 1, we know that computing $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C}\subseteq[n], |\mathcal{C}| \le \log\ell+1$ (or alternatively $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C}\subseteq\mathcal{T}, |\mathcal{C}| \le \log\ell+1$) exactly will allow us to recover the support of all the unknown vectors in $\mathcal{V}$. However $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$ for all $\mathcal{C}\subseteq[n]\setminus\mathcal{T}$ provided $\mathcal{T}$ is computed correctly. Therefore, we can recover the support of all the unknown vectors in $\mathcal{V}$ with $\mathsf{T}\log\gamma^{-1}$ samples with probability at least $1-((\ell k)^{\log\ell+1}+n)\gamma$. Rewriting the previous statement so that the failure probability is γ leads to the statement of the lemma. □

Algorithm 6 EXACT SUPPORT RECOVERY USING ACCESS TO ESTIMATES OF $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (OR ALTERNATIVELY $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) THAT ARE CORRECT WITH HIGH PROBABILITY

Require: For $\mathcal{C}\subseteq[n]$, access to estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) that are correct with high probability.

1: For each $i\in[n]$, compute an estimate of $|\mathcal{S}(i)|$.
2: Compute $\mathcal{T} = \{i\in[n] \mid \text{estimate}(\mathcal{S}(i)) > 0\}$.
3: Compute estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) for all subsets $\mathcal{C}\subseteq\mathcal{T}, |\mathcal{C}| \le \log\ell+1$.
4: Compute $\mathsf{occ}(\mathcal{C},\mathbf{a})$ for all subsets $\mathcal{C}\subseteq\mathcal{T}, |\mathcal{C}| \le \log\ell+1, \mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ using the computed estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$).
5: Use Algorithm 10 to recover the support of all unknown vectors in $\mathcal{V}$.

Proof of Lemma 6. Note that $\mathsf{Trimmed}(\mathcal{V})$ is the largest subset of vectors in $\mathcal{V}\equiv\{\mathbf{v}^{(1)},\mathbf{v}^{(2)},\ldots,\mathbf{v}^{(\ell)}\}$ such that the support of any vector in $\mathsf{Trimmed}(\mathcal{V})$ is not contained within the support of any other vector in $\mathsf{Trimmed}(\mathcal{V})$. Let us fix a vector $\mathbf{v}\in\mathsf{Trimmed}(\mathcal{V})$. For any other vector $\mathbf{v}'\in\mathsf{Trimmed}(\mathcal{V})$ there must exist an index $i_{\mathbf{v},\mathbf{v}'}\in\mathsf{supp}(\mathbf{v})$ such that $i_{\mathbf{v},\mathbf{v}'}\notin\mathsf{supp}(\mathbf{v}')$. Clearly the vector $\mathbf{v}$ constrained to the set of indices $\mathcal{C}\triangleq\cup_{\mathbf{v}'\in\mathsf{Trimmed}(\mathcal{V}),\mathbf{v}'\neq\mathbf{v}}\{i_{\mathbf{v},\mathbf{v}'}\}$ is an all-one vector but $\mathbf{v}'_{|\mathcal{C}}\neq\mathbf{1}$ for all $\mathbf{v}'\in\mathcal{V},\mathbf{v}'\neq\mathbf{v}$. This is true for all vectors in $\mathsf{Trimmed}(\mathcal{V})$ and since $|\mathsf{Trimmed}(\mathcal{V})\setminus\{\mathbf{v}\}| \le \ell-1$, we must have $\mathsf{Trimmed}(\mathcal{V})$ to be $(\ell-1)-$good. □

Proof of Lemma 7. As stated in the Lemma, suppose it is known if $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n]$ satisfying $|\mathcal{C}| \leq s+1$. Further assume that the set of unknown vectors $\mathcal{V}$ is s-good. Consider any vector $\mathbf{v} \in \mathsf{Trimmed}(\mathcal{V})$. Since $\mathcal{V}$ is s-good, there must exist an ordered set $\mathcal{C} \subseteq [n]$ such that $\mathbf{v}_{|\mathcal{C}}$ is the all 1 vector but $\mathbf{v}'_{|\mathcal{C}}$ is not the all 1 vector for any other vector $\mathbf{v}' \in \mathsf{Trimmed}(\mathcal{V})$. Therefore, we must have $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$. But, on the other hand, notice that if $|\mathsf{Trimmed}(\mathcal{V})| \geq 2$, there must exist an index $j \in \cup_{\mathbf{v}\in\mathsf{Trimmed}(\mathcal{V})}\mathsf{supp}(\mathbf{v})$ such that $\left|\cap_{i\in\mathcal{C}\cup\{j\}}\mathcal{S}(i)\right| = 0$ since the support of $\mathbf{v}$ does not contain the support of all other vectors. Algorithm 7 precisely checks for this condition and therefore this completes the proof.

□

Algorithm 7 Partial Support recovery using the quantities $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$

Require: For every $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell$, the quantities $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$ are provided as input
1: Set $\mathcal{T} = \phi$
2: **while** There exists a set $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell - 1$ such that $\mathbf{v}_{|\mathcal{C}} \neq \mathbf{1}_{|\mathcal{C}|}$ and $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0] = 1$. **do**
3: Set $\mathcal{U} = \mathcal{C}$.
4: **for** $j \in [n] \setminus \mathcal{C}$ **do**
5: **if** $\mathbf{1}[\left|\cap_{i\in\mathcal{C}\cup\{j\}}\mathcal{S}(i)\right| > 0] = 1$ **then**
6: Set $\mathcal{U} \leftarrow \mathcal{U} \cup \{j\}$
7: **end if**
8: **end for**
9: Set $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{v}\}$ where $\mathbf{v} \in \{0,1\}^n$ and $\mathsf{supp}(\mathbf{v}) = \mathcal{U}$.
10: **end while**
11: Return $\mathcal{T}$.

Algorithm 8 Partial Support recovery using access to estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$ that are correct with high probability

Require: For $\mathcal{C} \subseteq [n]$, access to estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$ that are correct with high probability.
1: For each $i \in [n]$, compute an estimate of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$.
2: Compute $\mathcal{T} = \{i \in [n] \mid \text{estimate}(\mathbf{1}[|\mathcal{S}(i)| > 0])) = \text{True}\}$.
3: Compute estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0]$ for all subsets $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \leq \ell$.
4: Use Algorithm 7 to recover the support of all unknown vectors in $\mathcal{V}$.

Proof of Corollary 3. Again, we know that all vectors $\mathbf{v} \in \mathcal{V}$ satisfy $\|\mathbf{v}\|_0 \leq k$ as they are k-sparse. Therefore, in the first stage, by computing if $|\mathcal{S}(i)| > 0$ for all $i \in [n]$, we can recover the union of support of all the unknown vectors $\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v})$ by computing $\mathcal{T} = \{i \in [n] \mid \mathcal{S}(i) > 0\}$. The probability of failure in finding the union of support correctly is at most $n\gamma$. Once we recover $\mathcal{T}$ correctly, we compute $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \leq \ell$. The probability of failure for this event $(\ell k)^\ell\gamma$. From Lemma 8, we know that computing $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell$ exactly will allow us to recover the support of all the unknown vectors in $\mathcal{V}$. On the other hand, we will have $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$ for all $\mathcal{C} \subseteq [n] \setminus \mathcal{T}$ provided $\mathcal{T}$ is computed correctly. Therefore, we can achieve deduplicated support recovery of all the unknown vectors in $\mathcal{V}$ with $\mathsf{T}\log\gamma^{-1}$ samples with probability at least $1 - ((\ell k)^\ell + n)\gamma$. Rewriting, so that the failure probability is γ leads to the statement of the lemma. □

Proof of Lemma 8. Consider the special case when $|\mathsf{Trimmed}(\mathcal{V})| = 1$ i.e. there exists a particular vector $\mathbf{v}$ in $\mathcal{V}$ whose support subsumes the support of all the other unknown vectors in $\mathcal{V}$. In that case, for each set $\mathcal{C} \subseteq \cup_{\mathbf{v}\in\mathsf{Trimmed}(\mathcal{V})}\mathsf{supp}(\mathbf{v})$, $|\mathcal{C}| \leq \ell$, we must have that $\left|\bigcup_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ (as there is only a single vector in $\mathsf{Trimmed}(\mathcal{V})$). On the other hand, if $|\mathsf{Trimmed}(\mathcal{V})| \geq 2$, then we know that $\mathsf{Trimmed}(\mathcal{V})$ is $(\ell-1)$-good and therefore, for each vector $\mathbf{v} \in \mathsf{Trimmed}(\mathcal{V})$, there exists an ordered set and an index $\mathcal{C}, \{j\} \subseteq \cup_{\mathbf{v}\in\mathsf{Trimmed}(\mathcal{V})}\mathsf{supp}(\mathbf{v})$, $|\mathcal{C}| \leq \ell - 1$ such that $\mathcal{C}$ belongs to the support of $\mathbf{v}$ but does not belong to the support of any other vector; hence $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ but $\left|\bigcap_{i\in\mathcal{C}\cup\{j\}}\mathcal{S}(i)\right| = 0$. In other words, there exists a set of size ℓ that is a subset of the union of support of vectors in $\mathsf{Trimmed}(\mathcal{V})$ but there does not exist any unknown vector that has 1 in all the indices indexed by the aforementioned set. Again, Algorithm 7 precisely checks this conditions and therefore this completes the proof. □

Algorithm 9 ESTIMATE(m, B) Estimating $\mathbb{E}X$ for $X \sim \mathcal{P}$

Require: I.i.d samples $x^{(1)}, x^{(2)}, \ldots, x^{(m)} \sim \mathcal{P}$
1: Set $t = m/B$
2: **for** $i = 1, 2, \ldots, B$ **do**
3: Set Batch i to be the samples $x^{(j)}$ for $j \in \{it+1, it+2, \ldots, (i+1)t\}$.
4: Set $S_1^i = \sum_{j \in \text{ Batch } i} \frac{x^{(j)}}{t}$
5: **end for**
6: Return $\mathsf{median}(\{S_1^i\}_{i=1}^B)$

Lemma 13. *The set* $\mathsf{Trimmed}(\mathcal{V})$ *is unique.*

Proof. We will prove this lemma by contradiction. Suppose there exists two distinct sets $\mathcal{T}_1, \mathcal{T}_2 \subset \mathcal{V}$ such that $|\mathcal{T}_1| = |\mathcal{T}_2| = |\mathsf{Trimmed}(\mathcal{V})|$. Since $\mathcal{T}_1, \mathcal{T}_2$ are distinct, there must exist a vector $\mathbf{v} \in \mathcal{T}_2 \setminus \mathcal{T}_1$. If $\mathsf{supp}(\mathbf{v})$ is not contained with the support of some vector in $\mathcal{T}_1$ and there is no other vector in $\mathcal{V}$ whose support contains $\mathbf{v}$, then clearly, $\mathbf{v}$ can be added to $\mathcal{T}_1$ implying that $\mathcal{T}_1$ cannot be the largest deduplicated set. On the other hand, suppose $\mathsf{supp}(\mathbf{v})$ is contained within the support of some vector $\mathbf{v}'$ in $\mathcal{T}_1$. However, this implies that $\mathcal{T}_2$ cannot be a valid deduplicated set as the support of $\mathbf{v}$ is contained with the support of $\mathbf{v}'$ and therefore, $\mathbf{v}$ cannot belong to a deduplicated set. This implies that the vector $\mathbf{v}$ cannot exist without violating some constrained of $\mathsf{Trimmed}(\mathcal{V})$ and therefore, the set $\mathsf{Trimmed}(\mathcal{V})$ is unique. □

E Technical Lemmas

Lemma 14 (Hoeffding's inequality for bounded random variables)**.** *Let $X_1, X_2, \ldots, X_m$ be independent random variables strictly bounded in the interval $[a, b]$. Let $\mu = m^{-1} \sum_i \mathbb{E}X_i$. In that case, we must have*

$$\Pr\Big(\Big| \frac{1}{m} \sum_{i=1}^{m} X_i - \mu \Big| \geq t \Big) \leq 2 \exp\Big(- \frac{2mt^2}{(b-a)^2} \Big).$$

Lemma 15 (Gaussian concentration inequality)**.** *Consider a random variable Z distributed according to $\mathcal{N}(0, \sigma^2)$. In that case, we must have $\Pr(|Z| \geq t) \leq 2\exp(-t^2/2)$ for any $t > 0$.*

Lemma 16 (Gaussian anti-concentration inequality)**.** *Consider a random variable Z distributed according to $\mathcal{N}(0, \sigma^2)$. In that case, we must have $\Pr(|Z| \leq t) \leq \sqrt{\frac{2}{\pi}} \cdot \frac{t}{\sigma}$ for any $t < \sigma\sqrt{\pi}/\sqrt{2}$.*

Proof. By simple calculations, we can have

$$\Pr(|Z| < t) \leq \int_{-t}^{t} \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma}} dx \leq \sqrt{\frac{2}{\pi}} \cdot \frac{t}{\sigma}.$$

□

Lemma 17. *Suppose $|\cup_{\mathbf{v} \in \mathcal{V}} \mathsf{supp}(\mathbf{v})| \leq n/2$. In that case, we can compute $\cup_{\mathbf{v} \in \mathcal{V}} \mathsf{supp}(\mathbf{v})$ correctly using $O(\ell^2(R^2 + \sigma^2)(\log n)^3/\delta^2)$ samples with probability at least $1 - 1/\mathsf{poly}(n)$.*

Proof. For each $i \in [n]$, suppose we want to test whether $i \in \cup_{\mathbf{v} \in \mathcal{V}} \mathsf{supp}(\mathbf{v})$ or not. Consider the random variable $y^2 \mathbf{x}_i^2$ when $(\mathbf{x}, y) \sim \mathcal{P}_r$. Notice that

$$\mathbb{E}y^2\mathbf{x}_i^2 = \frac{1}{\ell} \sum_{\mathbf{v} \in \mathcal{V}} \mathbb{E}y^2\mathbf{x}_i^2 \mid \mathbf{v} = \frac{1}{\ell} \sum_{\mathbf{v} \in \mathcal{V}} \Big(\sum_{j \in [n]} \mathbf{v}_j^2 + 2\mathbf{v}_i^2 \Big) \begin{cases} = \frac{1}{\ell} \sum_{\mathbf{v} \in \mathcal{V}} ||\mathbf{v}||_2^2 \text{ if } |\mathcal{S}_{\mathcal{V}}(i)| = 0 \\ \geq \frac{1}{\ell} \sum_{\mathbf{v} \in \mathcal{V}} ||\mathbf{v}||_2^2 + \frac{2\delta^2}{\ell} \text{ if } |\mathcal{S}_{\mathcal{V}}(i)| \neq 0 \end{cases}$$

where the final inequality follows from the fact that the magnitude of any non-zero entry of any unknown vector must be at least δ. For simplicity of notation, we will denote $A = \frac{1}{\ell} \sum_{\mathbf{v} \in \mathcal{V}} ||\mathbf{v}||_2^2$ to be average norm of the

unknown vectors. We will estimate $\mathbb{E}y^2\mathbf{x}_i^2$ by computing the following sample average

$$\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\mathbf{x}_i^{(j)}\Big)^2.$$

From the definition of $\mathcal{P}_r$, we must have $y\sim\mathcal{N}(0,\zeta^2+\sigma^2), |\zeta|\le R$ since $\mathbf{v}\in\{0,1\}^n, ||\mathbf{v}||_2\le R$ for all $\mathbf{v}\in\mathcal{V}$. By using Gaussian concentration inequalities, we must have $\Pr(|y|>t)\le\exp(-t^2/2(R^2+\sigma^2))$. Therefore, with probability $1-n^{-10}$, we have $|y|<20\sqrt{R^2+\sigma^2}\log n$. Similarly, with probability $1-n^{-10}$, $|\mathbf{x}_i|$ is bounded from above by $20\log n$. a Subsequently, we use Hoeffding's inequality to say that

$$\Pr\Big(\Big|\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\mathbf{x}_i^{(j)}\Big)^2-\mathbb{E}y^2\mathbf{x}_i^2\Big|\ge\frac{\delta^2 2}{2\ell}\Big)\le\exp\Big(-\Omega\Big(\frac{m\delta^2}{\ell^2(R^2+\sigma^2)(\log n)^2}\Big)\Big).$$

Hence, with $m=O(\ell^2(R^2+\sigma^2)(\log n)^3/\delta^2)$ samples, we can estimate if $\big|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)\big|>0$ or not correctly with probability at least $1-1/\mathsf{poly}(n)$. We can take a union bound over all the n indices to estimate $\mathbb{E}y^2\mathbf{x}_i^2$ correctly within an additive error of $\delta^2/2\ell$ for all $i\in[n]$. We will cluster all the indices such that a pair of distinct indices $u,v\in[n]$ are in the same group if

$$\Big|\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\mathbf{x}_u^{(j)}\Big)^2-\frac{\ell}{m}\cdot\sum_{j=1}^{m}\Big(y^{(j)}\mathbf{x}_v^{(j)}\Big)^2\Big|\le\frac{\delta^2}{\ell}.$$

Clearly, any two indices $u,v\in[n]$ that satisfy $|\mathcal{S}_{\mathcal{V}}(u)|=|\mathcal{S}_{\mathcal{V}}(v)|=0$ must belong to the same cluster. Since the size of the union of the support is at most $n/2$, the largest cluster must correspond to the indices where the entry is zero in all the unknown vectors. Subsequently, all those indices that do not belong to the largest cluster (after the clustering step) must belong to $\cap_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v})$. Furthermore, no index $i\in[n]$ such that $|\mathcal{S}_{\mathcal{V}}(i)|\neq 0$ can belong to the largest cluster. This complete the proof of the lemma. □

Finally, we will also use the following well-known lemma stating that we can compute estimates of the expectation of any one-dimensional random variable with only a few samples similar to sub-gaussian random variables.

Lemma 18. *For a random variable $x\sim\mathcal{P}$, there exists an algorithm (see Algorithm 9 in Appendix D) that can compute an estimate u of $\mathbb{E}x$ such that $|u-\mathbb{E}x|\le\epsilon$ with $O(\log\gamma^{-1}\mathbb{E}x^2/\epsilon^2)$ with probability at least $1-\gamma$.*

Proof of Lemma 18. Suppose we obtain m independent samples $x^{(1)},x^{(2)},\ldots,x^{(m)}\sim\mathcal{P}$. We use the median of means trick to compute u, an estimate of $\mathbb{E}x$. We will partition m samples obtained from $\mathcal{P}$ into $B=\lceil m/m'\rceil$ batches each containing m' samples each. In that case let us denote S^j to be the sample mean of the j^{th} batch i.e.

$$S^j=\sum_{s\in\text{Batch }j}\frac{x^{(s)}}{m'}.$$

We will estimate the true mean $\mathbb{E}x$ by computing u where $u\triangleq\mathsf{median}(\{S^j\}_{j=1}^B)$. For a fixed batch j, we can use Chebychev's inequality to say that

$$\Pr\Big(\big|S^j-\mathbb{E}x\big|\ge\epsilon\Big)\le\frac{\mathbb{E}x^2}{t\epsilon^2}\le\frac{1}{3}$$

for $t=O(\mathbb{E}x^2/\epsilon^2)$. Therefore for each batch j, we define an indicator random variable $Z_j=\mathbf{1}[\big|S^j-\mathbb{E}x\big|\ge\epsilon]$ and from our previous analysis we know that the probability of Z_j being 1 is less than 1/3. It is clear that $\mathbb{E}\sum_{j=1}^B Z_j\le B/3$ and on the other hand $|u-\mathbb{E}x|\ge\epsilon$ iff $\sum_{j=1}^B Z_j\ge B/2$. Therefore, due to the fact that Z_j's are independent, we can use Chernoff bound to conclude the following:

$$\Pr\Big(|u-\mathbb{E}x|\ge\epsilon\Big)\le\Pr\Big(\Big|\sum_{j=1}^{B}Z_j-\mathbb{E}\sum_{j=1}^{B}Z_j\Big|\ge\frac{\mathbb{E}\sum_{j=1}^{B}Z_j}{2}\Big)\le 2e^{-B/36}.$$

Hence, for $B=36\log\gamma^{-1}$, the estimate u is at most ϵ away from the true mean $\mathbb{E}x$ with probability at least $1-\gamma$. Therefore the sufficient sample complexity is $m=O(\log\gamma^{-1}\mathbb{E}x^2/\epsilon^2)$. □

Algorithm 10 RECOVER p-IDENTIFIABLE SUPPORTS

Require: $|\mathsf{occ}(C, \boldsymbol{a})|$ for every $C \subset [n]$, $|C| = t$, $t \in \{p, p+1\}$, and every $\mathbf{a} \in \{0,1\}^p \cup \{0,1\}^{p+1}$.

Set count $= 1, i = 1$.
while count $\leq \ell$ **do**
  if $|\mathsf{occ}(C, \boldsymbol{a})| = w$, and $|\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))| \in \{0, w\}$ for all $j \in [n] \setminus C$ **then**
    Set $\mathsf{supp}(\mathbf{u}^i)|_C = \mathbf{a}$
    For every $j \in [n] \setminus C$, set $\mathsf{supp}(\mathbf{u}^i)|_j = b$, where $|\mathsf{occ}(C \cup \{j\}, (\boldsymbol{a}, b))| = w$.
    Set $\mathsf{Multiplicity}^i = w$.
    For all $\mathbf{t} \in \{0,1\}^p \cup \{0,1\}^{p+1}, S \subseteq [n]$ such that $|S| \in \{p, p+1\}$, update

$$|\mathsf{occ}(S, \mathbf{t})| \leftarrow |\mathsf{occ}(S, \mathbf{t})| - |\mathsf{occ}(C, \mathbf{a})| \times \mathbf{1}[\mathsf{supp}(\mathbf{u}^i)|_S = \mathbf{t}]$$

    count $=$ count $+ w$.
    $i = i + 1$.
  end if
end while
Return $\mathsf{Multiplicity}^j$ copies of $\mathsf{supp}(\mathbf{u}^j)$ for all $j < i$.

F Proof of Lemma 1 (Theorem 1 in [Gandikota et al., 2021])

We will start with a few additional notations and definitions:

For a set of unknown vectors $\mathcal{V} \equiv \{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^\ell\}$, let $\mathbf{A} \in \{0,1\}^{n \times \ell}$ denote the support matrix corresponding to $\mathcal{V}$ where each column vector $\mathbf{A}_i \in \{0,1\}^n$ represents the support of the i^{th} unknown vector $\mathbf{v}^i$.

Definition 5 (p-identifiable)**.** *The i^{th} column $\mathbf{A}_i$ of a binary matrix $\mathbf{A} \in \{0,1\}^{n \times \ell}$ with all distinct columns is called p-identifiable if there exists a set $S \subset [n]$ of at most p-indices and a binary string $\mathbf{a} \in \{0,1\}^p$ such that $\mathbf{A}_i|_S = \boldsymbol{a}$, and $\mathbf{A}_j|_S \neq \boldsymbol{a}$ for all $j \neq i$.*

A binary matrix $\mathbf{A} \in \{0,1\}^{n \times \ell}$ with all distinct columns is called p-identifiable if there exists a permutation $\sigma : [\ell] \to [\ell]$ such that for all $i \in [\ell]$, the sub-matrix $\mathbf{A}^i$ formed by deleting the columns indexed by the set $\{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}$ has at least one p-identifiable column.

Let $\mathcal{V}$ be set of ℓ unknown vectors in $\mathbb{R}^n$, and $\mathbf{A} \in \{0,1\}^{n \times \ell}$ be its support matrix. Let $\mathbf{B}$ be the matrix obtained by deleting duplicate columns of $\mathbf{A}$. The set $\mathcal{V}$ is called p-identifiable if $\mathbf{B}$ is p-identifiable.

Theorem (Theorem 2 in [Gandikota et al., 2021])**.** *Any $n \times \ell$, (with $n > \ell$) binary matrix with all distinct columns is p-identifiable for some $p \leq \log \ell$.*

Proof. Suppose $\mathbf{A}$ is the said matrix. Since all the columns of $\mathbf{A}$ are distinct, there must exist an index $i \in [n]$ which is not the same for all columns in $\mathbf{A}$. We must have $|\mathsf{occ}((i), a)| \leq \ell/2$ for some $a \in \{0,1\}$. Subsequently, we consider the columns of $\mathbf{A}$ indexed by the set $\mathsf{occ}((i), a)$ and can repeat the same step. Evidently, there must exist an index $j \in [n]$ such that $|\mathsf{occ}((i), \mathbf{a})| \leq \ell/4$ for some $\mathbf{a} \in \{0,1\}^2$. Clearly, we can repeat this step at most $\log \ell$ times to find $C \subset [n]$ and $\mathbf{a} \in \{0,1\}^{\leq \log \ell}$ such that $|\mathsf{occ}(C, \mathbf{a})| = 1$ and therefore the column in $\mathsf{occ}(C, \mathbf{a})$ is p-identifiable. We denote the index of this column as $\sigma(1)$ and form the sub-matrix $\mathbf{A}^1$ by deleting the column. Again, $\mathbf{A}^1$ has $\ell - 1$ distinct columns and by repeating similar steps, $\mathbf{A}^1$ has a column that is $\log(\ell - 1)$ identifiable. More generally, $\mathbf{A}^i$ formed by deleting the columns indexed in the set $\{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}$, has a column that is $\log(\ell - i)$ identifiable with the index (in $\mathbf{A}$) of the column having the unique sub-string (in $\mathbf{A}^i$) denoted by $\sigma(i)$. Thus the lemma is proved. □

Next, we present an algorithm (see Algorithm 10) for support recovery of all the ℓ unknown vectors $\mathcal{V} \equiv \{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^\ell\}$ when $\mathcal{V}$ is p-identifiable. In particular, we show that if $\mathcal{V}$ is p-identifiable, then computing $|\mathsf{occ}(C, \mathbf{a})|$ for every subset of p and $p+1$ indices is sufficient to recover the supports.

The proof follows from the observation that for any subset of p indices $C \subset [n]$, index $j \in [n] \setminus C$ and $\mathbf{a} \in \{0,1\}^p$, $|\mathsf{occ}(C, \mathbf{a})| = |\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))| + |\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 0))|$. Therefore if one of the terms in the RHS is 0 for all $j \in [n] \setminus C$, then all the vectors in $\mathsf{occ}(C, \mathbf{a})$ share the same support.

Also, if some two vectors $\mathbf{u}, \mathbf{v} \in \mathsf{occ}(C, \mathbf{a})$ do not have the same support, then there will exist at least one index $j \in [n] \setminus C$ such that $\mathbf{u} \in \mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))|$ and $\mathbf{v} \in \mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 0))$ or the other way round, and therefore $|\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))| \notin \{0, |\mathsf{occ}(C, \mathbf{a})|\}$. Algorithm 10 precisely checks for this condition. The existence of some vector $\mathbf{v} \in \mathcal{V}$ (p-identifiable column), a subset of indices $C \subset [n]$ of size p, and a binary sub-string $\mathbf{b} \in \{0,1\}^{\leq p}$ follows from the fact that $\mathcal{V}$ is p-identifiable. Let us denote the subset of unknown vectors with distinct support in $\mathcal{V}$ by $\mathcal{V}^1$.

Once we recover the p-identifiable column of $\mathcal{V}^1$, we mark it as $\mathbf{u}^1$ and remove it from the set (if there are multiple p-identifiable columns, we arbitrarily choose one of them). Subsequently, we can modify the $|\mathsf{occ}(\cdot)|$ values for all $S \subseteq [n], |S| \in \{p, p+1\}$ and $\mathbf{t} \in \{0,1\}^p \cup \{0,1\}^{p+1}$ as follows:

$$\left|\mathsf{occ}^2(S, \mathbf{t})\right| \triangleq |\mathsf{occ}(S, \mathbf{t})| - |\mathsf{occ}(C, \mathbf{b})| \times \mathbf{1}[\mathsf{supp}(\mathbf{u}^1)|_S = \mathbf{t}]. \tag{7}$$

Notice that, Equation 7 computes $\left|\mathsf{occ}^2(S, \mathbf{t})\right| = \left|\{\mathbf{v}^i \in \mathcal{V}^2 \mid \mathsf{supp}(\mathbf{v}^i)|_S = \mathbf{t}\}\right|$ where $\mathcal{V}^2$ is formed by deleting all copies of $\mathbf{u}^1$ from $\mathcal{V}$. Since $\mathcal{V}^1$ is p-identifiable, there exists a p-identifiable column in $\mathcal{V}^1 \setminus \{\mathbf{u}^1\}$ as well which we can recover. More generally for $q > 2$, if $\mathbf{u}^{q-1}$ is the p-identifiable column with the unique binary sub-string $\mathbf{b}^{q-1}$ corresponding to the set of indices C^{q-1}, we will have for all $S \subseteq [n], |S| \in \{p, p+1\}$ and $\mathbf{t} \in \{0,1\}^p \cup \{0,1\}^{p+1}$

$$|\mathsf{occ}^q(S, \mathbf{t})| \triangleq \left|\mathsf{occ}^{q-1}(S, \mathbf{t})\right| - \left|\mathsf{occ}^{q-1}(C^{q-1}, \mathbf{b}^{q-1})\right| \times \mathbf{1}[\mathsf{supp}(\mathbf{u}^{q-1})|_S = \mathbf{t}]$$

implying $|\mathsf{occ}^q(S, \mathbf{t})| = \left|\{\mathbf{v}^i \in \mathcal{V}^q \mid \mathsf{supp}(\mathbf{v}^i)|_S = \mathbf{t}\}\right|$ where $\mathcal{V}^q$ is formed deleting all copies of $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^{q-1}$ from $\mathcal{V}$. Applying these steps recursively and repeatedly using the property that $\mathcal{V}$ is p-identifiable, we can recover all the vectors present in $\mathcal{V}$.