Learning Mixtures of Discrete Product Distributions using Spectral Decompositions

Prateek Jain
Microsoft Research India, Bangalore
prajain@microsoft.com

Sewoong Oh
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
woh@illinois.edu

Abstract—We study the problem of learning a distribution from samples, when the underlying distribution is a mixture of product distributions over discrete domains. This problem is motivated by several practical applications such as crowdsourcing, recommendation systems, and learning Boolean functions. The existing solutions either heavily rely on the fact that the number of components in the mixtures is finite or have sample/time complexity that is exponential in the number of components. In this paper, we introduce a polynomial time/sample complexity method for learning a mixture of $r$ discrete product distributions over $\{1,2,\ldots,\ell\}^n$, for general $\ell$ and $r$. We show that our approach is statistically consistent and further provide finite sample guarantees.

We use techniques from the recent work on tensor decompositions for higher-order moment matching. A crucial step in these moment matching methods is to construct a certain matrix and a certain tensor with low-rank spectral decompositions. These matrices and tensors are typically estimated directly from the samples. The main challenge in learning mixtures of discrete product distributions is that these low-rank tensors cannot be obtained directly from the sample moments. Instead, we reduce the tensor estimation problem to: a) estimating a low-rank matrix using only off-diagonal block elements; and b) estimating a tensor using a small number of linear measurements. Leveraging on recent developments in matrix completion, we give an alternating minimization based method to estimate the low-rank matrix, and formulate the tensor completion problem as a least-squares problem.

I. INTRODUCTION

Consider the following generative model: select a mixture component $q$ with probability $w_q$ and then generate a sample $y \in \mathbb{R}^n$ from this discrete distribution, where each $y_i \sim D_{y_i}, 1 \leq i \leq n$ and $D_{y_i}$ is a discrete distribution over $[\ell]$; each $y_i$ is sampled independently. Alternatively, let $x_i = e_j$, if $y_i = j$, where $e_j$ is the $j$-th standard basis vector. Also, let $\pi_i \in \mathbb{R}^{\ell \times r}$ represent $D_{y_i}$ where $\pi_i(j, q) = \text{Prob}(y_i = j | \text{cluster} = q)$. Then, the discrete distribution can be succinctly represented by the transition matrix $\Pi \in \mathbb{R}^{\ell \times n \times r} = [\pi_1; \pi_2; \ldots; \pi_n]$ and the weights $W = [w_1, \ldots, w_r]$.

The above mentioned mixture distribution (of $\ell$-wise discrete distributions over product spaces) has been used to model several real-life problems in domains like crowdsourcing [DS79], genetics [SKH07], and recommendation systems [TM10]. For example in the crowd-sourcing application, this model is same as the popular Dawid and Skene [DS79] model: $x_i$ represents answer of the $i$-th worker to a multiple choice question (or task) of type $q \in [r]$. Given a question’s type $q$, each of the worker is assumed to answer independently; also, each question type $q$ is assumed to be selected uniformly at random with probability $w_q$. Now, typically the goal is either to learn type of each question (clustering) or to find out the “quality” of a worker for a given question type (i.e. learn $\Pi$).

In most of the applications, the above model has been studied in the context of either of the following problems:

- Learn mixture parameters $\{\pi_q\}_{q \in \{1,\ldots,r\}}$ and $\{w_q\}_{q \in \{1,\ldots,r\}}$ accurately and efficiently.
- Cluster the samples accurately and efficiently.

The above two mentioned problems are closely related. Historically, however, different algorithms have been proposed depending on which question is addressed. Also, for each of the problems, distinct measures of performances have been used to evaluate the proposed solution. In this paper, we propose an efficient method that can be used to solve both the above mentioned problems.

The first question of estimating the underlying parameters of the mixture components has been addressed in [KMR+94], [FM99], [FO98], where the error of a given algorithm is measured as the KL-divergence between the true distribution and the estimated distribution. More precisely, a mixture learning algorithm is said to be an accurate learning algorithm, if it outputs a mixture of product distribution, s.t., the following holds with probability $\geq 1 - \delta$:

$$D_{KL}(X \mid \hat{X}) \equiv \sum_x P(X = x) \log(P(X = x) / P(\hat{X} = x)) \leq \epsilon,$$

where $\epsilon, \delta \in (0, 1)$ are any given constants, and $X, \hat{X} \in \{0,1\}^{nm}$ denote the random vectors distributed according to the true and the estimated mixture distribution, respectively. Furthermore, the algorithm is said to efficient if its time complexity is polynomial in $n, r, \ell, 1/\epsilon$, and $\log(1/\delta)$.

This Probably Approximately Correct (PAC) style framework was first introduced by Kearns et al. [KMR+94], where they provided the first analytical result for a simpler problem of learning mixtures of Hamming balls, which is a special case of our model with $\ell = 2$. 
However, the running time of the proposed algorithm is super-polynomial \( O(n/\delta \log^2 r) \) and also assumes that one can obtain the exact probability of a sample \( y \). Freund and Mansour [FM99] were the first to addressed the sample complexity, but for the restrictive case of \( r = 2 \) and \( \ell = 2 \). For this case, their method’s has running time \( O(n^{3.5} \log(1/\delta)/\varepsilon^2) \) and sample complexity \( O(n^2 \log(1/\delta)/\varepsilon^2) \). Feldman, O’Donnell, and Servedio in [FOS08] generalized approach of [FM99] to arbitrary number of types \( r \) and arbitrary number of output labels \( \ell \). For general \( \ell \), their algorithm requires running time scaling as \( O((n\ell^3/\varepsilon^3)) \). Hence, the proposed algorithm is an efficient learning algorithm for all problem instances of \( \{\pi_q\}_{q \in [r]} \) and \( \{w_q\}_{q \in [r]} \), but for finite values of \( r = O(1) \) and \( \ell = O(1) \).

A breakthrough in Feldman et al.’s result is that their result holds for all problem instances, with no dependence on the minimum weight \( w_{\text{min}} \) or the condition number \( \sigma_1(\Pi W^{1/2})/\sigma_r(\Pi W^{1/2}) \), where \( \sigma_r(\Pi W^{1/2}) \) is the \( r \)-th singular value of \( \Pi W^{1/2} \). However, this comes at a cost of running time scaling exponentially in both \( r^3 \) and \( \ell \), which is unacceptable in practice for any value of \( r \) beyond two. Further, the running time is exponential for all problem instances, even when the problem parameters are well-behaved, with finite condition number.

In this paper, we alleviate this issue by proposing an efficient algorithm for well-behave mixture distributions. In particular, we give an algorithm with polynomial running time, and prove that it gives \( \varepsilon \)-accurate estimate for any problem instance that satisfy the following two conditions: a) \( w_q \) is strictly positive for all \( q \), b) \( \sigma_1(\Pi W^{1/2})/\sigma_r(\Pi W^{1/2}) \) is finite.

The existence of an efficient learning algorithm for all problem instances and parameters still remains an open problem, and it is conjectured in [FOS08] that “solving the mixture learning problem for any \( r = \omega(1) \) would require a major breakthrough in learning theory”.

The second question of clustering has been addressed in [CHRZ07], [CR08]. Chaudhuri et al. in [CHRZ07] introduced an iterative clustering algorithm but their method is restricted to the case of a mixture of two product distributions with binary outputs, i.e. \( r = 2 \) and \( \ell = 2 \). Chaudhuri and Rao in [CR08] proposed a spectral method for general \( r, \ell \). However, for the algorithm to correctly recover cluster of each sample w.h.p. the underlying mixture distribution should satisfy a certain ‘spreading’ condition. Moreover, the algorithm need to know the parameters characterizing the ‘spread’ of the distribution, which typically is not available apriori. The method doesn’t provide a guarantee for estimating the distribution. As is the case for the first problem, for clustering also, we provide an efficient algorithm for general \( \ell, r \), under the assumption that the condition number of \( \Pi W^{1/2} \) to be finite. This condition is not directly comparable with the spreading condition assumed in previous work. Our algorithm first estimates the mixture parameters and then uses the distance based clustering method of [AK01].

Our method for estimating the mixture parameters is based on the moment matching technique [AHK12], [AGMS12]. Here, typically second, third (and sometimes fourth) moment of the true distribution is estimated using the given samples. Then, using the spectral decomposition of the second moment one develops certain whitening operators that reduce the higher-order moment tensors to orthogonal tensors. Such higher order tensors are then decomposed using a power-method based method [AGH12] to obtain the required distribution parameters.

While such a technique is generic and applies to several popular models [HK13], [AGH+12], for many of the models the moments themselves constitute the “correct” intermediate quantity that can be used for whitening and tensor decomposition. However, because there are dependencies in the \( \ell \)-wise model (for example, \( x_1 \) to \( x_\ell \) are correlated), the higher-order moments are “incomplete” versions of the intermediate quantities that we require (see (1), (3)). Hence, we need to complete these moments so as to use them for estimating distribution parameters \( \Pi, W \).

Completion of the “incomplete” second moment, gives rise to a low-rank matrix completion problem where the block-diagonal elements are missing. For this problem, we propose an alternating minimization based method and, borrowing techniques from the recent work of [JNS13], we prove that alternating minimization is able to complete the second moment exactly. We would like to note that our alternating minimization result also solves a generalization of the low-rank+diagonal decomposition problem of [SCPW12]. Moreover, unlike trace-norm based method of [SCPW12], which in practice is computationally expensive, our method is efficient, requires only one SVD step, and is robust to noise as well.

We reduce the completion of the “incomplete” third moment to a simple least squares problem that is robust as well. Using techniques from our second moment completion method, we can analyze an alternating minimization method also for the third moment case as well. However, for the mixture problem we can exploit the structure to reduce the problem to an efficient least squares problem with closed form solution.

Next, we present our method (see Algorithm 1) that combines the estimates from the above mentioned steps to estimate the distribution parameters \( \Pi, W \) (see Theorem III.2, Theorem III.3). After estimating the model parameters \( \Pi, W \), we also show that the KL-divergence measure (popularly used for the problem (a) ) and the clustering error measure (used for the problem (b)) can also be shown to be small, in fact the excess error goes to zero deterministically, as the number of samples approaches infinite (see Corollary III.4 Corollary III.5).
II. RELATED WORK

Learning mixture of distributions is an important problem with several applications such as clustering, crowdsourcing, community detection etc. One of the most popular problems in this domain is that of learning a mixture of Gaussians, where recently there has been several interesting results using both spectral as well as moment-matching based techniques [VW04], [AK01], [MV10], [HK13].

Another popular mixture distribution arises in topic models, where each word \( x_i \) is selected from a \( \ell \)-sized dictionary. Several recent results show that such a model can also be learned efficiently using spectral as well as moments based methods [RSS11], [AHK12], [AGM12]. However, there is a crucial difference between the general mixture of product distribution that we consider and the topic model distribution. Given a topic (or question) \( q \), each of the words \( x_i \) in the topic model have exactly the same probability. That is, \( \pi_i = \pi, \forall i \). In contrast, for our problem, \( \pi_i \neq \pi_j, i \neq j \), in general.

Learning mixture of discrete distribution over product spaces has been popular in modeling several real-life problems such as crowd-sourcing, recommendation systems etc. However, as discussed in the previous section, most of the existing results for this problem are designed for the case of small \( \ell \) or \( r \) (number of mixture components). Now for several practical problems [KOS13], \( \ell \) can be large and hence existing methods either do not apply or are very inefficient. In this work, we propose first provably efficient method for learning mixture of discrete distributions for general \( \ell \) and \( r \).

Our method is based on tensor decomposition methods for moment matching that have been used to obtain several interesting results for learning mixture distributions. For example, [HK13] provided the first method to learn mixture of Gaussians without any separation assumption. Similarly, [AHK12] introduced a method for learning mixture of HMMs, and also for topic models. Using similar techniques, another interesting result has been obtained for the problem of independent component analysis (ICA) [AGMS12], [GR12], [HK13].

As mentioned in the previous section, most of the above mentioned methods rely on applying a fixed whitening operator obtained using the second moment estimate and then tensor decomposition reveals the true parameters of the distribution. However, in a mixture of \( \ell \)-way distribution that we consider, the second or the third moment do not reveal all the “required” entries and hence the standard whitening operators do not succeed for our problem. We handle this problem by completing the second moment using an alternating minimization method. Our proof for the alternating minimization method closely follows the analysis of [JNS13]. However, [JNS13] handled a matrix completion problem where the entries are missing uniformly at random, while in our case the block diagonal elements are missing.

A. Notation

Typically, we denote a matrix or a tensor by an uppercase letter (e.g. \( M \)) while a vector is denoted by a small-case letter (e.g. \( v \)). \( M_i \) denotes the \( i \)-th column of matrix \( M \). \( M_{ij} \) denotes the \((i,j)\)-th entry of matrix \( M \) and \( M_{i,j,k} \) denotes the \((i,j,k)\)-th entry of the 3rd order tensor \( M \). \( A^T \) denotes the transpose of matrix \( A \), i.e., \( A^T_{ij} = A_{ji} \). \([k] = \{1, ..., k\} \) denotes the set of first \( k \) integers. \( e_i \) denotes the \( i \)-th standard basis vector.

If \( M \in \mathbb{R}^{m \times d} \), then \( M^m (1 \leq m \leq n) \) denotes the \( m \)-th block of \( M \), i.e., \((m-1)\ell+1 \) to \( m\ell \)-th rows of \( M \). Operator \( \otimes \) denote outer product. For example, \( H = v_1 \otimes v_2 \otimes v_3 \) denote a rank-one tensor such that \( H_{abc} = (v_1)_a \cdot (v_2)_b \cdot (v_3)_c \). For a symmetric third-order tensor \( T \in \mathbb{R}^{d \times d \times d} \), define an \( r \times r \times r \) dimensional operation with respect to a matrix \( R \in \mathbb{R}^{d \times r} \) as

\[
T[R, R, R] \equiv \sum_{i_1, i_2, i_3 \in [d]} T_{i_1, i_2, i_3} R_{i_1, j_1} R_{i_2, j_2} R_{i_3, j_3} (e_{j_1} \otimes e_{j_2} \otimes e_{j_3}).
\]

\[ \|A\| = \|A\|_2 \] denotes the spectral norm of \( A \). That is, \( \|A\|_2 = \max_{\|x\|_2=1} \|Ax\| \), \( \|A\|_F \) denotes the Frobenius norm of \( A \), i.e., \( \|A\|_F = \sqrt{\sum_{i_1, i_2, ..., i_p} A_{i_1, i_2, ..., i_p}^2} \). \( M = U \Sigma V^T \) denotes the singular value decomposition (SVD) of \( M \), i.e., \( \sigma_r(M) \) denotes the \( r \)-th singular value of \( M \). Also, wlog, assume that \( \sigma_1 \geq \sigma_2 \cdots \geq \sigma_r \).

III. MAIN RESULTS

In this section, we present our main results for estimating the mixture weights \( w_{ij}, 1 \leq q \leq r \) and the probability matrix \( \Pi \) of the mixture distribution. Our estimation method is based on the moment-matching technique that has been popularized by several recent results [AHK12], [HKZ12], [HK13], [AGH12]. However, our method differs from the existing methods in the following crucial aspects: we propose (a) a matrix completion approach to estimate the second moments from samples (Algorithm 2); and (b) a least squares approach with an appropriate change of basis to estimate the third moments from samples (Algorithm 3). These approaches provide robust algorithms to estimating the moments and might be of independent interest to a broad range of applications in the domain of learning mixture distributions.
The key step in our method is estimation of the following two quantities:

\[ M_2 = \sum_{q \in [r]} w_q (\pi_q \otimes \pi_q) = \Pi \Pi^T \in \mathbb{R}^{fn \times fn}, \] (1)
\[ M_3 = \sum_{q \in [r]} w_q (\pi_q \otimes \pi_q \otimes \pi_q) \in \mathbb{R}^{fn \times fn \times fn}, \] (2)

where \( W \) is a diagonal matrix s.t. \( W_{qq} = w_q \), and \( \Pi = U \Sigma V^T \) denotes SVD of \( \Pi \).

Now, as is standard in the moment based methods, we exploit spectral structure of \( M_2, M_3 \) to recover the latent parameters \( \Pi, W \). The following theorem presents a method for estimating \( \Pi, W \), assuming \( M_2, M_3 \) are estimated exactly:

**Theorem III.1.** Let \( M_2, M_3 \) be as defined in (1), (2). Also, let \( M_2 = U M_3 U^T M_2 \) be the eigenvalue decomposition of \( M_2 \). Now, define \( G = M_3 [U M_3^{-1/2} U M_3^{-1/2}] \), \( V^G = [v_1^G v_2^G \ldots v_r^G] \in \mathbb{R}^{r \times r} \). Let \( \lambda_q \) be the eigenvalues and eigenvectors obtained by the orthogonal tensor decomposition of \( G \) (see [AGH+12]), i.e.,

\[ G = \sum_{q=1}^r \lambda_q^G (v_q^G \otimes v_q^G \otimes v_q^G). \]

Then, \( \Pi = U M_3 U^T (V^G)^T \Lambda^G \), and \( W = (\Lambda^G)^{-2} \), where \( \Lambda^G \in \mathbb{R}^{r \times r} \) is a diagonal matrix with \( \lambda_q^G \). The above theorem reduces the problem of estimation of mixture parameters \( \Pi, W \) to that of estimating \( M_2, M_3 \). Typically, in moment based methods, quantities similar to \( M_2, M_3 \) can be estimated directly using the second moment or third moment of the distribution, which can be estimated efficiently using the provided data samples. However, the block-diagonal entries of \( M_2, M_3 \) cannot be directly computed from these sample moments.

For example, the expected value of a diagonal entry at \( j \)-th coordinate is \( \mathbb{E}[x_i x_j^T]_{i,j} = \mathbb{E}[x_i,x_j] = \sum_{q \in [r]} w_q \Pi_{i,j} q \), where the corresponding entry for \( M_2 \) is \( M_2_{i,j} = \sum_{q \in [r]} w_q (\Pi_{i,j})^q \).

To recover these unknown \( \ell \times \ell \) block-diagonal entries of \( M_2 \), we use an alternating minimization algorithm. Our algorithm writes \( M_2 \) in a bi-linear form and solves for each factor of the bi-linear form using the computed off-diagonal blocks of \( M_2 \). We then prove that this algorithm exactly recovers the missing entries when we are given the exact second moment. For estimating \( M_3 \), we reduce the problem of estimating unknown block-diagonal entries of \( M_3 \) to a least squares problem that can be solved efficiently.

Concretely, to get a consistent estimate of \( M_2 \), we pose it as a matrix completion problem, where we use the off-block-diagonal entries of the second moment, which we know are consistent, to estimate the missing entries. Precisely, let

\[ \Omega_2 \equiv \{(i,j) \subseteq [fn] \times [fn] \mid \lfloor i \rfloor_\ell \neq \lfloor j \rfloor_\ell \}, \]

be the indices of the off-block-diagonal entries, and define a masking operator as:

\[ \mathcal{P}_{\Omega_2}(A)_{i,j} = \begin{cases} A_{i,j} , & \text{if } (i,j) \in \Omega_2 , \\ 0 , & \text{otherwise} \end{cases}. \] (3)

Now, using the fact that \( M_2 \) has rank at most \( r \), we find a rank-\( r \) estimate that explains the off-block-diagonal entries using an alternating minimization algorithm defined in Section IV.

\[ \hat{M}_2 \equiv \text{MatrixAltMin} \left( \frac{2}{|S|} \sum_{t \in [|S|/2]} x_t x_t^T , \Omega_2 , r , T \right) \] (4)

where \( \{x_1, \ldots, x_{|S|}\} \) is the set of observed samples, and \( T \) is the number of iterations.

Similarly for the tensor \( M_3 \), the sample third moment does not converge to \( M_3 \). However, the off-block diagonal entries do converge to the corresponding entries of \( M_3 \). That is, let

\[ \Omega_3 \equiv \{(i,j,k) \subseteq [fn] \times [fn] \times [fn] \mid \lfloor i \rfloor_\ell \neq \lfloor j \rfloor_\ell \neq \lfloor k \rfloor_\ell \neq \lfloor i \rfloor_\ell \}, \]

be the indices of the off-block-diagonal entries, and define the following masking operator:

\[ \mathcal{P}_{\Omega_3}(A)_{i,j,k} = \begin{cases} A_{i,j,k} , & \text{if } (i,j,k) \in \Omega_3 , \\ 0 , & \text{otherwise} \end{cases}. \] (5)

Then, we have consistent estimates for \( \mathcal{P}_{\Omega_3}(M_3) \) from the sample third moment.

Now, in the case of \( M_3 \), we do not explicitly compute \( G \). Instead, we estimate a \( r \times r \times \) tensor \( \hat{G} \), similar to \( G \), using the following least squares formulation that uses only off-diagonal blocks of \( \mathcal{P}_{\Omega_2}(M_3) \). That is,

\[ \hat{G} \equiv \text{TensorsLS} \left( \frac{2}{|S|} \sum_{t \in [|S|/2]} x_t \otimes x_t \otimes x_t , \Omega_3 , \hat{U}_{M_2} , \hat{\Sigma}_{M_2} \right), \]

where \( \hat{M}_2 = \hat{U}_{M_2} \hat{\Sigma}_{M_2} \hat{U}_{M_2}^T \) is the eigenvalue decomposition of \( \hat{M}_2 \). After estimation of \( \hat{G} \), similar to Theorem III.1, we use the whitening and tensor decomposition to estimate \( \Pi, W \). See Algorithm 1 for a pseudo-code of our approach.

**Remark:** Note that we use a new set of \( |S|/2 \) samples to estimate the third moment. This sub-sampling helps us in our analysis, as it ensures independence of the samples \( x_{|S|/2+1}, \ldots, x_{|S|} \) from the output of the alternating minimization step (4).

The next theorem shows that the moment matching approach (Algorithm 1) is consistent. Let \( \hat{W} = \text{diag}(\hat{w}_1, \ldots, \hat{w}_r) \) and \( \hat{\Pi} = (\hat{\pi}_1, \ldots, \hat{\pi}_r) \) denote the estimates obtained using Algorithm 1. Also, let \( \mu \) denote the block-incoherence of \( M_2 = \Pi \Pi^T \) as defined in (7).

**Theorem III.2.** Assume that the sample second and the third moments are exact, i.e.,

\[ \mathcal{P}_{\Omega_2} \left( \frac{2}{|S|} \sum_{t \in [|S|/2]} x_t x_t^T \right) = \mathcal{P}_{\Omega_2}(M_2) \] and

\[ \mathcal{P}_{\Omega_3} \left( \frac{2}{|S|} \sum_{t \in [|S|/2+1]} x_t \otimes x_t \otimes x_t \right) = \mathcal{P}_{\Omega_3}(M_3) \].
Algorithm 1 Spectral-Dist: Moment method for Mixture of Discrete Distribution

1: Input: Samples \( \{x_t\}_{t \in S} \)
2: \( \hat{M}_2 \leftarrow \text{MATRIXALTMIN} \left( \left( \frac{2}{|S|} \sum_{t \in [1, \ldots, |S|/2]} x_t x_t^T \right), \Omega_2, r, r \right) \) (see Algorithm 2)
3: Compute eigenvalue decomposition of \( \hat{M}_2 = \tilde{U}_M \Sigma_M \tilde{U}_M^T \)
4: \( \hat{G} \leftarrow \text{TENSORLS} \left( \left( \frac{2}{|S|} \sum_{t \in [1, \ldots, |S|/2+1]} x_t \otimes x_t \otimes x_t \right), \Omega_3, \tilde{U}_M, \tilde{S}_M \right) \) (see Algorithm 3)
5: Compute orthogonal tensor decomposition of \( \hat{G} = \sum_{q \in [r]} \hat{\Lambda}_q^2 (\hat{v}_q^G \otimes \hat{v}_q^G \otimes \hat{v}_q^G) \) using Power-method of [AGH+12]
6: Output: \( \Pi = \tilde{U}_M \Sigma_M \hat{G} \hat{G}^T \tilde{W} \), where \( \hat{G} = (\hat{\Lambda}_q^2)^{-2} \)

let \( T = \infty \) for the MATRIXALTMIN procedure and let \( n \geq C_1 \sigma_1(M_2)^5 \mu^5 \nu^{3.5} \sigma_2(M_2)^3 \) for a global constant \( C > 0 \). Then, there exists a permutation \( P \) over \( [r] \) such that, for all \( q \in [r] \),

\[
\pi_q = \hat{\pi}_{P(q)} \quad \text{and} \quad w_q = \hat{w}_{P(q)}
\]

We now provide a finite sample version of the above theorem.

Theorem III.3 (Finite sample bound). There exists positive constants \( C_0, C_1, C_2, C_3 \) and a permutation \( P \) on \( [r] \) such that if \( n \geq C_0 \sigma_1(M_2)^5 \mu^5 \nu^{3.5} / \sigma_2(M_2)^3 \) then for any \( \varepsilon_M \leq \frac{C_2}{T} < 1/2 \) and for a large enough sample size:

\[
|S| \geq \frac{C_0 \log(2n/\delta)/|M_2|^2}{\sigma_2(M_2)^4 \cdot \varepsilon_M^2},
\]

the following holds for all \( q \in [r] \), with probability at least 1 \( - 2\delta \):

\[
|\hat{w}_{P(q)} - w_q| \leq C_3 \varepsilon_M \frac{1}{\sqrt{w_{min}}},
\]

\[
||\hat{\pi}_{P(q)} - \pi_q|| \leq C_3 \varepsilon_M \sqrt{\frac{\sigma_1(M_2)^4 (1 + \varepsilon_M \sigma_2(M_2))}{w_{min}}}.
\]

Further, Algorithm 1 runs in time \( \text{poly}(n, \ell, r, 1/\varepsilon, \log(1/\delta), 1/w_{min}, \sigma_1(M_2)/\sigma_2(M_2)) \).

Note that, the estimated \( \hat{\pi}_q \)'s and \( \hat{w}_q \)'s using Algorithm 1 do not necessarily define a valid probability measure: they can take negative values and might not sum to one. We can process the estimates further to get a valid probability distribution, and show that the estimated mixture distribution is close in Kullback-Leibler divergence to the original one. Let \( \varepsilon_w = C_3 \varepsilon_M / \sqrt{w_{min}} \). We first set

\[
\hat{w}'_q = \begin{cases} 
\hat{w}_q & \text{if } \hat{w}_q \geq \varepsilon_w, \\
\varepsilon_w & \text{if } \hat{w}_q < \varepsilon_w,
\end{cases}
\]

and set mixture weights \( \hat{w}_q = \hat{w}_q' / \sum q' \hat{w}_q' \). Similarly, let

Algorithm 2 MATRIXALTMIN: Alternating Minimization for Matrix Completion

1: Input: \( S_2 = \frac{2}{|S|} \sum_{t \in [1, \ldots, |S|/2]} x_t x_t^T \), \( \Omega_2, r, T \)
2: Initialize \( \ell n \times r \) dimensional matrix \( U_0 \leftarrow \text{top-r eigenvectors of } P_{\Omega_2}(S_2) \)
3: for all \( t = 1 \) to \( T - 1 \)
4: \( \tilde{U}_{t+1} = \arg \min_U \|P_{\Omega_2}(S_2) - P_{\Omega_2}(U U_t^T)\|_F \)
5: \( \|U_{t+1}R_{t+1}\| = \|U_{t+1}^T]\) (QR decomposition)
6: end for
7: Output: \( \hat{M}_2 = (\hat{U}_T)(U_{T-1})^T \)

\[
\varepsilon_\pi = C_3 \varepsilon_M \sqrt{\frac{\sigma_1(M_2)^4 (1 + \varepsilon_M \sigma_2(M_2))}{w_{min}}}
\]

and set

\[
\hat{\pi}_{q,j} = \begin{cases} 
\hat{\pi}_{q,j} \quad & \text{if } \hat{\pi}_{q,j} \geq \varepsilon_\pi, \\
\varepsilon_\pi & \text{if } \hat{\pi}_{q,j} < \varepsilon_\pi
\end{cases}
\]

for all \( q \in [n] \), \( p \in [l] \), and \( j \in [d] \), and normalize it to get valid distributions \( \tilde{\pi}_{q,j} = \hat{\pi}_{q,j} / \sum_p \hat{\pi}_{q,p} \). Let \( Z \) denote a random vector in \( \{0, 1\}^n \) obtained by first selecting a random type \( q \) with probability \( \tilde{w}_q \) and then drawing from a random vector according to \( \tilde{\pi}_q \).

Corollary III.4 (KL-divergence bound). Under the hypotheses of Theorem III.3 there exists a positive constant C such that if \( |S| \geq C n^2 \log((\ell n/\delta)) / |M_2|^2 (n^6 \ell^{12} \tau + (1/\eta^2)) / (\sigma_2(M_2)^4 \eta^4 w_{min}) \), then Algorithm 1 with the above post-processing produces a r-mixture distribution \( \hat{Z} \) that, with probability at least 1 \( - \delta \), satisfies : \( D_{KL}(Z||\hat{Z}) \leq \eta \).

Moreover, we can show that the “type” of each data point can also be recovered accurately.

Corollary III.5 (Clustering bound). Define:

\[
\bar{\varepsilon} \equiv \max_{i,j \in [r]} \left\{ \left\| \pi_i - \pi_j \right\|_F - 2||\Pi||_F \sqrt{2 \log(r/\delta)} \right\}.
\]

Under the hypotheses of Theorem III.3 there exists a positive numerical constant C such that if \( \bar{\varepsilon} > 0 \) and \( |S| \geq C n^2 \log((\ell n/\delta)) / |M_2|^2 r / (\sigma_2(M_2)^4 \bar{\varepsilon}^2 w_{min}) \), then with probability at least 1 \( - \delta \), the distance based clustering algorithm of [AK07] computes a correct clustering of the samples.

IV. ALGORITHM

In this section, we describe our algorithm in detail. As mentioned in the previous section, our algorithm first estimates \( M_2 \) using the alternating minimization procedure. Recall that the second moment of the data given by \( S_2 \) cannot estimate the block-diagonal entries of \( M_2 \). That is, in the case of infinite samples, \( P_{\Omega_2}(S_2) = P_{\Omega_2}(M_2) \). However, to apply the “whitening” operator to third order tensor (see Theorem III.1) we need to estimate \( M_2 \).

In general it is not possible to estimate \( M_2 \) from \( P_{\Omega_2}(M_2) \) as one can fill any entries in the block-diagonal entries. Fortunately, we can avoid such a case when \( M_2 \)
Algorithm 3: Least Squares method for Tensor Estimation

1: Input: $S_3 = \frac{2}{\lVert S \rVert} \sum_{t \in \{S_1/2, \ldots, |S_1|\}} \{x_t \otimes x_t \otimes x_t\}$, $\Omega_3$, $\bar{U}_{M_2}$, $\bar{S}_{M_2}$
2: Define operator $\tilde{\nu} : \mathbb{R}^{r \times r \times r} \to \mathbb{R}^{n \times n \times n}$ as given in (9)
3: Define $\tilde{A} : \mathbb{R}^{r \times r \times r} \to \mathbb{R}^{r \times r \times r}$ s.t. $\tilde{A}(Z) = \tilde{\nu}(Z)[\bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}, \bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}, \bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}]$
4: Output: $\hat{G} = \arg\min_Z \|\tilde{A}(Z) - P_{\Omega_3}(S_3) [\bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}, \bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}, \bar{U}_{M_2} \bar{S}_{M_2}^{-1/2}]\|_F$

For estimating $M_2$, the noise $E$ in the off-block-diagonal entries are due to insufficient sample size. We can precisely bound how large the sampling noise is in the following lemma.

Lemma 1. Let $S_2 = \frac{2}{\lVert S \rVert} \sum_{t \in \{1, \ldots, |S|/2\}} x_t x_t^T$ be the sample co-variance matrix. Also, let $E = \|P_{\Omega_3}(S_2) - P_{\Omega_3}(M_2)\|_2$. Then,

$$\|E\|_2 \leq 8\sqrt{n^2 \log(n/\delta)}/|S|,$$

for all $i \in [n]$. The above theorem shows that $M_2$ can be recovered exactly from infinite many samples, if $n \geq \frac{\mu^2 \sigma(M)^2}{\delta \sigma(M)}$. Furthermore, using Lemma 1 $M_2$ can be recovered approximately, with sample size $|S| = O(n^2(\ell + r)/\delta \sigma(M)^2)$. Now, recovering $M_2 = \Pi W^T$ recovers the left-singular space of $M$, i.e., range($U$). However, we still need to recover $W$ and the right-singular space of $M$, i.e., range($V$).

To this end, we can estimate the tensor $\bar{M}_3$, “whiten” the tensor using $\bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}$. Recall that, $\bar{M}_3 = \bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}$, and then use tensor decomposition techniques to solve for $V, W$. However, we show that estimating $M_3$ is not necessary, we can directly estimate the “whitened” tensor by solving a system of linear equations. In particular, we design an operator $\tilde{A} : \mathbb{R}^{r \times r \times r} \to \mathbb{R}^{r \times r \times r}$ such that $\tilde{A}(G) \approx P_{\Omega_3}(S_3) [\bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}, \bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}, \bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}]$, where

$$G = \frac{1}{\sqrt{m}} \sum_{q \in [r]} (R_3 \pi_q \otimes R_3 \pi_q \otimes R_3 \pi_q), \text{ and } R_3 = \bar{S}_{M_3}^{-1/2} \bar{U}_{M_3}^T M_2 W^{1/2}.$$

Moreover, we show that $\tilde{A}$ is nearly-isometric. Hence, we can efficiently estimate $G$, using the following system of equations:

$$\hat{G} = \arg\min_Z \|\tilde{A}(Z) - P_{\Omega_3}(S_3) [\bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}, \bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}, \bar{U}_{M_3} \bar{S}_{M_3}^{-1/2}]\|_F.$$

Let $\mu$ and $\mu_1$ denote the block-incoherence of $M_2$ and $\bar{M}_2$ respectively, as defined in (7).
following holds with probability at least $1 - \delta$:

$$
\|\hat{G} - G\|_F \leq \left( \frac{12\mu_1^3\varepsilon^3 r_1(M_2)^3}{n_q^3\varepsilon^3 \sigma_r(M_2)^3} + \frac{6\varepsilon^3 r_1^3}{\sigma_r(M_2)^3} \sqrt{\frac{2\log(1/\delta)}{|S|}} \right),
$$

for $\varepsilon \equiv \|M_2 - M_2\|_2$.

Next, we apply the tensor decomposition method of [AGH09] to decompose obtained tensor, $\hat{G}$, and obtain $\hat{R}_3, \hat{W}$ that approximates $R_3$ and $W$. We then use the obtained estimate $\hat{R}_3, \hat{W}$ to estimate $\Pi$; see Algorithm 1 for the details. In particular, using Theorem IV.1 [V.1] and Theorem IV.2 [V.2] Algorithm 1 provides the following estimate for $\Pi$:

$$
\hat{\Pi} = \hat{U}_{M_2} \hat{S}_{M_2}^{1/2} \hat{R}_3 \hat{W}^{-1/2} \approx \hat{U}_{M_2} \hat{U}_T M_2 \Pi.
$$

Now, $\|\hat{\Pi} - \Pi\|_2$ can be used by the above equation along with the fact that $\text{range}(\hat{U}_{M_2}) \approx \text{range}(\Pi)$. See longer version of the paper for a detailed proof.

V. APPLICATIONS IN CROWD-SOURCING

Crowd-sourcing has emerged as an effective paradigm for solving large-scale data-processing tasks in domains where humans have an advantage over computers. Examples include image classification, video annotation, data entry, optical character recognition, and translation. For tasks with discrete choice outputs, one of the most widely used model is the Dawid-Skene model introduced in [DS79]: each expert $j$ is modeled through a $r \times r$ confusion matrix $\pi^{(j)}$ where $\pi^{(j)}_{pq}$ is the probability that the expert answers $q$ when the true label is $p$. This model was developed to study how different clinicians give different diagnosis, even when they are presented with the same medical chart. This is a special case, with $\ell = r$, of the mixture model studied in this paper.

Historically, a greedy algorithm based on Expectation-Maximization has been widely used for inference [DS79], [SFB+05], [HZ98], [SPI08], but with no understanding of how the performance changes with the problem parameters and sample size. Recently, spectral approaches were proposed and analyzed with provable guarantees. For a simple case when there are only two labels, i.e., $r = \ell = 2$, Ghosh et al. in [GKM11] and Karger et al. in [KOST11a] analyzed a spectral approach of using the top singular vector for clustering under Dawid-Skene model. The model studied in these work is a special case of our model with $r = \ell = 2$ and $w = [1/2, 1/2]$, and $\pi^{(j)} = \begin{bmatrix} p_j & 1 - p_j \\ 1 - p_j & p_j \end{bmatrix}$. Let $q = (1/n) \sum_{j \in [n]} 2(p_j - 1)^2$, then it follows that $\sigma_1(M_2) = (1/2)n$ and $\sigma_2(M_2) = (1/2)nq$. It was proved in [GKM11], [KOST11a] that if we project each data point $x_i$ onto the second singular vector of $S_2$ the empirical second moment, and make a decision based on the sign of this projection, we get good estimates with the probability of misclassification scales as $O(1/\sigma_r(M_2))$.

More recently, Karger et al. in [KOST11b] proposed a new approach based on a message-passing algorithm for computing the top singular vectors, and improved this misclassification bound to an exponentially decaying $O(e^{-C\sigma_r(M_2)})$ for some positive numerical constant $C$. However, these approaches highly rely on the fact that there are only two ground truth labels, and the algorithm and analysis cannot be generalized. These spectral approaches has been extended to general $r$ in [KOST13] with misclassification probability scaling as $O(r/\sigma_r(M_2))$, but this approach still uses the existing binary classification algorithms as a black box and tries to solve a series of binary classification tasks.

Furthermore, existing spectral approaches use $S_2$ directly for inference. This is not consistent, since even if infinite number of samples are provided, this empirical second moment does not converge to $M_2$. Instead, we use recent developments in matrix completion to recover $M_2$ from samples, thus providing a consistent estimator. Hence, we provide a robust clustering algorithm for crowd-sourcing and provide estimates for the mixture distribution with provable guarantees. Corollary III.5 shows that with large enough samples, the misclassification probability of our approach scales as $O(re^{-C(r\sigma_r(M_2)^2/n)})$ for some positive constant $C$. This is an exponential decay and is a significant improvement over the known error bound of $O(r/\sigma_r(M_2))$.

VI. CONCLUSION

We presented a method for learning a mixture of $\ell$-wise discrete distribution with distribution parameters $\Pi, W$. Our method shows that assuming $n \geq C_1\sigma_1(\Pi^{1/2}W^{1/2})^2 \sigma_r(M_2)^{\ell}/\varepsilon^2$, and the number of samples to be $|S| \geq C_2\sigma_1(\Pi^{1/2}W^{1/2})^{\ell}/\varepsilon^2$, we have $\|\hat{\Pi} - \Pi\|_2 \leq \varepsilon_1/\varepsilon_2$ where $\varepsilon_1 \equiv \sigma_1(M_2)/\sigma_r(M_2), \varepsilon_2 \equiv \sigma_r(M_2)^{\ell}$. Note that our algorithm does not require any separability condition on the distribution, is consistent for infinite samples, and is robust to noise as well. That is, our analysis can be easily extended to the noisy case, where there is a small amount of noise in each sample.

Our sample complexity bounds include the condition number of the distribution $\kappa$ which implies that our method requires $\kappa$ to be at most $\text{poly}(\ell, r)$. This makes our method unsuitable for the problem of learning Boolean functions [FOS08]. However, it is not clear if is possible to design an efficient algorithm with sample complexity independent of the condition number. We leave further study of the dependence of sample complexity on the condition number as a topic for future research.

Another drawback of our method is that $n$ is required to be $n = \Omega(r^4)$. We believe that this condition is natural, as one cannot recover the distribution for $n = 1$. However, establishing tight information theoretic lower bound on $n$ (w.r.t. $\ell, r$) is still an open problem.
For the crowd-sourcing application, the current error bound for clustering translates into $O(e^{-Crq^2})$ when $r = 2$. This is not as strong as the best known error bound of $O(e^{-Cnq^2})$, since $q$ always less than one. The current analysis and algorithm for clustering needs to be improved to get an error bound of $O(\text{e}^{-C\text{min}r(M^2)})$ for general $r$ such that it gives optimal error rate for the special case of $r = 2$.

The sample complexity also depends on $1/w_{\text{min}}$, which we believe is unnecessary. If there is a component with small mixing weight, we should be able to ignore such component smaller than the sample noise level and still guarantee the same level accuracy. To this end, we need an adaptive algorithm that detects the number of components that are non-trivial and this is a subject of future research.

More fundamentally, all of the moment matching methods based on the spectral decompositions suffer from the same restrictions. It is required that the underlying tensors have rank equal to the number of components, and the condition number needs to be small. However, the problem itself is not necessarily more difficult when the condition number is larger.

Finally, we believe that our technique of completion of the second and the higher order moments should have itself is not necessarily more difficult when the condition number needs to be small. However, the problem condition needs to be improved such that it gives optimal error rate for the special case of $r = 2$.

References


