On the Efficient Implementation of High Accuracy Optimality of Profile Maximum Likelihood

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper we provide an efficient algorithm to compute a universal plug-in estimator for symmetric properties of distributions that is sample optimal up to accuracy $\epsilon \gg n^{-1/3}$, where n is the sample size. Our estimator is based on profile-maximum-likelihood (PML) and improves upon the previous best accuracy threshold of $\epsilon \gg n^{-1/4}$ achievable by the polynomial time computable PML-based universal estimator of [ACSS21, ACSS20]. Further, this reaches a theoretical limit as [Han21] shows that a broad class of PML-based estimators (containing the new one we provide) are sub optimal in the regime $\epsilon \ll n^{-1/3}$.

1 Introduction

Given n independent samples $y_1, ..., y_n \in \mathcal{D}$ from an unknown discrete distribution $\mathbf{p} \in \Delta^{\mathcal{D}}$ the problem of estimating properties of $\mathbf{p}$, e.g. entropy, distance to uniformity, support size and coverage are among the most fundamental in statistics and learning. More broadly the problem of estimating *symmetric properties* of distributions $\mathbf{p}$ (i.e. properties invariant to label permutations) are well studied and have numerous applications [Cha84, BF93, CCG+12, TE87, Für05, KLR99, PBG+01, DS13, RCS+09, GTPB07, HHRB01].

Over the past decade, symmetric property estimation has received significant attention and there have been many improvements to the time and sample complexity for estimating different properties, e.g. support [VV11b, WY15], coverage [ZVV+16, OSW16], entropy [VV11b, WY16, JVHW15], and distance to uniformity [VV11a, JHW16]. Towards unifying the attainment of computationally-efficient, sample-optimal estimators a striking work of [ADOS16] provided a universal plug-in approach based on a (approximate) profile maximum likelihood (PML) distribution, that (approximately) maximizes the likelihood of the observed profile (i.e. multiset of observed frequencies).

Formally, [ADOS16] showed that given n i.i.d. samples from an unknown distribution $\mathbf{p}$, if there exists an estimator for a symmetric property f achieving accuracy ϵ and failure probability δ, then this PML-based plug-in approach achieves error 2ϵ with failure probability $\delta e^{3\sqrt{n}}$. This result further implied a sample optimal unified approach for estimating various symmetric properties including support size and coverage, entropy, distance to uniformity when the estimation error $\epsilon \gg n^{-1/4}$, as the failure probability δ for estimating these properties is roughly $\exp(-\Omega(n\epsilon^2))$. Interestingly, follow up work of [HS20] improved this analysis and showed that the failure probability of PML is at most $\delta^{1-c}\exp(-n^{1/3+c})$, for any constant $c > 0$. This improved result further implied a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/3}$, which was later shown to be tight by [Han21], meaning that PML is known to be not sample optimal in the regime $\epsilon \ll n^{-1/3}$. We also remark that the results in [ADOS16, HS20] hold for any β-approximate PML[1] for $\beta > \exp(-n^{1-c'})$ and

[1] β-approximate PML is a distribution that achieves a multiplicative β-approximation to the PML objective.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

any constant $c' > 0$, with a failure probability $\delta^{1-c}\exp(n^{1/3+c} + n^{1-c'})$, which further implies a sample optimal estimator in the regime $\epsilon \gg n^{-\min(1/3,c/2)}$ for properties with estimation failure probability less than $\exp(-\Omega(\epsilon^2 n))$. Therefore, note that better approximation leads to a better failure probability, which further implies a larger range of ϵ for which the estimator is sample optimal.

On the computational aspects of approximate PML distributions, [CSS19a] provided the first efficient algorithm with a non-trivial approximation guarantee of $\exp(-n^{2/3}\log n)$, which further implied a sample optimal universal estimator for $\epsilon \gg n^{-1/6}$. This was then improved by [ACSS21] which showed how to compute PML to higher accuracy of $\exp(-\sqrt{n}\log n)$ in polynomial time thereby achieving a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/4}$.

In light of these results, the key open problem remaining is to close the gap between the $\epsilon \gg n^{-1/3}$-threshold for which PML is statistically optimal and the best polynomial-time obtaining PML-based universal estimator for $\epsilon \gg n^{-1/4}$. In this work we ask:

Is there an efficient approximate PML based estimator that is sample competitive for $\epsilon \gg n^{-1/3}$.

In this paper we answer this question in the affirmative. In particular, we give an efficient PML based estimator that has failure probability at most $\delta^{1-c}\exp(n^{1/3+c} + n^{1-c'})$, equivalently, a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/3}$. Note that this result is tight in the sense that PML is known to be not optimal in the regime $\epsilon \ll n^{-1/3}$. In fact [Han21] show that, no reasonable estimator is optimal beyond the regime, therefore we efficiently implement the regime which is known to be sample optimal. Refer to Section 2 for details.

To obtain this result we depart slightly from the previous approaches in [ADOS16, CSS19a, ACSS21]. Rather than directly compute an approximate PML distribution we show that a weaker notion of approximation suffices to get us the desired universal estimator. We propose a notion of a β-*weak approximate PML distribution* and show that an $\exp(-n^{1/3}\log n)$ weak approximate PML achieves the desired failure probability of $\delta^{1-c}\exp(n^{1/3+c})$ for any constant $c > 0$. Further, we provide an efficient algorithm to compute such an $\exp(-n^{1/3}\log n)$ weak approximate PML distribution. This notion of weak approximation is motivated by [HS20] and our paper can be viewed as an efficient algorithmic instantiation of [HS20].

Ultimately, our algorithms use the convex relaxation presented in [CSS19a, ACSS21] and provide a new rounding algorithm. An important step in the previous best result [ACSS20] and the present work is a matrix rounding procedure which controls the approximation guarantee. At a high level, the guarantee for the rounding procedure in [ACSS20] is exponential in the sum of matrix dimensions. In the present work, we need to round a rectangular matrix with approximation exponential in the smaller dimension, which is infeasible for arbitrary matrices. Our key technical innovation is to introduce a *swap* operation (see Section 4.1) which facilitates such an approximation guarantee. In addition to better approximation guarantee, our algorithm also exhibits better run times (see Section 2).

Organization: We introduce preliminaries in Section 1.1. In Section 2, we state all our main results and comparison it to prior work. In Section 2, we cover the related work. In Section 3, we provide the convex relaxation to PML studied in [CSS19a, ACSS21]. Finally in section Section 4, we provide a proof sketch of our main computational result. Many proofs are then differed to the appendix.

1.1 Preliminaries

For matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{s\times t}$, we say $\mathbf{A} \leq \mathbf{B}$ if and only if $\mathbf{A}_{ij} \leq \mathbf{B}_{ij}$ for all $i \in [s]$ and $j \in [t]$. Throughout this paper, we assume we receive a sequence of n independent samples from a distribution $\mathbf{p} \in \Delta^{\mathcal{D}}$, where $\Delta^{\mathcal{D}}$ is the set of all discrete distributions supported on domain $\mathcal{D}$. We let $[a, b]$ and $[a, b]_{\mathbb{R}}$ denote the interval of integers and reals $\geq a$ and $\leq b$ respectively, so $\Delta^{\mathcal{D}} \stackrel{\text{def}}{=} \{\mathbf{q} \in [0,1]^{\mathcal{D}}_{\mathbb{R}} | \left\|q\right\|_1 = 1\}$. Let $\mathcal{D}^n$ be the set of all length n sequences and $y^n \in \mathcal{D}^n$ be one such sequence with y^n_i denoting its ith element. Let $\mathbf{f}(y^n, x) \stackrel{\text{def}}{=} |\{i \in [n] \mid y^n_i = x\}|$ and $\mathbf{p}_x$ be the frequency and probability of $x \in \mathcal{D}$ respectively. For a sequence $y^n \in \mathcal{D}^n$, let $\mathbf{M} = \{\mathbf{f}(y^n, x)\}_{x\in\mathcal{D}} \setminus \{0\}$ be the set of all its non-zero distinct frequencies and $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{|\mathbf{M}|}$ be these distinct frequencies. The *profile* of a sequence y^n denoted $\phi = \Phi(y^n)$ is a vector in $\mathbb{Z}^{|\mathbf{M}|}$, where $\phi_j \stackrel{\text{def}}{=} |\{x \in \mathcal{D} \mid \mathbf{f}(y^n, x) = \mathbf{m}_j\}|$ is the number of domain elements with frequency

$\mathbf{m}_j$. We call n the length of profile ϕ and let Φ^n denote the set of all profiles of length n. The probability of observing sequence y^n and profile ϕ with respect to a distribution $\mathbf{p}$ are as follows: $\mathbb{P}(\mathbf{p}, y^n) = \prod_{x \in \mathcal{D}} \mathbf{p}_x^{\mathbf{f}(y^n, x)}$ and $\mathbb{P}(\mathbf{p}, \phi) = \sum_{\{y^n \in \mathcal{D}^n \mid \Phi(y^n) = \phi\}} \mathbb{P}(\mathbf{p}, y^n)$.

A distribution $\mathbf{p}_\phi \in \Delta^{\mathcal{D}}$ is a *profile maximum likelihood* (PML) distribution for profile $\phi \in \Phi^n$ if $\mathbf{p}_\phi \in \operatorname{argmax}_{\mathbf{p} \in \Delta^{\mathcal{D}}} \mathbb{P}(\mathbf{p}, \phi)$. Further, a distribution $\mathbf{p}_\phi^\beta$ is a *β-approximate PML* distribution if $\mathbb{P}(\mathbf{p}_\phi^\beta, \phi) \geq \beta \cdot \mathbb{P}(\mathbf{p}_\phi, \phi)$. For a distribution $\mathbf{p}$ and n, let $\mathbf{X}$ be a random variable that takes value $\phi \in \Phi^n$ with probability $\mathbb{P}(\mathbf{p}, \phi)$. The distribution of $\mathbf{X}$ depends only on $\mathbf{p}$ and n and we call $H(\mathbf{X})$ (entropy of $\mathbf{X}$) the *profile entropy* with respect to $(\mathbf{p}, n)$ and denote it by $H(\Phi^n, \mathbf{p})$. We use $\widetilde{O}(\cdot)$, $\widetilde{\Omega}(\cdot)$ notation to hide all polylogarithmic factors in n and N. We let $a_n \gg b_n$ to denote that $a_n \in \Omega(b_n n^c)$ or $b_n \in O(n^{-c} a_n)$, for some small constant $c > 0$.

Probability discretization: Let $\mathbf{R} \stackrel{\text{def}}{=} \{\mathbf{r}_i\}_{i \in [1, \ell]}$ be a finite discretization of the probability space, where $\mathbf{r}_i \in [0, 1]_{\mathbb{R}}$ and $\ell \stackrel{\text{def}}{=} |\mathbf{R}|$. We call $\mathbf{q} \in [0, 1]_{\mathbb{R}}^{\mathcal{D}}$ a *pseudo-distribution* if $\|\mathbf{q}\|_1 \leq 1$ and a *discrete pseudo-distribution* with respect to $\mathbf{R}$ if all its entries are in $\mathbf{R}$ as well. We use $\Delta_{pseudo}^{\mathcal{D}}$ and $\Delta_{\mathbf{R}}^{\mathcal{D}}$ to denote the set of all pseudo-distributions and discrete pseudo-distributions with respect to $\mathbf{R}$ respectively. For all probability terms defined involving distributions $\mathbf{p}$, we extend those definitions to pseudo distributions $\mathbf{q}$ by replacing $\mathbf{p}_x$ with $\mathbf{q}_x$ everywhere. For convenience we refer to $\mathbb{P}(\mathbf{q}, \phi)$ for any pseudo-distribution $\mathbf{q}$ as the "probability" of profile ϕ with respect to $\mathbf{q}$.

2 Results

Here we provide the main results of our work. In our first result we show that the following weaker notion of approximate PML suffices to obtain the desired universal estimator.

Definition 2.1. Given a profile ϕ, we call a distribution $\mathbf{p}' \in \Delta^{\mathcal{D}}$ *β-approximate PML distribution with respect to $\boldsymbol{R}$* if it satisfies the following: $\mathbb{P}(\mathbf{p}', \phi) \geq \beta \cdot \max_{\mathbf{q} \in \Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, \phi\right)$.

In the above definition we are maximizing over $\Delta_{\mathbf{R}}^{\mathcal{D}}$ instead of $\Delta^{\mathcal{D}}$ and the previous known approximate PML definition corresponds to choose the set $\mathbf{R} = [0, 1]_{\mathbb{R}}$. Using our new definition, we show that for a specific choice of the discretization set $\mathbf{R}$, a distribution $\mathbf{p}'$ that is an approximate PML with respect to $\mathbf{R}$ suffices to obtain a universal estimator; this result is formally stated below.

Theorem 2.2 (Competitiveness of an approximate PML w.r.t $\mathbf{R}$)**.** *For symmetric property f suppose there exists an estimator $\hat{f}$ that takes input a profile $\phi \in \Phi^n$ drawn from $\boldsymbol{p} \in \Delta^{\mathcal{D}}$ and satisfies,*

$$\mathbb{P}\left(|f(\boldsymbol{p}) - \hat{f}(\phi)| \geq \epsilon\right) \leq \delta \ .$$

then for $\boldsymbol{R} \stackrel{\text{def}}{=} \{\frac{1}{2n^2}, \ldots, \frac{1}{2n^2}(1+\alpha)^i, \ldots 1\}$ where $\alpha = n^{-1/3}$, a pseudo distribution $\boldsymbol{q}' \in \Delta_{\boldsymbol{R}}^{\mathcal{D}}$ such that $\boldsymbol{q}'/\|\boldsymbol{q}'\|_1$ is an $\exp(-O(n^{1/3} \log n))$-approximate PML with respect to the $\boldsymbol{R}$ satisfies,

$$\mathbb{P}\left(|f(\frac{\boldsymbol{q}'}{\|\boldsymbol{q}'\|_1}) - \hat{f}(\phi)| \geq \epsilon\right) \leq \delta^{1-c} \exp(n^{1/3+c}), \quad \textit{for any constant } c > 0 \ .$$

The proof of the above theorem is implicit in the analysis of [HS20], however we provide a short simpler proof using their continuity lemma (Lemma A.1). Note that the bound on the failure probability we get is asymptotically same as that of exact PML from [HS20], which is known to be tight [Han21]. Furthermore, to achieve such an improved failure probability bound all we need is an approximate PML distributions with respect to $\mathbf{R}$, for some $\mathbf{R}$ which is of small size. Taking advantage of this fact and building upon [CSS19a, ACSS21], we provide a new rounding algorithm that outputs the desired approximate PML distribution with respect to $\mathbf{R}$.

Theorem 2.3 (Computation of an approximate PML w.r.t $\mathbf{R}$)**.** *Given a profile ϕ with k distinct frequencies and $\boldsymbol{R}$, there exists an algorithm that runs in time $O\left(\frac{n}{\min(k, |\boldsymbol{R}|)}\left(\min(|\boldsymbol{R}|, n/k)k^\omega + \min(|\boldsymbol{R}|, k)k^2\right)\right)$, where ω is the matrix multiplication constant and returns a pseudo distribution $\boldsymbol{q}' \in \Delta_{\boldsymbol{R}}^{\mathcal{D}}$ such that,*

$$\mathbb{P}\left(\frac{\boldsymbol{q}'}{\|\boldsymbol{q}'\|_1}, \phi\right) \geq \exp\left(-O(\min(k, |\boldsymbol{R}|) \log n)\right) \cdot \max_{\boldsymbol{q} \in \Delta_{\boldsymbol{R}}^{\mathcal{D}}} \mathbb{P}\left(\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_1}, \phi\right) \ .$$

When $\mathbf{R} = [0, 1]$, our algorithm computes an $\exp(-O(k \log n))$ approximate PML distribution, therefore our result is at least as good as the previous best known approximate PML algorithm due to [ACSS20]. In comparison to [ACSS20], our rounding algorithm is much simpler and practical. We provide a more detailed comparison to it and other approximate PML algorithms in Section 2.

Applications Our main results have several applications which we discuss here. First note that, combining Theorem 2.2 and 2.3 we immediately get an efficient universal estimator with a failure probability at most $\delta^{1-c} \exp(n^{1/3+c})$, for any constant $c > 0$. For many interesting class of symmetric properties the parameter δ above is exponentially small as stated below in the lemma.

Lemma 2.4 (Lemma 2 in [ADOS16], Theorem 3 in [HS20])**.** *For distance to uniformity, entropy, support size and coverage, any* 1*-Lipschitz property and sorted* ℓ_1 *distance there exists an estimator that is sample optimal and the failure probability is at most* $\exp(\epsilon^2 n^{1-\alpha})$ *for any constant* $\alpha > 0$.

Therefore, we get a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/3}$.

Lemma 2.5 (Efficient sample optimal universal estimator)**.** *For* $\epsilon \gg n^{-1/3}$ *and symmetric properties such as, distance to uniformity, entropy, support size and coverage, any* 1*-Lipschitz property and sorted* ℓ_1 *distance there exists an efficient approximate PML based estimator that is sample optimal and has failure probability upper bounded by* $\exp(-n^{1/3})$.

As our work computes an $\exp(-O(k \log n))$ approximate PML, we also efficiently implement the profile entropy [HO20] (defined in Section 1.1) and pseudo PML results [CSS19b, HO19].

Lemma (Informal) *We can efficiently implement the profile entropy and pseudo PML results.*

The efficient version of these results were already known in [ACSS20], however, our algorithm is simpler and has better run time guarantees than [ACSS20].

Tightness of our result Recall [HS20] showed that the failure probability of a (approximate) PML based estimator is upper bounded by $\delta^{1-c} \exp(-n^{1/3+c})$, for any constant $c > 0$. This result further implied a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/3}$ for various symmetric properties including the 1-Lipschitz properties (lemma 2.5). In our work, we design an algorithm that efficiently recover these results and a natural question to ask here is if these results can be improved.

However in a very recent work, [Han21] showed that PML is not sample optimal in estimating every 1-Lipschitz property in the regime $\epsilon \ll n^{-1/3}$. In fact, the analysis in [Han21] holds for any (reasonable) universal plug-in based estimator that outputs a distribution $\hat{\mathbf{p}}$ satisfying,

$$\max_{\mathbf{p} \in \Delta^{\mathcal{D}}} \mathbb{E}\|\mathbf{p} - \hat{\mathbf{p}}\|_1^{\text{sorted}} \leq A(n)\sqrt{k/n}\,,$$

where $A(n) \leq n^{\gamma}$ for every $\gamma > 0$ and $\|\mathbf{p} - \mathbf{q}\|_1^{\text{sorted}} \stackrel{\text{def}}{=} \min_{\text{permutations } \sigma} \|\mathbf{p} - \mathbf{q}_\sigma\|_1$ denotes the sorted ℓ_1 distance between $\mathbf{p}$ and $\mathbf{q}$. In other words, if an estimator is based on a reasonably good estimate of the true distribution $\mathbf{p}$ (in terms of sorted-ℓ_1 distance), then it cannot be sample optimal for every 1-Lipschitz property.

Comparison to earlier approximate PML algorithms All the provable approximate PML algorithms have two key steps: 1) Solve a convex approximation to the PML. 2) Round the fractional optimal solution from step 1 to a valid approximate PML.

The convex approximation to PML was first provided in [CSS19a] and a better analysis for it is shown in [ACSS21]. In particular, [CSS19a] and [ACSS21] showed that an integral optimal solution to step 1 approximates the PML up to accuracy $\exp(-n^{2/3} \log n)$ and $\exp(-\min(k, |\mathbf{R}|) \log n)$ respectively, where k and $|\mathbf{R}|$ are the number of distinct frequencies and distinct probability values respectively. In addition to the loss from convex approximation, the previous algorithms also incurred a loss in the rounding step. The loss in the rounding step for the previous works is bounded by $\exp(-n^{2/3} \log n)$ [CSS19a], $\exp(-\sqrt{n} \log n)$ [ACSS21] and $\exp(-k \log n)$ [ACSS20].

In our work, we show that, there exists a choice of $\mathbf{R}$ that is of a small size ($|\mathbf{R}| \leq n^{1/3}$) and suffices to get the desired universal estimator. As $|\mathbf{R}| \leq n^{1/3}$, we only incur a loss of $\exp(-n^{1/3} \log n)$ in the first step. Furthermore for the second step, we provide a new simpler and a practical rounding algorithm with a better approximation loss of $\exp(-O(\min(k, |\mathbf{R}|) \log n)) \in \exp(-O(n^{-1/3} \log n))$.

Regarding the run times, both [ACSS20] and our approach have run times of the form $\mathcal{T}_{\text{solve}} + \mathcal{T}_{\text{sparsify}} + \mathcal{T}_{\text{round}}$, where the terms correspond to the time required to solve the convex program, sparsify and round a solution respectively. Therefore, in our algorithm we pay the same cost as [ACSS20] for the first two steps but our run time guarantees are superior to theirs in the rounding step. In particular, the run time of [ACSS20] is shown to be a large polynomial and is not practical as their approach requires enumerating all the approximate min cuts in some graph.

Other related work PML was introduced by [OSS+04]. Many heuristic approaches have been proposed to compute approximate PML, such as the EM algorithm in [OSS+04], an algebraic approaches in [ADM+10], Bethe approximation in [Von12] and [Von14], and a dynamic programming approach in [PJW17]. For the broad applicability of PML in property testing and to estimate other symmetric properties please refer to [HO19]. Please refer to [CSS19b, HO19] for a different variant of PML. Also, refer to [HO20] for more details related to profile entropy.

There have been other approaches for designing universal estimators, e.g. [VV11b] based on [ET76], [HJW18] based on local moment matching, and variants of PML by [CSS19b, HO19] that weakly depend on the property. Optimal sample complexities for estimating many symmetric properties were also obtained by constructing property specific estimators, e.g. support [VV11b, WY15], support coverage [OSW16, ZVV+16], entropy [VV11b, WY16, JVHW15], distance to uniformity [VV11a, JHW16], sorted ℓ_1 distance [VV11a, HJW18], Renyi entropy [AOST14, AOST17], KL divergence [BZLV16, HJW16] and others.

Limitations of our work One of the limitations of all the provable approximate PML algorithms [CSS19a, ACSS21, ACSS20] (including ours) is that they require the solution of a convex program that approximates the PML objective and there are no known practical algorithms for this task. All these previous works use the CVX solver which is not practical for large sample instances; note that our results hold for small error regimes which lead to large sample instances. Therefore, designing a practical algorithm to solve the convex program is an important future research direction. As discussed above, local moment matching (LLM) based approach is another universal approach for property estimation. It is unclear which of the two (PML or LLM) can lead to practical algorithms.

3 Convex relaxation to PML

Here we restate the convex program from [CSS19a] that approximates the PML objective. The current best analysis of this convex program is in [ACSS21]. We first describe the notation and later state several results from [CSS19a, ACSS21] that capture the guarantees of the convex program.

Notation: For any matrices $\mathbf{X} \in \mathbb{R}^{a\times c}$ and $\mathbf{Y} \in \mathbb{R}^{b\times c}$, we let $\text{concat}(\mathbf{X}, \mathbf{Y})$ denote the matrix $\mathbf{W} \in \mathbb{R}^{(a+b)\times c}$ that satisfies, $\mathbf{W}_{i,j} = \mathbf{X}_{i,j}$ for all $i \in [a]$ and $j \in [c]$ and $\mathbf{W}_{a+i,j} = \mathbf{Y}_{ij}$ for all $i \in [b]$ and $j \in [c]$. Recall we let $\mathbf{R} \stackrel{\text{def}}{=} \{\mathbf{r}_i\}_{i\in[\ell]}$ be a finite discretization of the probability space, where $\mathbf{r}_i \in [0,1]_{\mathbb{R}}$ and $\ell \stackrel{\text{def}}{=} |\mathbf{R}|$. Let $\mathbf{r} \in [0,1]^{\ell}_{\mathbb{R}}$ be a vector whose i'th element is equal to $\mathbf{r}_i$. The following lemma captures the loss due to the probability discretization.

Lemma 3.1 (Lemma 4.4 in [CSS19a]). *Let $\boldsymbol{R} = \{\frac{1}{2n^2}, \ldots, \frac{1}{2n^2}(1+\alpha)^i, \ldots, 1\}$. For any profile $\phi \in \Phi^n$ and distribution $\boldsymbol{p} \in \Delta^{\mathcal{D}}$, there exists a pseudo distribution $\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}$ that satisfies $\mathbb{P}(\boldsymbol{p}, \phi) \geq \mathbb{P}(\boldsymbol{q}, \phi) \geq \exp(-\alpha n - 6)\,\mathbb{P}(\boldsymbol{p}, \phi)$ and therefore, $\max_{\boldsymbol{p}\in\Delta^{\mathcal{D}}} \mathbb{P}(\boldsymbol{p}, \phi) \geq \max_{\boldsymbol{q}\in\Delta^{\mathcal{D}}_{\boldsymbol{R}}} \mathbb{P}(\boldsymbol{q}, \phi) \geq \exp(-\alpha n - 6) \max_{\boldsymbol{p}\in\Delta^{\mathcal{D}}} \mathbb{P}(\boldsymbol{p}, \phi)$.*

For any probability discretization set $\mathbf{R}$, profile ϕ and pseudo distribution $\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}$, define:

$$\mathbf{Z}^{\phi}_{\mathbf{R}} \stackrel{\text{def}}{=} \left\{ \mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0} \;\middle|\; \mathbf{X}\mathbf{1} \in \mathbb{Z}^{\ell}, [\mathbf{X}^{\top}\mathbf{1}]_j = \phi_j \text{ for all } j \in [1,k] \text{ and } \mathbf{r}^{\top}\mathbf{X}\mathbf{1} \leq 1 \right\}, \tag{1}$$

$$\mathbf{Z}^{\phi,frac}_{\mathbf{R}} \stackrel{\text{def}}{=} \left\{ \mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0} \;\middle|\; [\mathbf{X}^{\top}\mathbf{1}]_j = \phi_j \text{ for all } j \in [1,k] \text{ and } \mathbf{r}^{\top}\mathbf{X}\mathbf{1} \leq 1 \right\}, \tag{2}$$

where the 0'th column corresponds to domain elements with frequency 0 (unseen) and we use $\mathbf{m}_0 \stackrel{\text{def}}{=} 0$. W.l.o.g., we assume $\mathbf{m}_0 < \mathbf{m}_1 < \cdots < \mathbf{m}_k$. Let $\mathbf{C}_{ij} \stackrel{\text{def}}{=} \mathbf{m}_j \log \mathbf{r}_i$ for all $i \in [\ell]$ and $j \in [0,k]$. The objective of the convex program is define as follows: for any $\mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0}$ define,

$$\mathbf{g}(\mathbf{X}) \stackrel{\text{def}}{=} \exp\Big(\sum_{i\in[\ell], j\in[0,k]} [\mathbf{C}_{ij}\mathbf{X}_{ij} - \mathbf{X}_{ij}\log\mathbf{X}_{ij}] + \sum_{i\in[\ell]} [\mathbf{X}\mathbf{1}]_i \log[\mathbf{X}\mathbf{1}]_i \Big). \tag{3}$$

For any $\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}$, the function $\mathbf{g}(\mathbf{X})$ approximates the $\mathbb{P}(\mathbf{q}, \phi)$ term and is stated below.

Lemma 3.2 (Theorem 6.7 and Lemma 6.9 in [ACSS21])**.** *Let $\boldsymbol{R}$ be a probability discretization set. For any profile $\phi \in \Phi^n$ with k distinct frequencies the following statements hold for $\alpha = \min(k, |\boldsymbol{R}|) \log n$: $\exp(-O(\alpha)) \cdot C_\phi \cdot \max_{\boldsymbol{X} \in \boldsymbol{Z}^{\phi}_{\boldsymbol{R}}} \boldsymbol{g}(\boldsymbol{X}) \leq \max_{\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}} \mathbb{P}(\boldsymbol{q}, \phi) \leq \exp(O(\alpha)) \cdot C_\phi \cdot \max_{\boldsymbol{X} \in \boldsymbol{Z}^{\phi}_{\boldsymbol{R}}} \boldsymbol{g}(\boldsymbol{X})$ and $\max_{\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}} \mathbb{P}(\boldsymbol{q}, \phi) \leq \exp(O(\min(k, |\boldsymbol{R}|) \log n)) \cdot C_\phi \cdot \max_{\boldsymbol{X} \in \boldsymbol{Z}^{\phi, frac}_{\boldsymbol{R}}} \boldsymbol{g}(\boldsymbol{X})$, where $C_\phi \stackrel{\text{def}}{=} \frac{n!}{\prod_{j \in [1,k]} (\boldsymbol{m}_j!)^{\phi_j}}$ is a term that only depends on the profile*[2].

The proof of convexity for the function $\mathbf{g}(\mathbf{X})$ and a running time analysis to solve the convex program are provided in [CSS19a]. For any $\mathbf{X} \in \mathbf{Z}^{\phi}_{\mathbf{R}}$, a pseudo-distributions associated with it is defined below.

Definition 3.3. For any $\mathbf{X} \in \mathbf{Z}^{\phi}_{\mathbf{R}}$, the discrete pseudo-distribution $\mathbf{q}_{\mathbf{X}}$ associated with $\mathbf{X}$ and $\mathbf{R}$ is defined as follows: For any arbitrary $[\mathbf{X}\mathbf{1}]_i$ number of domain elements assign probability $\mathbf{r}_i$. Further $\mathbf{p}_{\mathbf{X}} \stackrel{\text{def}}{=} \mathbf{q}_{\mathbf{X}} / \|\mathbf{q}_{\mathbf{X}}\|_1$ is the distribution associated with $\mathbf{X}$ and $\mathbf{R}$.

Note that $\mathbf{q}_{\mathbf{X}}$ is a valid pseudo-distribution because of the third condition in Equation (1) and these pseudo distributions $\mathbf{p}_{\mathbf{X}}$ and $\mathbf{q}_{\mathbf{X}}$ satisfy the following lemma.

Lemma 3.4 (Theorem 6.7 in [ACSS21])**.** *Let $\boldsymbol{R}$ and $\phi \in \Phi^n$ be any probability discretization set and profile respectively. For any $\boldsymbol{X} \in \boldsymbol{Z}^{\phi}_{\boldsymbol{R}}$, the discrete pseudo distribution $\boldsymbol{q}_{\boldsymbol{X}}$ and distribution $\boldsymbol{p}_{\boldsymbol{X}}$ associated with $\boldsymbol{X}$ and $\boldsymbol{R}$ satisfy: $\exp(-O(k \log n)) C_\phi \cdot \boldsymbol{g}(\boldsymbol{X}) \leq \mathbb{P}(\boldsymbol{q}_{\boldsymbol{X}}, \phi) \leq \mathbb{P}(\boldsymbol{p}_{\boldsymbol{X}}, \phi)$.*

4 Approximate PML algorithm

Here we provide a proof sketch of Theorem 2.3 and provide a description of the algorithm that proves it. The description of our approximate PML algorithm is as follows.

Algorithm 1 ApproximatePML($\phi, \mathbf{R}$)

1: $\mathbf{X} = \text{argmax}_{\mathbf{Y} \in \mathbf{Z}^{\phi, frac}_{\mathbf{R}}} \log \mathbf{g}(\mathbf{Y})$.
2: $\mathbf{X}' = \text{Sparsify}(\mathbf{X})$.
3: $(\mathbf{A}, \mathbf{B}) = \text{SwapMatrixRound}(\mathbf{X}')$.
4: $(\mathbf{X}_{\text{final}}, \mathbf{R}_{\text{final}}) = \text{Create}(\mathbf{A}, \mathbf{B}, \mathbf{R})$
5: Let $\mathbf{p}'$ be the distribution with respect to $\mathbf{X}_{\text{final}}$ and $\mathbf{R}_{\text{final}}$ (See Definition 3.3).
6: Return $\mathbf{q} = \text{Discretize}(\mathbf{p}', \phi, \mathbf{R})$

The above algorithm includes several steps and in the following we provide guarantee for each of these steps. We later use these guarantees to prove our final theorem (Theorem 2.3).

Lemma 4.1 ([CSS19a, ACSS21])**.** *Step 1 of the algorithm can be implemented in $\widetilde{O}(|\boldsymbol{R}|\, k^2 + |\boldsymbol{R}|^2\, k)$ time and the maximizer $\boldsymbol{X}$ satisfies: $C_\phi \cdot \boldsymbol{g}(\boldsymbol{X}) \geq \exp(O(-\min(k, |\boldsymbol{R}|) \log n)) \max_{\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}} \mathbb{P}(\boldsymbol{q}, \phi)$.*

Lemma 4.2 (Lemma 4.3 in [ACSS20])**.** *For any $\boldsymbol{X} \in \boldsymbol{Z}^{\phi, frac}_{\boldsymbol{R}}$, the algorithm $\text{Sparsify}(\boldsymbol{X})$ runs in time $\widetilde{O}(|\boldsymbol{R}|\, k^\omega)$ and outputs $\boldsymbol{X}' \in \boldsymbol{Z}^{\phi, frac}_{\boldsymbol{R}}$ such that: $\boldsymbol{g}(\boldsymbol{X}') \geq \boldsymbol{g}(\boldsymbol{X})$ and $\left|\{i \in [\ell] \mid [\boldsymbol{X}' \overrightarrow{1}]_i > 0\}\right| \leq k+1$.*

Please refer to [ACSS20] for the description of the procedure Sparsify. The usage of this procedure is that one can assume $|\mathbf{R}| \leq k+1$ as we can ignore the zero rows of the matrix $\mathbf{X}$. To explain our next step, we need to define a new operation called the swap.

Definition 4.3. Given a matrix $\mathbf{A}$, indices $i_1 < i_2$, $j_1 < j_2$ and a parameter $\epsilon \geq 0$, the operation $\text{swap}(\mathbf{A}, i_1, i_2, j_1, j_2, \epsilon)$ outputs a matrix $\mathbf{A}'$ that satisfies,

$$\mathbf{A}'_{ij} = \begin{cases} \mathbf{A}_{i,j} + \epsilon \text{ for } i = i_1,\ j = j_1 & \mathbf{A}_{i,j} - \epsilon \text{ for } i = i_1,\ j = j_2\ , \\ \mathbf{A}_{i,j} - \epsilon \text{ for } i = i_2,\ j = j_1 & \mathbf{A}_{i,j} + \epsilon \text{ for } i = i_2,\ j = j_2\ , \\ \mathbf{A}_{ij} \text{ otherwise.} \end{cases} \tag{4}$$

[2] The theorem statement in [ACSS21] is only written with an approximation factor of $\exp(O(k \log n))$ instead of $\exp(O(\min(k, |\mathbf{R}|) \log n))$. However their proof provides a stronger approximation factor which is upper bounded by the non-negative rank of the probability matrix, which in turn is upper bounded by the minimum of distinct frequencies and distinct probabilities.

Furthermore, we say $\mathbf{A}'$ is x-swap distance from $\mathbf{A}$, if $\mathbf{A}'$ can be obtained from $\mathbf{A}$ through a sequence of swap operations and the summation of the value ϵ's in these operations is at most x. That is, there is a set of parameters $\{(i_1^{(s)}, i_2^{(s)}, j_1^{(s)}, j_2^{(s)}, \epsilon^{(s)})\}_{s\in[t]}$, where $\sum_{s\in[t]} \epsilon^{(s)} \leq x$, such that $\mathbf{A}^{(s)} = \mathrm{swap}(\mathbf{A}^{(s-1)}, i_1^{(s)}, i_2^{(s)}, j_1^{(s)}, j_2^{(s)}, \epsilon^{(s)})$ for $s \in [t]$, where $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{A}^{(t)} = \mathbf{A}'$.

Recall that our objective $\mathbf{g}(\mathbf{X})$ contains two terms: 1) the linear term $\sum_{i\in[\ell],j\in[0,k]} \mathbf{C}_{ij}\mathbf{X}_{ij}$. 2) the entropy term $\sum_{i\in[\ell]}[\mathbf{X}\overrightarrow{1}]_i \log[\mathbf{X}\overrightarrow{1}]_i - \sum_{i\in[\ell],j\in[0,k]} \mathbf{X}_{ij}\log \mathbf{X}_{ij}$. The swap operation always increases the first term, and in the following lemma we bound the loss due to the second term.

Lemma 4.4. *If $\boldsymbol{A}' \in \mathbb{R}^{\ell\times[0,k]}$ is x-swap distance from $\boldsymbol{A} \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,frac}$, then, $\boldsymbol{A}' \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,frac}$ and $\boldsymbol{g}(\boldsymbol{A}') \geq \exp(-O(x\log n))\,\boldsymbol{g}(\boldsymbol{A})$.*

One of the main contributions of our work is the following lemma, where we repeatedly apply swap operation to recover a matrix $\mathbf{A}$ which exhibits several nice properties as stated below.

Lemma 4.5. *For any matrix $\boldsymbol{A} \in \mathbb{R}^{s\times t}$ $(s \leq t)$ that satisfies $\boldsymbol{A}^\top\overrightarrow{1} \in \mathbb{Z}_{\geq 0}^t$. The algorithm* SwapMatrixRound *runs in $O(s^2t)$ time and returns matrices $\boldsymbol{A}'$ and $\boldsymbol{B}$ such that,*

- *$\boldsymbol{A}'$ is $O(s)$-swap distance from $\boldsymbol{A}$, $\boldsymbol{A}'\overrightarrow{1} = \boldsymbol{A}\overrightarrow{1}$ and $\boldsymbol{A}'^\top\overrightarrow{1} = \boldsymbol{A}^\top\overrightarrow{1}$.*
- *$0 \leq \boldsymbol{B}_{ij} \leq \boldsymbol{A}'_{ij}$ for all $i \in [s]$ and $j \in [t]$, $\boldsymbol{B}\overrightarrow{1} \in \mathbb{Z}_{\geq 0}^s$, $\boldsymbol{B}^\top\overrightarrow{1} \in \mathbb{Z}_{\geq 0}^t$ and $\|\boldsymbol{A}' - \boldsymbol{B}\|_1 \leq O(s)$.*

The above lemma helps us modify our matrix $\mathbf{X}$ to a new matrix $\mathbf{A}$ that we can round using the Create procedure. The guarantees of this procedure are summarized below.

Lemma 4.6 (Lemma 6.13 in [ACSS21])**.** *For any $\boldsymbol{A} \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,frac} \subseteq \mathbb{R}_{\geq 0}^{\ell\times[0,k]}$ and $\boldsymbol{B} \in \mathbb{R}_{\geq 0}^{\ell\times[0,k]}$ such that $\boldsymbol{B} \leq \boldsymbol{A}$, $\boldsymbol{B}\overrightarrow{1} \in \mathbb{Z}^\ell$, $\boldsymbol{B}^\top\overrightarrow{1} \in \mathbb{Z}^{[0,k]}$ and $\|\boldsymbol{A} - \boldsymbol{B}\|_1 \leq t$. The algorithm* Create$(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R})$ *runs in time $O(\ell k)$ and returns a solution $\boldsymbol{A}'$ and a probability discretization set $\boldsymbol{R}'$ such that $|\boldsymbol{R}'| \leq |\boldsymbol{R}| + \min(k+1, t)$, $\boldsymbol{A}' \in \boldsymbol{Z}_{\boldsymbol{R}'}^{\phi}$ and $\boldsymbol{g}(\boldsymbol{A}') \geq \exp\left(-O\left(t\log n\right)\right)\boldsymbol{g}(\boldsymbol{A})$.*

As our final goal is to return a distribution in $\Delta_{\mathbf{R}}^{\mathcal{D}}$. We also need the following discretization lemma.

Lemma 4.7. *The function* Discretize *takes input a distribution $\boldsymbol{p} \in \Delta^{\mathcal{D}}$ with ℓ' distinct probability values, a profile ϕ, a discretization set of the form $\boldsymbol{R} = \{\frac{1}{2n^2}, \ldots, \frac{1}{2n^2}(1+\alpha)^i, \ldots, 1\}$ for some $\alpha \in (0,1)$ and outputs a pseudo distribution $\boldsymbol{q} \in \Delta_{\boldsymbol{R}}^{\mathcal{D}}$ such that,*

$$\mathbb{P}\left(\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_1}, \phi\right) \geq \exp(-O(\min(k, |\boldsymbol{R}|) + \min(k, \ell') + \alpha^2 n)\log n)\mathbb{P}\left(\boldsymbol{p}, \phi\right). \tag{5}$$

In Section 5, we use the guarantees stated above for each step of Algorithm 1 to prove Theorem 2.3. The description of the function Discretize is specified in the proof of Lemma 4.7. We describe the procedure SwapMatrixRound and provide a proof sketch of Lemma 4.5 in Section 4.1.

4.1 Description of SwapMatrixRound and comparison to [ACSS20]

Here we describe the procedure SwapMatrixRound and compare our rounding algorithm to [ACSS20]. Our approximate PML algorithm has four main steps (1-4) and we differ from [ACSS20] in the key Step 3. This step in [ACSS20] invokes a procedure called MatrixRound that takes input a matrix $\mathbf{A} \in \mathbb{R}^{\ell\times[0,k]}$ and outputs a matrix $\mathbf{B} \in \mathbb{R}^{\ell\times[0,k]}$ such that: $\mathbf{B} \leq \mathbf{A}$, $\mathbf{B}\overrightarrow{1} \in \mathbb{Z}_{\geq 0}^\ell$, $\mathbf{B}^\top\overrightarrow{1} \in \mathbb{Z}_{\geq 0}^{[0,k]}$ and $\|\mathbf{A} - \mathbf{B}\|_1 \leq O(\ell + k)$. Such a matrix $\mathbf{B}$ is crucial as the procedure Create uses such a matrix to round fractional row sums of matrix $\mathbf{A}$ to integral values. The error incurred in these two steps is at most $\exp(O(\|\mathbf{A} - \mathbf{B}\|_1 \log n)) \in \exp(O((\ell + k)\log n))$. As the procedure Sparsify allows us to assume $\ell \leq k+1$, we get an $\exp(-k\log n)$ approximate PML using [ACSS20]. However, the setting that we are interested in is when $\ell \ll k$; for instance when $\ell \in O(n^{1/3})$ and $k \in \Theta(\sqrt{n})$. In these settings, we desire an $\exp(-O(\min(\ell, k)\log n)) \in \exp(-O(\ell\log n))$ approximate PML. In order to get such an improved approximation using [ACSS20], we need a matrix $\mathbf{B}$ satisfying the earlier mentioned inequalities along with $\|\mathbf{A} - \mathbf{B}\|_1 \leq O(\min(k, \ell))$. However, such a matrix $\mathbf{B}$ may not exist for arbitrary matrices $\mathbf{A}$ and the best guarantees any algorithm can achieve is $\|\mathbf{A} - \mathbf{B}\|_1 \in O(\ell + k)$.

To overcome this, we introduce a new procedure called $\mathrm{SwapMatrixRound}$ that takes as input, a matrix $\mathbf{A}$ and transforms it to a new matrix $\mathbf{A}'$ that satisfies: $\mathbf{g}(\mathbf{A}') \geq \exp(-O(\min(k,\ell)\log n))\mathbf{g}(\mathbf{A})$. Furthermore, this transformed matrix $\mathbf{A}'$ exhibits a matrix $\mathbf{B}$ that satisfies the guarantees: $\mathbf{B} \leq \mathbf{A}'$, $\mathbf{B}\overrightarrow{1} \in \mathbb{Z}^{\ell}_{\geq 0}$, $\mathbf{B}^{\top}\overrightarrow{1} \in \mathbb{Z}^{k}_{\geq 0}$ and $\|\mathbf{A}' - \mathbf{B}\|_1 \leq O(\ell)$. These matrices $\mathbf{A}'$ and $\mathbf{B}$ are nice as we can invoke the procedure Create, which would output a valid distribution with required guarantees. In the following we provide a description of the algorithm that finds these matrices $\mathbf{A}'$ and $\mathbf{B}$.

Algorithm 2 $\mathrm{SwapMatrixRound}(\mathbf{A})$

Let $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{D}^{(0)} = 0$.
for $r = 1 \ldots \ell$ **do**
  $(\mathbf{Y}, j) = \mathrm{PartialRound}(\mathbf{A}^{(r-1)}, r)$
  $\mathbf{A}^{(r)} = \mathrm{RoundiRow}(\mathbf{Y}, j, r)$.
  $\mathbf{D}^{(r)} = \mathbf{D}^{(r-1)} + \mathbf{Y} - \mathbf{A}^{(r)}$.
end for
Return $\mathbf{A}' = \mathbf{D}^{(\ell)} + \mathbf{A}^{(\ell)}$ and $\mathbf{B} = \mathbf{A}^{(\ell)}$.

Our algorithm includes two main subroutines: $\mathrm{PartialRound}$ and $\mathrm{RoundiRow}$. At each iteration i, the procedure $\mathrm{PartialRound}$ considers row i and modifies it by repeatedly applying the swap operation. This modified row is nice as the procedure $\mathrm{RoundiRow}$ can round this row to have an integral row sum while not affecting the rows in $[i-1]$. By iterating through all rows, we get the required matrices $\mathbf{A}'$ and $\mathbf{B}$ that satisfy the required guarantees. In the remainder, we formally state the guarantees achieved by the procedures $\mathrm{PartialRound}$ and $\mathrm{RoundiRow}$.

Lemma 4.8. *The algorithm* $\mathrm{PartialRound}$ *takes in inputs* $\boldsymbol{X} \in \mathbb{R}^{\ell \times [0,k]}_{\geq 0}$ *and* $i \in [\ell - 1]$ *that satisfies the following,* $[\boldsymbol{X}\overrightarrow{1}]_{i'} \in \mathbb{Z}_{\geq 0}$ *for all* $i' \in [1, i-1]$ *and* $[\boldsymbol{X}^{\top}\overrightarrow{1}]_j \in \mathbb{Z}_{\geq 0}$ *for all* $j \in [0,k]$*, and outputs a matrix* $\boldsymbol{Y} \in \mathbb{R}^{\ell \times [0,k]}_{\geq 0}$ *and an index* j' *such that:*

- $\boldsymbol{Y}$ *is within* 3*-swap distance from* $\boldsymbol{X}$.
- $\boldsymbol{Y}_{ij'} \geq o$ *and* $\sum_{i'=1}^{i-1} \boldsymbol{Y}_{i'j'} + \boldsymbol{Y}_{ij'} - o \in \mathbb{Z}_{\geq 0}$,

where $o = [\boldsymbol{X}\overrightarrow{1}]_i - \lfloor [\boldsymbol{X}\overrightarrow{1}]_i \rfloor$. *Furthermore, the running time of the algorithm is* $O(\ell k)$.

Lemma 4.9. *The algorithm* $\mathrm{RoundiRow}$ *takes in inputs* $\boldsymbol{Y} \in \mathbb{R}^{\ell \times [0,k]}_{\geq 0}$*, an column index* $j \in [0,k]$ *and a row index* $i \in [\ell - 1]$ *such that:* $\boldsymbol{Y}^{\top}\overrightarrow{1} \in \mathbb{Z}^{[0,k]}_{\geq 0}$, $\boldsymbol{Y}_{ij} \geq o$ *and* $\sum_{i'=1}^{i-1} \boldsymbol{Y}_{i'j} + \boldsymbol{Y}_{ij} - o \in \mathbb{Z}_{\geq 0}$, *where* $o = [\boldsymbol{Y}\overrightarrow{1}]_i - \lfloor [\boldsymbol{Y}\overrightarrow{1}]_i \rfloor$. *Outputs a matrix* $\boldsymbol{X} \in \mathbb{R}^{\ell \times [0,k]}_{\geq 0}$ *such that,*

- $\boldsymbol{X} \leq \boldsymbol{Y}$
- $\|\boldsymbol{X} - \boldsymbol{Y}\|_1 \leq 1$
- $[\boldsymbol{X}\overrightarrow{1}]_{i'} = [\boldsymbol{Y}\overrightarrow{1}]_{i'}$ *for all* $i' \in [i-1]$, $[\boldsymbol{X}\overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$
- $\boldsymbol{X}^{\top}\overrightarrow{1} \in \mathbb{Z}^{[0,k]}_{\geq 0}$.

We defer the description of all the missing procedures and the missing proofs to appendix.

5 Proof of Theorem 2.3

Proof of theorem 2.3. We first show the following two inequalities: $\mathbf{X}_{\text{final}} \in \mathbf{Z}^{\phi}_{\mathbf{R}_{\text{final}}}$ and $\mathbf{g}(\mathbf{X}_{\text{final}}) \geq \exp(-O(\min(k, |\mathbf{R}|)\log n))\mathbf{g}(\mathbf{X})$. By Lemma 4.1, the Step 1 of Algorithm 1 returns a solution $\mathbf{X} \in \mathbf{Z}^{\phi,frac}_{\mathbf{R}}$ that satisfies,

$$C_{\phi} \cdot \mathbf{g}(\mathbf{X}) \geq \exp\left(O\left(-\min(k, |\mathbf{R}|)\log n\right)\right) \max_{\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\mathbf{q}, \phi)\,. \tag{6}$$

By Lemma 4.2, the Step 2 of Algorithm 1 takes input $\mathbf{X}$ and outputs $\mathbf{X}'$ such that

$$\mathbf{X}' \in \mathbf{Z}^{\phi,frac}_{\mathbf{R}} \text{ and } \mathbf{g}(\mathbf{X}') \geq \mathbf{g}(\mathbf{X}), \tag{7}$$

and $\left|\{i \in [\ell] \mid [\mathbf{X}'\overrightarrow{1}]_i > 0\}\right| \leq k+1$. As the matrix $\mathbf{X}'$ has at most $k+1$ non-zero rows, without loss of generality we can assume $|\mathbf{R}| \leq k+1$ (by discarding zero rows).

As matrix $\mathbf{X}' \in \mathbf{Z}^{\phi,frac}_{\mathbf{R}}$, we have that $\mathbf{X}'$ has integral column sums and by invoking Lemma 4.5 with parameters $s = |\mathbf{R}|$ and $t = k+1$, we get matrices $\mathbf{A}$ and $\mathbf{B}$ that satisfy guarantees of Lemma 4.5.

As $[\mathbf{A}\overrightarrow{1}]_i = [\mathbf{X}'\overrightarrow{1}]_i$ for all $i \in [\ell]$, $[\mathbf{A}^\top\overrightarrow{1}]_j = [\mathbf{X}'^\top\overrightarrow{1}]_j$ for all $j \in [0,k]$ and $\mathbf{X}' \in \mathbf{Z}_{\mathbf{R}}^{\phi,frac}$, we immediately get that $\mathbf{A} \in \mathbf{Z}_{\mathbf{R}}^{\phi,frac}$. Further note that $\mathbf{A}$ is within $O(|\mathbf{R}|) \in O(\min(|\mathbf{R}|,k))$-swap distance from $\mathbf{X}'$ and by Lemma 4.4 we get that $\mathbf{g}(\mathbf{A}) \geq \exp(-O(\min(|\mathbf{R}|,k)\log n))\mathbf{g}(\mathbf{X}')$. To summarize, we showed the following inequalities,

$$\mathbf{A} \in \mathbf{Z}_{\mathbf{R}}^{\phi,frac} \text{ and } \mathbf{g}(\mathbf{A}) \geq \exp(-O(\min(|\mathbf{R}|,k)\log n))\mathbf{g}(\mathbf{X}') \ . \tag{8}$$

Note that, Lemma 4.5 also outputs a matrix $\mathbf{B}$ that satisfies: $\mathbf{B} \leq \mathbf{A}$, $\mathbf{B}\overrightarrow{1} \in \mathbb{Z}^\ell$, $\mathbf{B}^\top\overrightarrow{1} \in \mathbb{Z}^{[0,k]}$ and $\|\mathbf{A}-\mathbf{B}\|_1 \leq O(\min(|\mathbf{R}|,k))$. These matrices $\mathbf{A}$ and $\mathbf{B}$ satisfy the conditions of Lemma 4.6 with parameter value $t = O(\min(|\mathbf{R}|,k))$, and the procedure Create taken in input matrices $\mathbf{A},\mathbf{B}$ and returns a solution $(\mathbf{X}_{\text{final}}, \mathbf{R}_{\text{final}})$ such that $|\mathbf{R}_{\text{final}}| \leq |\mathbf{R}| + \min(\mathbf{R},k) \leq 2|\mathbf{R}|$ and,

$$\mathbf{X}_{\text{final}} \in \mathbf{Z}_{\mathbf{R}_{\text{final}}}^{\phi} \text{ and } \mathbf{g}(\mathbf{X}_{\text{final}}) \geq \exp(-O(\min(|\mathbf{R}|,k)\log n))\mathbf{g}(\mathbf{A}) \ . \tag{9}$$

As $\mathbf{X}_{\text{final}} \in \mathbf{Z}_{\mathbf{R}_{\text{final}}}^{\phi}$, by definition Definition 3.3 and Lemma 3.2, the distribution $\mathbf{p}'$ satisfies,

$$\begin{aligned}
\mathbb{P}(\mathbf{p}',\phi) &\geq \exp(-O(\min(k,|\mathbf{R}_{\text{final}}|)\log n))C_\phi\mathbf{g}(\mathbf{X}_{\text{final}}) \geq \exp(-O(\min(k,|\mathbf{R}|))\log n))C_\phi\mathbf{g}(\mathbf{A}) \ , \\
&\geq \exp(-O(\min(k,|\mathbf{R}|)\log n))C_\phi\mathbf{g}(\mathbf{X}') \geq \exp(-O(\min(k,|\mathbf{R}|)\log n))C_\phi\mathbf{g}(\mathbf{X}) \ , \\
&\geq \exp(-O(\min(k,|\mathbf{R}|)\log n)) \max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\mathbf{q},\phi) \ .
\end{aligned}$$

In the second inequality we used Equation (9) and $|\mathbf{R}_{\text{final}}| \leq 2|\mathbf{R}|$. In the third, fourth and fifth inequalities, we used Equation (8), Equation (7) and Equation (6) respectively.

Recall we need a distribution that approximately maximizes $\max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi)$ instead of just $\max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\mathbf{q},\phi)$. In the remainder of the proof we provide a procedure to output such a distribution.

For any constant $c > 0$, let $c\cdot\mathbf{R} \stackrel{\text{def}}{=} \{c\cdot\mathbf{r}_i \mid \mathbf{r}_i \in \mathbf{R}\}$. For any $\mathbf{q} \in \Delta_{\mathbf{R}}^{\mathcal{D}}$, as $\|\mathbf{q}\|_1$ satisfies: $\mathbf{r}_{min} \leq \|\mathbf{q}\|_1 \leq 1$, we get that,

$$\max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) = \max_{c\in[1,1/\mathbf{r}_{min}]_{\mathbb{R}}} \max_{\mathbf{q}\in\Delta_{c\cdot\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\mathbf{q},\phi) \ . \tag{10}$$

The above expression holds as the maximizer $\mathbf{q}^*$ of the left hand side satisfies: $\mathbf{q}^* \in \Delta_{(1/\|\mathbf{q}^*\|_1)\cdot\mathbf{R}}^{\mathcal{D}}$. Define $C \stackrel{\text{def}}{=} \{(1+\beta)^i\}_{i\in[a]}$ for some $\beta \in o(1)$, where $a \in O(\frac{1}{\beta}\log(1/\mathbf{r}_{min}))$ is such that $\mathbf{r}_{min}(1+\beta)^a = 1$. For any constant $c \in [1,1/\mathbf{r}_{min}]_{\mathbb{R}}$, note that there exists a constant $c' \in C$ such that $c(1-\beta) \leq c' \leq c$. Furthermore, for any distribution $\mathbf{q} \in \Delta_{\mathbf{R}}^{\mathcal{D}}$ with $\|\mathbf{q}\|_1 = 1/c$, note that the distribution $\mathbf{q}' = c'\mathbf{q} \in \Delta_{c'\cdot\mathbf{R}}^{\mathcal{D}}$ and satisfies: $\mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) = \mathbb{P}(c\cdot\mathbf{q},\phi) = \mathbb{P}(\frac{c}{c'}\mathbf{q}',\phi) = \left(\frac{c}{c'}\right)^n\mathbb{P}(\mathbf{q}',\phi)$. Therefore we get that, $\mathbb{P}(\mathbf{q}',\phi) = \left(\frac{c'}{c}\right)^n\mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) \geq (1-\beta)^n\mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) \geq \exp(-2\beta n)\mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi)$. Combining this analysis with Equation (10) we get that,

$$\max_{c\in C} \max_{\mathbf{q}\in\Delta_{c\cdot\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\mathbf{q},\phi) \geq \exp(-2\beta n) \max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi). \tag{11}$$

For each $c > 0$ as $|\mathbf{R}| = |c\cdot\mathbf{R}|$, our algorithm (Algorithm 1) returns a distribution $\mathbf{p}_c$ that satisfies,

$$\mathbb{P}(\mathbf{p}_c,\phi) \geq \exp(-O(\min(k,|\mathbf{R}|)\log n)) \max_{\mathbf{q}\in\Delta_{c\cdot\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\mathbf{q},\phi) \ .$$

Let $\mathbf{p}^*$ be the distribution that achieves the maximum objective value to our convex program among the distributions $\{\mathbf{p}_c\}_{c\in C}$. Then note that $\mathbf{p}^*$ satisfies: $\mathbb{P}(\mathbf{p}^*,\phi) \geq \exp(-O(\min(k,|\mathbf{R}|)\log n) - 2\beta n)\max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi)$. Substituting $\beta = \frac{\min(k,|\mathbf{R}|)}{n}$ in the previous expression, we get,

$$\mathbb{P}(\mathbf{p}^*,\phi) \geq \exp(-O(\min(k,|\mathbf{R}|)\log n)) \max_{\mathbf{q}\in\Delta_{\mathbf{R}}^{\mathcal{D}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) \ .$$

As each of our distributions $\mathbf{p}_c$ (including $\mathbf{p}^*$) have the number of distinct probability values upper bounded by $2|\mathbf{R}|$, by applying Lemma 4.7, we get a pseudo distribution $\mathbf{q} \in \Delta_{\mathbf{R}}^{\mathcal{D}}$ with the desired guarantees. The final run time of our algorithm is $O(|C|\mathcal{T}_1) \in O(\frac{n}{\min(k,|\mathbf{R}|)}\cdot\mathcal{T}_1)$, where $\mathcal{T}_1$ is the time to implement Algorithm 1. Further note that by Lemma 3.1, without loss of generality we can assume $|\mathbf{R}| \leq n/k$. As all the steps of Algorithm 1 are polynomial in n, our final running time follows from the run times of each step and we conclude the proof. □

References

[ACSS20] Nima Anari, Moses Charikar, Kirankumar Shiragur, and Aaron Sidford. Instance based approximations to profile maximum likelihood. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[ACSS21] Nima Anari, Moses Charikar, Kirankumar Shiragur, and Aaron Sidford. The bethe and sinkhorn permanents of low rank matrices and implications for profile maximum likelihood. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 93–158. PMLR, 2021.

[ADM+10] J. Acharya, H. Das, H. Mohimani, A. Orlitsky, and S. Pan. Exact calculation of pattern probabilities. In *2010 IEEE International Symposium on Information Theory*, pages 1498–1502, June 2010.

[ADOS16] Jayadev Acharya, Hirakendu Das, Alon Orlitsky, and Ananda Theertha Suresh. A unified maximum likelihood approach for optimal distribution property estimation. *CoRR*, abs/1611.02960, 2016.

[AOST14] Jayadev Acharya, Alon Orlitsky, Ananda Theertha Suresh, and Himanshu Tyagi. The complexity of estimating rényi entropy. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1855–1869, 2014.

[AOST17] Jayadev Acharya, Alon Orlitsky, Ananda Theertha Suresh, and Himanshu Tyagi. Estimating renyi entropy of discrete distributions. *IEEE Trans. Inf. Theor.*, 63(1):38–56, January 2017.

[BF93] John Bunge and Michael Fitzpatrick. Estimating the number of species: a review. *Journal of the American Statistical Association*, 88(421):364–373, 1993.

[BZLV16] Y. Bu, S. Zou, Y. Liang, and V. V. Veeravalli. Estimation of kl divergence between large-alphabet distributions. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1118–1122, July 2016.

[CCG+12] Robert K Colwell, Anne Chao, Nicholas J Gotelli, Shang-Yi Lin, Chang Xuan Mao, Robin L Chazdon, and John T Longino. Models and estimators linking individual-based and sample-based rarefaction, extrapolation and comparison of assemblages. *Journal of plant ecology*, 5(1):3–21, 2012.

[Cha84] A Chao. Nonparametric estimation of the number of classes in a population. scandinavianjournal of statistics11, 265-270. *Chao26511Scandinavian Journal of Statistics1984*, 1984.

[CSS19a] Moses Charikar, Kirankumar Shiragur, and Aaron Sidford. Efficient profile maximum likelihood for universal symmetric property estimation. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 780–791, New York, NY, USA, 2019. ACM.

[CSS19b] Moses Charikar, Kirankumar Shiragur, and Aaron Sidford. A general framework for symmetric property estimation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12447–12457. Curran Associates, Inc., 2019.

[DS13] Timothy Daley and Andrew D Smith. Predicting the molecular complexity of sequencing libraries. *Nature methods*, 10(4):325, 2013.

[ET76] Bradley Efron and Ronald Thisted. Estimating the number of unseen species: How many words did shakespeare know? *Biometrika*, 63(3):435–447, 1976.

[Für05] Johannes Fürnkranz. Web mining. In *Data mining and knowledge discovery handbook*, pages 899–920. Springer, 2005.

[GTPB07] Zhan Gao, Chi-hong Tseng, Zhiheng Pei, and Martin J Blaser. Molecular analysis of human forearm superficial skin bacterial biota. *Proceedings of the National Academy of Sciences*, 104(8):2927–2932, 2007.

[Han21] Yanjun Han. On the high accuracy limitation of adaptive property estimation. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 910–918. PMLR, 13–15 Apr 2021.

[HHRB01] Jennifer B Hughes, Jessica J Hellmann, Taylor H Ricketts, and Brendan JM Bohannan. Counting the uncountable: statistical approaches to estimating microbial diversity. *Appl. Environ. Microbiol.*, 67(10):4399–4406, 2001.

[HJW16] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax estimation of KL divergence between discrete distributions. *CoRR*, abs/1605.09124, 2016.

[HJW18] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Local moment matching: A unified methodology for symmetric functional estimation and distribution estimation under wasserstein distance. *arXiv preprint arXiv:1802.08405*, 2018.

[HO19] Yi Hao and Alon Orlitsky. The Broad Optimality of Profile Maximum Likelihood. *arXiv e-prints*, page arXiv:1906.03794, Jun 2019.

[HO20] Yi Hao and Alon Orlitsky. Profile entropy: A fundamental measure for the learnability and compressibility of discrete distributions, 2020.

[HS20] Yanjun Han and Kirankumar Shiragur. The optimality of profile maximum likelihood in estimating sorted discrete distributions, 2020.

[JHW16] J. Jiao, Y. Han, and T. Weissman. Minimax estimation of the l1 distance. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 750–754, July 2016.

[JVHW15] J. Jiao, K. Venkat, Y. Han, and T. Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, May 2015.

[KLR99] Ian Kroes, Paul W Lepp, and David A Relman. Bacterial diversity within the human subgingival crevice. *Proceedings of the National Academy of Sciences*, 96(25):14547–14552, 1999.

[OSS+04] A. Orlitsky, S. Sajama, N. P. Santhanam, K. Viswanathan, and Junan Zhang. Algorithms for modeling distributions over large alphabets. In *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, pages 304–304, 2004.

[OSW16] Alon Orlitsky, Ananda Theertha Suresh, and Yihong Wu. Optimal prediction of the number of unseen species. *Proceedings of the National Academy of Sciences*, 113(47):13283–13288, 2016.

[PBG+01] Bruce J Paster, Susan K Boches, Jamie L Galvin, Rebecca E Ericson, Carol N Lau, Valerie A Levanos, Ashish Sahasrabudhe, and Floyd E Dewhirst. Bacterial diversity in human subgingival plaque. *Journal of bacteriology*, 183(12):3770–3783, 2001.

[PJW17] D. S. Pavlichin, J. Jiao, and T. Weissman. Approximate Profile Maximum Likelihood. *ArXiv e-prints*, December 2017.

[RCS+09] Harlan S Robins, Paulo V Campregher, Santosh K Srivastava, Abigail Wacher, Cameron J Turtle, Orsalem Kahsai, Stanley R Riddell, Edus H Warren, and Christopher S Carlson. Comprehensive assessment of t-cell receptor β-chain diversity in $\alpha\beta$ t cells. *Blood*, 114(19):4099–4107, 2009.

[TE87] Ronald Thisted and Bradley Efron. Did shakespeare write a newly-discovered poem? *Biometrika*, 74(3):445–455, 1987.

[Von12] Pascal O. Vontobel. The bethe approximation of the pattern maximum likelihood distribution. pages 2012–2016, 07 2012.

[Von14] P. O. Vontobel. The bethe and sinkhorn approximations of the pattern maximum likelihood estimate and their connections to the valiant-valiant estimate. In *2014 Information Theory and Applications Workshop (ITA)*, pages 1–10, Feb 2014.

[VV11a] G. Valiant and P. Valiant. The power of linear estimators. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 403–412, Oct 2011.

[VV11b] Gregory Valiant and Paul Valiant. Estimating the unseen: An n/log(n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 685–694, New York, NY, USA, 2011. ACM.

[WY15] Y. Wu and P. Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *ArXiv e-prints*, April 2015.

[WY16] Y. Wu and P. Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, June 2016.

[ZVV+16] James Zou, Gregory Valiant, Paul Valiant, Konrad Karczewski, Siu On Chan, Kaitlin Samocha, Monkol Lek, Shamil Sunyaev, Mark Daly, and Daniel G. MacArthur. Quantifying unobserved protein-coding variants in human populations provides a roadmap for large-scale sequencing projects. *Nature Communications*, 7:13293 EP–, Oct 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 2
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]