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

Moses Charikar
Stanford University
moses@cs.stanford.edu

Zhihao Jiang
Stanford University
faebdc@stanford.edu

Kirankumar Shiragur
Stanford University
shiragur@stanford.edu

Aaron Sidford
Stanford University
sidford@stanford.edu

Abstract

We provide an efficient unified plug-in approach for estimating symmetric properties of distributions given n independent samples. Our estimator is based on profile-maximum-likelihood (PML) and is sample optimal for estimating various symmetric properties when the estimation error $\epsilon \gg n^{-1/3}$. This result improves upon the previous best accuracy threshold of $\epsilon \gg n^{-1/4}$ achievable by polynomial time computable PML-based universal estimators [ACSS21, ACSS20]. Our estimator reaches a theoretical limit for universal symmetric property estimation as [Han21] shows that a broad class of universal estimators (containing many well known approaches including ours) cannot be sample optimal for every 1-Lipschitz property when $\epsilon \ll n^{-1/3}$.

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. Further, 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 been studied extensively 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 [ADOS17] 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, [ADOS17] showed that given $y_1, ..., y_n$ 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 $\delta \exp(3\sqrt{n})$. As the failure probability δ for many estimators for well-known properties (e.g. support size and coverage, entropy, and distance to uniformity) is roughly $\exp(-\epsilon^2 n)$, this result implied a sample optimal unified approach for estimating these properties when the estimation error $\epsilon \gg n^{-1/4}$.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

This result of [ADOS17] laid the groundwork for a line of work on the study of computational and statistical aspects of PML-based approaches to symmetric property estimation. For example, follow up work of [HS21] improved the analysis of [ADOS17] and showed that the failure probability of PML is at most $\delta^{1-c}\exp(-n^{1/3+c})$, for any constant $c > 0$ and therefore it is sample optimal in the regime $\epsilon \gg n^{-1/3}$. The condition $\epsilon \gg n^{-1/3}$ on the optimality of PML is tight [Han21], in the sense that, PML is known to be not sample optimal in the regime $\epsilon \ll n^{-1/3}$. In fact, no estimator (that obeys some mild conditions), is sample optimal for estimating all symmetric properties in the regime $\epsilon \ll n^{-1/3}$; see Section 2 after Theorem 2.6 for details.

We also remark that the statistical guarantees in [ADOS17, HS21] hold for any β-approximate PML[1] for suitable values of β. In particular, [HS21] showed that any β-approximate PML for $\beta > \exp(-n^{1-c'})$ and any constant $c' > 0$, has a failure probability of $\delta^{1-c}\exp(n^{1/3+c} + n^{1-c'})$ for any constant $c > 0$. These results further imply a sample optimal estimator in the regime $\epsilon \gg n^{-\min(1/3, c'/2)}$ for properties with failure probability less than $\exp(-\epsilon^2 n)$. Note that better approximation leads to a larger range of ϵ for which the estimator is sample optimal.

Regarding computational aspects of PML, [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 efficiently compute PML to higher accuracy of $\exp(-\sqrt{n}\log n)$ thereby achieving a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/4}$. The current best polynomial time approximate PML algorithm by [ACSS20] achieves an accuracy of $\exp(-k\log n)$, where k is the number of distinct observed frequencies. Although this result achieves better instance based statistical guarantees, in the worst case it still only implies a sample optimal universal estimator in the regime $\epsilon \gg n^{-1/4}$.

In light of these results, a key open problem is to close the gap between the regimes $\epsilon \gg n^{-1/3}$ and $\epsilon \gg n^{-1/4}$, where the former is the regime in which PML based estimators are statistically optimal and the later is the regime where efficient PML based estimators exist. In this work we ask:

Is there an efficient approximate PML-based estimator that is sample optimal 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 $\delta^{1-c}\exp(n^{1/3+c} + n^{1-c'})$, and consequently is sample optimal in the regime $\epsilon \gg n^{-1/3}$. As remarked, this result is tight in the sense that PML and a broad class of estimators are known to be not optimal in the regime $\epsilon \ll n^{-1/3}$.

To obtain this result we depart slightly from the previous approaches in [ADOS17, CSS19a, ACSS21]. Rather than directly compute an approximate PML distribution we compute a weaker notion of approximation which we show suffices to get us the desired universal estimator. We propose a notion of a *β-weak approximate PML distribution* inspired by [HS21] 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 an $\exp(-n^{1/3}\log n)$-weak approximate PML distribution. Our paper can be viewed as an efficient algorithmic instantiation of [HS21].

Ultimately, our algorithms use the convex relaxation presented in [CSS19a, ACSS21] and provide a new rounding algorithm. We differ from the previous best $\exp(-k\log n)$ approximate PML algorithm [ACSS20] only in the matrix rounding procedure which controls the approximation guarantee. At a high level, the approximation 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 an approximation exponential in the smaller dimension, which may be 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 a better approximation guarantee than [ACSS20], our algorithm also exhibits better run times (see Section 2).

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

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

1.1 Preliminaries

General notation: For matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{s\times t}$, we use $\mathbf{A} \leq \mathbf{B}$ to denote that $\mathbf{A}_{ij} \leq \mathbf{B}_{ij}$ for all $i \in [s]$ and $j \in [t]$. We let $[a, b]$ and $[a, b]_{\mathbb{R}}$ denote the interval $\geq a$ and $\leq b$ of integers and reals respectively. 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 $c > 0$.

Throughout this paper, we assume we receive a sequence of n independent samples from a distribution $\mathbf{p} \in \Delta^{\mathcal{D}}$, where $\Delta^{\mathcal{D}} \stackrel{\text{def}}{=} \{\mathbf{q} \in [0,1]^{\mathcal{D}}_{\mathbb{R}} | \left\|q\right\|_1 = 1\}$ is the set of all discrete distributions supported on domain $\mathcal{D}$. Let $\mathcal{D}^n$ be the set of all length n sequences of elements of $\mathcal{D}$ and for $y^n \in \mathcal{D}^n$ let 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 for distribution $\mathbf{p}$ are $\mathbb{P}(\mathbf{p}, y^n) = \prod_{x\in\mathcal{D}} \mathbf{p}_x^{\mathbf{f}(y^n,x)}$ and $\mathbb{P}(\mathbf{p}, \phi) = \sum_{\{y^n\in\mathcal{D}^n \mid \Phi(y^n)=\phi\}} \mathbb{P}(\mathbf{p}, y^n)$.

Profile maximum likelihood: 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 $\mathbf{p}^\beta_\phi$ is a β-*approximate PML* distribution if $\mathbb{P}(\mathbf{p}^\beta_\phi, \phi) \geq \beta \cdot \mathbb{P}(\mathbf{p}_\phi, \phi)$. For a distribution $\mathbf{p}$ and a length n, let $\mathbf{X}$ be a random variable that takes value $\phi \in \Phi^n$ with probability $\mathbb{P}(\mathbf{p}, \phi)$. We call $H(\mathbf{X})$ (entropy of $\mathbf{X}$) the *profile entropy* with respect to $(\mathbf{p}, n)$ and denote it by $H(\Phi^n, \mathbf{p})$.

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 $\mathbf{q} \in [0,1]^{\mathcal{D}}_{\mathbb{R}}$ 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 $\Delta^{\mathcal{D}}_{\text{pseudo}}$ and $\Delta^{\mathcal{D}}_{\mathbf{R}}$ to denote the set of all pseudo-distributions and discrete pseudo-distributions with respect to $\mathbf{R}$ respectively. In our work, we use the following most commonly used [CSS19a, ACSS21, ACSS20] probability discretization set. For any $\alpha > 0$,

$$\mathbf{R}_{n,\alpha} \stackrel{\text{def}}{=} \{1\} \cup \left\{ \frac{1}{2n^2}(1 + n^{-\alpha})^i \mid \text{for all } i \in \mathbb{Z}_{\geq 0} \text{ such that } \frac{1}{2n^2}(1 + n^{-\alpha})^i \leq 1 \right\} . \tag{1}$$

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.

See Appendix A for the definition of an estimator and optimal sample complexity.

2 Results

Here we provide our main results. In our first result (Theorem 2.2), we show that a weaker notion of approximate PML suffices to obtain the desired universal estimator. Later we show that these weaker approximate PML distributions can be efficiently computed (Theorem 2.3).

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 $\mathbb{P}(\mathbf{p}', \phi) \geq \beta \cdot \max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, \phi\right)$.

The above definition generalizes β-approximate PML distributions which is simply the special case when $\mathbf{R} = [0,1]_{\mathbb{R}}$. Using our new definition, we show that for a specific choice of the discretization set $\mathbf{R}_{n,1/3}$, a distribution $\mathbf{p}'$ that is an approximate PML with respect to $\mathbf{R}_{n,1/3}$ 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 $\widehat{f}$ that takes input a profile $\phi \in \Phi^n$ drawn from $\boldsymbol{p} \in \Delta^{\mathcal{D}}$ and satisfies,*

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

then for $\boldsymbol{R} = \boldsymbol{R}_{n,1/3}$ (See Equation (1)*), a discrete pseudo distribution $\boldsymbol{q}' \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}$ such that $\boldsymbol{q}'/\|\boldsymbol{q}'\|_1$ is an $\exp(-O(|\boldsymbol{R}|\log n))$-approximate PML distribution with respect to the $\boldsymbol{R}$ satisfies,*

$$\mathbb{P}\left(\left|f\left(\frac{\boldsymbol{q}'}{\|\boldsymbol{q}'\|_1}\right) - f(\boldsymbol{p})\right| \geq 2\epsilon\right) \leq \delta^{1-c}\exp(O(n^{1/3+c})), \quad \textit{for any constant } c > 0\,. \tag{2}$$

The proof of the above theorem is implicit in the analysis of [HS21], however we provide a short simpler proof using their continuity lemma (Lemma B.1). Note that the bound on the failure probability we get is the same asymptotically as that of exact PML from [HS21], which is known to be tight [Han21]. Furthermore, to achieve such an improved failure probability bound all we need is an approximate PML distribution 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 $\mathbf{R}$)**.** *We provide an algorithm that given a probability discretization set $\boldsymbol{R} = \boldsymbol{R}_{n,\alpha}$ for $\alpha > 0$ (See Equation* (1)*) and a profile ϕ with k distinct frequencies, runs in time $\widetilde{O}\left(|\boldsymbol{R}| + \frac{n}{\min(k,|\boldsymbol{R}|)}\left(\min(|\boldsymbol{R}|, n/k)k^{\omega} + \min(|\boldsymbol{R}|, k)k^2\right)\right)$, where $\omega < 2.373$ is the current matrix multiplication constant [Wil12, Gal14, AW21] and returns a pseudo distribution $\boldsymbol{q}' \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}$ such that,*

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

When $\mathbf{R} = \mathbf{R}_{n,1}$, 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 simpler and we suspect, more practical. We provide a more detailed comparison to it later in this section.

Applications: Our main results have several applications which we discuss here. First note that, combining Theorem 2.2 and 2.3 immediately yields the following corollary.

Corollary 2.4 (Efficient unified estimator)**.** *Given a profile $\phi \in \Phi^n$ with k distinct frequencies, we can compute an approximate PML distribution $\boldsymbol{q}'$ that satisfies Equation* (2) *in Theorem 2.2 in time $\widetilde{O}\left(\frac{n}{\min(k,n^{1/3})}\left(\min(n^{1/3}, n/k)k^{\omega} + \min(n^{1/3}, k)k^2\right)\right)$.*

For many symmetric properties the failure probability is exponentially small as stated below.

Lemma 2.5 (Lemma 2 in [ADOS17], Theorem 3 in [HS21])**.** *For distance to uniformity, entropy, support size and coverage, 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 $\alpha > 0$.*

The above result combined with Corollary 2.4, immediately yields the following theorem.

Theorem 2.6 (Efficient sample optimal unified estimator)**.** *There exists an efficient approximate PML-based estimator that for $\epsilon \gg n^{-1/3}$ and symmetric properties such as, distance to uniformity, entropy, support size and coverage, and sorted ℓ_1 distance achieves optimal sample complexity and has failure probability upper bounded by $\exp(-n^{1/3})$.*

As our work computes an $\exp(-O(k\log n))$ approximate PML, we recover efficient version of Lemma 2.3 and Theorem 2.4 from [ACSS20]. The first result uses $\exp(-O(k\log n))$ approximate PML algorithm to efficiently implement an estimator that has better statistical guarantees based on profile entropy [HO20] (See Section 1.1). The second result provides an efficient implementation of the PseudoPML estimators [CSS19b, HO19]. Please refer to the respective papers for further details.

Tightness of our result: Recall that [HS21] showed that the failure probability of an (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 (Theorem 2.6). In our work, we efficiently recover these results and a natural question to ask here is if these results can be improved.

As remarked earlier, [Han21] showed that the condition for optimality of PML ($\epsilon \gg n^{-1/3}$) is in some sense tight. More formally, they showed that PML is not sample optimal in estimating every

1-Lipschitz property in the regime $\epsilon \ll n^{-1/3}$. In fact, the results in [Han21] hold more broadly for any 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. Furthermore, many well-known universal estimators including PML and LLM [HJW18] indeed provide a reasonably good estimate of the true distribution and therefore cannot be sample optimal in the regime $\epsilon \ll n^{-1/3}$. Please refer to [Han21] for further details.

Comparison to approximate PML algorithms: All prior provable approximate PML algorithms [CSS19a, ACSS21, ACSS20] have two key steps: (Step 1) solve a convex approximation to the PML and (Step 2) round the (fractional) solution to a valid approximate PML distribution.

A 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 (Step 2). 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 $\mathbf{R}$ (=$\mathbf{R}_{n,1/3}$) that is of small size ($|\mathbf{R}| \leq n^{1/3}$) and suffices to get the desired universal estimator. As $|\mathbf{R}| \leq n^{1/3}$, our approach only incur a loss of $\exp(-\min(k, |\mathbf{R}|)\log n) \in \exp(-n^{1/3}\log n)$ in the convex approximation step (Step 1). Furthermore for the rounding step (Step 2), 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 ours 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. 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 as a large polynomial and perhaps not practical as their approach requires enumerating all the approximate min cuts. In contrast, our algorithm has a run time that is subquadratic.

Other related work PML was introduced by [OSS$^+$04]. Many heuristic approaches have been proposed to compute approximate PML, such as the EM algorithm in [OSS$^+$04], an algebraic approaches in [ADM$^+$10], Bethe approximation in [Von12] and [Von14], and a dynamic programming approach in [PJW17]. For the broad applicability of PML in property testing and to estimate other symmetric properties please refer to [HO19]. Please refer to [HO20] for details related to profile entropy. Other approaches for designing universal estimators are: [VV11b] based on [ET76], [HJW18] based on local moment matching, and variants of PML by [CSS19b, HO19] that weakly depend on the target property that we wish to estimate. Optimal sample complexities for estimating many symmetric properties were also obtained by constructing property specific estimators, e.g. support [VV11b, WY15], support coverage [OSW16, ZVV$^+$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 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 such 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 $\mathrm{concat}(\mathbf{X}, \mathbf{Y})$ 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 $\mathbf{r}_i$.

Lemma 3.1 (Lemma 4.4 in [CSS19a]). *Let $\boldsymbol{R} = \boldsymbol{R}_{n,\alpha}$ for some $\alpha > 0$. 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:

$$\mathbf{Z}^{\phi}_{\mathbf{R}} \stackrel{\text{def}}{=} \left\{ \mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0} \;\middle|\; \mathbf{X}\mathbf{1} \in \mathbb{Z}^{\ell}, [\mathbf{X}^{\top}\mathbf{1}]_j = \phi_j \text{ for all } j \in [1,k] \text{ and } \mathbf{r}^{\top}\mathbf{X}\mathbf{1} \leq 1 \right\}, \tag{3}$$

$$\mathbf{Z}^{\phi,\mathrm{frac}}_{\mathbf{R}} \stackrel{\text{def}}{=} \left\{ \mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0} \;\middle|\; [\mathbf{X}^{\top}\mathbf{1}]_j = \phi_j \text{ for all } j \in [1,k] \text{ and } \mathbf{r}^{\top}\mathbf{X}\mathbf{1} \leq 1 \right\}. \tag{4}$$

The j'th column corresponds to frequency m_j and we use $\mathbf{m}_0 \stackrel{\text{def}}{=} 0$ to capture the unseen elements. Without loss of generality, 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 optimization problem is follows: for any $\mathbf{X} \in \mathbb{R}^{\ell\times[0,k]}_{\geq 0}$ define,

$$\mathbf{g}(\mathbf{X}) \stackrel{\text{def}}{=} \exp\Big(\sum_{i\in[\ell], j\in[0,k]} [\mathbf{C}_{ij}\mathbf{X}_{ij} - \mathbf{X}_{ij}\log \mathbf{X}_{ij}] + \sum_{i\in[\ell]} [\mathbf{X}\mathbf{1}]_i \log [\mathbf{X}\mathbf{1}]_i \Big)\ . \tag{5}$$

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 $\max_{\boldsymbol{q}\in\Delta^{\mathcal{D}}_{\boldsymbol{R}}} \mathbb{P}(\boldsymbol{q}, \phi) \leq \exp(O(\min(k, |\boldsymbol{R}|)\log n)) \cdot C_{\phi} \cdot \max_{\boldsymbol{X}\in\boldsymbol{Z}^{\phi,\mathrm{frac}}_{\boldsymbol{R}}} \boldsymbol{g}(\boldsymbol{X})$, 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 concavity 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 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 (3) 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 a probability discretization set and a profile with k distinct frequencies. 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: $\exp(-O(k\log n))\, C_{\phi} \cdot \boldsymbol{g}(\boldsymbol{X}) \leq \mathbb{P}(\boldsymbol{q}_{\boldsymbol{X}}, \phi) \leq \mathbb{P}(\boldsymbol{p}_{\boldsymbol{X}}, \phi)$.*

[2] The theorem statement in [ACSS21] is only written with an approximation factor of $\exp(O(k\log n))$. 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. Therefore the theorem statement in [ACSS21] holds with a much stronger approximation guarantee of $\exp(O(\min(k, |\mathbf{R}|)\log n))$.

4 Approximate PML algorithm

Here we provide a proof sketch of Theorem 2.3 and provide a rounding algorithm that proves it. Our rounding algorithm takes as input a matrix $\mathbf{X} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}}$ which may have fractional row sums and round it to integral values. This new rounded matrix $\mathbf{X}_{\mathrm{final}}$ corresponds to our approximate PML distribution (See Definition 3.3). The description of our algorithm is as follows.

Algorithm 1 ApproximatePML($\phi, \mathbf{R} = \mathbf{R}_{n,\alpha}$)

1: Let $\mathbf{X}$ be any solution that satisfies, $\log \mathbf{g}(\mathbf{X}) \geq \max_{\mathbf{Y} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}}} \log \mathbf{g}(\mathbf{Y}) - O\left(\min(k, |\mathbf{R}|) \log n\right)$.
2: $\mathbf{X}' = \mathrm{sparsify}(\mathbf{X})$.
3: $(\mathbf{A}, \mathbf{B}) = \mathrm{swapmatrixround}(\mathbf{X}')$.
4: $(\mathbf{X}_{\mathrm{final}}, \mathbf{R}_{\mathrm{final}}) = \mathrm{create}(\mathbf{A}, \mathbf{B}, \mathbf{R})$
5: Let $\mathbf{p}'$ be the distribution with respect to $\mathbf{X}_{\mathrm{final}}$ and $\mathbf{R}_{\mathrm{final}}$ (See Definition 3.3).
6: Return $\mathbf{q} = \mathrm{discretize}(\mathbf{p}', \phi, \mathbf{R})$

We now provide a guarantee for each of these lines of Algorithm 1. We later use these guarantees to prove our final theorem (Theorem 2.3). The guarantees of the approximate maximizer $\mathbf{X}$ computed in the first step of the algorithm are summarized in the following lemma.

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

The guarantees of the second step of our algorithm are summarized in the following lemma. Please refer to [ACSS20] for the description of the procedure $\mathrm{sparsify}$. We use this procedure so that we can assume $|\mathbf{R}| \leq k + 1$ as we can ignore the zero rows of the matrix $\mathbf{X}$.

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

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 $\mathrm{swap}(\mathbf{A}, i_1, i_2, j_1, j_2, \epsilon)$ outputs a matrix $\mathbf{A}'$ that satisfies,

$$\mathbf{A}'_{ij} = \begin{cases} \mathbf{A}_{i,j} + \epsilon \text{ for } i = i_1,\ j = j_1 & \mathbf{A}_{i,j} - \epsilon \text{ for } i = i_1,\ j = j_2\ , \\ \mathbf{A}_{i,j} - \epsilon \text{ for } i = i_2,\ j = j_1 & \mathbf{A}_{i,j} + \epsilon \text{ for } i = i_2,\ j = j_2\ , \\ \mathbf{A}_{ij} \text{ otherwise.} \end{cases} \tag{6}$$

Definition 4.4 (Swap distance)**.** $\mathbf{A}'$ 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, i.e. 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 $\mathbf{A}^{(t)} = \mathbf{A}'$.

The following lemma directly follows from Definition 4.3 and Definition 4.4.

Lemma 4.5. *For any matrices $\boldsymbol{A}, \boldsymbol{A}' \in \mathbb{R}^{s \times t}$, if $\boldsymbol{A}'$ is x-swap distance from $\boldsymbol{A}$ for some $x \geq 0$, then $\boldsymbol{A}'\overrightarrow{1} = \boldsymbol{A}\overrightarrow{1}$ and $\boldsymbol{A}'^\top\overrightarrow{1} = \boldsymbol{A}^\top\overrightarrow{1}$.*

Recall that our objective $\mathbf{g}(\mathbf{X})$ contains two terms: (1) the linear term $\sum_{i \in [\ell], j \in [0,k]} \mathbf{C}_{ij}\mathbf{X}_{ij}$ and (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 4.6. *If $\boldsymbol{A}' \in \mathbb{R}^{\ell \times [0,k]}$ is x-swap distance from $\boldsymbol{A} \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,\mathrm{frac}}$, then, $\boldsymbol{A}' \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,\mathrm{frac}}$ 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 4.7. *For any matrix $\boldsymbol{A} \in \mathbb{R}^{s\times t}$ $(s \le t)$ that satisfies $\boldsymbol{A}^\top \overrightarrow{1} \in \mathbb{Z}^t_{\ge 0}$. The algorithm* swapmatrixround *runs in $O(s^2 t)$ 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 $\boldsymbol{A}'^\top \overrightarrow{1} = \boldsymbol{A}^\top \overrightarrow{1}$.*
- *$0 \le \boldsymbol{B}_{ij} \le \boldsymbol{A}'_{ij}$ for all $i \in [s]$ and $j \in [t]$, $\boldsymbol{B}\overrightarrow{1} \in \mathbb{Z}^s_{\ge 0}$, $\boldsymbol{B}^\top \overrightarrow{1} \in \mathbb{Z}^t_{\ge 0}$ and $\|\boldsymbol{A}' - \boldsymbol{B}\|_1 \le O(s)$.*

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 4.8 (Lemma 6.13 in [ACSS21])**.** *For any $\boldsymbol{A} \in \boldsymbol{Z}^{\phi,\text{frac}}_{\boldsymbol{R}} \subseteq \mathbb{R}^{\ell\times[0,k]}_{\ge 0}$ and $\boldsymbol{B} \in \mathbb{R}^{\ell\times[0,k]}_{\ge 0}$ such that $\boldsymbol{B} \le \boldsymbol{A}$, $\boldsymbol{B}\overrightarrow{1} \in \mathbb{Z}^\ell$, $\boldsymbol{B}^\top\overrightarrow{1} \in \mathbb{Z}^{[0,k]}$ and $\|\boldsymbol{A} - \boldsymbol{B}\|_1 \le t$. 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 $|\boldsymbol{R}'| \le |\boldsymbol{R}| + \min(k+1, t)$, $\boldsymbol{A}' \in \boldsymbol{Z}^{\phi}_{\boldsymbol{R}'}$ and $\boldsymbol{g}(\boldsymbol{A}') \ge \exp\left(-O\left(t\log n\right)\right)\boldsymbol{g}(\boldsymbol{A})$.*

As our final goal is to return a distribution in $\Delta^{\mathcal{D}}_{\mathbf{R}}$, we also use the following discretization lemma.

Lemma 4.9. *The function* discretize *takes as input a distribution $\boldsymbol{p} \in \Delta^{\mathcal{D}}$ with ℓ' distinct probability values, a profile ϕ, a discretization set of the form $\boldsymbol{R} = \boldsymbol{R}_{n,\alpha}$ for some $\alpha > 0$ and outputs a pseudo distribution $\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}$ such that: $\mathbb{P}\left(\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_1}, \phi\right) \ge \exp(-O(\min(k, |\boldsymbol{R}|) + \min(k, \ell') + \alpha^2 n)\log n)\mathbb{P}\left(\boldsymbol{p}, \phi\right)$.*

In Section 5, we use the guarantees stated above for each line of Algorithm 1 to prove Theorem 2.3. The description of the function discretize is specified in the proof of Lemma 4.9. We describe the procedure swapmatrixround and provide a proof sketch of Lemma 4.7 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]. Both of [ACSS20] and our approximate PML algorithm have four main lines (1-4); we differ from [ACSS20] in the key Line 3. This line in [ACSS20] invokes a procedure called matrixround that takes as 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: $\mathbf{B} \le \mathbf{A}$, $\mathbf{B}\overrightarrow{1} \in \mathbb{Z}^\ell_{\ge 0}$, $\mathbf{B}^\top\overrightarrow{1} \in \mathbb{Z}^{[0,k]}_{\ge 0}$ and $\|\mathbf{A} - \mathbf{B}\|_1 \le O(\ell + k)$. Such a matrix $\mathbf{B}$ is crucial as the procedure create uses $\mathbf{B}$ 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 $\ell \le k+1$, 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 $\|\mathbf{A}-\mathbf{B}\|_1 \le O(\min(k, \ell))$. However, such a matrix $\mathbf{B}$ may not exist for arbitrary matrices $\mathbf{A}$ and the best guarantee any algorithm can achieve is $\|\mathbf{A} - \mathbf{B}\|_1 \in O(\ell + k)$.

To overcome this, we introduce a new procedure called swapmatrixround that takes as input, a matrix $\mathbf{A}$ and transforms it to a new matrix $\mathbf{A}'$ that satisfies: $\mathbf{g}(\mathbf{A}') \ge \exp(-O(\min(k, \ell)\log n))\mathbf{g}(\mathbf{A})$. Furthermore, this transformed matrix $\mathbf{A}'$ exhibits a matrix $\mathbf{B}$ that satisfies the guarantees: $\mathbf{B} \le \mathbf{A}'$, $\mathbf{B}\overrightarrow{1} \in \mathbb{Z}^\ell_{\ge 0}$, $\mathbf{B}^\top\overrightarrow{1} \in \mathbb{Z}^k_{\ge 0}$ and $\|\mathbf{A}' - \mathbf{B}\|_1 \le O(\ell)$. These matrices $\mathbf{A}'$ and $\mathbf{B}$ are nice in that 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 swapmatrixround$(\mathbf{A})$

1: Let $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{D}^{(0)} = 0$.
2: **for** $r = 1 \ldots \ell$ **do**
3: $\quad(\mathbf{Y}, j) = \text{partialRound}(\mathbf{A}^{(r-1)}, r)$
4: $\quad\mathbf{A}^{(r)} = \text{roundiRow}(\mathbf{Y}, j, r)$.
5: $\quad\mathbf{D}^{(r)} = \mathbf{D}^{(r-1)} + \mathbf{Y} - \mathbf{A}^{(r)}$.
6: **end for**
7: Return $\mathbf{A}' = \mathbf{D}^{(\ell)} + \mathbf{A}^{(\ell)}$ and $\mathbf{B} = \mathbf{A}^{(\ell)}$.

Our algorithm includes two main subroutines: partialRound and roundiRow. At each iteration i, the procedure partialRound considers row i and modifies it by repeatedly applying the swap operation. This modified row is nice as the procedure 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 partialRound and roundiRow.

Lemma 4.10. *The algorithm* partialRound *takes as inputs* $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *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* $\boldsymbol{Y} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *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 $O(\ell k)$*.*

Note that by Lemma 4.5, if $\mathbf{Y}$ is within 3-swap distance from $\mathbf{X}$, then $\mathbf{Y}\overrightarrow{1} = \mathbf{X}\overrightarrow{1}$ and $\mathbf{Y}^\top \overrightarrow{1} = \mathbf{X}^\top \overrightarrow{1}$.

Lemma 4.11. *The algorithm* roundiRow *takes as inputs* $\boldsymbol{Y} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$*, an column index* $j \in [0,k]$ *and a row index* $i \in [\ell - 1]$ *such that:* $\boldsymbol{Y}^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^{[0,k]}$*,* $\boldsymbol{Y}_{ij} \geq o$ *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* $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *such that,*

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

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

5 Proof of Main Result (Theorem 2.3)

Here we put together the results from the previous sections to prove, Theorem 2.3.

Proof of Theorem 2.3. Algorithm 1 achieves the guarantees of Theorem 2.3. In the remainder of the proof, we combine the guarantees of each step of the algorithm to prove the theorem. Toward this end, 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 Line 1 of Algorithm 1 returns a solution $\mathbf{X} \in \mathbf{Z}^{\phi,\text{frac}}_{\mathbf{R}}$ that satisfies,

$$C_\phi \cdot \mathbf{g}(\mathbf{X}) \geq \exp\left(-O\left(\min(k, |\mathbf{R}|)\log n\right)\right) \max_{\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\mathbf{q}, \phi)\ . \tag{7}$$

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

$$\mathbf{X}' \in \mathbf{Z}^{\phi,\text{frac}}_{\mathbf{R}} \text{ and } \mathbf{g}(\mathbf{X}') \geq \mathbf{g}(\mathbf{X}), \tag{8}$$

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 $\mathbf{X}' \in \mathbf{Z}^{\phi,\text{frac}}_{\mathbf{R}}$, we have that $\mathbf{X}'$ has integral column sums and by invoking Lemma 4.7 with parameters $s = |\mathbf{R}|$ and $t = k+1$, we get matrices $\mathbf{A}$ and $\mathbf{B}$ that satisfy guarantees of Lemma 4.7. 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 $\mathbf{X}' \in \mathbf{Z}^{\phi,\text{frac}}_{\mathbf{R}}$, we immediately get that $\mathbf{A} \in \mathbf{Z}^{\phi,\text{frac}}_{\mathbf{R}}$. Further note that $\mathbf{A}$ is within $O(|\mathbf{R}|) = O(\min(|\mathbf{R}|, k))$-swap distance from $\mathbf{X}'$ and by Lemma 4.6 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,

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

Note that, Lemma 4.7 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 4.8 with parameter value $t = O(\min(|\mathbf{R}|,k))$. Therefore, the procedure create takes 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,

$$\mathbf{X}_{\text{final}} \in \mathbf{Z}^{\phi}_{\mathbf{R}_{\text{final}}} \text{ and } \mathbf{g}(\mathbf{X}_{\text{final}}) \geq \exp(-O(\min(|\mathbf{R}|,k)\log n))\mathbf{g}(\mathbf{A}) \ . \tag{10}$$

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

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

In the second inequality we used Equation (10) and $|\mathbf{R}_{\text{final}}| \leq 2|\mathbf{R}|$. In the third, fourth and fifth inequalities, we used Equation (9), Equation (8) and Equation (7) respectively.

Recall we need a distribution that approximately maximizes $\max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi)$ instead of just $\max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\mathbf{q},\phi)$. 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 $\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}$, as $\|\mathbf{q}\|_1$ satisfies: $\mathbf{r}_{\min} \leq \|\mathbf{q}\|_1 \leq 1$, we get that,

$$\max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi) = \max_{c\in[1,1/\mathbf{r}_{\min}]_{\mathbb{R}}} \max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{c\cdot\mathbf{R}}} \mathbb{P}(\mathbf{q},\phi) \ . \tag{11}$$

The above expression holds as the maximizer $\mathbf{q}^*$ of the left hand side satisfies: $\mathbf{q}^* \in \Delta^{\mathcal{D}}_{(1/\|\mathbf{q}^*\|_1)\cdot\mathbf{R}}$. 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 $\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}$ with $\|\mathbf{q}\|_1 = 1/c$, note that the distribution $\mathbf{q}' = c'\mathbf{q} \in \Delta^{\mathcal{D}}_{c'\cdot\mathbf{R}}$ 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 (11) we get that,

$$\max_{c\in C} \max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{c\cdot\mathbf{R}}} \mathbb{P}(\mathbf{q},\phi) \geq \exp(-2\beta n) \max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi). \tag{12}$$

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

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

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: $\mathbb{P}(\mathbf{p}^*,\phi) \geq \exp(-O(\min(k,|\mathbf{R}|)\log n) - 2\beta n)\max_{\mathbf{q}\in\Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi)$. Substituting $\beta = \frac{\min(k,|\mathbf{R}|)}{n}$ in the previous expression, we get,

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

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 4.9, we get a pseudo distribution $\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}$ 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 lines of Algorithm 1 are polynomial in n, our final running time follows from the run times of each line and we conclude the proof. □

Acknowledgments and Disclosure of Funding

We would like to thank the reviewers for their valuable feedback. Researchers on this project were supported by an Amazon Research Award, a Dantzig-Lieberman Operations Research Fellowship, a Google Faculty Research Award, a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research gift, a Simons-Berkeley Research Fellowship, a Simons Investigator Award, a Sloan Research Fellowship and a Stanford Data Science Scholarship.

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.

[ADOS17] Jayadev Acharya, Hirakendu Das, Alon Orlitsky, and Ananda Theertha Suresh. A unified maximum likelihood approach for estimating symmetric properties of discrete distributions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 11–21. PMLR, 06–11 Aug 2017.

[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.

[AW21] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 522–539. SIAM, 2021.

[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.

[Gal14] François Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 296–303. ACM, 2014.

[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. *ArXiv e-prints*, 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 e-prints, 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 distributions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[HS21] Yanjun Han and Kirankumar Shiragur. On the competitive analysis and high accuracy optimality of profile maximum likelihood. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1317–1336. SIAM, 2021.

[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, arXiv:1712.07177*, 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. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2012–2016, 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.

[Wil12] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, page 887–898, New York, NY, USA, 2012. Association for Computing Machinery.

[WY15] Y. Wu and P. Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *ArXiv e-prints, arXiv:1504.01227*, 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]

A Additional definitions

Optimal sample complexity The sample complexity of an estimator $\widehat{f} : \mathcal{D}^n \to \mathbb{R}$ when estimating a property $f : \Delta^{\mathcal{D}} \to \mathbb{R}$ for distributions in a collection $P \subseteq \Delta^{\mathcal{D}}$, is the number of samples $\widehat{f}$ needs to determine f with accuracy ϵ and low failure probability δ for all distributions in P. Specifically,

$$C^{\widehat{f}}(f, P, \delta, \epsilon) \stackrel{\text{def}}{=} \min\{n \mid \mathbb{P}\left(|f(p) - \widehat{f}(x^n)| \geq \epsilon\right) \leq \delta \text{ for all } p \in P\}.$$

The sample complexity of estimating f is the lowest sample complexity of any estimator,

$$C^*(f, P, \delta, \epsilon) = \min_{\widehat{f}} C^{\widehat{f}}(f, P, \delta, \epsilon).$$

In the paper, the dependency on δ is typically de-emphasized and δ is assumed to be $1/3$.

B Proof of Theorem 2.2

Here we provide the proof of Theorem 2.2. The proof of this statement is implicit in the analysis presented in [HS21]. Here we provide a simpler and short proof that uses the continuity lemma presented in [HS21]. For convenience, we restate this key lemma in our notation.

Lemma B.1 (Lemma 2 in [HS21])**.** *Let $A \geq 2$,$c_0 \in (0, 1), r, s$ be arbitrary constants with $0 < s < r \leq 1/2$ and let $\boldsymbol{R} = \boldsymbol{R}_{n,r}$. Then there exists a constant $c = c(A, c_0, r, s) > 0$ such that for any distribution $\boldsymbol{p} \in \Delta^{\mathcal{D}}$, there exists a pseudo distribution $\boldsymbol{q} \in \Delta^{\mathcal{D}}_{\boldsymbol{R}}$ such that: for all $S \subseteq \Phi^n$, it holds that,*

$$\mathbb{P}(\boldsymbol{p}, S) \geq \mathbb{P}\left(\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_1}, S\right)^{1/(1-c_o n^{-s})} \exp(-cn^{1-2r+s})$$

$$\mathbb{P}\left(\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_1}, S\right) \geq \mathbb{P}(\boldsymbol{p}, S)^{1/(1-c_o n^{-s})} \exp(-cn^{1-2r+s})$$

Proof of Theorem 2.2. Let $\mathbf{p}$ be the hidden distribution. Given a profile ϕ, let $\mathbf{q}_\phi \in \Delta^{\mathcal{D}}_{\mathbf{R}}$ be any pseudo distribution that satisfies,

$$\begin{aligned}\mathbb{P}\left(\frac{\mathbf{q}_\phi}{\|\mathbf{q}_\phi\|_1}, \phi\right) &\geq \exp(-O(|\mathbf{R}| \log n)) \max_{\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, \phi\right) \\ &\geq \exp(-O(n^{-1/3} \log^2 n)) \max_{\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, \phi\right) .\end{aligned}$$

As $L \stackrel{\text{def}}{=} |\Delta^{\mathcal{D}}_{\mathbf{R}}| \leq \exp(O(n^{1/3} \log^2 n))$, we use $\mathbf{q}_1, \ldots \mathbf{q}_L$ to denote the pseudo distributions in $\Delta^{\mathcal{D}}_{\mathbf{R}}$. Let $G \stackrel{\text{def}}{=} \{\phi \in \Phi^n \mid |f(\mathbf{p}) - \widehat{f}(\phi)| \leq \epsilon\}$, that is, the set of all profiles where the estimator succeeds. Also let $S_i = \{\phi \in G \mid \mathbf{q}_\phi = \mathbf{q}_i\}$. Using these definitions in the remainder of the proof, we upper bound the failure probability of our estimator.

$$\begin{aligned}\mathbb{P}\left(\left|f(\frac{\mathbf{q}_\phi}{\|\mathbf{q}_\phi\|_1}) - f(\mathbf{p})\right| > \epsilon\right) &\leq \mathbb{P}(\mathbf{p}, \Phi^n \backslash G) + \sum_{\{i \in [1,|L|] \mid |f(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}) - f(\mathbf{p})| > \epsilon\}} \mathbb{P}(\mathbf{p}, S_i) \ , \\ &\leq \delta + \sum_{\{i \in [1,|L|] \mid |f(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}) - f(\mathbf{p})| > \epsilon\}} \mathbb{P}(\mathbf{p}, S_i) \ .\end{aligned}$$

In the above inequality we used that the failure probability of the estimator is at most δ, that is, $\mathbb{P}(\mathbf{p}, \Phi^n \backslash G) \leq \delta$. First note that from the definitions of $\mathbf{q}_i$, S_i and $\mathbf{q}_\phi$, we have that,

$$\mathbb{P}\left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}, S_i\right) \geq \exp(-O(n^{-1/3} \log^2 n)) \max_{\mathbf{q} \in \Delta^{\mathcal{D}}_{\mathbf{R}}} \mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, S_i\right) .$$

Further applying Lemma B.1 with $r = 1/3$ and s with any tiny constant, we get that,

$$\mathbb{P}\left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}, S_i\right) \geq \exp(-O(n^{-1/3} \log n)) \mathbb{P}(\mathbf{p}, S_i)^{1+O(n^{-c'})} \exp(-O(n^{1/3+c'})) \ ,$$

for any tiny constant $c' > 0$. The above expression further simplifies to,

$$\mathbb{P}\left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}, S_i\right) \geq \exp(-O(n^{1/3+c'}))\mathbb{P}\left(\mathbf{p}, S_i\right)^{1+O(n^{-c'})} .$$

Suppose S_i is set such that, $\mathbb{P}\left(\mathbf{p}, S_i\right) \geq \delta^{\frac{1}{1+O(n^{-c'})}} \exp(\frac{O(n^{1/3+c'})}{1+O(n^{-c'})})$, then note that $\mathbb{P}\left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}, S_i\right) > \delta$. As the estimator provided by the conditions of the lemma succeeds on all the profiles in S_i, we that $|f(\mathbf{p}) - \widehat{f}(\phi)| \leq \epsilon$ for all $\phi \in S_i$. Suppose $|f(\mathbf{p}) - f(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1})| > 2\epsilon$, then by triangle inequality this would imply $|f(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}) - \widehat{f}(\phi)| > \epsilon$ for all $\phi \in S_i$. However note that, $\mathbb{P}\left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}, S_i\right) > \delta$, and this would imply that the failure probability of the estimator is greater than δ when the underlying distribution is $\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1}$; a contradiction. Therefore, it should be the case that $|f(\mathbf{p}) - f(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_1})| \leq 2\epsilon$ for all S_i that satisfy $\mathbb{P}\left(\mathbf{p}, S_i\right) \geq \delta^{\frac{1}{1+O(n^{-c'})}} \exp(\frac{O(n^{1/3+c'})}{1+O(n^{-c'})})$ and our failure probability is upper bounded by,

$$\mathbb{P}\left(\left|f(\frac{\mathbf{q}_\phi}{\|\mathbf{q}_\phi\|_1}) - f(\mathbf{p})\right| > \epsilon\right) \leq \delta + |L|\delta^{1-O(n^{-c'})} \exp(O(n^{1/3+c'})) .$$

Further, substituting the value of $|L| \leq \exp(O(n^{1/3}\log^2 n))$, we get our desired result and we conclude our proof. □

C Other results

Lemma C.1. *For any two vectors $u, v \in \mathbb{R}_{\geq 0}^{[0,k]}$, the following inequality holds,*

$$(u^T\overrightarrow{1}\log u^T\overrightarrow{1} + v^T\overrightarrow{1}\log v^T\overrightarrow{1}) - \sum_{i\in[0,k]}(u_i\log u_i + v_i\log v_i) \leq w^T\overrightarrow{1}\log w^T\overrightarrow{1} - \sum_{i\in[0,k]} w_i\log w_i$$

where $w = u + v$.

Proof. For any $x \in \mathbb{R}_{\geq 0}^{[0,k]}$, let $f(x) \stackrel{\text{def}}{=} x^T\overrightarrow{1}\log x^T\overrightarrow{1} - \sum_{i\in[0,k]} x_i\log x_i$. Note that $f(x)$ is concave [CSS19a] and furthermore, $f(c\cdot x) = c\cdot f(x)$.

Let $w' = \frac{1}{2}u + \frac{1}{2}v$, applying concavity we get that,

$$f(w') \geq \frac{1}{2}f(u) + \frac{1}{2}f(v),$$

As $f(w) = 2f(w')$, combined with above inequality we have our proof. □

Lemma C.2. *For many matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{\ell\times[0,k]}$ (where $\|\boldsymbol{X}\|_1, \|\boldsymbol{Y}\|_1, k, \ell \leq O(n^2)$) such that $\|\boldsymbol{X} - \boldsymbol{Y}\|_1 \leq \alpha$, we have that,*

$$\left|\sum_{i\in\ell, j\in[0,k]} \boldsymbol{X}_{ij}\log\boldsymbol{X}_{ij} - \sum_{i\in\ell, j\in[0,k]} \boldsymbol{Y}_{ij}\log\boldsymbol{Y}_{ij}\right| \leq O(\alpha\log n) + O(\log n) .$$

Furthermore,

$$\left|\sum_{i\in\ell}[\boldsymbol{X}\overrightarrow{1}]_i\log[\boldsymbol{X}\overrightarrow{1}]_i - \sum_{i\in\ell}[\boldsymbol{Y}\overrightarrow{1}]_i\log[\boldsymbol{Y}\overrightarrow{1}]_i\right| \leq O(\alpha\log n) + O(\log n) .$$

Proof. Note that $x\log x$ is $O(\log n)$-Lipshcitz for $x \geq \frac{1}{n^2}$, and for any $0 \leq x_1, x_2 \leq \frac{1}{n^2}$, we have $|x_1\log x_1 - x_2\log x_2| \leq O(\frac{1}{n^2}\log n)$. As a result, for any non-negative x_1, x_2 that are $O(n^2)$, we have

$$|x_1\log x_1 - x_2\log x_2| \leq \left(\frac{1}{n^2} + |x_1 - x_2|\right)\cdot O(\log n)$$

Note that $\sum_{i\in\ell,j\in[0,k]} \mathbf{X}_{ij} \le O(n^2)$ and $\sum_{i\in\ell,j\in[0,k]} \mathbf{Y}_{ij} \le O(n^2)$. We have

$$\left|\sum_{i\in\ell,j\in[0,k]} \mathbf{X}_{ij}\log\mathbf{X}_{ij} - \sum_{i\in\ell,j\in[0,k]} \mathbf{Y}_{ij}\log\mathbf{Y}_{ij}\right| \le \sum_{i\in\ell,j\in[0,k]} |\mathbf{X}_{ij}\log\mathbf{X}_{ij} - \mathbf{Y}_{ij}\log\mathbf{Y}_{ij}|$$

$$\le \sum_{i\in\ell,j\in[0,k]} (\frac{1}{n^2} + |\mathbf{X}_{ij} - \mathbf{Y}_{ij}|)O(\log n) \le (1 + \|\mathbf{X}-\mathbf{Y}\|_1)\cdot O(\log n) \le O(\alpha\log n) + O(\log n)$$

and similarly,

$$\left|\sum_{i\in\ell}[\mathbf{X}\overrightarrow{1}]_i\log[\mathbf{X}\overrightarrow{1}]_i - \sum_{i\in\ell}[\mathbf{Y}\overrightarrow{1}]_i\log[\mathbf{Y}\overrightarrow{1}]_i\right| \le \sum_{i\in\ell}\left|[\mathbf{X}\overrightarrow{1}]_i\log[\mathbf{X}\overrightarrow{1}]_i - [\mathbf{Y}\overrightarrow{1}]_i\log[\mathbf{Y}\overrightarrow{1}]_i\right|$$

$$\le \sum_{i\in\ell}(\frac{1}{n^2} + |[\mathbf{X}\overrightarrow{1}]_i - [\mathbf{Y}\overrightarrow{1}]_i|)O(\log n) \le (\frac{1}{n} + \|\mathbf{X}-\mathbf{Y}\|_1)O(\log n) \le O(\alpha\log n) + O(\log n)$$

We conclude the proof. □

D Description and guarantees of partialRound

Here we provide the description of partialRound and its guarantees are summarized in Lemma 4.10.

Algorithm 3 partialRound$(\mathbf{X}, i)$

1: $\mathbf{Z} = \text{split}(\mathbf{X}, i)$.
2: $(\mathbf{W}, j) = \text{partial_round_special}(\mathbf{Z}, i+1)$.
3: $\mathbf{Y} = \text{combine}(\mathbf{W}, i)$.
4: Return $(\mathbf{Y}, j)$.

Lemma 4.10. *The algorithm* partialRound *takes as inputs* $\boldsymbol{X} \in \mathbb{R}_{\ge 0}^{\ell\times[0,k]}$ *and* $i \in [\ell-1]$ *that satisfies the following,* $[\boldsymbol{X}\overrightarrow{1}]_{i'} \in \mathbb{Z}_{\ge 0}$ *for all* $i' \in [1, i-1]$ *and* $[\boldsymbol{X}^\top\overrightarrow{1}]_j \in \mathbb{Z}_{\ge 0}$ *for all* $j \in [0,k]$*, and outputs a matrix* $\boldsymbol{Y} \in \mathbb{R}_{\ge 0}^{\ell\times[0,k]}$ *and an index* j' *such that:*

- $\boldsymbol{Y}$ *is within* 3*-swap distance from* $\boldsymbol{X}$*.*
- $\boldsymbol{Y}_{ij'} \ge o$ *and* $\sum_{i'=1}^{i-1}\boldsymbol{Y}_{i'j'} + \boldsymbol{Y}_{ij'} - o \in \mathbb{Z}_{\ge 0}$*, where* $o = [\boldsymbol{X}\overrightarrow{1}]_i - \lfloor[\boldsymbol{X}\overrightarrow{1}]_i\rfloor$*.*

Furthermore, the running time of the algorithm is $O(\ell k)$*.*

Our algorithm invokes several subroutines and in the following we provide description for each of these and also state their guarantees. The subroutine partial_round_special provides an algorithm that proves the Lemma 4.10 in the special case of $[\mathbf{X}\overrightarrow{1}]_i \in [0,1)$. The description of this subroutine and its guarantees are stated below. For convenience, we first introduce some new notation and a new operation trans.

Notation: For any matrix $\mathbf{X} \in \mathbb{R}_{\ge 0}^{\ell\times[0,k]}$ and indices i_1, i_2, j_1, j_2, we define $\mathbf{X}(i_1 : i_2, j_1 : j_2) \stackrel{\text{def}}{=} \sum_{i=i_1}^{i_2}\sum_{j=j_1}^{j_2}\mathbf{X}_{ij}$ and say $(i,j) \in [i_1 : i_2, j_1 : j_2]$ if $i_1 \le i \le i_2$ and $j_1 \le j \le j_2$. Therefore, $\mathbf{X}(i_1 : i_2, j_1 : j_2) = \sum_{(i,j)\in[i_1:i_2,j_1:j_2]}\mathbf{X}_{ij}$. Also note that when $i_1 \ge i_2 + 1$ or $j_1 \ge j_2 + 1$, we are summing over a empty set and $\mathbf{X}(i_1 : i_2, j_1 : j_2) \stackrel{\text{def}}{=} 0$.

Lemma D.1 (Guarantees of trans)**.** *Let* $\boldsymbol{X} \in \mathbb{R}_{\ge 0}^{\ell\times[0,k]}$*. For any indices* $i_1 \le i_2 + 1 \le i_3 \le i_4 + 1$*,* $j_1 \le j_2 + 1 \le j_3 \le j_4 + 1$*, and* $0 \le v \le \min\{\boldsymbol{X}(i_3 : i_4, j_1 : j_2), \boldsymbol{X}(i_1 : i_2, j_3 : j_4)\}$*, there exists* $\boldsymbol{Y} \in \mathbb{R}_{\ge 0}^{\ell\times[0,k]}$ *such that,* $\boldsymbol{Y}$ *is within* v*-swap distance from* $\boldsymbol{X}$*, and*

- $\boldsymbol{Y}(i_1 : i_2, j_1 : j_2) = \boldsymbol{X}(i_1 : i_2, j_1 : j_2) + v$ • $\boldsymbol{Y}(i_3 : i_4, j_3 : j_4) = \boldsymbol{X}(i_3 : i_4, j_3 : j_4) + v$,
- $\boldsymbol{Y}(i_1 : i_2, j_3 : j_4) = \boldsymbol{X}(i_1 : i_2, j_3 : j_4) - v$ • $\boldsymbol{Y}(i_3 : i_4, j_1 : j_2) = \boldsymbol{X}(i_3 : i_4, j_1 : j_2) - v$.
- $\boldsymbol{Y}_{ij} = \boldsymbol{X}_{ij}\ \ \forall (i,j) \notin [i_1 : i_2, j_1 : j_2] \cup [i_1 : i_2, j_3 : j_4] \cup [i_3 : i_4, j_1 : j_2] \cup [i_3 : i_4, j_3 : j_4]$.

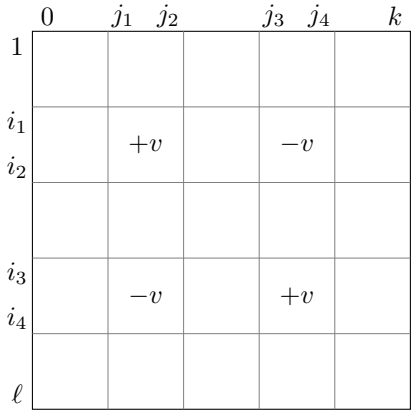

Figure 1: trans operation. It increases two blocks by v respectively, and decreases two blocks by v respectively.

Furthermore, we define operation $\mathrm{trans}(\boldsymbol{X}, v, i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)$ *whose output is* $\boldsymbol{Y}$ *that satisfies properties above, and the running time of process* trans *is* $O(\ell k)$.

We defer the proof for lemma D.1 to appendix D.1. The description of our subroutine $\mathrm{partial_round_special}$ is as follows.

Algorithm 4 $\mathrm{partial_round_special}(\mathbf{X}, i)$

$j = \min\{r : \mathbf{X}(1:\ell, 0:r) > \mathbf{X}(1:i-1, 0:k)$.
$t = \mathbf{X}(1:i-1, 0:k) - \mathbf{X}(1:\ell, 0:j-1)$.
$\mathbf{Y} = \mathbf{X}$.
$\mathbf{Y} = \mathrm{trans}(\mathbf{Y}, \mathbf{Y}(i:i, 0:j-1), 1, i-1, i, i, 0, j-1, j, k)$.
if $\mathbf{Y}(1:i-1, j:j) \geq t$ **then**
    $v = \mathbf{Y}(1:i-1, j:j) - \lfloor \mathbf{Y}(1:i-1, j:j) \rfloor$.
    $\mathbf{Y} = \mathrm{trans}(\mathbf{Y}, \min\{v+1, \mathbf{Y}(1:i-1, j:j) - t\}, 1, i-1, i+1, \ell, 0, j-1, j, j)$.
else
    $v = \lceil \mathbf{Y}(1:i-1, j:j) \rceil - \mathbf{Y}(1:i-1, j:j)$.
    $\mathbf{Y} = \mathrm{trans}(\mathbf{Y}, v, 1, i-1, i, \ell, j, j, j+1, k)$.
end if
$\mathbf{Y} = \mathrm{trans}(\mathbf{Y}, \mathbf{Y}(i:i, j+1:k), i, i, i+1, \ell, j, j, j+1, k)$.
Return $(\mathbf{Y}, j)$.

Lemma D.2 (Guarantee of $\mathrm{partial_round_special}$)**.** *The algorithm* $\mathrm{partial_round_special}$ *takes in inputs* $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *and* $i \in [1, \ell-1]$ *that satisfy the conditions of Lemma 4.10 with an additions assumption that* $\sum_{j' \in [0,k]} \boldsymbol{X}_{ij'} \in [0,1)$ *and outputs a matrix* $\boldsymbol{Y} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *and an index* j *that satisfy the guarantees of Lemma 4.10. Furthermore, the running time of this procedure is* $O(\ell k)$.

We defer the proof for lemma D.2 to appendix D.2. To extend the above algorithm for the general case, we define simple operations split and $\mathrm{combine}$ that we define next. Intuitively, the operation $\mathrm{combine}$ combines i-th and $(i+1)$-th rows by adding them up, and split splits i-th row into two rows so that the summation of one row is an integer and the summation of the other row is less than one.

Definition D.3 (Combine)**.** For any $\mathbf{W} \in \mathbb{R}_{\geq 0}^{(\ell+1) \times [0,k]}$ and $t \in [1, \ell]$, $\mathbf{Y} \stackrel{\mathrm{def}}{=} \mathrm{combine}(\mathbf{W}, t) \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ is defined as follows,

$$\mathbf{Y}_{i,j} = \begin{cases} \mathbf{W}_{i,j} & i \leq t-1 \\ \mathbf{W}_{i,j} + \mathbf{W}_{(i+1),j} & i = t \\ \mathbf{W}_{(i+1),j} & i \geq t+1 \end{cases}$$

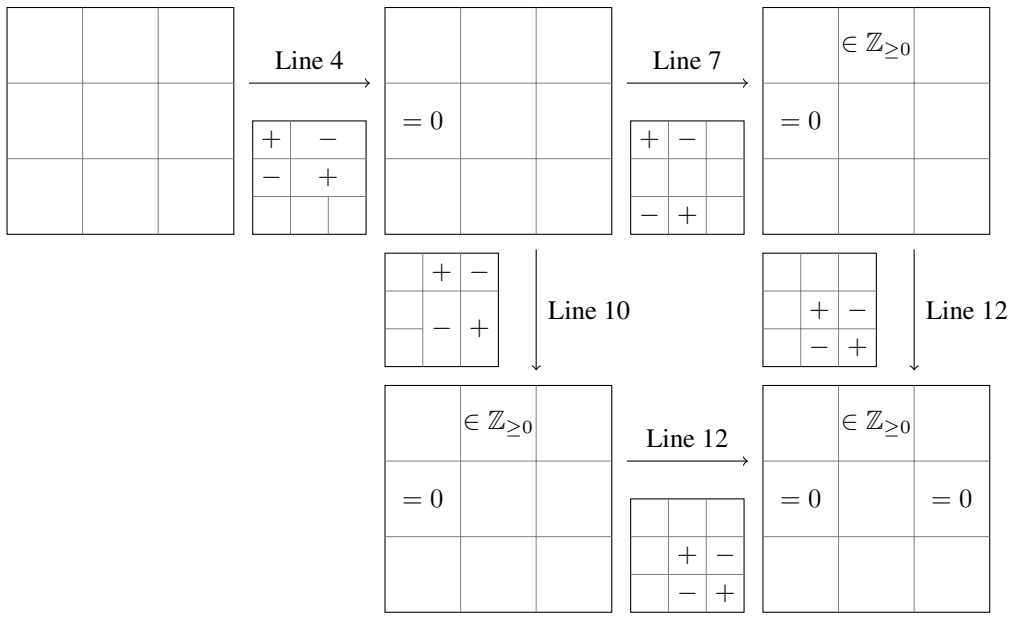

Figure 2: Algorithm 4. We use a 3×3 partitioned matrix to denote $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$, where the upper (middle / lower) row denotes the first $(i-1)$ rows (the i-th row / the last $(\ell - i)$ rows) of $\mathbf{Y}$, and the left (middle / right) column denotes the 0-th to $(j-1)$-th columns (the j-th column / the last $(k-j)$ columns) of $\mathbf{Y}$. Symbol $+$ or $-$ means increasing or decreasing some elements in the block. Symbol $= 0$ means all the elements in the block are zero. Symbol $\in \mathbb{Z}_{\geq 0}$ means the sum of the block is a non-negative integer.

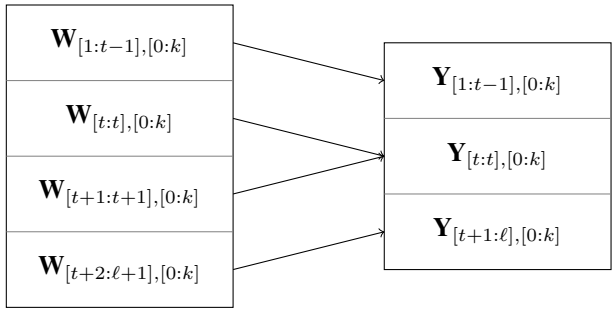

Figure 3: $\mathrm{combine}$ operation. It combines the t-th row and the $(t+1)$-th row by adding them up

Definition D.4 (Split)**.** Let $\mathbf{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$, $r(t) := \sum_{j \in [0,k]} \mathbf{X}_{tj}$ and $s(t) := \min\{j : \sum_{j' \leq j} \mathbf{X}_{tj'} \geq \lfloor r(t) \rfloor\}$. We define $\mathrm{split}(\mathbf{X}, t)$ to be $\mathbf{Z} \in \mathbb{R}_{\geq 0}^{(\ell+1) \times [0,k]}$ where,

$$\mathbf{Z}_{i,j} = \mathbf{X}_{i,j}, \ \ i \leq t-1 \qquad\qquad \mathbf{Z}_{i,j} = \mathbf{X}_{(i-1),j}, i \geq t+2$$

$$\mathbf{Z}_{t,j} = \begin{cases} \mathbf{X}_{t,j}, & j < s(t) \\ \lfloor r(t) \rfloor - \sum_{j' \leq j-1} \mathbf{X}_{t,j'}, & j = s(t) \\ 0, & j > s(t) \end{cases} \qquad \mathbf{Z}_{t+1,j} = \begin{cases} 0, & j < s(t) \\ \sum_{j' \leq j} \mathbf{X}_{t,j'} - \lfloor r(t) \rfloor, & j = s(t) \\ \mathbf{X}_{t,j}, & j > s(t) \end{cases}$$

We remark that $\mathbf{Z}$ has the following properties: 1) $\mathrm{combine}(\mathbf{Z}, t) = \mathbf{X}$, 2) $\sum_{j \in [0,k]} \mathbf{Z}_{tj} = \lfloor r(t) \rfloor$, 3) $\sum_{j \in [0,k]} \mathbf{Z}_{(t+1),j} = r(t) - \lfloor r(t) \rfloor$, which is a real number in $[0, 1)$.

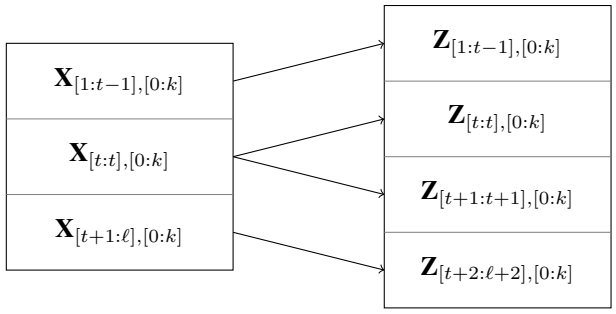

Figure 4: split operation. It splits the t-th row into two rows, where the sum of one row is an integer, and the sum of the other row is less than one.

D.1 Proof for lemma D.1

Proof for Lemma D.1. We define $r(\mathbf{X})$ to be $\left\{(i,j) \in [i_3 : i_4, j_1 : j_2] \middle| \mathbf{X}_{ij} > 0\right\}$, and $s(\mathbf{X})$ to be $\left\{(i,j) \in [i_1 : i_2, j_3 : j_4] \middle| \mathbf{X}_{ij} > 0\right\}$. Note that $r(\mathbf{X})$ is also dependent on i_3, i_4, j_1 and j_2, but they are fixed in this proof, and $s(\mathbf{X})$ is also dependent on i_1, i_2, j_3 and j_4. We prove the lemma by induction.

When $|r(\mathbf{X})| + |s(\mathbf{X})| \leq 1$, one of $|r(\mathbf{X})|$ and $|s(\mathbf{X})|$ is 0, so $v = 0$, and $\mathbf{Y} = \mathbf{X}$ is a feasible output.

Now we assume the lemma is true when $|r(\mathbf{X})| + |s(\mathbf{X})| < t$, and show it is still true when $|r(\mathbf{X})| + |s(\mathbf{X})| = t$, where $t \geq 2$.

If $|r(\mathbf{X})| = 0$ or $|s(\mathbf{X})| = 0$, we have $v = 0$, and $\mathbf{Y} = \mathbf{X}$ is a feasible output.

If $|r(\mathbf{X})| > 0$ and $|s(\mathbf{X})| > 0$, we draw arbitrary elements $(c, b) \in r(\mathbf{X})$ and $(a, d) \in s(\mathbf{X})$.

Let $u = \min\{\mathbf{X}_{cb}, \mathbf{X}_{ad}, v\}$. Let $\mathbf{W} = \mathrm{swap}(\mathbf{X}, a, c, b, d, u)$, and we have

- $\mathbf{W}$ is u-swap distance from $\mathbf{X}$, (13)
- $\mathbf{W}(i_1 : i_2, j_1 : j_2) = \mathbf{X}(i_1 : i_2, j_1 : j_2) + u$ • $\mathbf{W}(i_3 : i_4, j_3 : j_4) = \mathbf{X}(i_3 : i_4, j_3 : j_4) + u$, (14)
- $\mathbf{W}(i_1 : i_2, j_3 : j_4) = \mathbf{X}(i_1 : i_2, j_3 : j_4) - u$ • $\mathbf{W}(i_3 : i_4, j_1 : j_2) = \mathbf{X}(i_3 : i_4, j_1 : j_2) - u$. (15)
- $\forall (i,j) \notin [i_1 : i_2, j_1 : j_2] \cup [i_1 : i_2, j_3 : j_4] \cup [i_3 : i_4, j_1 : j_2] \cup [i_3 : i_4, j_3 : j_4], \mathbf{W}_{ij} = \mathbf{X}_{ij}$. (16)

If $u = v$, $\mathbf{Y} = \mathbf{W}$ is a feasible output. Otherwise, either $u = \mathbf{X}_{cb}$ or $u = \mathbf{X}_{ad}$, so either $\mathbf{W}_{cb}$ or $\mathbf{W}_{ad}$ becomes zero. Note that (a, b) and (c, d) are not in $[i_1 : i_2, j_3 : j_4] \cup [i_3 : i_4, j_1 : j_2]$, so we have $|r(\mathbf{W})| + |s(\mathbf{W})| \leq |r(\mathbf{X})| + |s(\mathbf{X})| - 1$. Let $\mathbf{Y} = \mathrm{trans}(\mathbf{W}, v - u, i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)$, and it is a feasible output because of the following properties.

- $\mathbf{Y}$ is $(u + (v - u) = v)$-swap distance from $\mathbf{X}$,
- $\mathbf{Y}(i_1 : i_2, j_1 : j_2) = \mathbf{W}(i_1 : i_2, j_1 : j_2) + (v - u) = \mathbf{X}(i_1 : i_2, j_1 : j_2) + v$,
- $\mathbf{Y}(i_3 : i_4, j_3 : j_4) = \mathbf{W}(i_3 : i_4, j_3 : j_4) + (v - u) = \mathbf{X}(i_3 : i_4, j_3 : j_4) + v$,
- $\mathbf{Y}(i_1 : i_2, j_3 : j_4) = \mathbf{W}(i_1 : i_2, j_3 : j_4) - (v - u) = \mathbf{X}(i_1 : i_2, j_3 : j_4) - v$,
- $\mathbf{Y}(i_3 : i_4, j_1 : j_2) = \mathbf{W}(i_3 : i_4, j_1 : j_2) - (v - u) = \mathbf{X}(i_3 : i_4, j_1 : j_2) - v$,
- $\forall (i,j) \notin [i_1 : i_2, j_1 : j_2] \cup [i_1 : i_2, j_3 : j_4] \cup [i_3 : i_4, j_1 : j_2] \cup [i_3 : i_4, j_3 : j_4], \mathbf{Y}_{ij} = \mathbf{W}_{ij} = \mathbf{X}_{ij}$,

where the relation between $\mathbf{Y}$ and $\mathbf{W}$ follows from inductive assumption, and the relation between $\mathbf{W}$ and $\mathbf{X}$ follows from Equation (13) to (16).

To implement trans, it is sufficient to enumerate all the elements in $r(\mathbf{X})$ and $s(\mathbf{X})$, so the running time can be $O(\ell k)$.

□

D.2 Proof for Lemma D.2

Lemma D.2 is a special case of Lemma 4.10, with an additional input constraint $\sum_{j' \in [0,k]} \mathbf{X}_{ij'} \in [0, 1)$. In the output $\mathbf{Y}$, the i-th row only has one positive element $\mathbf{Y}_{ij}$. In algorithm 4, Line 4 makes $\mathbf{Y}_{ij'}$ where $j' < j$ be zero, Line 12 makes $\mathbf{Y}_{ij'}$ where $j' > j$ be zero, and Line 7 and Line 10 make $\sum_{i'=1}^{i-1} \mathbf{Y}_{i'j}$ to be an integer.

Proof for Lemma D.2. There are mainly two parts in this proof. The former part proves the properties of output $\mathbf{Y}$ based on Lemma D.1, and the latter part proves the parameters are valid when calling trans in Algorithm 4.

First part of the proof In $\mathrm{partial_round_special}(\mathbf{X}, i)$, $\mathbf{Y}(i : i, 0 : j-1)$ only changes at Line 4, when it decreases by $\mathbf{Y}(i : i, 0 : j-1)$, so we have $\mathbf{Y}(i : i, 0 : j-1) = 0$ at Line 13. $\mathbf{Y}(i : i, j+1 : k)$ does not change after Line 12, where it decreases by $\mathbf{Y}(i : i, j+1 : k)$, so we have $\mathbf{Y}(i : i, j+1 : k) = 0$ at Line 13. Note that $\mathbf{Y}_{ij} = \sum_{j' \in [0,k]} \mathbf{X}_{ij'}$, so the output $\mathbf{Y}$ satisfies

$$\mathbf{Y}_{ij} = \sum_{j' \in [0,k]} \mathbf{X}_{ij'}$$

$\mathbf{Y}(1 : i-1, j : j)$ does not change after Line 7 or Line 10. If the algorithm goes to Line 7, $\mathbf{Y}(1 : i-1, j : j)$ decreases by $v+1$ or $\mathbf{Y}(1 : i-1, j : j) - t$, which means $\mathbf{Y}(1 : i-1, j : j) = \max\{(\lfloor \mathbf{Y}(1 : i-1, j : j) \rfloor - 1), t\}$ at Line 13. If the algorithm goes to Line 10, $\mathbf{Y}(1 : i-1, j : j)$ increases by v, which means $\mathbf{Y}(1 : i-1, j : j) = \lceil \mathbf{Y}(1 : i-1, j : j) \rceil$ at Line 13. In both cases, $\mathbf{Y}(1 : i-1, j : j)$ is a non-negative integer at Line 13, so the output $\mathbf{Y}$ satisfies

$$\sum_{i'=1}^{i-1} \mathbf{Y}_{i'j} \in \mathbb{Z}_{\geq 0}$$

We use three trans operations on $\mathbf{Y}$ which initially equals to $\mathbf{X}$, and we have $v \leq 1$ in each operation $\mathrm{trans}(\mathbf{X}, v, i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)$, so $\mathbf{Y}$ is 3-swap distance from $\mathbf{X}$.

We use $O(1)$ trans operations on $\mathbf{Y}$, so the algorithm takes $O(\ell k)$ time.

Second part of the proof We also need to show $0 \leq v \leq \min\{\mathbf{X}(i_3 : i_4, j_1 : j_2), \mathbf{X}(i_1 : i_2, j_3 : j_4)\}$ in each operation $\mathrm{trans}(\mathbf{X}, v, i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)$ to ensure the operations are valid.

Based on the definition of j and t, we have

$$\begin{aligned} &\mathbf{X}(1 : i-1, 0 : k) = \mathbf{X}(1 : \ell, 0 : j-1) + t, \\ &0 \leq t < \mathbf{Y}(1 : \ell, j : j). \end{aligned}$$

Note that t and $\mathbf{Y}(1 : \ell, j : j)$ are both integers, so we have

$$t \leq \mathbf{Y}(1 : \ell, j : j) - 1.$$

Before Line 4, $\mathbf{Y} = \mathbf{X}$, and we have $\mathbf{Y}(1 : i-1, 0 : k) \geq \mathbf{Y}(1 : \ell, 1 : j-1)$, which implies

$$\mathbf{Y}(1 : i-1, j : k) \geq \mathbf{Y}(i : \ell, 0 : j-1) \geq \mathbf{Y}(i : i, 0 : j-1),$$

which implies Line 4 is valid.

Before Line 7, we have $\mathbf{Y}(1 : i-1, 0 : k) = \mathbf{Y}(1 : \ell, 0 : j-1) + t$. Also, we have $\mathbf{Y}(i : i, 0 : j-1) = 0$ because of Line 4. So, we have

$$\mathbf{Y}(i+1 : \ell, 0 : j-1) \geq \mathbf{Y}(1 : i-1, j : j) - t,$$

which implies Line 7 is valid.

Before Line 10, we have $\mathbf{Y}(1:\ell, j:j) \geq t$. Note that t is an integer, so when $\mathbf{Y}(1:i-1, j:j) < t$, we have

$$\mathbf{Y}(i:\ell, j:j) = \mathbf{Y}(1:\ell, j:j) - \mathbf{Y}(1:i-1, j:j) \geq t - \mathbf{Y}(1:i-1, j:j) \geq v.$$

We also have,

$$\begin{aligned}\mathbf{Y}(1:i-1, j+1:k) &= \mathbf{Y}(1:i-1, 0:k) - \mathbf{Y}(1:i-1, 0:j-1) - \mathbf{Y}(1:i-1, j:j) \\ &\geq (\mathbf{Y}(1:i-1, 0:k) - \mathbf{Y}(1:\ell, 0:j-1)) - \mathbf{Y}(1:i-1, j:j) \\ &= t - \mathbf{Y}(1:i-1, j:j) \geq v.\end{aligned}$$

These two formulas imply Line 10 is valid.

After Line 7, we have either $\mathbf{Y}(1:i-1, j:j) = t$ (when $v+1 < \mathbf{Y}(1:i-1, j:j) - t$) or $\mathbf{Y}(i+1:\ell, j:j) \geq 1$ (the other case). After Line 10, we have $\mathbf{Y}(1:i-1, j:j) \leq t$. Since t is an integer, when $\mathbf{Y}(1:i-1, j:j) \leq t$, we have

$$\mathbf{Y}(i:\ell, j:j) = \mathbf{Y}(1:\ell, j:j) - \mathbf{Y}(1:i-1, j:j) \geq 1.$$

As a result, we always have $\mathbf{Y}(i:\ell, j:j) \geq 1$ before Line 12. Note that $\mathbf{Y}(i:i, 0:k) < 1$ and $\mathbf{Y}(i:i, 0:j-1) = 0$, so we have

$$\mathbf{Y}(i+1:\ell, j:j) = \mathbf{Y}(i:\ell, j:j) - \mathbf{Y}_{ij} > \mathbf{Y}(i:i, 0:k) - \mathbf{Y}_{ij} = \mathbf{Y}(i:i, j+1:k),$$

which implies Line 12 is valid. □

D.3 Proof for Lemma 4.10

Proof of Lemma 4.10. We prove the properties mentioned in Lemma 4.10 one by one. Firstly we have

- $\mathbf{Y}_{ij} \geq \mathbf{W}_{(i+1),j} = \sum_{j' \in [0,k]} \mathbf{Z}_{(i+1),j'} = o$,
- $\sum_{i'=1}^{i-1} \mathbf{Y}_{i'j} + \mathbf{Y}_{ij} - o = \sum_{i'=1}^{i+1} \mathbf{W}_{i'j} - o = \sum_{i'=1}^{i} \mathbf{W}_{i'j} \in \mathbb{Z}_{\geq 0}$,

where the relation between $\mathbf{W}$ and $\mathbf{Z}$ is from Lemma D.2, and the relation between $\mathbf{Y}$ and $\mathbf{W}$ is from the definition of $\mathrm{combine}$.

Next we show $\mathbf{Y}$ is $(3v)$-swap distance from $\mathbf{X}$. By lemma D.2, there is a set of parameters $\{(a_s, b_s, c_s, d_s, \epsilon_i)\}_{s\in[t]}$, where $\sum_{s\in[t]} \epsilon_s = 3v$, such that $A_s = \mathrm{swap}(A_{s-1}, a_s, b_s, c_s, d_s, \epsilon_s)$ for $s \in [t]$, and $A_0 = \mathbf{Z}$, $A_t = \mathbf{W}$.

Let $a'_s = a_s - \mathbb{I}(a_s > i)$ and $c'_s = c_s - \mathbb{I}(c_s > i)$ for $i \in [t]$, where $\mathbb{I}(\cdot)$ is the indicator function. Let $A'_0 = \mathbf{X}$ and $A'_s = \mathrm{swap}(A'_{s-1}, a'_s, b_s, c'_s, d_s, \epsilon_s)$ for $s \in [t]$. Then we have $A'_t = \mathbf{Y}$, which implies $\mathbf{Y}$ is $(3v)$-swap distance from $\mathbf{X}$.

We proved all the conditions of the lemma and we conclude the proof.

□

E Description and guarantees of $\mathrm{roundiRow}$

Here we provide a description of the subroutine $\mathrm{roundiRow}$.

Algorithm 5 $\mathrm{roundiRow}(\mathbf{Y}, j, i)$

1: Let matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$.
2: Assign $\mathbf{X}_{i'j'} = \mathbf{Y}_{i'j'}$ for all $i' \in [\ell],\ j' \in [0,k]$ and $j' \neq j$; and $\mathbf{X}_{i'j} = \mathbf{Y}_{i'j}$ for all $i' \in [1, i-1]$.

3: Assign $\mathbf{X}_{ij} = \mathbf{Y}_{ij} - o$ and residue $= 1 - o$.
4: **for** $i' = i+1 \ldots \ell$ **do**
5: Let $w_{i'} = \min(\mathbf{Y}_{i'j}, \text{residue})$.
6: $\mathbf{X}_{i'j} = \mathbf{Y}_{i'j} - w_{i'}$ and residue $=$ residue $- w_{i'}$.
7: **if** residue $= 0$ **then**
8: exit the for loop.
9: **end if**
10: **end for**
11: **Return X**.

The guarantees of the above algorithm are summarized in the lemma below.

Lemma 4.11. *The algorithm* roundiRow *takes as inputs* $\boldsymbol{Y} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$*, an column index* $j \in [0,k]$ *and a row index* $i \in [\ell - 1]$ *such that:* $\boldsymbol{Y}^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^{[0,k]}$*,* $\boldsymbol{Y}_{ij} \geq o$ *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* $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ *such that,*

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

Proof. We first show that $\mathbf{X} \in \mathbb{R}^{\ell \times [0,k]}$ is a non-negative matrix. Note that $\mathbf{X}_{i'j'} = \mathbf{Y}_{i'j'} - o$ if $i' = i$, $j' = j$ or else $\mathbf{X}_{i'j'}$ takes one of the values in the set $\{\mathbf{Y}_{i'j'}, \mathbf{Y}_{i'j'} - \min(\mathbf{Y}_{i'j'}, \text{residue})\}$. As $\mathbf{Y}_{ij} \geq o$, we get that $\mathbf{X}_{i'j'} \geq 0$ for all $i' \in [\ell]$ and $j' \in [0,k]$. Furthermore, it is immediate that, $\mathbf{X} \leq \mathbf{Y}$.

Consider any $i' \in [1, i-1]$ and it is immediate that (Line 2) $[\mathbf{X}\overrightarrow{1}]_{i'} = [\mathbf{Y}\overrightarrow{1}]_{i'} \in \mathbb{Z}_{\geq 0}$. Furthermore, $[\mathbf{X}\overrightarrow{1}]_i = [\mathbf{Y}\overrightarrow{1}]_i - o = \lfloor [\mathbf{Y}\overrightarrow{1}]_i \rfloor \in \mathbb{Z}_{\geq 0}$.

Consider any $j' \in [0,k]$ such that $j' \neq j$ and note that $\mathbf{X}_{i'j'} = \mathbf{Y}_{i'j'}$ for all $i' \in [\ell]$. Therefore, we get that $[\mathbf{X}^\top \overrightarrow{1}]_{j'} = [\mathbf{Y}^\top \overrightarrow{1}]_{j'} \in \mathbb{Z}_{\geq 0}$. Now consider the index j,

$$\sum_{i' \in [\ell]} \mathbf{X}_{i'j} + 1 = \sum_{i' \in [1,i-1]} \mathbf{X}_{i'j} + \sum_{i' \in [i,\ell]} \mathbf{X}_{i'j} + 1 = \sum_{i' \in [1,i-1]} \mathbf{Y}_{i'j} + \sum_{i' \in [i,\ell]} (\mathbf{Y}_{i'j} - w_{i'}) + 1,$$
$$= \sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - o - \sum_{i' \in [i+1,\ell]} w_{i'} + 1$$

The second and third equalities follow from Lines 6 and 3 of the algorithm respectively. In the following we show that, $\sum_{i' \in [i+1,\ell]} w_{i'} = 1 - o$. To show this equality all we need is to show that residue $= 0$ at the end of the loop, which holds when $\sum_{i' \in [i+1,\ell]} \mathbf{Y}_{i'j} \geq 1 - o$. As $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} \in \mathbb{Z}_{\geq 0}$, $o \in (0,1)_{\mathbb{R}}$ and $z \stackrel{\text{def}}{=} \sum_{i'=1}^{i} \mathbf{Y}_{i'j} - o \in \mathbb{Z}_{\geq 0}$, we get that,

$$\sum_{i' \in [i+1,\ell]} \mathbf{Y}_{i'j} = \sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - \sum_{i' \in [1,i]} \mathbf{Y}_{i'j} = \sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z - o \text{ for some integer } z \in \mathbb{Z}_{\geq 0},$$

In the following we show that $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z \geq 1$. As $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z \in \mathbb{Z}_{\geq 0}$ all we need is to show that $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z > 0$. Suppose $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z = 0$, then $\sum_{i' \in [i+1,\ell]} \mathbf{Y}_{i'j} = \sum_{i' \in [\ell]} \mathbf{Y}_{i'j} - z - o = -o < 0$, which is a contradiction as $\mathbf{Y}_{i'j'} \geq 0$ for all $i' \in [\ell]$, $j \in [0,k]$. Therefore we get that for all $j' \in [0,k]$,

$$\sum_{i' \in [\ell]} \mathbf{X}_{i'j'} + 1 = \sum_{i' \in [\ell]} \mathbf{Y}_{i'j'} \ . \tag{17}$$

As $\mathbf{X} \in \mathbb{R}_{\geq 0}^{\ell \times [0,k]}$ and $\sum_{i' \in [\ell]} \mathbf{Y}_{i'j} \in \mathbb{Z}_{\geq 0}$ (requirements of the lemma) we get that, $\sum_{i' \in [\ell]} \mathbf{X}_{i'j} \in \mathbb{Z}_{\geq 0}$. Therefore,

$$\mathbf{X}^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^{[0,k]} .$$

In the remainder of the proof we show that $\|\mathbf{X} - \mathbf{Y}\|_1 \leq 1$. Recall, earlier we showed that: $\mathbf{X} \leq \mathbf{Y}$, $[\mathbf{X}^\top \overrightarrow{1}]_{j'} = [\mathbf{Y}^\top \overrightarrow{1}]_{j'}$ for all $j' \neq j$ and $[\mathbf{X}^\top \overrightarrow{1}]_j + 1 = [\mathbf{Y}^\top \overrightarrow{1}]_j$. Combining these inequalities together, we immediately get that, $\|\mathbf{X} - \mathbf{Y}\|_1 = 1$.

We proved all the conditions of the lemma and we conclude the proof.

□

F Guarantees of swapmatrixround

To prove Lemma 4.7, we show a stronger version of the lemma.

Lemma F.1. *For any matrix $\boldsymbol{A} \in \mathbb{R}^{s \times t}$ $(s \leq t)$ that satisfies $\boldsymbol{A}^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^t$. In algorithm* swapmatrixround*, for all $r \in \{0\} \cup [\ell]$,*

- *$(\boldsymbol{D}^{(r)} + \boldsymbol{A}^{(r)})$ is $(3r)$-swap distance from $\boldsymbol{A}$.*
- *$\boldsymbol{D}_{ij}^{(r)} \geq 0, \boldsymbol{A}_{ij}^{(r)} \geq 0$ for all $i \in [s]$ and $j \in [t]$, $[\boldsymbol{A}^{(r)} \overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$ for all $i \in [r]$, $(\boldsymbol{A}^{(r)})^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^t$ and $\|\boldsymbol{D}^{(r)}\|_1 \leq r$.*

Proof. We prove this lemma through induction.

Base case i=0 In this case $\mathbf{A}^{(0)} = \mathbf{A}$, $\mathbf{D}^{(0)} = 0$, we have that all the conditions of the lemma are immediately satisfied.

Inductive step r We assume that the lemma conditions hold for $r-1$ and prove for the case r. We use $\mathbf{Y}^{(r)}$ to denote the value of $\mathbf{Y}$ in algorithm swapmatrixround when the enumerator is r. We verify the properties claimed by the lemma one by one, and we use the properties related to partialRound and roundiRow given by Lemma 4.10 and Lemma 4.11 respectively.

Since $(\mathbf{Y}^{(r)}, j) = \text{partialRound}(\mathbf{A}^{(r-1)}, r)$, $\mathbf{Y}^{(r)}$ is 3-swap distance from $\mathbf{A}^{(r-1)}$, which implies $(\mathbf{Y}^{(r)} + \mathbf{D}^{(r-1)})$ is 3-swap distance from $(\mathbf{A}^{(r-1)} + \mathbf{D}^{(r-1)})$. Note that $(\mathbf{A}^{(r)} + \mathbf{D}^{(r)}) = (\mathbf{Y}^{(r)} + \mathbf{D}^{(r-1)})$, and $(\mathbf{A}^{(r-1)} + \mathbf{D}^{(r-1)})$ is $(3(r-1))$-swap distance from $\mathbf{A}$, so we have $(\mathbf{D}^{(r)} + \mathbf{A}^{(r)})$ is $(3r)$-swap distance from $\mathbf{A}$.

Since $\mathbf{A}^{(r)} = \text{roundiRow}(\mathbf{Y}^{(r)}, j, r)$, we have $\mathbf{A}_{ij}^{(r)} \geq 0$ and $(\mathbf{Y}^{(r)} - \mathbf{A}^{(r)})_{ij} \geq 0$ for all $i \in [s]$ and $j \in [t]$. Furthermore, since $\mathbf{D}_{ij}^{(r-1)} \geq 0$ for all $i \in [s]$ and $j \in [t]$, and $\mathbf{D}^{(r)} = (\mathbf{Y}^{(r)} - \mathbf{A}^{(r)}) + \mathbf{D}^{(r-1)}$, we have $\mathbf{D}_{ij}^{(r)} \geq 0$ for all $i \in [s]$ and $j \in [t]$.

Since $(\mathbf{Y}^{(r)}, j) = \text{partialRound}(\mathbf{A}^{(r-1)}, r)$, we have $[\mathbf{Y}^{(r)} \overrightarrow{1}]_i = [\mathbf{A}^{(r-1)} \overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$ for $i \in [r-1]$. Since $\mathbf{A}^{(r)} = \text{roundiRow}(\mathbf{Y}^{(r)}, j, r)$, we have $[\mathbf{A}^{(r)} \overrightarrow{1}]_i = [\mathbf{Y}^{(r)} \overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$ for $i \in [r-1]$, and $[\mathbf{A}^{(r)} \overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$ for $i = r$. As a result, $[\mathbf{A}^{(r)} \overrightarrow{1}]_i \in \mathbb{Z}_{\geq 0}$ for $i \in [r]$.

Since $\mathbf{A}^{(r)} = \text{roundiRow}(\mathbf{Y}^{(r)}, j, r)$, we have $(\mathbf{A}^{(r)})^\top \overrightarrow{1} \in \mathbb{Z}_{\geq 0}^t$.

Since $\mathbf{A}^{(r)} = \text{roundiRow}(\mathbf{Y}^{(r)}, j, r)$, we have $\|\mathbf{Y}^{(r)} - \mathbf{A}^{(r)}\|_1 \leq 1$. Then we have $\|\mathbf{D}^{(r)}\|_1 \leq \|\mathbf{Y}^{(r)} - \mathbf{A}^{(r)}\|_1 + \|\mathbf{D}^{(r-1)}\|_1 \leq r$ because $\mathbf{D}^{(r)} = (\mathbf{Y}^{(r)} - \mathbf{A}^{(r)}) + \mathbf{D}^{(r-1)}$.

□

Corollary F.2. *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 *returns matrices $\boldsymbol{A}'$ and $\boldsymbol{B}$ such that,*

- *$\boldsymbol{A}'$ is $O(s)$-swap distance from $\boldsymbol{A}$.*

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

$\mathbf{A}'$ is $O(s)$-swap distance from $\mathbf{A}$, and it implies $\mathbf{A}'\overrightarrow{1} = \mathbf{A}\overrightarrow{1}$ and $\mathbf{A}'^\top \overrightarrow{1} = \mathbf{A}^\top \overrightarrow{1}$.

We run $O(s)$ times $\mathrm{partialRound}$, $\mathrm{roundiRow}$, and matrix addition and subtraction, so the running time is $O(s^2 t)$.

We proved all the conditions of Lemma 4.7 and we conclude the proof.

G Proof for Lemma 4.9

Proof. Let $\{\mathbf{r}'_i\}_{i \in [\ell']}$ be the set of distinct probability values of distribution $\mathbf{p}$. Let $\mathbf{X} \in \mathbb{R}^{\ell' \times [0,k]}_{\geq 0}$ be the maximizer of the following optimization problem,

$$\max_{\mathbf{Y} \in \mathbf{Z}^\phi_{\mathbf{p}}} \mathbf{g}(\mathbf{Y}) \ .$$

By Lemma 3.2, the maximizer $\mathbf{X}$ satisfies,

$$C_\phi \cdot \mathbf{g}(\mathbf{X}) \geq \exp(-O(\min(k, \ell')) \log n) \mathbb{P}(\mathbf{p}, \phi) \ ,$$

In the following we consider multiple distributions with different probabilities to denote the probability values explicitly, we extend the notation of $\mathbf{g}(\mathbf{X})$ to $\mathbf{g}(\mathbf{X}, \mathbf{r})$. Given the solution $\mathbf{X}$, we now maximize over the probability values, that is, solve,

$$\max_{\mathbf{r}'' \in [0,1]^{\ell'}} \mathbf{g}(\mathbf{X}, \mathbf{r}'') \stackrel{\text{def}}{=} \sum_{i \in [\ell'], j \in [0,k]} (\mathbf{m}_j \log \mathbf{r}''_i \mathbf{X}_{ij} - \mathbf{X}_{ij} \log \mathbf{X}_{ij}) + \sum_{i \in [\ell']} [\mathbf{X}\mathbf{1}]_i \log [\mathbf{X}\mathbf{1}]_i \ ,$$
$$\text{such that } \sum_{i \in [\ell']} [\mathbf{X}\mathbf{1}]_i \mathbf{r}''_i = 1 \ .$$

The optimum solution $\mathbf{r}^*$ satisfies $\sum_{i \in [\ell']} [\mathbf{X}\mathbf{1}]_i \mathbf{r}^*_i = 1$ and,

$$\mathbf{r}^*_i = \frac{\sum_{j \in [0,k]} \mathbf{m}_j \mathbf{X}_{ij}}{[\mathbf{X}\overrightarrow{1}]_i \sum_{i' \in [\ell']} \sum_{j \in [0,k]} \mathbf{m}_j \mathbf{X}_{i'j}} = \frac{\sum_{j \in [0,k]} \mathbf{m}_j \mathbf{X}_{ij}}{n [\mathbf{X}\overrightarrow{1}]_i} \ .$$

Furthermore,

$$C_\phi \mathbf{g}(\mathbf{X}, \mathbf{r}^*) \geq C_\phi \mathbf{g}(\mathbf{X}, \mathbf{r}') \geq \exp(-O(\min(k, \ell')) \log n) \mathbb{P}(\mathbf{p}, \phi) \ . \tag{18}$$

Substituting values of $\mathbf{r}^*$ in $\mathbf{g}(\mathbf{X}, \mathbf{r}^*)$ we get,

$$\mathbf{g}(\mathbf{X}, \mathbf{r}^*) = n \sum_{i \in [\ell']} [\mathbf{X}\overrightarrow{1}]_i \mathbf{r}^*_i \log \mathbf{r}^*_i - \sum_{i \in [\ell'], j \in [0,k]} \mathbf{X}_{ij} \log \mathbf{X}_{ij} + \sum_{i \in [\ell']} [\mathbf{X}\mathbf{1}]_i \log [\mathbf{X}\mathbf{1}]_i \tag{19}$$

We now construct a pseudo distribution $\mathbf{q}$ by rounding down the probability values into set $\mathbf{R}$ as follows: for each $i \in [\ell]$, the number of elements with probability value $\mathbf{r}_i$ is equal to $\sum_{\{i' \in [\ell'] \mid \lfloor \mathbf{r}^*_{i'} \rfloor_{\mathbf{R}} = \mathbf{r}_i\}} [\mathbf{X}\overrightarrow{1}]_{i'}$, where $\lfloor y \rfloor_{\mathbf{R}} \stackrel{\text{def}}{=} \max_{\{x \in \mathbf{R} \mid x \leq y\}} x$. Define a new solution $\mathbf{X}' \in \mathbb{R}^{[\ell] \times [0,k]}$ as follows: $\mathbf{X}'_{ij} \stackrel{\text{def}}{=} \sum_{\{i' \in [\ell'] \mid \lfloor \mathbf{r}^*_{i'} \rfloor_{\mathbf{R}} = \mathbf{r}_i\}} \mathbf{X}_{i'j}$. Let $\beta = \sum_{i' \in [\ell']} \lfloor \mathbf{r}^*_{i'} \rfloor_{\mathbf{R}} [\mathbf{X}\overrightarrow{1}]_{i'} = \sum_{i \in [\ell]} \mathbf{r}_i [\mathbf{X}'\overrightarrow{1}]_i$ and note that $\|\mathbf{q}\|_1 = \beta$. Now note that the following inequality immediately holds,

$$\mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1}, \phi\right) = \mathbb{P}(\mathbf{q}, \phi) \beta^{-n} \text{ and } \mathbb{P}(\mathbf{q}, \phi) \geq C_\phi \mathbf{g}(\mathbf{X}', \mathbf{r}). \tag{20}$$

Now consider,

$$\begin{aligned} \log \mathbf{g}(\mathbf{X}', \mathbf{r}) &= \sum_{i \in [\ell], j \in [0,k]} (\mathbf{m}_j \log \mathbf{r}_i \mathbf{X}'_{ij} - \mathbf{X}'_{ij} \log \mathbf{X}'_{ij}) + \sum_{i \in [\ell]} [\mathbf{X}'\mathbf{1}]_i \log [\mathbf{X}'\mathbf{1}]_i, \\ &\geq \sum_{i \in [\ell'], j \in [0,k]} \mathbf{m}_j \log \lfloor \mathbf{r}^*_{i'} \rfloor_{\mathbf{R}} \mathbf{X}_{ij} - \sum_{i \in [\ell'], j \in [0,k]} \mathbf{X}_{ij} \log \mathbf{X}_{ij} + \sum_{i \in [\ell']} [\mathbf{X}\mathbf{1}]_i \log [\mathbf{X}\mathbf{1}]_i \ . \end{aligned} \tag{21}$$

Consider the first term in the above expression,

$$\sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log\lfloor \mathbf{r}^*_{i'}\rfloor_{\mathbf{R}} \mathbf{X}_{ij} = \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \frac{\lfloor \mathbf{r}^*_i\rfloor_{\mathbf{R}}}{\beta} \mathbf{X}_{ij} + \log\beta \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \mathbf{X}_{ij},$$
$$= \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \frac{\lfloor \mathbf{r}^*_i\rfloor_{\mathbf{R}}}{\beta} \mathbf{X}_{ij} + n\log\beta \tag{22}$$

To simplify the above expression we define α_i as, $\mathbf{r}^*_i(1+\alpha_i) = \frac{\lfloor \mathbf{r}^*_i\rfloor_{\mathbf{R}}}{\beta}$, then we get that,

$$\sum_{i\in[\ell']} \mathbf{r}^*_i(1+\alpha_i)[\mathbf{X}\overrightarrow{1}]_i = \sum_{i\in[\ell']} \frac{\lfloor \mathbf{r}^*_i\rfloor_{\mathbf{R}}}{\beta}[\mathbf{X}\overrightarrow{1}]_i = 1.$$

However, we also have that, $\sum_{i\in[\ell']} \mathbf{r}^*_i[\mathbf{X}\overrightarrow{1}]_i = 1$. Combining both we get that, $\sum_{i\in[\ell']} \alpha_i \mathbf{r}^*_i[\mathbf{X}\overrightarrow{1}]_i = 0$. Substituting this equality in Equation 22, we get,

$$\begin{aligned}
\sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log\lfloor \mathbf{r}^*_{i'}\rfloor_{\mathbf{R}} \mathbf{X}_{ij} &= \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \frac{\lfloor \mathbf{r}^*_i\rfloor_{\mathbf{R}}}{\beta} \mathbf{X}_{ij} + n\log\beta \\
&= \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i(1+\alpha_i)\mathbf{X}_{ij} + n\log\beta \\
&= \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i\mathbf{X}_{ij} + \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log(1+\alpha_i)\mathbf{X}_{ij} + n\log\beta \\
&\geq \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i\mathbf{X}_{ij} + \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j (\alpha_i - \alpha_i^2)\mathbf{X}_{ij} + n\log\beta \\
&\geq \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i\mathbf{X}_{ij} - \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \alpha_i^2\mathbf{X}_{ij} + n\log\beta\,.
\end{aligned} \tag{23}$$

In the fourth inequality we used the inequality $\log(1+x) \geq x - x^2$, when $x \in (0,1)$. In the last inequality we used $\sum_{i\in[\ell']} \alpha_i \mathbf{r}^*_i[\mathbf{X}\overrightarrow{1}]_i = 0$. Now note that each of the α_i satisfy, $|\alpha_i| \leq O(\alpha)$. Substituting it in the above expression we get,

$$\sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log\lfloor \mathbf{r}^*_{i'}\rfloor_{\mathbf{R}} \mathbf{X}_{ij} \geq \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i\mathbf{X}_{ij} - \alpha^2 n + n\log\beta, \tag{24}$$

where in the above inequality we used $\sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j\mathbf{X}_{ij} = \sum_{j\in[0,k]} \mathbf{m}_j\phi_j = n$. Combining the above inequality with Equation (21), we get,

$$\begin{aligned}
\log\mathbf{g}(\mathbf{X}',\mathbf{r}) &\geq \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log\lfloor \mathbf{r}^*_{i'}\rfloor_{\mathbf{R}} \mathbf{X}_{ij} - \sum_{i\in[\ell'],j\in[0,k]} \mathbf{X}_{ij}\log\mathbf{X}_{ij} + \sum_{i\in[\ell']} [\mathbf{X}\mathbf{1}]_i \log[\mathbf{X}\mathbf{1}]_i, \\
&\geq \sum_{i\in[\ell'],j\in[0,k]} \mathbf{m}_j \log \mathbf{r}^*_i\mathbf{X}_{ij} - \sum_{i\in[\ell'],j\in[0,k]} \mathbf{X}_{ij}\log\mathbf{X}_{ij} + \sum_{i\in[\ell']} [\mathbf{X}\mathbf{1}]_i \log[\mathbf{X}\mathbf{1}]_i - \alpha^2 n + n\log\beta, \\
&= \log\mathbf{g}(\mathbf{X},\mathbf{r}^*) - \alpha^2 n + n\log\beta
\end{aligned} \tag{25}$$

Combining everything we get the following,

$$\begin{aligned}
\log\mathbb{P}(\mathbf{p},\phi) - \log\mathbb{P}\left(\frac{\mathbf{q}}{\|\mathbf{q}\|_1},\phi\right) &= \log\mathbb{P}(\mathbf{p},\phi) - \log\mathbb{P}(\mathbf{q},\phi) + n\log\beta, \\
&\leq O(\min(k,\ell')\log n) + \log(C_\phi\mathbf{g}(\mathbf{X},\mathbf{r}^*)) - \log(C_\phi\mathbf{g}(\mathbf{X}',\mathbf{r})) + n\log\beta, \\
&\leq O(\min(k,\ell')\log n) + \alpha^2 n\,.
\end{aligned}$$

In the first, second and third inequalities we use Equation (18), Equation (20) and Equation (25) respectively. We conclude the proof. □

H Proof for Lemma 4.6

We firstly show a special case of Lemma 4.6

Lemma H.1. *If* $\boldsymbol{A}' = \mathrm{swap}(\boldsymbol{A}, i_1, i_2, j_1, j_2, \epsilon)$ *where* $\boldsymbol{A}, \boldsymbol{A}' \in \mathbb{R}^{\ell \times [0,k]}$ *and* $\boldsymbol{A} \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,\mathrm{frac}}$*, then,* $\boldsymbol{A}' \in \boldsymbol{Z}_{\boldsymbol{R}}^{\phi,\mathrm{frac}}$ *and* $\boldsymbol{g}(\boldsymbol{A}') \geq \exp(-4\epsilon \log n)\, \boldsymbol{g}(\boldsymbol{A})$.

Proof. Since $\mathbf{A}' = \mathrm{swap}(\mathbf{A}, i_1, i_2, j_1, j_2, \epsilon)$, we have $\mathbf{A}'^\top \overrightarrow{1} = \mathbf{A}^\top \overrightarrow{1}$ and $\mathbf{A}' \overrightarrow{1} = \mathbf{A} \overrightarrow{1}$. As $\mathbf{A} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}}$, we immediately get that $\mathbf{A}' \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}}$ and we also have

$$
\begin{aligned}
\frac{\mathbf{g}(\mathbf{A}')}{\mathbf{g}(\mathbf{A})} &= \frac{\exp\Big(\sum_{i\in[\ell],j\in[0,k]} \left[\mathbf{C}_{ij}\mathbf{A}'_{ij} - \mathbf{A}'_{ij}\log \mathbf{A}'_{ij}\right] + \sum_{i\in[\ell]} [\mathbf{A}'\mathbf{1}]_i \log[\mathbf{A}'\mathbf{1}]_i\Big)}{\exp\Big(\sum_{i\in[\ell],j\in[0,k]} \left[\mathbf{C}_{ij}\mathbf{A}_{ij} - \mathbf{A}_{ij}\log \mathbf{A}_{ij}\right] + \sum_{i\in[\ell]} [\mathbf{A}\mathbf{1}]_i \log[\mathbf{A}\mathbf{1}]_i\Big)} \\
&= \frac{\exp\Big(\sum_{i\in[\ell],j\in[0,k]} \left[\mathbf{C}_{ij}\mathbf{A}'_{ij} - \mathbf{A}'_{ij}\log \mathbf{A}'_{ij}\right]\Big)}{\exp\Big(\sum_{i\in[\ell],j\in[0,k]} \left[\mathbf{C}_{ij}\mathbf{A}_{ij} - \mathbf{A}_{ij}\log \mathbf{A}_{ij}\right]\Big)} \\
&\geq \frac{\exp\Big(\left(\mathbf{C}_{i_1j_1} - \mathbf{C}_{i_1j_2} - \mathbf{C}_{i_2j_1} + \mathbf{C}_{i_2j_2}\right)\epsilon\Big)}{\exp\Big(\mathbf{A}'_{i_1j_1}\log \mathbf{A}'_{i_1j_1} - \mathbf{A}_{i_1j_1}\log \mathbf{A}'_{i_1j_1} + \mathbf{A}'_{i_2j_2}\log \mathbf{A}'_{i_2j_2} - \mathbf{A}_{i_2j_2}\log \mathbf{A}'_{i_2j_2}\Big)} \\
&\geq \frac{1}{\exp\Big(2\epsilon\log(n+1)\Big)} \geq \exp\Big(-4\epsilon\log n\Big),
\end{aligned}
$$

where the last inequality holds because because $\frac{d}{dx}\left(x\log(x)\right) = (\log(x)+1)$ and

$$
\begin{aligned}
\mathbf{C}_{i_1j_1} - \mathbf{C}_{i_1j_2} - \mathbf{C}_{i_2j_1} + \mathbf{C}_{i_2j_2} &= \mathbf{m}_{j_1}\mathbf{r}_{i_1} - \mathbf{m}_{j_2}\mathbf{r}_{i_1} - \mathbf{m}_{j_1}\mathbf{r}_{i_2} + \mathbf{m}_{j_2}\mathbf{r}_{i_2} \\
&= (\mathbf{m}_{j_2} - \mathbf{m}_{j_1})(\mathbf{r}_{i_2} - \mathbf{r}_{i_1}) \geq 0.
\end{aligned}
$$

□

Proof for Lemma 4.6. Since $\mathbf{A}'$ is x-swap distance from $\mathbf{A}$, there exists a set of parameters denoted by $\{(i_1^{(s)}, i_2^{(s)}, j_1^{(s)}, j_2^{(s)}, \epsilon^{(s)})\}_{s\in[t]}$, where $\sum_{s\in[t]} \epsilon^{(s)} \leq x$, s.t $\mathbf{A}^{(s)} = \mathrm{swap}(\mathbf{A}^{(s-1)}, i_1^{(s)}, i_2^{(s)}, j_1^{(s)}, j_2^{(s)}, \epsilon^{(s)})$ for $s \in [t]$, where $\mathbf{A}^{(0)} = \mathbf{A}$, $\mathbf{A}^{(t)} = \mathbf{A}'$.

We apply Lemma H.1 on $\mathbf{A}^{(i)}$ and $\mathbf{A}^{(i-1)}$ for each $i \in [t]$, and we have

$$
\mathbf{A}^{(0)} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}} \Rightarrow \mathbf{A}^{(1)} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}} \Rightarrow \mathbf{A}^{(2)} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}} \Rightarrow \cdots \Rightarrow \mathbf{A}^{(t)} \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}},
$$

$$
\mathbf{g}(\mathbf{A}^{(t)}) \geq \exp\Big(-4\epsilon^{(t)}\log n\Big)\, \mathbf{g}(\mathbf{A}^{(t-1)}) \geq \cdots \geq \exp\left(-4\left(\sum_{s\in[t]} \epsilon^{(s)}\right)\log n\right)\, \mathbf{g}(\mathbf{A}^{(0)}),
$$

which implies $\mathbf{A}' \in \mathbf{Z}_{\mathbf{R}}^{\phi,\mathrm{frac}}$ and $\mathbf{g}(\mathbf{A}') \geq \exp\left(-O\left(x\log n\right)\right)\, \mathbf{g}(\mathbf{A})$. We conclude the proof. □