Learning Discrete Distributions from Untrusted Batches

Mingda Qiao[1] and Gregory Valiant*[2]

1 **Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China**
qmd14@mails.tsinghua.edu.cn

2 **Computer Science Department, Stanford University, Stanford, USA**
valiant@stanford.edu

Abstract

We consider the problem of learning a discrete distribution in the presence of an ϵ fraction of malicious data sources. Specifically, we consider the setting where there is some underlying distribution, p, and each data source provides a batch of $\geq k$ samples, with the guarantee that at least a $(1-\epsilon)$ fraction of the sources draw their samples from a distribution with total variation distance at most η from p. We make no assumptions on the data provided by the remaining ϵ fraction of sources – this data can even be chosen as an adversarial function of the $(1-\epsilon)$ fraction of "good" batches. We provide two algorithms: one with runtime exponential in the support size, n, but polynomial in k, $1/\epsilon$ and $1/\eta$ that takes $O((n+k)/\epsilon^2)$ batches and recovers p to error $O(\eta + \epsilon/\sqrt{k})$. This recovery accuracy is information theoretically optimal, to constant factors, even given an infinite number of data sources. Our second algorithm applies to the $\eta = 0$ setting and also achieves an $O(\epsilon/\sqrt{k})$ recover guarantee, though it runs in $\text{poly}((nk)^k)$ time. This second algorithm, which approximates a certain tensor via a rank-1 tensor minimizing ℓ_1 distance, is surprising in light of the hardness of many low-rank tensor approximation problems, and may be of independent interest.

1998 ACM Subject Classification Probability and Statistics

Keywords and phrases robust statistics, information-theoretic optimality

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.47

1 Introduction

Consider the following real-world problem: suppose there are millions of people texting away on their phones, and you wish to learn the distribution of words corresponding to a given mis-typed word, or the distribution of words that follow a given sequence, etc. The challenge of this setup is twofold. First, each person provides far too little data to accurately learn these distributions based solely on one person's data, hence a successful learning or estimation algorithm must combine data from different sources. Second, these sources are heterogeneous – some people have wider fingers than others, and the nature of typos likely differs between people. Further complicating this heterogeneity is the very real possibility that a small but not negligible fraction of the sources might be operated by adversarial agents whose goal is to embarrass the learning algorithm either as a form of corporate sabotage or as a publicity stunt. In much the same way as "Google bombing" or "link bombing" was used to associate specific websites with certain terms – perhaps the most famous of which

* Gregory's work is supported by ONR award N00014-17-1-2562 and NSF CAREER award CCF-1351108.

© Mingda Qiao and Gregory Valiant;
licensed under Creative Commons License CC-BY
9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 47; pp. 47:1–47:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

resulting in the George W. Bush biography being the top Google and Yahoo! hit when searching for the term "miserable failure" back in Sept. 2006 – it seems likely that certain groups of people would collectively attempt to influence the auto-correct or auto-suggest responses to certain misspellings.

This general problem of learning or estimation given data supplied by a large number of individuals has gained attention in the more practical communities under the keyword *Federated Learning* (see e.g. [24, 17] and Google research blog post [23]), and raises a number of pertinent questions: What notions of privacy can be maintained while leveraging everyone's data? Do we train a single model or try to personalize models for each user?

In this work, we consider a basic yet fundamental problem in this space: learning a discrete distribution given access to batches of samples, where an unknown $(1-\epsilon)$ fraction of batches are drawn i.i.d. from distributions that each has total variation (equivalently, ℓ_1) distance at most η from some target distribution, p, and we make no assumptions about the data contained in the remaining ϵ fraction of batches. The data in this "bad" ϵ fraction of the batches can even be chosen adversarially as a function of the "good" data. This problem is also a natural problem in the space of "robust statistics", and the recent line of recent work from the theory community on estimation and learning with untrusted data (e.g. [20, 7, 31, 6, 30, 8]), and we outline these connections in Section 1.2.

We begin by summarizing our main results and discuss the connections with related work. We conclude the introduction with a discussion of several relevant directions for future research.

1.1 Summary of Results

The following theorem characterizes our tight information theoretic result for robustly learning a discrete distribution in the setting where a certain fraction of batches of data are arbitrarily corrupted:

▶ **Theorem 1.** *Let p denote a distribution supported on n domain elements, and fix parameters $\epsilon \in (0, 1/900)$, $\eta > 0$, failure probability $\delta \in (0,1)$, and integer $k \geq 1$. Suppose we have access to $m = O((n + k + \log(1/\delta))/\epsilon^2)$ batches of data such that at least $m(1-\epsilon)$ of the batches consist of $\geq k$ i.i.d. draws from some distribution with ℓ_1 distance at most η from p. There exists an algorithm that runs in time* $\mathrm{poly}\,(2^n, k, 1/\epsilon, 1/\eta, \log(1/\delta))$ *and, with probability at least $1-\delta$, returns a distribution $\hat{p}$ satisfying $\|p-\hat{p}\|_1 = O(\eta + \epsilon/\sqrt{k})$, where the "O" notation hides an absolute constant.*

The recovery guarantees of the above theorem are information theoretically optimal, even given infinite data, as the following lower bound formalizes:

▶ **Theorem 2.** *In the setup of Theorem 1 for integers $k \geq 1$ and $n \geq 2$ and parameters $\epsilon \in (0, 1/2)$ and $\eta \in [0, 1/4)$, no algorithm can, with probability greater than $1/2$, return a distribution $\hat{p}$ satisfying $\|p - \hat{p}\|_1 < 2\eta + \epsilon/\sqrt{2k}$, even in the limit as the number of batches, m, tends to infinity.*

To provide some intuition for these results, consider the setting where $\eta = 0$, namely where the $(1-\epsilon)$ fraction of "good" batches all consist of k i.i.d. draws from p. In this case, the above results claim an optimal recovery error of $\Theta(\epsilon/\sqrt{k})$. Intuitively, this is due to the following "tensorization" property of the total variation distance, which we show in Appendix B: given two distributions p and q with $\|p-q\|_1 = \alpha = O(1/\sqrt{k})$, the ℓ_1 distance between their kth tensor products, $\|p^{\otimes k} - q^{\otimes k}\|_1$, is at least $\Theta(\alpha\sqrt{k})$. Here, the kth tensor product $p^{\otimes k}$ denotes the n^k sized object, indexed by a k-tuple $i_1, \ldots, i_k$, with

$p^{\otimes k}_{i_1,\ldots,i_k} = \prod_{j=1}^{k} p(i_j)$. This lower bound on the distance between the kth tensor products implies that, even though the adversary can alter the observed k-fold product distribution of p by ϵ (w.r.t. the ℓ_1 or total variation distance), any two k-tensors with distance $\leq \epsilon$ that each correspond to the kth tensor products of distributions will correspond to distributions whose distance is only $O(\epsilon/\sqrt{k})$, motivating our claimed recovery guarantees.

The algorithm to which the guarantees of Theorem 1 apply proceeds by reducing the problem at hand to $\leq 2^n$ instances of the problem of learning a Bernoulli random variable in this setting of untrusted batches. Specifically, for each of the $\leq 2^n$ possible subsets of the observed domain elements, the algorithm attempts to estimate the probability mass that p assigns to that set. The algorithm then combines these $\leq 2^n$ estimates to estimate p. In Section 1.3, we discuss the possibility of an efficient variant of this algorithm that proceeds via estimating these weights for only a polynomial number of subsets. We also note that the requirement that $\epsilon \leq 1/900$ can be relaxed slightly at the expense of readability of the proof.

Our proof of the lower bound (Theorem 2) proceeds by providing a construction of an explicit pair of indistinguishable instances, one corresponding to a distribution p, one corresponding to a distribution q, with $\|p-q\|_1 = 4\eta + 2\epsilon/\sqrt{2k}$. Each instance consists of two parts – a distribution from which the "good" batches of data are drawn, and a distribution over k-tuples of samples from which the ϵ fraction of "bad" batches are drawn. For the pair of instances constructed, the mixture of the good and bad batches corresponding to p is identical to the mixture of the good and bad batches corresponding to q.

In addition to the algorithm of Theorem 1, which runs in time exponential in the support size, we provide a second algorithm in the setting where $\eta = 0$, with runtime $(nk)^{O(k)}$ that also achieves the information theoretically optimal recovery error of $O(\epsilon/\sqrt{k})$.

▶ **Theorem 3.** *As in the setup of Theorem 1, given $m = (nk)^{O(k)} \log(1/\delta)/\epsilon^2$ batches of samples, of which a $(1-\epsilon)$ fraction consist of $\geq k$ i.i.d. draws from a distribution p, supported on $\leq n$ elements, there is an algorithm that runs in time* $\mathrm{poly}((nk)^k, 1/\epsilon, \log(1/\delta))$ *which, with probability at least $1-\delta$, returns a distribution $\hat{p}$ such that $\|p-\hat{p}\|_1 = O(\epsilon/\sqrt{k})$.*

The algorithm to which the above theorem applies proceeds by forming the k-tensor corresponding to the empirical distribution over the m batches, with each batch regarded as a k-tuple. The algorithm then efficiently computes a rank-1 k-tensor which approximates this empirical tensor in the (element-wise) ℓ_1 sense. The ability to efficiently (in time polynomial in the size of the k-tensor) compute this rank-1 approximation is rather surprising, and crucially leverages the structure of the empirical k-tensor guaranteed by the setup of our problem. Indeed the general problem of finding the best rank-1 approximation (in either an ℓ_1 or ℓ_2 sense) of a k-tensor is NP-hard for $k \geq 3$ [13]. Additionally, even for various "nice" random distributions of 3-tensors, efficient algorithms for related rank-1 approximation problems would yield efficient algorithms for recovering planted cliques of size $o(\sqrt{n})$ from $G(n, 1/2)$ [9].[1]

1.2 Related Work

There are a number of relevant lines of related work, including work from the theoretical computer science and information theory communities on learning, estimating and testing properties of distributions or collections of distributions, the classical work on "robust

[1] Specifically, [9] shows that given an $n \times n \times n$ 3-tensor A constructed from an instance of planted-clique, the ability to efficiently find a unit vector v that nearly maximizes $\max_{\|v\|=1} \sum_{i,j,k} v_i v_j v_k A_{i,j,k}$ would yield an algorithm for finding cliques of size at least $O(n^{1/3}\,\mathrm{polylog}(n))$ planted in $G(n, 1/2)$.

statistics", the recent line of work from the computer science community on robust learning and estimation with untrusted data and "adversarial" machine learning, and the very recent work from the more applied community on "federated learning". From a technical perspective, there are also connections between our work and the body of work on tensor factorizations, and low rank matrix and tensor approximations with respect to the ℓ_1 norm. We briefly summarize these main lines of related work.

1.2.1 Learning and Testing Discrete Distributions

The problem of learning a discrete distribution given access to independent samples has been intensely studied over the past century (see, e.g. [10, 18] and the references therein). For distributions supported on n elements, given access to k i.i.d. samples, minimax optimal loss rates as a function of n and k have been considered for various loss functions, including KL-divergence [4], ℓ_2 [15], χ^2 loss [15]. For ℓ_1 error (equivalently, "total variation distance" or "statistical distance"), it is easy to show that the expected worst-case error is $\Theta(\sqrt{n/k})$, and the recent work [15] establishes the exact first-order coefficients of this loss. Beyond this worst-case setting, there has also been significant work considering this learning problem with the goal of developing "instance optimal" algorithms that leverage whatever structure might be present in the given distribution (see e.g. [28, 35]).

In the context of work on testing distributional properties from the theoretical computer science community, the work closest to this current paper is the work of Levi, Ron, and Rubinfeld [21], which considers the following task: given access to independent draws from each of m distributions, distinguish whether all m distributions are identical (or very close), versus the case that there is significant variation between the distributions – namely that the average distance to any single distribution is at least a constant. We note that in this work, as in much of the distribution testing literature, the results are extremely sensitive to assumption that in the "yes" case, the distributions are all extremely close to one distribution. For example, the task of distinguishing whether the distributions have average distance at most α_1 from a single distribution, versus having average distance at least α_2, seems to be a significantly harder problem when α_1 is not asymptotically smaller than α_2.[2] In a related vein, the recent work [32] considers the setting of drawing k i.i.d. samples from each of m possibly heterogeneous distributions, and shows that the *set* of such distributions can be accurately learned. For example, if each distribution is a Bernoulli random variable with the ith distribution corresponding to some probability p_i, the histogram of the p_i's can be recovered to error $O(1/k)$, rather than the $\Theta(1/\sqrt{k})$ that would be given by the empirical estimates. Both [21] and [32], however, crucially rely on the assumption that the data consists of independent draws from the m distributions, and the algorithms and results do not extend to the present setting where some non-negligible fraction of the data is arbitrary/adversarial.

1.2.2 Robust Statistics, Learning with Unreliable Data, and Adversarial Learning

The problem of estimation and learning in the presence of contaminated or outlying data points has a history of study in the Statistics community, dating back to early work of Tukey [33] (see also the surveys [14, 12]). Recently, these problems have gained attention

[2] For example, testing whether two distributions supported on $\leq n$ elements are identical, vs have distance at least 0.1 requires $\Theta(n^{2/3})$ samples, whereas distinguishing whether the two distributions have distance ≤ 0.1 versus ≥ 0.9 requires $\Theta(n/\log n)$ samples. [2, 5, 34].

from both the applied and theoretical computer science communities. On the applied side, the interest in "adversarial" machine learning stems from the very real vulnerabilities of many deployed learning systems to "data poisoning" attacks in which an adversary selectively alters or plants a small amount of carefully chosen training data that significantly impacts the resulting trained model (see e.g. [27, 26, 36]).

On the theory side, recent work has revisited some of the classical robust statistics settings with an eye towards 1) establishing the information theoretic dependencies between the fraction of corrupted data, dimension of the problem, and achievable accuracy of returned model or estimate, and 2) developing computationally tractable algorithms that approach or achieve these information theoretic bounds. These recent works include [20, 7, 8] who consider the problem of robustly estimating the mean and covariance of high-dimensional Gaussians (and other distributions such as product distributions), and [31] who consider a sparse ratings aggregation setting. Other learning problems, such as robustly learning halfspaces [16], robust linear regression [3] and more general convex optimization [6] have also been considered in similar settings where some fraction of the data is drawn from a distribution of interest, and no assumptions are made about the remaining data. This latter work and [25] also considers several models for which strong positive results can be attained even when the majority of the data is arbitrary (i.e. $\epsilon > 1/2$).

The present setting, in which data arrives in batches, with some batches corresponding to "good" data represents a practically relevant instance of this more general robust learning problem, and we are not aware of previous work from the theory community that explicitly considers it.[3] The recent attention on "federated learning" [24, 17, 23] focuses on a similar setting where data is presented in batches (corresponding to individual users), although the current emphasis is largely on privacy and communication concerns, as opposed to robustness.

1.2.3 Low Rank Approximations in ℓ_1

The algorithm underlying Theorem 3 efficiently computes a rank-1 approximation of the empirical k-tensor of data, where the recovered rank-1 tensor is close in the (element-wise) ℓ_1 sense. Both the problems of computing the best low-rank ℓ_1 approximation of a *matrix*, and the problems of computing the best rank-1 approximation of a k-tensor for $k \geq 3$ are NP-hard [11, 13]. Nevertheless, recent work has provided efficient algorithms for returning good (in a competitive-analysis sense) low-rank approximations of matrices in the ℓ_1 norm [29], and for efficiently recovering low-rank approximations of special classes of tensors, including those with random or "incoherent" factors (see e.g. [19, 22]). Still, for many tensor decomposition problems over various families of random nearly low-rank tensors – such as those obtained from instances of planted-clique [9] – these decomposition problems remain mysterious.

1.3 Future directions and discussion

Two concrete open directions raised by this work are 1) understanding the computational hardness of this basic learning question, and 2) extending the information theoretic and algorithmic results to the setting where a *minority* of the batches of data are "good", which generalizes the problem of robustly learning a mixture of distributions. Before discussing these problems, we note that there are a number of other practically relevant related questions,

[3] One could view each batch of k samples as a single k-dimensional draw from a product of multinomial distributions, or as a k-sparse sample from an n-dimensional product distribution, although the results of previous work do not yield strong results for these settings.

including the extent to which this problem can be solved while maintaining some notion of privacy for each batch/user, and considering other basic learning and estimation tasks in this batched robust setting.

1.3.1 An Efficient Algorithm?

Recall that the algorithm of Theorem 1 has a linear sample complexity (treating ϵ as fixed), yet requires a runtime exponential in n. The algorithm of Theorem 3 requires more data, but runs in time $(nk)^{O(k)}$, and only applies to the clean but less practically relevant setting where $\eta = 0$. One natural question is whether there exists an algorithm that achieves both the linear sample complexity and the $(nk)^{O(k)}$ runtime. Alternately, does there exist an algorithm with data requirements and runtime that are polynomial in all the parameters, $n, k, 1/\epsilon$, and achieves the optimal recovery guarantees even in the $\eta = 0$ regime? Or are there natural barriers to such efficiency or connections to other problems that we believe to be computationally intractable?

One natural approach to yielding an efficient algorithm is via a variant of our first algorithm. At a high level, our algorithm proceeds by recovering $\epsilon/\sqrt{k}$-accurate estimates of the probability mass of each of the 2^n possible subsets of the n domain elements, and then recovering a distribution consistent with these estimates via a linear program. A natural approach to improving the running time of the algorithm is to find an efficient separation oracle for this linear program. Alternately, one could even imagine recovering $\epsilon/\sqrt{k}$-accurate estimates of the mass of a random poly(k, n) sized set of subsets of the domain and then using these to recover the distribution. If the error in each estimate were independent, say distributed according to $N(0, \epsilon^2/k)$, then a polynomial (and even linear!) number of such measurements would suffice to recover the distribution to error $O(\epsilon/\sqrt{k})$. On the other hand, if an adversary is allowed to choose the errors in each measurement, trivially, an exponential number would be required. In our setting, however, the adversaries are somewhat limited in their ability to corrupt the measurements, and it is plausible that such an approach could work, perhaps both in practice and theory.

1.3.2 The small-α regime

Our positive results (Theorems 1 and 3) assume that the fraction of adversarial data, ϵ, is relatively small (at most of order 10^{-3}). While this assumption can be slightly relaxed by carefully adjusting the constants in the proofs, it is also natural to ask: what can we do if ϵ is much larger and even close to 1? This regime is examined in [6, 25]. It is clearly impossible to estimate the distribution in question to a nontrivial precision in the setting where $\epsilon \geq 1/2$ due to the symmetry between the real distribution and the one chosen by the adversary. Nevertheless, two learning frameworks were proposed for this case in [6]: the *list-decodable learning* framework (first introduced by Balcan et al. [1]), in which the learning algorithm is allowed to output multiple answers to the learning problem (perhaps $1/(1-\epsilon)$), and the *semi-verified* model, where a small amount of reliable/verified data is available. It seems plausible that variants of our algorithms that attempt to recover a rank $1/(1-\epsilon)$ approximation to the empirical tensor might be successful, though the algorithm and analysis are certainly not immediate.

1.4 Organization of Paper

In Section 2 we establish our information theoretic lower bound, Theorem 2, which bounds the accuracy of the recovered distribution and applies even in the infinite data setting. Section 3 describes our information theoretically optimal algorithm that has runtime exponential in the

support size, n, and establishes Theorem 1. Section 4 describes our more efficient algorithm, with runtime $(nk)^{O(k)}$, which directly approximates the k-tensor of observations via a rank-1 tensor, establishing Theorem 3. We conclude this section with a brief summary of the notation that will be used throughout the remainder of the paper.

1.5 Notation

Throughout, we use p to denote a discrete distribution of support size at most n. Without loss of generality, we assume that the support of p is $[n] = \{1, 2, \ldots, n\}$. Thus, p can also be interpreted as a probability vector $(p_1, p_2, \ldots, p_n) \in \mathbb{R}^n$, where p_i denotes the probability of element i. For a set $S \subseteq [n]$, we adopt the shorthand notation $p(S) = \sum_{i \in S} p_i$ to denote the total probability mass assigned to elements of S.

Let $p^{\otimes k}$ denote the k-fold product distribution corresponding to p, which is the kth tensor power of vector p. Each entry of $p^{\otimes k}$ is indexed by k indices $i_1, i_2, \ldots, i_k \in [n]$, and

$$p^{\otimes k}_{i_1, i_2, \ldots, i_k} = p_{i_1} p_{i_2} \cdots p_{i_k}.$$

More generally, a n^k-tensor is a k-dimensional array in which each entry is indexed by k elements in $[n]$. The *marginal* of n^k-tensor A is the vector $a \in \mathbb{R}^n$ defined as

$$a_i = \sum_{i_2, \ldots, i_k \in [n]} A_{i, i_2, \ldots, i_k}.$$

The i-th *slice* of A is the n^{k-1}-tensor obtained by restricting the first index of A to i.

A probability vector (resp. probability tensor) is a vector (resp. tensor) whose entries are nonnegative and sum to 1. Note that a probability n^k-vector defines a probability distribution on $[n]^k$. Moreover, its marginal is the marginal distribution of the first component, and its i-th slice (after normalization) is the conditional distribution of the other $k-1$ components given that the first component equals i.

The learning algorithm is given m batches of data, each of which is a k-tuple in $[n]^k$. Among the m batches, $m(1-\epsilon)$ are "good" in the sense that each of them is drawn from $\tilde{p}^{\otimes k}$ for some distribution $\tilde{p}$ with $\Delta(p, \tilde{p}) \leq \eta$.[4] The other $m\epsilon$ batches are chosen arbitrarily after the $m(1-\epsilon)$ good batches are drawn. The objective is to output a distribution q such that $\Delta(p, q)$ is small. Here $\Delta(p, q)$ stands for the total variation distance between p and q, i.e., one half of the ℓ_1-distance between vectors p and q:

$$\Delta(p, q) \coloneqq \frac{1}{2} \|p - q\|_1 = \frac{1}{2} \sum_{i=1}^{n} |p_i - q_i| = \max_{S \subseteq [n]} [p(S) - q(S)].$$

2 Information Theoretic Lower Bound

In this section we establish Theorem 2, showing that it is impossible to learn p to an $o(\eta + \epsilon/\sqrt{k})$ precision in ℓ_1 distance, even for distributions supported on 2 domain elements. This lower bound holds even if the learning algorithm is given unlimited computation power and access to infinitely many batches, i.e., the input of the algorithm is simply the mixture distribution $(1-\epsilon)P + \epsilon N$, where P is a mixture of k-fold product distributions of distributions that are η-close to p in the total variation distance, and N is a probability n^k-tensor that corresponds to the distribution of the adversarial batches.

[4] The distribution $\tilde{p}$ may vary for different good batches.

▶ **Lemma 4.** *For integer $k \geq 1$ and parameters $\epsilon \in (0, 1/2)$ and $\eta \in [0, 1/4)$, there are Bernoulli distributions p, q, p', q', together with two probability 2^k-tensors $N^{(p)}$ and $N^{(q)}$, such that:*

1. $\Delta(p, q) = 2\eta + \epsilon/\sqrt{2k}$.
2. $\Delta(p, p') = \Delta(q, q') = \eta$. *item* $(1-\epsilon)p'^{\otimes k} + \epsilon N^{(p)} = (1-\epsilon)q'^{\otimes k} + \epsilon N^{(q)}$.

Lemma 4 implies that given the distribution $(1-\epsilon)p'^{\otimes k} + \epsilon N^{(p)} = (1-\epsilon)q'^{\otimes k} + \epsilon N^{(q)}$ as input, the algorithm cannot determine whether the underlying distribution is p or q. This establishes the $\Omega(\eta + \epsilon/\sqrt{k})$ lower bound in Theorem 2.

Proof of Lemma 4. Since $\epsilon/\sqrt{2k} < \epsilon < 1/2$, by Lemma 11, for Bernoulli distributions p' and q' with means $(1-\epsilon/\sqrt{2k})/2$ and $(1+\epsilon/\sqrt{2k})/2$, it holds that $\Delta(p', q') = \epsilon/\sqrt{2k}$ and $\Delta(p'^{\otimes k}, q'^{\otimes k}) \leq \epsilon$. Let p and q be Bernoulli distributions with means $(1-\epsilon/\sqrt{2k})/2 - \eta$ and $(1+\epsilon/\sqrt{2k})/2 + \eta$.[5] Then the distributions clearly satisfy the first two conditions.

Let A be the entrywise maximum of $p'^{\otimes k}$ and $q'^{\otimes k}$, i.e., $A_i = \max(p'^{\otimes k}_i, q'^{\otimes k}_i)$ for every $i \in [2]^k$. Let α denote the sum of entries in A. Then, $\alpha = 1 + \Delta(p'^{\otimes k}, q'^{\otimes k}) \leq 1 + \epsilon \leq 1/(1-\epsilon)$. Define $N^{(p)} = \left[A/\alpha - (1-\epsilon)p'^{\otimes k}\right]/\epsilon$ and $N^{(q)} = \left[A/\alpha - (1-\epsilon)q'^{\otimes k}\right]/\epsilon$. Then the third condition is met, and it remains to prove that $N^{(p)}$ and $N^{(q)}$ are probability tensors.

Note that the elements in $N^{(p)}$ sum to $(\alpha/\alpha - (1-\epsilon)\cdot 1)/\epsilon = 1$. Moreover, since $\alpha \leq 1/(1-\epsilon)$, $N^{(p)}_i = A_i/\alpha - (1-\epsilon)p'^{\otimes k}_i \geq (1-\epsilon)(A_i - p'^{\otimes k}_i) \geq 0$ for any $i \in [2]^k$. This shows that $N^{(p)}$ and $N^{(q)}$ are probability tensors and finishes the proof. ◀

3 An Information Theoretically Optimal Algorithm

In this section, we present an algorithm that approximates the distribution to an information theoretically optimal $O(\eta + \epsilon/\sqrt{k})$ accuracy using $O((n + k + \ln(1/\delta))/\epsilon^2)$ batches. The algorithm runs in time $\mathrm{poly}(2^n, k, 1/\epsilon, 1/\eta, \ln(1/\delta))$. In particular, the algorithm is computationally efficient if the distribution has a relatively small support.

Our approach is to reduce the problem to learning Bernoulli distributions: we estimate the probability mass of any subset of the support to an $O(\eta + \epsilon/\sqrt{k})$ accuracy, and output a distribution that is consistent with the measurements. Then the total variation distance between our output and the true distribution would also be upper bounded by $O(\eta + \epsilon/\sqrt{k})$.

In the following we define a subroutine that efficiently estimates $p(S)$, the probability mass that p assigns to set $S \subseteq [n]$. Given batches $x_1, x_2, \ldots, x_m \in [n]^k$, the algorithm counts the number of batches that contain exactly i elements in S $(0 \leq i \leq k)$ and obtains a distribution f^S over $\{0, 1, \ldots, k\}$. Then it outputs $(i+2)\eta$ for some $0 \leq i \leq 1/\eta - 4$ as the estimation, if there is a mixture of binomial distributions with success probabilities in interval $[i\eta, (i+4)\eta]$ such that its ℓ_1 distance to f^S is upper bounded by $O(\epsilon)$.

Note that the mathematical program (8) can be transformed to an equivalent linear program, and therefore can be efficiently solved.

The following lemma bounds the difference between the estimation $\mathsf{BinomialEst}((x_i)_{i\in[m]}, S, \epsilon, \eta)$ and $p(S)$.

▶ **Lemma 5.** *Suppose that $S \subseteq [n]$, $\epsilon \in (0, 1/900)$, and $\eta, \delta \in (0, 1)$. For some $m = O((k + \ln(1/\delta))/\epsilon^2)$, $x_1, x_2, \ldots, x_m$ are m batches chosen as in the setup of Theorem 1. With probability $1-\delta$ over the randomness of the $m(1-\epsilon)$ good batches,*

$$\left|\mathsf{BinomialEst}((x_i)_{i\in[m]}, S, \epsilon, \eta) - p(S)\right| \leq 3\eta + 60\epsilon/\sqrt{k}.$$

[5] p and q are well defined since $\epsilon/(2\sqrt{2k}) + \eta < 1/4 + 1/4 = 1/2$.

Algorithm 1: $\mathsf{BinomialEst}((x_i)_{i\in[m]}, S, \epsilon, \eta)$

Input: Batches $x_1, x_2, \ldots, x_m \in [n]^k$, set $S \subseteq [n]$, and parameters ϵ, η.
Output: An estimation of $p(S)$.

```
for i = 1, 2, ..., m do
    cnt_i ← Σ_{j=1}^{k} I{x_{i,j} ∈ S};
for i = 0, 1, ..., k do
    f_i^S ← (1/m) Σ_{j=1}^{m} I{cnt_j = i};
tot ← 4η/(ε/k);
for i = 0, 1, ..., 1/η − 4 do
    return (i + 2)η if the following mathematical program is feasible:
```

$$\begin{aligned} \text{find} \quad & \alpha_0, \alpha_1, \ldots, \alpha_{\text{tot}} \\ \text{subject to} \quad & \Delta\left(\sum_{j=0}^{\text{tot}} \alpha_j B\left(k, i\eta + j\epsilon/k\right), f^S\right) \le 2\epsilon \\ & \sum_{j=0}^{\text{tot}} \alpha_j = 1 \\ & \alpha_j \ge 0, \ \forall j \in \{0, 1, \ldots, \text{tot}\} \end{aligned} \tag{1}$$

Here the "O" notation hides an absolute constant.

Proof of Lemma 5. Let $I \subseteq [m]$ denote the indices of the $m(1-\epsilon)$ good batches and let $\overline{I}$ be its complement. Define distributions $\hat{p}^S$ and δ^S over $\{0, 1, \ldots, k\}$ as: $\hat{p}_i^S = \frac{1}{|I|}\sum_{j\in I} \mathbb{I}\{\text{cnt}_j = i\}$ and $\delta_i^S = \frac{1}{|\overline{I}|}\sum_{j\in \overline{I}} \mathbb{I}\{\text{cnt}_j = i\}$. Then the distribution f^S defined in Algorithm 1 can be rewritten as $f^S = (1-\epsilon)\hat{p}^S + \epsilon\delta^S$.

Concentration of $\hat{p}^S$. For each good batch $j \in I$, let θ_j denote the probability of set S in the actual distribution from which batch j is drawn. By the assumption that the underlying distribution of each good batch is η-close to the target p, we have $|\theta_j - p(S)| \le \eta$. Define p^S as the uniform mixture of binomial distributions $B(k, \theta_j)$ for $j \in I$, i.e., $p^S = \frac{1}{|I|}\sum_{j\in I} B(k, \theta_j)$. Since $\hat{p}^S$ denotes the frequency of $|I| = \Omega(m)$ samples, exactly one of which is drawn from $B(k, \theta_j)$ for each $j \in I$, for some $m = O((k + \ln(1/\delta))/\epsilon^2)$, it holds with probability $1-\delta$ that $\Delta(p^S, \hat{p}^S) \le \epsilon/2$. It follows that

$$\Delta(p^S, f^S) \le \Delta(p^S, \hat{p}^S) + \Delta(\hat{p}^S, f^S) = \Delta(p^S, \hat{p}^S) + \epsilon\Delta(\hat{p}^S, \delta^S) \le 3\epsilon/2.$$

Feasibility of mathematical program (8). Let $i^* = \lfloor p(S)/\eta \rfloor - 2$. We show that with probability $1-\delta$, the mathematical program (8) is feasible for $i = i^*$.
Since $|(i^*+2)\eta - p(S)| \le \eta$ and $|\theta_j - p(S)| \le \eta$, it holds for any $j \in I$ that $i^*\eta \le \theta_j \le (i^*+4)\eta$. Let $\tilde{\theta}_j$ be the value among $\{i^*\eta + t\epsilon/k : t \in \{0, 1, \ldots, \text{tot}\}\}$ that is closest to θ_j. By definition, we have $|\tilde{\theta}_j - \theta_j| \le \epsilon/(2k)$, which implies that

$$\Delta(B(k, \theta_j), B(k, \tilde{\theta}_j)) \le k \cdot \epsilon/(2k) = \epsilon/2.$$

Let $\tilde{p}^S = \frac{1}{|I|}\sum_{j\in I} B(k,\tilde{\theta}_j)$. Then we have

$$\Delta(p^S,\tilde{p}^S) \le \frac{1}{|I|}\sum_{j\in I}\Delta(B(k,\theta_j),B(k,\tilde{\theta}_j)) \le \epsilon/2,$$

and it follows that with probability $1-\delta$,

$$\Delta(\tilde{p}^S,f^S) \le \Delta(p^S,\tilde{p}^S)+\Delta(p^S,f^S) \le \epsilon/2+3\epsilon/2 = 2\epsilon.$$

Note that this naturally defines a feasible solution to the mathematical program (8) for $i=i^*$: for each $t\in\{0,1,\ldots,\mathrm{tot}\}$, $\alpha_t = \frac{1}{|I|}\sum_{j\in I}\mathbb{I}\left\{\tilde{\theta}_j = i^*\eta + t\epsilon/k\right\}$.

Approximation guarantee. Let x_0 denote $\mathsf{BinomialEst}((x_i)_{i\in[m]},S,\epsilon,\eta)$. In the following we prove that, with probability $1-\delta$, x_0 is a good approximation of $p(S)$.

Recall that $p^S = \frac{1}{|I|}\sum_{j\in I}B(k,\theta_j)$ is a mixture of binomial distributions with success probabilities in $[p(S)-\eta,p(S)+\eta]$. Moreover, by definition of procedure $\mathsf{BinomialEst}$, f^S is 2ϵ-close to a mixture of binomial distributions with success probabilities that lie in $[x_0-2\eta,x_0+2\eta]$. Then the inequality $\Delta(p^S,f^S)\le 3\epsilon/2$, which holds with probability $1-\delta$, further implies that the two mixtures above are 4ϵ-close to each other.

Let $\epsilon' = 60\epsilon/\sqrt{k}$. Note that our assumption $\epsilon < 1/900$ implies that $\epsilon' < 1/(15\sqrt{k})$. Now suppose for a contradiction that $|x_0-p(S)| > 3\eta+\epsilon'$. Without loss of generality, we have $x_0 > p(S)+3\eta+\epsilon'$, or equivalently, $x_0-2\eta > p(S)+\eta+\epsilon'$. Applying the contrapositive of Lemma 13 with parameters $\epsilon=\epsilon'$, $p=p(S)+\eta$ and $q=x_0-2\eta$ gives a contradiction. ◀

Now we prove our main theorem by Lemma 5 and a reduction to the estimation of Bernoulli random variables.

Proof of Theorem 1. We compute $\hat{p}(S) = \mathsf{BinomialEst}((x_i)_{i\in[m]},S)$ for each $S\subseteq[n]$, and then output an arbitrary feasible solution (if any) to the following linear program:

$$\begin{aligned}
\text{find}\quad & q\in\mathbb{R}^n\\
\text{subject to}\quad & \left|\sum_{i\in S} q_i - \hat{p}(S)\right| \le 3\eta + 60\epsilon/\sqrt{k},\ \forall S\subseteq[n]\\
& \sum_{i=1}^n q_i = 1\\
& q_i \ge 0,\ \forall i\in[n]
\end{aligned} \tag{2}$$

The algorithm described above involves solving a linear program with n variables and $O(2^n)$ constraints, as well as 2^n calls to the subroutine $\mathsf{BinomialEst}$, each of which takes polynomial time. Thus the whole algorithm runs in $\mathrm{poly}(2^n,k,1/\epsilon,1/\eta,\log(1/\delta))$ time.

Let $\delta_0 = \delta/2^n$. By Lemma 5 and a union bound, for some $m = O\left((k+\ln(1/\delta_0))/\epsilon^2\right) = O\left((n+k+\ln(1/\delta))/\epsilon^2\right)$, it holds with probability $1-2^n\cdot\delta_0 = 1-\delta$ that, for any $S\subseteq[n]$, $|\hat{p}(S)-p(S)| \le 3\eta+60\epsilon/\sqrt{k}$. This implies that linear program (2) is feasible with probability $1-\delta$. Moreover, let q be any feasible solution to (2). Then it holds for any $S\subseteq[n]$ that

$$|p(S)-q(S)| \le |p(S)-\hat{p}(S)| + |q(S)-\hat{p}(S)| \le 6\eta + 120\epsilon/\sqrt{k}.$$

And hence, from the definition of total variation distance, $\Delta(p,q) = O(\eta+\epsilon/\sqrt{k})$. ◀

4 A Tensor-Based Algorithm

In this section, we present a different algorithm that efficiently learns the distribution in the small-batch regime where k is a constant, and where all the good batches are drawn from the actual distribution p, i.e., $\eta = 0$. The algorithm works on the *frequency tensor* A defined by the m batches, i.e., $A_{i_1,i_2,\ldots,i_k}$ is the fraction of batches whose set of samples equal $(i_1, i_2, \ldots, i_k)$.

The following recursive function $\mathsf{DistSet}(n, k, A)$ takes tensor A and outputs a set of "guesses" of the distribution.

Algorithm 2: $\mathsf{DistSet}(n, k, A)$

Input: n, k, and n^k-tensor A.
Output: A set of n-dimensional vectors.

```
if k = 1 then
    return {A};
for i = 1, 2, ..., n do
    A_i ← the normalized i-th slice of A;
    S_i ← DistSet(n, k − 1, A_i);
a ← the marginal of A;
return S_1 ∪ S_2 ∪ ··· ∪ S_n ∪ {a};
```

The following lemma states that given sufficiently many batches, $\mathsf{DistSet}(n, k, A)$ contains a vector that is $O(\epsilon/k)$-close to the real distribution p in the total variation distance.

▶ **Lemma 6.** *Suppose that $\epsilon \in (0, 1/2)$ and vector p is a probability distribution on $[n]$. Let $A \in \mathbb{R}^{n^k}$ be the frequency tensor of m batches, among which $m(1-\epsilon)$ are drawn from $p^{\otimes k}$ and the other $m\epsilon$ batches are arbitrary and may depend on the $m(1-\epsilon)$ good batches. Then, with probability $1-\delta$ (over the randomness in the good batches),*

$$\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q) \le \frac{6\epsilon}{k} + O\left(\sqrt{\frac{n^k \cdot k! \cdot (n + k \ln n + \ln(1/\delta))}{m}}\right).$$

Proof of Lemma 6. Let $\delta_0 = \delta/(n+1)^{k-1}$. We recursively define functions $f_{n,1}(\epsilon, m)$ through $f_{n,k}(\epsilon, m)$ for $\epsilon, m > 0$ as follows:

$$f_{n,1}(\epsilon, m) = 3\epsilon + C\sqrt{\frac{n + \ln(1/\delta_0)}{m}} \tag{3}$$

and for $t \ge 2$,

$$f_{n,t}(\epsilon, m) = \max\left(\frac{3\epsilon}{t} + C\sqrt{\frac{n + \ln(1/\delta_0)}{m}}, f_{n,t-1}\left(1 - (1-\epsilon)^{(t-1)/t}, \frac{m}{nt}\right)\right). \tag{4}$$

Here C is an absolute constant to be determined later.

The following claim states that $\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q)$ is upper bounded by $f_{n,k}(\epsilon, m(1-\epsilon))$ with high probability.

▶ **Claim 1.** *Suppose that $t \in [k]$, $\epsilon \in (0, 1/2)$, and $A \in \mathbb{R}^{n^t}$ is the frequency tensor of a data set, in which at least m batches are drawn from $p^{\otimes t}$, and the fraction of these batches is at least $1-\epsilon$. Then, with probability $1 - (n+1)^{t-1} \cdot \delta_0$, it holds that*

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \le f_{n,t}(\epsilon, m).$$

The following claim further upper bounds $f_{n,k}(\epsilon, m)$.

▶ **Claim 2.** *For any $\epsilon \in (0, 1/2)$ and $m > 0$,*

$$f_{n,k}(\epsilon, m) \leq \frac{6\epsilon}{k} + C\sqrt{\frac{n^k \cdot k! \cdot (n + \ln(1/\delta_0))}{m}}.$$

Claims 1 and 2 together imply that with probability $1 - (n+1)^{k-1} \cdot \delta_0 = 1 - \delta$, it holds that

$$\begin{aligned}\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q) &\leq f_{n,k}(\epsilon, m(1-\epsilon)) \\ &\leq \frac{6\epsilon}{k} + O\left(\sqrt{\frac{n^k \cdot k! \cdot (n + k \ln n + \ln(1/\delta))}{m}}\right).\end{aligned}$$

◀

Lemma 6 implies that given

$$m = O(n^k \cdot k! \cdot (n + k \ln n + \ln(1/\delta)) \cdot k^2/\epsilon^2) = (nk)^{O(k)} \ln(1/\delta)/\epsilon^2$$

batches, the minimum distance is upper bounded by $O(\epsilon/k)$. Now we prove Theorem 3 by leveraging this approximation guarantee of procedure DistSet.

Proof of Theorem 3. Let $q_0 = \operatorname{argmin}_{q \in \mathsf{DistSet}(n,k,A)} \Delta(A, q^{\otimes k})$ and $q_1 = \operatorname{argmin}_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q)$. Note that q_0 can be computed from A in $(nk)^{O(k)}/\epsilon^2$ time. In the following we prove that q_0 is a good approximation of p. By Lemma 6, for some $m = (nk)^{O(k)} \ln(1/\delta)/\epsilon^2$, it holds with probability $1 - \delta/2$ that $\Delta(p, q_1) \leq 7\epsilon/k$, and thus, $\Delta(p^{\otimes k}, {q_1}^{\otimes k}) \leq k\Delta(p, q_1) \leq 7\epsilon$. We write A as $A = (1-\epsilon)\hat{P} + \epsilon N$, where $\hat{P}$ and N are the frequency tensor of the "good" and "bad" batches, respectively. Since $\hat{P}$ denotes the frequency among $m(1-\epsilon) = \Omega(m)$ samples drawn from $p^{\otimes k}$, and the support of $p^{\otimes k}$ is of size n^k, for some $m = O\left((n^k + \ln(1/\delta))/\epsilon^2\right)$, it holds with probability $1 - \delta/2$ that $\Delta(\hat{P}, p^{\otimes k}) \leq \epsilon$. Therefore,

$$\Delta(A, p^{\otimes k}) \leq \Delta(A, \hat{P}) + \Delta(\hat{P}, p^{\otimes k}) = \epsilon\Delta(\hat{P}, N) + \Delta(\hat{P}, p^{\otimes k}) \leq 2\epsilon.$$

By a union bound, it holds with probability $1 - \delta$ that

$$\Delta(A, {q_1}^{\otimes k}) \leq \Delta(A, p^{\otimes k}) + \Delta(p^{\otimes k}, {q_1}^{\otimes k}) \leq 2\epsilon + 7\epsilon = 9\epsilon.$$

Furthermore, by definition of q_0, $\Delta(A, {q_0}^{\otimes k}) \leq \Delta(A, {q_1}^{\otimes k}) \leq 9\epsilon$, and thus

$$\Delta(p^{\otimes k}, {q_0}^{\otimes k}) \leq \Delta(A, p^{\otimes k}) + \Delta(A, {q_0}^{\otimes k}) \leq 2\epsilon + 9\epsilon = 11\epsilon.$$

Let $\epsilon' = 165\epsilon/\sqrt{k} < 1/(15\sqrt{k})$. Since $\Delta(p^{\otimes k}, {q_0}^{\otimes k}) \leq 11\epsilon = \epsilon'\sqrt{k}/15$, applying the contrapositive of Lemma 12 with parameter ϵ' yields that, with probability at least $1 - \delta$, $\Delta(p, q_0) \leq \epsilon' = O(\epsilon/\sqrt{k})$. ◀

References

1 Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Symposium on Theory of Computing (STOC)*, pages 671–680, 2008.

2 Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 259–269. IEEE, 2000.

3 K. Bhatia, P. Jain, and P. Kar. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 721–729, 2015.

4 Dietrich Braess and Thomas Sauer. Bernstein polynomials and learning theory. *Journal of Approximation Theory*, 128(2):187–206, 2004.

5 Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1193–1203. Society for Industrial and Applied Mathematics, 2014.

6 Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Symposium on Theory of Computing (STOC)*, pages 47–60, 2017.

7 Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS)*, pages 655–664, 2016.

8 Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. *arXiv preprint arXiv:1703.00893*, 2017.

9 Alan Frieze and Ravi Kannan. A new approach to the planted clique problem. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 2. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

10 E Gilbert. Codes based on inaccurate source probabilities. *IEEE Transactions on Information Theory*, 17(3):304–314, 1971.

11 Nicolas Gillis and Stephen A Vavasis. On the complexity of robust pca and l1-norm low-rank matrix approximation. *arXiv preprint arXiv:1509.09236*, 2015.

12 Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons, 2011.

13 Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

14 Peter J Huber and Elvezio M Ronchetti. *Robust statistics*, volume 2. John Wiley & Sons, 2009. doi:10.1002/9780470434697.

15 Sudeep Kamath, Alon Orlitsky, Dheeraj Pichapati, and Ananda Theertha Suresh. On learning distributions from their samples. In *Conference on Learning Theory (COLT)*, pages 1066–1100, 2015.

16 Adam R Klivans, Philip M Long, and Rocco A Servedio. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10(Dec):2715–2740, 2009.

17 Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL: https://arxiv.org/abs/1610.05492.

18 Raphail Krichevsky and Victor Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

19 Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

20 Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS)*, pages 665–674, 2016.

21 Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9(8):295–347, 2013.

22 Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 438–446. IEEE, 2016.

23 Brendan McMahan and Daniel Ramage. `https://research.google.com/pubs/pub44822.html`, 2017.

24 H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

25 Michela Meister and Gregory Valiant. A data prism: Semi-verified learning in the small-alpha regime. *arXiv preprint arXiv:1708.02740*, 2017.

26 Blaine Nelson, Battista Biggio, and Pavel Laskov. Understanding the risk factors of learning in adversarial environments. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 87–92. ACM, 2011.

27 James Newsome, Brad Karp, and Dawn Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, pages 81–105. Springer, 2006.

28 Alon Orlitsky and Ananda Theertha Suresh. Competitive distribution estimation: Why is good-turing good. In *Advances in Neural Information Processing Systems*, pages 2143–2151, 2015.

29 Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise l1-norm error. *Algorithms*, 1:2, 2017.

30 Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. *arXiv preprint arXiv:1703.04940*, 2017.

31 Jacob Steinhardt, Gregory Valiant, and Moses Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4439–4447, 2016.

32 Kevin Tian, Weihao Kong, and Gregory Valiant. Learning populations of parameters. In *Neural Information Processing Systems (to appear)*, 2017.

33 John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, 2:448–485, 1960.

34 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*, pages 685–694. ACM, 2011.

35 Gregory Valiant and Paul Valiant. Instance optimal learning of discrete distributions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 142–155. ACM, 2016.

36 Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

A Technical Lemmas

We prove a few useful technical lemmas.

▶ **Lemma 7.** *For any $\epsilon \in [0, 1/2]$ and $\alpha \in [0, 1]$, $(1-\epsilon)^\alpha \geq 1 - 2\alpha\epsilon$.*

Proof of Lemma 7. Let $f(x) = (1-x)^\alpha - (1-2\alpha x)$. Since $f'(x) = \alpha[2-(1-x)^{\alpha-1}] \geq 0$ for any $x \in [0, 1/2]$ and $\alpha \in [0, 1]$, it holds for any $\epsilon \in [0, 1/2]$ that $f(\epsilon) \geq f(0) = 0$, which proves the lemma. ◀

▶ **Lemma 8.** *For any $a > 0$, the function $(1 - a/x)^x$ is increasing on $(a, +\infty)$.*

Proof of Lemma 8. Let $f(x) = x\ln(1 - a/x)$. Then for $x \in (a, +\infty)$, we have $f'(x) = \ln\left(1 - \frac{a}{x}\right) + \frac{a}{x-a}$ and $f''(x) = -\frac{a^2}{x(x-a)^2} < 0$. Since $\lim_{x\to+\infty} f'(x) = 0$, f' is positive on $(a, +\infty)$, which proves the lemma. ◀

▶ **Lemma 9.** *Suppose that n and m are positive integers such that $m \geq \max(n, 2)$. Then for any $\alpha \in [0, 1.1\sqrt{n}]$, $\left(1+\frac{\alpha}{n}\right)^n \left(1-\frac{\alpha}{m}\right)^m \geq \frac{1}{7}$.*

Proof of Lemma 9. For $x \in [0, 1.1\sqrt{n}]$, define

$$f_{n,m}(x) = n \ln\left(1+\frac{x}{n}\right) + m \ln\left(1-\frac{x}{m}\right).$$

Note that $f_{n,m}$ is well-defined since $x \leq 1.1\sqrt{n} \leq 1.1\sqrt{m} < m$. Moreover,

$$f'_{n,m}(x) = 1/(1+x/n) - 1/(1-x/m) \leq 0.$$

Thus, it remains to prove the inequality for $\alpha = 1.1\sqrt{n}$, i.e.,

$$\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m \geq \frac{1}{7}. \tag{5}$$

If $n \geq 2$, we lower bound the lefthand side of (5) by

$$\begin{aligned}\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m &\geq \left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{n}\right)^n \\ &= \left(1-\frac{1.21}{n}\right)^n \\ &\geq \left(1-\frac{1.21}{2}\right)^2 \\ &\geq \frac{1}{7}.\end{aligned}$$

Here the first and third steps follow from Lemma 8. For $n = 1$, we have

$$\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m = 2.1 \times \left(1-\frac{1.1}{m}\right)^m \geq 2.1 \times \left(1-\frac{1.1}{2}\right)^2 \geq \frac{1}{7}.$$ ◀

▶ **Lemma 10.** *For any $k \in \mathbb{N}$ and $t \in \{1, \ldots, k-1\}$,*

$$\binom{k}{t}\left(\frac{t}{k}\right)^t \left(\frac{k-t}{k}\right)^{k-t} \geq \frac{1}{3\sqrt{t}}.$$

Proof of Lemma 10. Stirling's approximation states that $\sqrt{2\pi n} \cdot (n/e)^n \leq n! \leq e\sqrt{n} \cdot (n/e)^n$ for any positive integer n. Thus,

$$\binom{k}{t} = \frac{k!}{t!(k-t)!} \geq \frac{\sqrt{2\pi k}}{e^2\sqrt{t(k-t)}} \cdot \frac{(k/e)^k}{(t/e)^t[(k-t)/e]^{k-t}} = \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{k}{t(k-t)}} \cdot \frac{k^k}{t^t(k-t)^{k-t}}.$$

We conclude that

$$\binom{k}{t}\left(\frac{t}{k}\right)^t \left(\frac{k-t}{k}\right)^{k-t} \geq \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{k}{t(k-t)}} \geq \frac{1}{3\sqrt{t}}.$$ ◀

B Tensorization of the Total Variation Distance

In this section we prove two inequalities regarding the total variation distance between k-fold product distributions.

▶ **Lemma 11.** *For any $\epsilon \in (0, 1/2)$ and $k \in \mathbb{N}$, there exist Bernoulli distributions P and Q such that:*
1. $\Delta(P, Q) = \epsilon$.
2. $\Delta(P^{\otimes k}, Q^{\otimes k}) \le \epsilon\sqrt{2k}$.

In particular, Bernoulli distributions with means $(1-\epsilon)/2$ and $(1+\epsilon)/2$ satisfy the above conditions.

Proof of Lemma 11. Let P and Q be Bernoulli distributions with means $(1-\epsilon)/2$ and $(1+\epsilon)/2$, respectively. Clearly, we have $\Delta(P, Q) = \epsilon$ and

$$\mathrm{KL}(P, Q) = \epsilon \ln \frac{1+\epsilon}{1-\epsilon} \le \frac{2\epsilon^2}{1-\epsilon} \le 4\epsilon^2.$$

Here the second step applies the inequality $\ln(1+x) \le x$. By Pinsker's inequality,

$$\Delta(P^{\otimes k}, Q^{\otimes k}) \le \sqrt{\frac{1}{2}\mathrm{KL}(P^{\otimes k}, Q^{\otimes k})} = \sqrt{\frac{k}{2}\mathrm{KL}(P, Q)} \le \epsilon\sqrt{2k}.$$ ◀

The next lemma shows that for sufficiently small ϵ, the $O(\sqrt{k})$ ratio between $\Delta(P^{\otimes k}, Q^{\otimes k})$ and $\Delta(P, Q)$ in Lemma 11 is tight (up to a constant factor).

▶ **Lemma 12.** *Suppose that $k \in \mathbb{N}$ and $\epsilon \in \left(0, 1/(15\sqrt{k})\right)$. P and Q are two distributions on the same support such that $\Delta(P, Q) \ge \epsilon$. Then, $\Delta(P^{\otimes k}, Q^{\otimes k}) \ge \frac{\epsilon\sqrt{k}}{15}$.*

To prove Lemma 12, we have the following weaker claim for the special case of Bernoulli distributions.

▶ **Lemma 13.** *Suppose that $k \in \mathbb{N}$, $\epsilon \in (0, 1/(15\sqrt{k}))$, $p, q \in [0, 1]$ and $q - p \ge \epsilon$. $\tilde{P}$ is a mixture of binomial distributions with k trials and success probabilities in $[0, p]$, i.e., $\tilde{P} = \sum_{i=1}^{\mathrm{tot}} \alpha_i B(k, p_i)$ for some nonnegative weights $\alpha_1, \ldots, \alpha_{\mathrm{tot}}$ and probabilities $p_1, \ldots, p_{\mathrm{tot}}$ in $[0, p]$, such that $\sum_{i=1}^{\mathrm{tot}} \alpha_i = 1$. $\tilde{Q}$ is a mixture of binomial distributions with k trials and success probabilities in $[q, 1]$. Then, $\Delta(\tilde{P}, \tilde{Q}) \ge \frac{\epsilon\sqrt{k}}{15}$.*

We prove Lemma 12 by a reduction to Bernoulli distributions.

Proof of Lemma 12. Suppose that P and Q share the support $[n]$. Define $\pi : [n] \to \{0, 1\}$ as $\pi(x) = \begin{cases} 1, & P(x) \ge Q(x), \\ 0, & P(x) < Q(x). \end{cases}$ Let p and q the means of $\pi(P)$ and $\pi(Q)$. Without loss of generality, $p \le q$. By construction, we have $|p - q| = \Delta(P, Q) \ge \epsilon$, and

$$\begin{aligned}\Delta(B(k, p), B(k, q)) &= \Delta((\pi(P_1), \pi(P_2), \ldots, \pi(P_k)), (\pi(Q_1), \pi(Q_2), \ldots, \pi(Q_k))) \\ &\le \Delta((P_1, P_2, \ldots, P_k), (Q_1, Q_2, \ldots, Q_k)) \\ &= \Delta(P^{\otimes k}, Q^{\otimes k}),\end{aligned}$$

where $(P_i)_{i\in[k]}$ and $(Q_i)_{i\in[k]}$ are independent copies of P and Q. Here the second step applies the data processing inequality. Applying Lemma 13 with $\tilde{P} = B(k, p)$ and $\tilde{Q} = B(k, q)$ yields that $\Delta(P^{\otimes k}, Q^{\otimes k}) \ge \Delta(B(k, p), B(k, q)) \ge \epsilon\sqrt{k}/15$. ◀

Now we turn to the more technical proof of Lemma 13.

Proof of Lemma 13. If $k < 10$, the inequality trivially follows from $\Delta(\tilde{P}, \tilde{Q}) \geq |p - q| > \epsilon\sqrt{k}/15$. Thus we assume that $k \geq 10$ in the following proof. Without loss of generality, we have $p \leq 1/2$ (otherwise we prove the lemma for $p' = 1 - q \leq 1/2$ and $q' = 1 - p$).

Let $t = \lfloor p(k-1) \rfloor$. Note that $t \leq (k-1)/2$ and $t \leq p(k-1) < t + 1$. Define function $f(x) = \sum_{j=0}^{t} \binom{k}{j} x^j (1-x)^{k-j}$. Note that

$$\begin{aligned} f'(x) &= k \sum_{j=0}^{t} \left[\binom{k-1}{j-1} x^{j-1}(1-x)^{k-j} - \binom{k-1}{j} x^j (1-x)^{k-j-1} \right] \\ &= -k \binom{k-1}{t} x^t (1-x)^{k-1-t} \leq 0, \end{aligned} \tag{6}$$

and thus f is non-increasing.

Since $f(x)$ is the probability that the binomial distribution $B(k, x)$ assigns to set $\{0, 1, \ldots, t\}$, $\tilde{P}(\{0, 1, \ldots, t\})$ can be written as a weighted average of $f(p_i)$'s for $p_1, p_2, \ldots \in [0, p]$. Similarly, $\tilde{Q}(\{0, 1, \ldots, t\})$ is a weighted average of $f(q_i)$'s for $q_1, q_2, \ldots \in [q, 1]$. Since $\tilde{P}(\{0, 1, \ldots, t\}) - \tilde{Q}(\{0, 1, \ldots, t\})$ is a lower bound on $\Delta(\tilde{P}, \tilde{Q})$, it remains to show that $f(p_i) - f(q_i) \geq \epsilon\sqrt{k}/15$ for any $p_i \leq p$ and $q_i \geq q$. The monotonicity of f and the fact that $q \geq p + \epsilon$ further imply that it suffices to prove $f(p) - f(p + \epsilon) \geq \epsilon\sqrt{k}/15$. We prove the inequality in the following two cases.

Case 1: $t = 0$. In this case, we have $0 \leq p < 1/(t-1)$. Note that

$$f(p) - f(p + \epsilon) = -\epsilon f'(x) = \epsilon k (1-x)^{k-1}$$

for some $x \in (p, p + \epsilon)$. If $\epsilon \leq 1/k$, we have $x \leq p + \epsilon \leq 2/(k-1)$, and

$$f(p) - f(p + \epsilon) \geq \epsilon k \left(1 - \frac{2}{k-1}\right)^{k-1} \geq \epsilon k \left(1 - \frac{2}{9}\right)^{9} \geq \frac{\epsilon k}{10}. \tag{7}$$

Here the second step follows from Lemma 8 and the assumption that $k \geq 10$. It then follows that $f(p) - f(p + \epsilon) \geq \epsilon\sqrt{k}/15$.

If $\epsilon \geq 1/k$, by Inequality (7) and the assumption that $\epsilon < 1/(15\sqrt{k})$,

$$f(p) - f(p + \epsilon) \geq f(p) - f(p + 1/k) \geq \frac{1}{10} \geq \frac{\epsilon\sqrt{k}}{15}.$$

Case 2: $t > 0$. Let $x_0 = t/(k-1)$. By Equation (6) and Lemma 10, we have

$$|f'(x_0)| = k \binom{k-1}{t} \left(\frac{t}{k-1}\right)^t \left(\frac{k-1-t}{k-1}\right)^{k-1-t} \geq \frac{k}{3\sqrt{t}}.$$

For any $x \in [p, p + \epsilon]$, we can write $x = (t + \alpha)/(k-1)$ for some $\alpha \geq 0$. Then,

$$\frac{|f'(x)|}{|f'(x_0)|} = \left(1 + \frac{\alpha}{t}\right)^t \left(1 - \frac{\alpha}{k-1-t}\right)^{k-1-t}$$

Since $t \leq (k-1)/2$ and $k \geq 10$, we have $k - 1 - t \geq \max(t, (k-1)/2) \geq \max(t, 2)$. Applying Lemma 9 with $n = t$ and $m = k - 1 - t$ yields that

$$\left(1 + \frac{\alpha}{t}\right)^t \left(1 - \frac{\alpha}{k-1-t}\right)^{k-1-t} \geq \frac{1}{7}$$

for any $\alpha \in \left[0, 1.1\sqrt{t}\right]$. Consequently, $|f'(x)| \geq \frac{|f'(x_0)|}{7} \geq \frac{k}{21\sqrt{t}}$ for any $x \in [x_0, x_0 + 1.1\sqrt{t}/(k-1)]$.
Let l be the length of the intersection of $[p, p+\epsilon]$ and $[x_0, x_0 + 1.1\sqrt{t}/(k-1)]$. By our choice of t, we have $t \leq p(k-1) < t+1$, and thus, $x_0 \leq p < x_0 + 1/(k-1)$. This implies that

$$l = \min(p+\epsilon, x_0 + 1.1\sqrt{t}/(k-1)) - p \geq \min(\epsilon, \sqrt{t}/[10(k-1)]).$$

Therefore, we conclude that

$$f(p) - f(p+\epsilon) \geq l \cdot \frac{k}{21\sqrt{t}} \geq \frac{1}{21} \min\left(\frac{\epsilon k}{\sqrt{t}}, \frac{1}{10}\right) \geq \frac{1}{21} \min\left(\epsilon\sqrt{2k}, \frac{1}{10}\right) \geq \frac{\epsilon\sqrt{k}}{15}.$$

Here the last step holds since our assumption $\epsilon < 1/(15\sqrt{k})$ implies that $\epsilon\sqrt{2k} < 1/10$. ◀

C Missing Proofs from Section 4

C.1 Proof of Claim 1

Proof of Claim 1. By assumption, the frequency tensor A can be written as $A = (1-\epsilon)\hat{P} + \epsilon N$, where $\hat{P}$ and N are probability n^t-tensors. In particular, $\hat{P}$ denotes the frequency among the m "good" samples drawn from $p^{\otimes t}$, while N denotes the frequency among the other batches. We prove the lemma by induction on t.

Base case. When $t = 1$, we have $\mathsf{DistSet}(n, t, A) = \{A\}$, and thus,

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) = \Delta(p, A) \leq \Delta(p, \hat{P}) + \Delta(\hat{P}, A) = \Delta(p, \hat{P}) + \epsilon\Delta(\hat{P}, N) \leq \Delta(p, \hat{P}) + \epsilon.$$

It is well-known that with probability $1 - \delta_0$, $\Delta(p, \hat{P}) \leq C\sqrt{\frac{n + \ln \delta_0^{-1}}{m}}$ for some absolute constant C. Therefore,

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \leq C\sqrt{\frac{n + \ln \delta_0^{-1}}{m}} + \epsilon \leq f_{n,1}(\epsilon, m).$$

Inductive step. Let $\hat{p}$ and δ be the marginals of $\hat{P}$ and N, respectively. Then the marginal of A is given by $a = (1-\epsilon)\hat{p} + \epsilon\delta$. Moreover, the i-th slice of A after normalization is given by

$$A_i = \frac{(1-\epsilon)\hat{p}_i \cdot \hat{P}_i + \epsilon\delta_i \cdot N_i}{(1-\epsilon)\hat{p}_i + \epsilon\delta_i} = (1-\epsilon_i')\hat{P}_i + \epsilon_i' N_i,$$

where $\epsilon_i' = \frac{\epsilon\delta_i}{(1-\epsilon)\hat{p}_i + \epsilon\delta_i}$. Moreover, A_i contains $m_i' = m \cdot \hat{p}_i$ samples drawn from $p^{\otimes(t-1)}$. Let $\alpha = (1 - \sqrt[t]{1-\epsilon})/\epsilon$ and $\beta = 1/t$. We consider the following two cases.

Case 1: For every $i \in [n]$, either $\delta_i/\hat{p}_i \geq 1 - \alpha$ or $\hat{p}_i \leq \beta/n$.
Since $a \in \mathsf{DistSet}(n, t, A)$, we have

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \leq \Delta(p, a) \leq \Delta(p, \hat{p}) + \Delta(\hat{p}, a) = \Delta(p, \hat{p}) + \epsilon\Delta(\hat{p}, \delta).$$

Moreover, since for any $i \in [n]$, either $\delta_i \geq (1-\alpha)\hat{p}_i$ or $\hat{p}_i \leq \beta/n$, it holds that $\hat{p}_i - \delta_i \leq \alpha\hat{p}_i + \beta/n$, and thus

$$\Delta(\hat{p}, \delta) = \sum_{i=1}^{n} \max(\hat{p}_i - \delta_i, 0) \leq \sum_{i=1}^{n} (\alpha\hat{p}_i + \beta/n) = \alpha + \beta.$$

It follows that

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \leq \Delta(p,\hat{p}) + \epsilon(\alpha+\beta) = \Delta(p,\hat{p}) + 1 - \sqrt[t]{1-\epsilon} + \epsilon/t \leq \frac{3\epsilon}{t} + \Delta(p,\hat{p}).$$

Here the last step $\sqrt[t]{1-\epsilon} \geq 1 - 2\epsilon/t$ follows from Lemma 7.

Case 2: For some $i \in [n]$, both $\delta_i/\hat{p}_i < 1-\alpha$ and $\hat{p}_i > \beta/n$ hold. In this case, we have

$$\epsilon_i' = \frac{\epsilon(\delta_i/\hat{p}_i)}{1 - \epsilon + \epsilon(\delta_i/\hat{p}_i)} \leq \frac{\epsilon(1-\alpha)}{1-\epsilon\alpha} = 1 - (1-\epsilon)^{(t-1)/t}$$

and $m_i' = m \cdot \hat{p}_i \geq \beta m/n = m/(nt)$. We have the following bound:

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \leq \min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q).$$

Combining the two cases shows that $\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q)$ is upper bounded by either $3\epsilon/t + \Delta(p,\hat{p})$ or $\min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q)$ for some $i \in [n]$ such that $\epsilon_i' \leq 1 - (1-\epsilon)^{(t-1)/t}$ and $m_i' \geq m/(nt)$. According to the induction hypothesis, for each $i \in [n]$, it holds with probability $1 - (n+1)^{t-2} \cdot \delta_0$ that

$$\min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q) \leq f_{n,t-1}\left(\epsilon_i', m_i'\right).$$

Moreover, with probability $1-\delta_0$, $\Delta(p,\hat{p}) \leq C\sqrt{\frac{n+\ln\delta_0^{-1}}{m}}$. By a union bound, with probability $1 - n \cdot (n+1)^{t-2}\delta_0 - \delta_0 \geq 1 - (n+1)^{t-1}\delta_0$, it holds that

$$\begin{aligned}
&\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \\
&\leq \max\left(\frac{3\epsilon}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m}}, f_{n,t-1}\left(1-(1-\epsilon)^{(t-1)/t}, \frac{m}{nt}\right)\right) \\
&= f_{n,t}(\epsilon, m).
\end{aligned}$$

This completes the inductive step. ◀

C.2 Proof of Claim 2

Proof of Claim 2. Define $(\epsilon_k, m_k) = (\epsilon, m)$ and $(\epsilon_{t-1}, m_{t-1}) = \big(1 - (1-\epsilon_t)^{(t-1)/t}, m_t/(nt)\big)$. Then by Equations (3) and (4),

$$\begin{aligned}
&f_{n,k}(\epsilon_k, m_k) \\
&\quad = \max\left(\frac{3\epsilon_k}{k} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_k}}, f_{n,k-1}(\epsilon_{k-1}, m_{k-1})\right) \\
&\quad = \max\left(\frac{3\epsilon_k}{k} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_k}}, \frac{3\epsilon_{k-1}}{k-1} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_{k-1}}}, f_{n,k-2}(\epsilon_{k-2}, m_{k-2})\right) \\
&\quad = \cdots \\
&\quad = \max_{t\in[k]}\left(\frac{3\epsilon_t}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_t}}\right).
\end{aligned}$$

Moreover, by Lemma 7 and a simple induction, $\epsilon_t = 1-(1-\epsilon)^{t/k} \le 2t\epsilon/k$ and $m_t = m/(n^{k-t}A_k^{k-t})$, where A_n^m denotes $n(n-1)\cdots(n-m+1)$. It follows that for any $t \in [k]$,

$$\begin{aligned}
\frac{3\epsilon_t}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_t}} &\le \frac{3}{t}\cdot\frac{2t\epsilon}{k} + C\sqrt{\frac{n^{k-t}A_k^{k-t}(n+\ln\delta_0^{-1})}{m}} \\
&\le \frac{6\epsilon}{k} + C\sqrt{\frac{n^k\cdot k!\cdot(n+\ln\delta_0^{-1})}{m}}.
\end{aligned}$$

◀