Optimal Robust Learning of Discrete Distributions from Batches

Ayush Jain [1] **Alon Orlitsky** [1]

Abstract

Many applications, including natural language processing, sensor networks, collaborative filtering, and federated learning, call for estimating discrete distributions from data collected in batches, some of which may be untrustworthy, erroneous, faulty, or even adversarial. Previous estimators for this setting ran in exponential time, and for some regimes required a suboptimal number of batches. We provide the first polynomial-time estimator that is optimal in the number of batches and achieves essentially the best possible estimation accuracy.

1. Introduction

1.1. Motivation

Estimating discrete distributions from their samples is a fundamental modern-science tenet. (Kamath et al., 2015) showed that as the number of sample s grows, a k-symbol distribution can be learned to expected L_1 distance $\sim \sqrt{2(k-1)/(\pi s)}$ that we call the *information-theoretic limit*.

In many applications, some samples are inadvertently or maliciously corrupted. A simple and intuitive example shows that this erroneous data limits the extent to which a distribution can be learned, even with infinitely many samples.

Consider the extremely simple case of just two possible binary distributions: $(1, 0)$ and $(1 - \beta, \beta)$. An adversary who observes a $1 - \beta$ fraction of the samples and can determine the rest, could use the observed samples to learn the underlying distribution, and set the remaining samples to make the distribution appear to be $(1 - \beta, \beta)$. By the triangle inequality, even with arbitrarily many samples, any estimator for p incurs an L_1 loss $\geq \beta$ for at least one of the two distributions. We call this the *adversarial lower bound*.

The example may seem to suggest a pessimistic conclusion. If an adversary can corrupt a β fraction of the data, a loss $\geq \beta$ is unavoidable. Fortunately, that is not necessarily so.

In many applications data is collected in batches, most of which are genuine, but some possibly corrupted. Here are a few examples. Data may be gathered by sensors, each providing a large amount of data, and some sensors may be faulty. The word frequency of an author may be estimated from several large texts, some of which are mis-attributed. Or user preferences may be learned by querying several users, but some users may intentionally bias their feedback.

Interestingly, even when a β-fraction of the batches are corrupted, the underlying distribution can be estimated to L_1 distance much lower than β. Consider for example just three n-sample batches, of which one is chosen adversarially. The underlying distribution can be learned from each genuine batch to expected L_1 distance $\sim \sqrt{2(k-1)/(\pi n)}$. It is easy to see that the average of the two estimates pairwise-closest in L_1 distance achieves a comparable expected distance that for large batch size n is much lower than β.

This raises the natural question of whether estimates from even more batches can be combined effectively to estimate distributions to within a distance that is not only much smaller than the β achieved when no batch information was utilized, but also significantly smaller than the $O(\sqrt{k/n})$ distance derived above when two batches were used. For example can the underlying distribution be learned to a small L_1 distance when, as in many practical examples, $n \leq k$?

To formalize the problem, (Qiao & Valiant, 2017) considered learning a k-symbol distribution p whose samples are provided in batches of size $\geq n$. A total of m batches are provided, of which a fraction $\leq \beta$ may be arbitrarily and adversarially corrupted, while in every other batch b the samples are drawn according a distribution p_b satisfying $||p_b - p||_1 \leq \eta$, allowing for the possibility that slightly different distributions generate samples in each batch.

For this adversarial batch setting, they showed that for any alphabet size $k \geq 2$, and any number m of batches, the lowest achievable L_1 distance is $\geq \eta + \frac{\beta}{\sqrt{2n}}$. We refer to this as the *adversarial batch lower bound*.

For $\beta < 1/900$, they also derived an estimation algorithm that approximates p to L_1 distance $O(\max\{\eta + \beta/\sqrt{n}, \sqrt{(n+k)/(nm)}\})$, achieving the adversarial batch lower

[1] University of California, San Diego. Correspondence to: Ayush Jain <ayjain@eng.ucsd.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

bound, for m large enough. Surprisingly therefore, not only can the underlying distribution be approximated to L_1 distance $O(\sqrt{k/n})$ that falls below β, but the distance diminishes as $\beta/\sqrt{n}$, independent of the alphabet size k.

Yet, the algorithm in (Qiao & Valiant, 2017) had three significant drawbacks. 1) it runs in time exponential in the alphabet size, hence impractical for most relevant applications; 2) its guarantees are limited to very small fractions of corrupted batches $\beta \geq 1/900$, hence do not apply to practically important ranges; 3) with m batches of size $\geq n$ each, the total number of samples is $\geq nm$, and for alphabet size $k \ll n$, the algorithm's distance guarantee falls short of the information-theoretic $\Theta(\sqrt{k/(nm)})$ limit.

In this paper we derive an algorithm that 1) runs in polynomial time in all parameters; 2) can tolerate any fraction of adversarial batches $\beta < 1/2$, though to derive concrete constant factors in the theoretical analysis, we assume $\beta \leq 0.4$; 3) achieves distortion $O(\max\{\eta + \beta\sqrt{\frac{\log(1/\beta)}{n}}, \sqrt{\frac{k}{nm}}\})$ that achieves the statistical limit in terms of the number nm of samples, and is optimal up to a small $O(\sqrt{\log(1/\beta)})$ factor from the adversarial batch lower bound.

The algorithm's computational efficiency, enables the first experiments of learning with adversarial batches. We tested the algorithm on simulated data with various adversarial-batch distributions and adversarial noise levels up to $\beta = 0.49$. The algorithm runs in a fraction of a second, and as shown in Section 3, estimates p nearly as well as an oracle that knows the identity of the adversarial batches.

To summarize, the algorithm runs in polynomial time, works for any adversarial fraction $\beta < 0.5$, is optimal in number of samples, and essentially optimal in batch size. It opens the door to practical robust estimation in sensor networks, federated learning, and collaborative filtering.

1.2. Problem Formulation

Let Δ_k be the collection of all distributions over $[k] = \{1, \ldots, k\}$. The L_1 distance between two distributions $p, q \in \Delta_k$ is

$$||p - q||_1 \triangleq \sum_{i \in [k]} |p(i) - q(i)| = 2 \cdot \max_{S \subseteq [k]} |p(S) - q(S)|.$$

We would like to estimate an unknown *target distribution* $p \in \Delta_k$ to a small L_1 distance from samples, some of which may be corrupted or even adversarial.

Specifically, let B be a collections of m batches of n samples each. Among these batches is an unknown collection of *good batches* $B_G \subseteq B$; each batch $b \in B_G$ in this collection has n independent samples $X_1^b, X_2^b, ..., X_n^b \sim p_b$ with $||p_b - p||_1 \leq \eta$. Furthermore, the batches of samples in B_G are independent of each other.

For the special case where $\eta = 0$, all samples in the good batches are generated by the target distribution $p = p_b$. Since the proofs and techniques are essentially the same for $\eta = 0$ and $\eta > 0$, for simplicity of presentation we assume that $\eta = 0$. We briefly discuss, at the end, how these results translate to the case $\eta > 0$.

The remaining set $B_A = B \setminus B_G$ of *adversarial batches* consists of arbitrary n samples each, that may even be chosen by an adversary, possibly based on the samples in the good batches. Let $\alpha = |B_G|/m$, and $\beta = |B_A|/m = 1 - \alpha$ be the fractions of good and adversarial batches, respectively.

Our goal is to use the m batches to return a distribution p^* such that $||p^* - p||_1$ is small or equivalently $|p(S) - p^*(S)|$ is small for all $S \subseteq [k]$.

1.3. Result Summary

In section 2 we derive a polynomial-time algorithm that returns an estimate p^* of p with the following properties.

Theorem 1. *For any given $\beta \leq 0.4$, n, k, and $m = \Omega(\frac{k}{\beta^2 \log(1/\beta)})$, Algorithm 2 runs in time polynomial in all parameters and its estimate p^* satisfies $||p^* - p||_1 \leq 100\beta\sqrt{\frac{\log(1/\beta)}{n}}$ with probability $\geq 1 - O(e^{-k})$.*

The theorem implies that our algorithm can achieve the adversarial lower bound to a small factor of $O(\sqrt{\log(1/\beta)})$ using the optimal number of samples. The next theorem shows that when the number of samples is not enough to achieve the adversarial batch lower bound our algorithm achieves the statistical lower bound.

Theorem 2. *For any given $\beta \leq 0.4$, n and k and m, Algorithm 2 runs in polynomial time, and its estimate p^* satisfies $||p^* - p||_1 \leq O(\max\{\beta\sqrt{\frac{\ln(1/\beta)}{n}}, \sqrt{\frac{k}{mn}}\})$ with probability $\geq 1 - O(e^{-k})$.*

The above theorem follows from Theorem 1 and a short proof appears in Appendix D.

Note that our polynomial time algorithm achieves the statistical limits for L_1 distance and achieves the adversarial batch lower bounds to a small multiplicative factor of $O(\sqrt{\log(1/\beta)})$.

1.4. Comparison to Recent Results and Techniques

In a paper concurrent and independent of this work, (Chen et al., 2019) propose an algorithm that uses the sum of squares methodology to estimate p to the same distance as ours. Their algorithm needs $\tilde{\mathcal{O}}(\frac{(nk)^{\mathcal{O}(\log(1/\beta))}}{\beta^4})$ batches and has a run-time $\tilde{\mathcal{O}}(\frac{(nk)^{\mathcal{O}(\log^2(1/\beta))}}{\beta^{\mathcal{O}(\log(1/\beta))}})$. Both the sample complexity and run time are much higher than ours, and is quasi-polynomial. They also considered certain structured distributions, namely t-piecewise degree-d polynomial, not addressed in this paper. For this distribution class they provide an algorithm with similar quasi-polynomial run time and the number of batches required was quasi-polylogarithmic in domain size k, and quasi-polynomial in other parameters.

In the follow up work (Jain & Orlitsky, 2020), we generalized our techniques to improve both the run time and the number of batches required for learning piece-wise polynomial distributions. We gave an algorithm that runs in polynomial time in all parameters and uses the number of batches $\Omega\Big(\frac{t\cdot d\cdot\sqrt{n}\cdot\log(n/\beta)}{\beta^3}\Big)$, which has an optimal linear dependence on t and d and is independent of domain size k. Further, we developed first algorithm for robust *classification* in a similar adversarial batch setting.

Another follow up work (Chen et al., 2020), concurrent and independent to (Jain & Orlitsky, 2020), combined their previous work (Chen et al., 2019) with the techniques presented here, and also obtained a polynomial time algorithm for learning piecewise-polynomial distributions, which requires $\Omega\Big(\frac{t^2\cdot d^2\log^3 k\cdot\text{polylog}(n/\beta)}{\beta^2}\Big)$ batches.

1.5. Other Related Work

The current results extend several long lines of work on learning distributions and their properties.

The best approximation of a distribution with a given number of samples was determined up to the exact first-order constant for KL loss (Braess & Sauer, 2004), and L_1 loss and χ^2 loss (Kamath et al., 2015). These settings do not allow adversarial examples, and some modification of the empirical estimates of the samples is often shown to be near optimal. This is not the case in the presence of adversarial samples, where the challenge is to devise algorithms that are efficient from both computational and sample viewpoints.

Our results also relate to classical robust-statistics work (Tukey, 1960; Huber, 1992). There has also been significant recent work leading to practical distribution learning algorithms that are robust to adversarial contamination of the data. For example, (Diakonikolas et al., 2016; Lai et al., 2016) presented algorithms for learning the mean and covariance matrix of high-dimensional sub-gaussian and other distributions with bounded fourth moments in presence of the adversarial samples. Their estimation guarantees are typically in terms of L_2, and do not yield the L_1- distance results required for discrete distributions.

The work was extended in (Charikar et al., 2017) to the case when more than half of the samples are adversarial. Their algorithm returns a small set of candidate distributions one of which is a good approximate of the underlying distribution. For more extensive survey on robust learning algorithms in the continuous setting, see (Steinhardt et al., 2017; Diakonikolas et al., 2019).

Another motivation for this work derives from the practical federated-learning problem, where information arrives in batches (McMahan et al., 2016; McMahan & Ramage, 2017).

1.6. Preliminaries

We introduce notation that will help outline our approach and will be used in rest of the paper.

Throughout the paper, we use B' to denote a sub-collection of batches in B and use B'_G and B'_A for a sub-collection of batches in B_G and B_A, respectively. And S is used to denote a subset of $[k]$, we abbreviate singleton set of $[k]$ such as $\{j\}$ by j.

For any batch $b\in B$, we let $\bar{\mu}_b$ denote the empirical measure defined by samples in batch b. And for any sub-collection of batches $B'\subseteq B$, let $\bar{p}_{B'}$ denote the empirical measure defined by combined samples in all the batches in B'. We use two different symbols to distinguish the empirical distribution defined by an individual batch and the empirical distribution defined by a sub-collection of batches. Let $\mathbf{1}_S(.)$ denote the indicator random variable for set S. Thus, for any subset $S\subseteq[k]$,

$$\bar{\mu}_b(S)\triangleq\frac{1}{n}\sum_{i\in[n]}\mathbf{1}_S(X_i^b)$$

and

$$\bar{p}_{B'}(S)\triangleq\frac{1}{|B'|n}\sum_{b\in B'}\sum_{i\in[n]}\mathbf{1}_S(X_i^b)=\frac{1}{|B'|}\sum_{b\in B'}\bar{\mu}_b(S).$$

Note that $\bar{p}_{B'}$ is the mean of the empirical measures $\bar{\mu}_b$ defined by the batches $b\in B'$. For subset $S\subseteq[k]$, let $\text{med}(\bar{\mu}(S))$ be the median of the set of estimates $\{\bar{\mu}_b(S): b\in B\}$. Note that the median has been computed using the estimates $\bar{\mu}_b(S)$ for all the batches in $b\in B$.

For $r\in[0,1]$, we let $\text{V}(r)\triangleq\frac{r(1-r)}{n}$, which we use to denote the variance of sum of n i.i.d. random variables distributed according to Bernoulli(r).

We pause briefly to note the following two properties of the function $\text{V}(r)$ that we use later.

$$\forall\, r,s\in[0,1],\ \text{V}(r)\le\frac{1}{4n}\text{ and }|\text{V}(r)-\text{V}(s)|\le\frac{|r-s|}{n}.\tag{1}$$

Here the second property made use of the fact that the derivative $|V'(r)|\le 1/n$, $\forall\, r\in[0,1]$.

For $b\in B_G$, $\mathbf{1}_S(X_i^b)$ for $i\in[n]$ are i.i.d. with distribution $\mathbf{1}_S(X_i^b)\sim$ Bernoulli$(p(S))$. For $b\in B_G$, since $\bar{\mu}_b(S)$ is average of $\mathbf{1}_S(X_i^b)$, $i \in [n]$, therefore,

$$E[\,\bar{\mu}_b(S)\,] = p(S) \quad \text{and} \quad E[(\bar{\mu}_b(S) - p(S))^2] = \mathrm{V}(p(S)).$$

For any collection of batches $B' \subseteq B$ and subset $S \subseteq [k]$, the empirical probability $\bar{\mu}_b(S)$ of S based on batches $b \in B'$ will differ for the different batches. The empirical variance of these empirical probabilities $\bar{\mu}_b(S)$ for batches $b \in B'$ is denoted as

$$\overline{\mathrm{V}}_{B'}(S) \triangleq \frac{1}{|B'|} \sum_{b \in B'} (\bar{\mu}_b(S) - \bar{p}_{B'}(S))^2.$$

1.7. Organization of the Paper

In Section 2 we present the algorithm, its analysis along with the key insights used in developing the algorithm. Section 3 reports the performance of the algorithm on experiments performed on the simulated data.

2. Algorithm and its Analysis

At a high level, our algorithm removes the adversarial batches — which are "outliers" — possibly losing a small number of good batches as well in the process. The outlier removal method forms the backbone of many robust learning algorithms. Notably (Diakonikolas et al., 2016; 2017) have used this idea to learn the mean of a high dimensional sub-gaussian distribution up to a small L_2 distance, even in an adversarial setting. The main challenge in designing a robust learning algorithm is actually the task of finding the outlier batches efficiently. Several new ideas are needed to identify the outlier batches in the setting considered here.

We begin by illustrating the difficulty of identifying the adversarial batches. Even if p is known, in general, one cannot determine whether a batch b has samples from p or from a distribution at a large L_1 distance from p. The key difficulty is that, for a batch having n samples from p, typically the difference between $\bar{\mu}_b(S)$ and $p(S)$ is large for some of the subsets among 2^k subsets of $[k]$. For example, consider batches of samples from a uniform distribution over k. The empirical distribution of the samples in any batch of size n is at an L_1 distance $\geq 2(1 - n/k)$, which for the distributions with large domain size k can be up to two, which is the maximum L_1 distance between two distributions. To address this challenge, we use the following observation.

For a fixed subset $S \subseteq [k]$ and a good batch $b \in B_G$, $\bar{\mu}_b(S)$ has a sub-gaussian distribution subG$(p(S), \frac{1}{4n})$ and the variance is $\mathrm{V}(p(S))$. Therefore, for a fixed subset S, most of the good batches assign the empirical probability $\bar{\mu}_b(S) \in p(S) \pm \tilde{O}(1/\sqrt{n})$. Moreover, the mean and the variance of $\bar{\mu}_b(S)$ for $b \in B_G$ converges to the expected values $p(S)$ and $\mathrm{V}(p(S))$, respectively.

The collection of batches B along with good batches also includes a sub-collection B_A of adversarial batches that constitute up to an $\beta-$fraction of B. If for adversarial batches $b \in B_A$, the average difference between $\bar{\mu}_b(S)$ and $p(S)$ is within a few standard deviations $\tilde{O}(\frac{1}{\sqrt{n}})$, then these adversarial batches can only deviate the overall mean of empirical probabilities $\bar{\mu}_b(S)$ by $\tilde{O}(\frac{\beta}{\sqrt{n}})$ from $p(S)$. Hence, the mean of $\bar{\mu}_b(S)$ will deviates significantly from $p(S)$ only if for a large number of adversarial batches $b \in B_A$ empirical probability $\bar{\mu}_b(S)$ differ from $p(S)$ by quantity much larger than the standard deviation $\tilde{O}(\frac{1}{\sqrt{n}})$.

We quantify this effect by defining the *corruption score*. For a subset $S \subseteq [k]$, let

$$\text{med}(\bar{\mu}(S)) \triangleq \text{median}\{\bar{\mu}_b(S) : b \in B\}.$$

For a subset $S \subseteq [k]$ and a batch b, *corruption score* $\psi_b(S)$ is defined as

$$\psi_b(S) \triangleq \begin{cases} 0, & \text{if } |\bar{\mu}_b(S) - \text{med}(\bar{\mu}(S))| \leq 3\sqrt{\frac{\ln(6e/\beta)}{n}}, \\ (\bar{\mu}_b(S) - \text{med}(\bar{\mu}(S)))^2, & \text{else.} \end{cases}$$

Because $p(S)$ is not known, the above definition use median of $\bar{\mu}_b(S)$ as its proxy.

From the preceding discussion, it follows that for a fixed subset $S \subseteq [k]$, corruption score of most good batches w.r.t. S is zero, and adversarial batches that may have a significant effect on the overall mean of empirical probabilities have high corruption score $\psi_b(S)$.

The *corruption score* of a sub-collection B' w.r.t. a subset S is defined as the sum of the *corruption score* of batches in it, namely

$$\psi(B', S) \triangleq \sum_{b \in B'} \psi_b(S).$$

A high corruption score of B' w.r.t. a subset S indicates the presence of many batches $b \in B'$ for which the difference $|\bar{\mu}_b(S) - \text{med}(\bar{\mu}(S))|$ is large. Finally, for a sub-collection B' we define *corruption* as

$$\psi(B') \triangleq \max_{S \subseteq [k]} \psi(B', S).$$

Note that removing batches from a sub-collection reduces corruption. We can simply make corruption zero by removing all batches, but we would lose all the information as well. The proposed algorithm reduces the corruption below a threshold by removing a few batches while not sacrificing too many good batches in the process.

The remainder of this section assumes that the sub-collection of good batches B_G satisfies certain deterministic conditions. Lemma 3 shows that the stated conditions hold with high probability for sub-collection of good batches in B_G.

Nothing is assumed about the adversarial batches, except that they form a $\leq \beta$ fraction of the overall batches B.

Conditions: Consider a collection of m batches B, each containing n samples. Among these batches, there is a collection $B_G \subseteq B$ of good batches of size $|B_G| \geq (1-\beta)m$ and a distribution $p \in \Delta_k$ such that the following deterministic conditions hold for all subsets $S \subseteq [k]$:

1. The median of the estimates $\{\bar{\mu}_b(S) : b \in B\}$ is not too far from $p(S)$.

$$|\text{med}(\bar{\mu}(S)) - p(S)| \leq \sqrt{\ln(6)/n}.$$

2. For all sub-collections $B'_G \subseteq B_G$ of good batches of size $|B'_G| \geq (1-\beta/6)|B_G|$,

$$|\bar{p}_{B'_G}(S) - p(S)| \leq \frac{\beta}{2}\sqrt{\frac{\ln(6e/\beta)}{n}},$$

$$\Big|\frac{1}{|B'_G|}\sum_{b \in B'_G}(\bar{\mu}_b(S) - p(S))^2 - \text{V}(p(S))\Big| \leq \frac{6\beta\ln(\frac{6e}{\beta})}{n}.$$

3. The corruption for good batches B_G is small, namely

$$\psi(B_G) \leq \frac{\beta m \ln(6e/\beta)}{n}.$$

Condition 1 and 3 above are self-explanatory. Condition 2 illustrates that for any sub-collection of good batches that retains all but a small fraction of good batches, empirical mean and variance estimate the actual values $p(S)$ and $\text{V}(p(S))$.

Lemma 3. *When samples in B_G come from p and $|B_G| = \Omega(\frac{k}{\beta^2\ln(1/\beta)})$, then conditions 1- 3 hold simultaneously with probability $\geq 1 - O(e^{-k})$.*

We prove the above lemma by using the observation that for $b \in B_G$, $\bar{\mu}_b(S)$ has a sub-gaussian distribution subG$(p(S), \frac{1}{4n})$, and it has variance $\text{V}(p(S))$. The proof is in Appendix A.

For easy reference, in the remaining paper, we will denote the upper bound in Condition 3 on the corruption of B_G as

$$\kappa_G \triangleq \frac{\beta m \ln(6e/\beta)}{n}.$$

Assuming that the above stated conditions hold, the next lemma bounds the L_1 distance between the empirical distribution $\bar{p}_{B'}$ and p for any sub-collection B' in terms of how large its corruption is compared to κ_G.

Lemma 4. *Suppose the conditions 1- 3 holds. Then for any B' such that $|B' \cap B_G| \geq (1 - \frac{\beta}{6})|B_G|$ and let $\psi(B') = t \cdot \kappa_G$, for some $t \geq 0$, then*

$$||\bar{p}_{B'} - p||_1 \leq (10 + 3\sqrt{t})\beta\sqrt{\frac{\ln(6e/\beta)}{n}}.$$

Observe that for any sub-collection B' retaining a major portion of good batches, from condition 2, the mean of $\bar{\mu}_b$ of the good batches $B' \cap B_G$ approximates p. Then showing that a small corruption score of B' w.r.t. all subsets S imply that the adversarial batches $B' \cap B_A$ have limited effect on $\bar{p}_{B'}$ proves the above lemma. A complete proof is in Appendix B.

We next exhibit a Batch Deletion procedure in Algorithm 1 that lowers the corruption score of a sub-collection B' w.r.t. a given subset S by deleting a few batches from the sub-Collection. This will be a subroutine of our main algorithm. Lemma 5 characterizes its performance.

Algorithm 1 Batch Deletion

```
1: Input: Sub-Collection of Batches $B'$, subset $S \subseteq [k]$,
   med=med($\bar{\mu}(S)$), and $\beta$.
2: Output: A collection $DEL \subseteq B'$ of batches to delete.
3: $DEL = \{\}$;
4: while $\psi(B', S) \geq 20\kappa_G$ do
5:     Samples batch $b \in B'$ such that probability of picking a batch $b \in B'$ is $\frac{\psi_b(S)}{\psi(B',S)}$;
6:     $DEL \leftarrow DEL \cup b$;
7:     $B' \leftarrow \{B' \setminus b\}$;
8: end while
9: return $(DEL)$;
```

Lemma 5. *For a given B' and subset S procedure 1 returns a sub-collection $DEL \subset B'$, such that*

1. *For subset S the corruption score $\psi(B' \setminus DEL, S)$ of the new sub-collection is $< 20\kappa_G$.*
2. *Each batch $b \in B'$ that gets included in DEL is an adversarial batch with probability ≥ 0.95.*
3. *The subroutine deletes at-least $\psi(B', S) - 20\kappa_G$ batches.*

Proof. Step 4 in the algorithm ensures the first property. Next, to prove property 2, we bound the probability of deleting a good batch as

$$\sum_{b \in B' \cap B_G} \frac{\psi_b(S)}{\psi(B', S)} \leq \frac{\sum_{b \in B_G}\psi_b(S)}{\psi(B', S)} \leq \frac{\kappa_G}{20\kappa_G},$$

here the last step follows from condition 3 and while loop conditional in step 4. Property 3 follows from the observation that the total corruption score reduced is $\geq (\psi(B', S) - 20\kappa_G)$ and corruption score of one batch is bounded as $\psi_b(S) \leq 1$. ∎

We will use procedure 1 to successively update B to decrease the corruption score for different subsets $S \subseteq [k]$.

The next lemma show that even after successive updates the resultant sub-collection retains most of the good batches.

Lemma 6. *Let B' be the sub-collection after applying any number of successive deletion updates suggested by the Algorithm 1 on B, for any sequence of input subsets $S_1, S_2, ... \subseteq [k]$, then $|B' \cap B_G| \geq (1-\beta/6)|B_G|$, with probability $\geq 1 - O(e^{-k})$.*

Therefore, one can make successive updates to the collection of all batches B by deleting the batches suggested by procedure 1 for all subsets in $S \subseteq [k]$ one by one. This will result in a sub-collection $B' \subseteq B$, which still has most of the good batches and corruption score $\psi(B', S)$ bounded w.r.t. each subset S. However, this will take time exponential in k as there are 2^k subsets, and therefore, we want a computationally efficient method to find a subsets S with high corruption score and use procedure 1 for only those subsets. Next, we derive a novel method to achieve this objective.

We start with the following observation. A high corruption score of sub-collection B' with respect to an affected subset S implies a higher empirical variance of $\bar\mu_b(S)$ for such S than the expected value of the variance of $\bar\mu_b(S)$. While an affected subset S the empirical variance $\overline{\mathrm{V}}_B(S)$ is higher than expected, it is not necessarily higher than the empirical variance observed for all non-affected subset. This is because $\mathrm{V}(p(S))$, the expected value of the variance of $\bar\mu_b(S)$, for some subsets S may be larger compared to the other. Hence, simply finding the subset S with the largest variance doesn't work.

We use the following key insight to address this. Recall that the mean of empirical probabilities $\bar\mu_b(S)$ for good batches $b \in B_G$ converges, or equivalently $\bar p_{B_G}(S) \to p(S)$. This implies that $\mathrm{V}(\bar p_{B_G}(S)) \to \mathrm{V}(p(S))$. Also, since the empirical variance $\overline{\mathrm{V}}_{B_G}(S)$ converges to $\mathrm{V}(p(S))$, we get $\overline{\mathrm{V}}_{B_G}(S) - \mathrm{V}(\bar p_{B_G}(S)) \to 0$. Therefore, without corruption by the adversarial batches the difference between two estimators of the variance would be small for all subsets $S \subseteq [k]$, and its large value, we show in Lemma 7, can reliably detect any significant adversarial corruption. This happens because empirical variance of $\bar\mu_b(S)$ depends on the second moment whereas the other estimator $\mathrm{V}(\bar p_{B'}(S))$ of variance depends on the mean of $\bar\mu_b(S)$, hence the corruption affects the second estimator less severely. The next Lemma shows that the difference between the two variance estimators for subset S can indicate the corruption score w.r.t. subset S

Lemma 7. *Suppose the conditions 1- 3 holds. Then for any $B' \subseteq B$ such that $|B' \cap B_G| \geq (1-\frac{\beta}{6})|B_G|$ and let $\psi(B', S) = t \cdot \kappa_G$ for some $t \geq 0$, then following holds.*

$$\overline{V}_{B'}(S) - V(\bar p_{B'}(S)) \leq \Big(t + 4\sqrt{t} + 28\Big)\kappa_G,$$

$$\overline{V}_{B'}(S) - V(\bar p_{B'}(S)) \geq \Big(0.5t - 8\sqrt{t} - 25\Big)\kappa_G.$$

The next Lemma shows that a subset for which $\overline{\mathrm{V}}_{B'}(S) - \mathrm{V}(\bar p_{B'}(S))$ is large, can be found using a polynomial-time algorithm. In subsection 2.2 we derive the algorithm. We refer to this algorithm as $Detection-Algorithm$. The next lemma characterizes the performance of this algorithm. In subsection 2.2, we show that the algorithm achieves the performance guarantees of the next Lemma.

Lemma 8. *$Detection-Algorithm$ has run time polynomial in number of batches in its input sub-collection B' and alphabet size k, and returns $S^*_{B'}$ such that*

$$\begin{aligned}|\overline{V}_{B'}(S^*_{B'}) - V(\bar p_{B'}(S^*_{B'}))| \\ \geq 0.56 \max_{S\subseteq[k]} |\overline{V}_{B'}(S) - V(\bar p_{B'}(S))|.\end{aligned}$$

This leads us to the Robust distribution Learning Algorithm 2. Theorem 9 characterizes its performance.

Algorithm 2 Robust Distribution Estimator

```
Input: All batches $b \in B$, batch size $n$, alphabet size $k$, and $\beta$.
Output: Estimate $p^*$ of the distribution $p$.
$i \leftarrow 1$ and $B'_i \leftarrow B$.
while True do
    $S^*_{B'_i} = Detection-Algorithm(B'_i)$
    if $|\Delta_{B'_i}(S^*_{B'_i})| \leq 75\kappa_G$ then
        Break;
    end if
    med $\leftarrow$ med$(\bar\mu(S^*_{B'_i}))$.
    $DEL \leftarrow$Batch-Deletion$(B'_i, S^*_{B'_i}$, med).
end while
return $(p^* \leftarrow \bar p_{B'_i})$.
```

Theorem 9. *Suppose the conditions 1- 3 holds. Then Algorithm 2 runs in polynomial time and with probability $\geq 1 - O(e^{-k})$ returns a sub-collection $B'_f \subseteq B$ such that $|B'_f \cap B_G| \geq (1-\frac{\beta}{6})|B_G|$ and for $p^* = \bar p_{B'_f}$,*

$$||p^* - p||_1 \leq 100\beta\sqrt{\frac{\ln(6e/\beta)}{n}}.$$

Outline of the Proof of Theorem 9: In each round of the algorithm, Subroutine $Detection-Algorithm$ finds subsets for which the difference between the two variance estimates is large. Lemma 7 implies that the corruption w.r.t. this subset is large. The deletion subroutine updates the sub-collection of batches by removing some batches from it and reduces the corruption w.r.t. the detected subset S.

The algorithm terminates when for some sub-collection B'_f subroutine $Detection-Algorithm$ returns a subset S

small difference between the two variance estimators. Then Lemma 8 implies that the difference is small for all subsets. Lemma 7 further implies that if the difference between the two variance estimators is small then the corruption is bounded w.r.t. all subsets for sub-collection B'_f. Finally, Lemma 4 bounds the L_1 distance between $\bar{p}_{B'_f}$ and p. □

Proof of Theorem 1: Combining Lemma 3 and Theorem 9 yields Theorem 1.

2.1. Extension to $\eta > 0$

Recall that when $\eta > 0$, for each good batch $b \in B_G$, the distribution p_b of samples in batch b is close to the common target distribution p, such that $||p_b - p|| \le \eta$, instead of necessarily being the same. For simplicity, we have given the algorithm and the proof for only $\eta = 0$. The algorithm and the proof naturally extend to this more general case; here we get an extra additive dependence on η for the bounds in the lemmas and the theorems, and for the parameters of the algorithm. And with this slight modification in the parameters algorithm estimates p to a distance $O(\eta + \beta\sqrt{\ln(1/\beta)/n})$, and has the same sample and time complexity.

2.2. Efficient Detection Algorithm

In this subsection, we derive the procedure $Detection - Algorithm$, that runs in the polynomial time and achieves the performance in Lemma 8.

Given a collection B' of batches, we construct two covariance matrices $C^{EV}_{B'}$ and $C^{EM}_{B'}$ of size $k \times k$.

For an alphabet size k, we can treat the empirical probabilities estimates $\bar{\mu}_b$ and $\bar{p}_{B'}$ as a k-dimensional vector such that j^{th} entry denote the empirical probability of the j^{th} symbol. Recall that $\bar{p}_{B'}$ is the mean of $\bar{\mu}_b$, $b \in B'$.

The first covariance matrix, $C^{EV}_{B'}$, is the covariance matrix of $\bar{\mu}_b$ for $b \in B'$, with entries for $j, l \in [k]$,

$$C^{EV}_{B'}(j,l) = \frac{1}{|B'|}\sum_{b\in B'}(\bar{\mu}_b(j) - \bar{p}_{B'}(j))(\bar{\mu}_b(l) - \bar{p}_{B'}(l)).$$

The second covariance matrix $C^{EM}_{B'}$, is an expected covariance matrix of $\bar{\mu}_b$ if samples in the batches b were drawn from the distribution $\bar{p}_{B'}$. Hence, its entries are

$$C^{EM}_{B'}(j,l) = -\frac{\bar{p}_{B'}(j)\bar{p}_{B'}(l)}{n} \text{ for } j,l \in [k],\ j \neq l,$$

and

$$C^{EM}_{B'}(j,j) = \frac{\bar{p}_{B'}(j)(1-\bar{p}_{B'}(j))}{n}.$$

Let $D_{B'}$ be the difference of the two matrices:

$$D_{B'} = C^{EV}_{B'} - C^{EM}_{B'}.$$

For a vector $x \in \{0,1\}^k$, let

$$S(x) \triangleq \{j \in [k] : x(j) = 1\},$$

be the subset of $[k]$ corresponding to the vector x.

Observations

1. The sum of elements in any row and or column for both the covariance matrices, and hence also for the difference matrix, is zero, hence

$$C^{EV}_{B'}\mathbf{1} = C^{EM}_{B'}\mathbf{1} = D_{B'}\mathbf{1} = \mathbf{0}.$$

Proof: We show for $C^{EV}_{B'}$, the proof for $C^{EM}_{B'}$ is similar. For any $j \in [k]$,

$$\begin{aligned}&\sum_{l\in[k]} C^{EV}_{B'}(j,l)\\ &= \frac{1}{|B'|}\sum_{l\in[k]}\sum_{b\in B'}(\bar{\mu}_b(j) - \bar{p}_{B'}(j))(\bar{\mu}_b(l) - \bar{p}_{B'}(l))\\ &= \sum_{b\in B'}(\bar{\mu}_b(j) - \bar{p}_{B'}(j))\sum_{l\in[k]}(\bar{\mu}_b(l) - \bar{p}_{B'}(l))\\ &= \sum_{b\in B'}(\bar{\mu}_b(j) - \bar{p}_{B'}(j))(1-1) = 0.\end{aligned}$$

2. It is easy to verify that for any vector $x \in \{0,1\}^k$,

$$\begin{aligned}\langle C^{EV}_{B'}, xx^\intercal\rangle &= \frac{1}{|B'|}\sum_{b\in B'}(\bar{\mu}_b(S(x)) - \bar{p}_{B'}(S(x)))^2\\ &= \overline{\mathrm{V}}_{B'}(S(x)),\end{aligned}$$

the empirical variance of $\bar{\mu}_b(S(x))$ for $b \in B'$. Similarly,

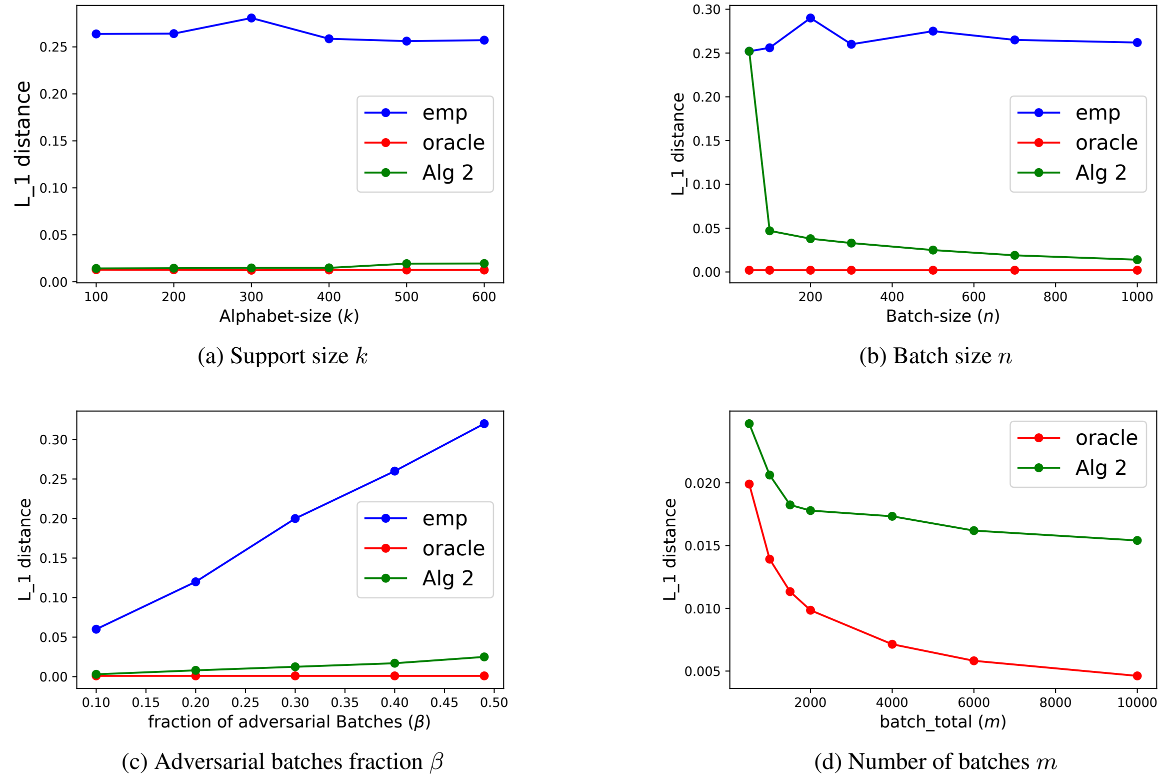
$$\begin{aligned}\langle C^{EM}_{B'}, xx^\intercal\rangle &= \frac{\bar{p}_{B'}(S(x))(1-\bar{p}_{B'}(S(x)))}{n}\\ &= \mathrm{V}(\bar{p}_{B'}(S(x))).\end{aligned}$$

Therefore,

$$\begin{aligned}\langle D_{B'}, xx^\intercal\rangle &= \langle C^{EV}_{B'} - C^{EM}_{B'}, xx^\intercal\rangle\\ &= \overline{\mathrm{V}}_{B'}(S(x)) - \mathrm{V}(\bar{p}_{B'}(S(x))).\end{aligned}$$

3. Note that $y \to \frac{1}{2}(y+\mathbf{1})$ is a 1-1 mapping from $\{-1,1\}^k \to \{0,1\}^k$, and that

$$\begin{aligned}&\langle C^{EV}_{B'}, \frac{1}{2}(y+\mathbf{1})\frac{1}{2}(y+\mathbf{1})^\intercal\rangle\\ &= \langle C^{EV}_{B'}, \frac{1}{4}(yy^\intercal + \mathbf{1}y^\intercal + y\mathbf{1}^\intercal + \mathbf{1}\mathbf{1}^\intercal)\rangle\\ &= \frac{1}{4}\langle C^{EV}_{B'}, yy^\intercal\rangle.\end{aligned}$$



(a) Support size k

(b) Batch size n

(c) Adversarial batches fraction β

(d) Number of batches m

Figure 1. L_1 estimation error with different Parameters

Let

$$y = \arg \max_{y\in\{-1,1\}^k} |\langle D_{B'}, yy^\intercal\rangle|.$$

Then from y one can recover the corresponding subset $S(x)$, with $x = \frac{1}{2}(y + \mathbf{1})$, maximizing

$$|\overline{\mathrm{V}}_{B'}(S(x)) - \mathrm{V}(\bar{p}_{B'}(S(x)))|.$$

In (Alon & Naor, 2004), Alon et al. derives a polynomial-time approximation algorithm for the above optimization problem. The algorithm first uses a semi-definite relaxation of the problem and then uses randomized integer rounding techniques based on Grothendieck's Inequality. Their algorithm recovers $y_{B'}$ such that

$$|\langle D_{B'}, y_{B'}y_{B'}^\intercal\rangle| \geq 0.56 \max_{y\in\{-1,1\}^k} |\langle D_{B'}, yy^\intercal\rangle|.$$

Let $x_{B'} = \frac{1}{2}(y + \mathbf{1})$. Then from observation 3 it follows that

$$|\langle D_{B'}, x_{B'}x_{B'}^\intercal\rangle| \geq 0.56 \max_{x\in\{0,1\}^k} |\langle D_{B'}, xx^\intercal\rangle|.$$

Therefore for $S^*_{B'} = S(x_{B'})$ we get

$$|\overline{\mathrm{V}}_{B'}(S^*_{B'}) - \mathrm{V}(\bar{p}_{B'}(S^*_{B'}))| \\ \geq 0.56 \max_{S\subseteq[k]} |\overline{\mathrm{V}}_{B'}(S) - \mathrm{V}(\bar{p}_{B'}(S))|.$$

3. Experiments

We evaluate the algorithm's performance on synthetic data.

We compare the estimator's performance with two others: 1) an oracle that knows the identity of the adversarial batches. The oracle ignores the adversarial batches and computes the empirical estimators based on remaining batches and is not affected by the presence of adversarial batches. The estimation error achieved by the oracle is the best one could get, even without the adversarial corruptions. 2) a naive-empirical estimator that computes the empirical distribution of all samples across all batches.

Two non-trivial estimators have been derived for this problem. Both have prohibitively large sample and/or computational complexity. The estimator in (Qiao & Valiant, 2017) has run time exponential in k, making it impractical. The time and sample complexities of the estimator in (Chen et al., 2019) are either super-polynomial or a high-degree polynomial, depending on the range of the parameters (k,n,$1/\beta$), rendering their simulation prohibitively high as well.

We tried different adversarial distributions and found that the major determining factor of the effectiveness of the adversarial batches is the distance between the adversarial distribution and the target distribution. If the adversarial distribution is too far, then adversarial batches are easier to detect. For this scenario our algorithm is even more effective than the performance limits shown in Theorem 1 and the performance between our algorithm and the oracle is almost indistinguishable. When the adversarial distribution is very close to the target distribution p, the adversarial batches don't affect the estimation error by much. The estimator has the worst performance when the adversary chooses the distribution of its batches at an optimal distance from target distribution. This optimal distance differs with the value of the algorithm's parameters. Hence for each choice of algorithm parameters, we tried adversarial distributions at varying distances and reported the worst performance of our estimator.

All experiments were performed on a laptop with a configuration of 2.3 GHz Intel Core i7 CPU and 16 GB of RAM. We choose the parameters for the algorithm by using a small simulation. We provide all codes and implementation details in the supplementary material.

We show four plots here. In each plot we vary one parameter and plot the L_1 loss incurred by all three estimators. For each experiment, we ran ten trials and reported the average L_1 distance achieved by each estimator.

For the first plot we fix batch-size $n = 1000$ and $\beta = 0.4$ and vary alphabet size k. We generate $m = k/(0.4)^2$ batches for each k. Our algorithm's performance show no significant change as the size of alphabet increases and its performance nearly matches the performance of the Oracle and outperforms the naive estimator by order of magnitudes.

In the the second plot we fix $\beta = 0.4$ and $k = 200$ and vary batch size n. We choose $m = 40 \times \frac{k}{\beta^2} \times \frac{1000}{n}$, this keeps the total number of samples $n \times m$, constant for different n. We see that the L_1 loss incurred by our estimator is much smaller than the naive empirical estimator and it diminishes as the batch size increases and comes very close to the performance of the oracle. Note that this roughly matches the decay $O(1/\sqrt{n})$ of L_1 error characterized in both the lower and the upper bounds.

For the next plot we fix batch size $n = 1000$ and $k = 200$. The number of good batches $(1-\beta)m = 400k$ is kept same. We vary the adversarial noise level and plot the performance of all estimators. We tested our estimator for fraction of adversarial batches as high as 0.49 and still our estimator recovered p to a good accuracy and in fact at the lower noise level it is essentially similar to the oracle and it increases (near) linearly with the noise level β as in Theorem 1,

In the last plot we fixed all other parameters $n = 1000$, $k = 200$, and $\beta = 0.4$ and varied the number of batches. We see that the performance of oracle keep improving as number of bathes increases. But for our algorithm it decreases initially but later it saturates as predicted by adversarial batch lower bound.

Acknowledgements

We thank Vaishakh Ravindrakumar and Yi Hao for their helpful comments in the prepration of this manuscript.

We are grateful to the National Science Foundation (NSF) for supporting this work through grants CIF-1564355 and CIF-1619448.

References

Alon, N. and Naor, A. Approximating the cut-norm via grothendieck's inequality. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 72–80. ACM, 2004.

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

Charikar, M., Steinhardt, J., and Valiant, G. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 47–60. ACM, 2017.

Chen, S., Li, J., and Moitra, A. Efficiently learning structured distributions from untrusted batches. *arXiv preprint arXiv:1911.02035*, 2019.

Chen, S., Li, J., and Moitra, A. Learning structured distributions from untrusted batches: Faster and simpler. *arXiv preprint arXiv:2002.10435*, 2020.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robust estimators in high dimensions without the computational intractability. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 655–664. IEEE, 2016.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 999–1008. JMLR. org, 2017.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

Jain, A. and Orlitsky, A. A general method for robust learning from batches. *arXiv preprint arXiv:2002.11099*, 2020.

Kamath, S., Orlitsky, A., Pichapati, D., and Suresh, A. T. On learning distributions from their samples. In *Conference on Learning Theory*, pp. 1066–1100, 2015.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 665–674. IEEE, 2016.

McMahan, H. B. and Ramage, D. https://research.google.com/pubs/pub44822.html. 2017.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Philippe, R. 18.s997 high-dimensional statistics. *Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA*, 2015.

Qiao, M. and Valiant, G. Learning discrete distributions from untrusted batches. *arXiv preprint arXiv:1711.08113*, 2017.

Steinhardt, J., Charikar, M., and Valiant, G. Resilience: A criterion for learning in the presence of arbitrary outliers. *arXiv preprint arXiv:1703.04940*, 2017.

Tukey, J. W. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pp. 448–485, 1960.