ROBUST LEARNING OF FIXED-STRUCTURE BAYESIAN NETWORKS IN NEARLY-LINEAR TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of learning Bayesian networks where an ϵ-fraction of the samples are adversarially corrupted. We focus on the fully-observable case where the underlying graph structure is known. In this work, we present the first nearly-linear time algorithm for this problem with a dimension-independent error guarantee. Previous robust algorithms with comparable error guarantees are slower by at least a factor of (d/ϵ), where d is the number of variables in the Bayesian network and ϵ is the fraction of corrupted samples. Our work establishes a direct connection between robust learning of Bayesian networks and robust mean estimation. By exploiting this connection, our algorithm and analysis are considerably simpler than those in previous work. As a subroutine in our algorithm, we develop a robust mean estimation algorithm whose runtime is nearly-linear in the number of nonzeros in the input, which may be of independent interest.

1 INTRODUCTION

Probabilistic graphical models (Koller & Friedman, 2009) offer an elegant and succinct way to represent structured high-dimensional distributions. The problem of inference and learning in probabilistic graphical models is an important problem that arises in many disciplines (see Wainwright & Jordan (2008) and the references therein), which has been studied extensively during the past decades (see, e.g., Chow & Liu (1968); Dasgupta (1997); Abbeel et al. (2006); Wainwright et al. (2006); Anandkumar et al. (2012); Santhanam & Wainwright (2012); Loh & Wainwright (2012); Bresler et al. (2013; 2014); Bresler (2015)).

Bayesian networks (Jensen & Nielsen, 2007) are an important family of probabilistic graphical models that represent conditional dependence by a directed graph (see Section 2 for a formal definition). In this paper, we study the problem of learning Bayesian networks where an ϵ-fraction of the samples are adversarially corrupted. We focus on the most fundamental setting: all variables are binary and observable, and the structure of the Bayesian network is given to the algorithm.

Formally, we work with the following corruption model:

Definition 1.1 (ϵ-Corrupted Set of Samples)**.** *Given $0 < \epsilon < 1/2$ and a distribution family $\mathcal{P}$ on $\mathbb{R}^d$, the algorithm first specifies the number of samples N, and N samples $X_1, X_2, \ldots, X_N$ are drawn from some unknown $P \in \mathcal{P}$. The adversary inspects the samples, the ground-truth distribution P, and the algorithm, and then replaces ϵN samples with arbitrary points. The set of N points is given to the algorithm as input. We say that a set of samples is ϵ-corrupted if it is generated by this process.*

This is a strong corruption model and it generalizes many existing models (including but not limited to, e.g., Huber's contamination model (Huber, 1964)). Given as input an ϵ-corrupted set of samples, we would like to design robust algorithms for learning Bayesian networks whose error guarantee is independent of the dimensionality d of the problem.

In the fully-observable fixed-structure setting, the problem is straightforward when there are no corruption. We know that the empirical estimator (which computes the empirical conditional probabilities) is sample efficient and runs in linear time (Dasgupta, 1997). The problem becomes much more challenging when there are corruptions.

Recent work (Diakonikolas et al., 2016) gave polynomial-time robust algorithms for learning binary product distributions with error $O(\epsilon\sqrt{\log 1/\epsilon})$. Subsequently, (Cheng et al., 2018) showed that it is

possible to learn Bayesian networks with similar error guarantees under mild assumptions. However, the algorithm in (Cheng et al., 2018) runs in time $\Omega(Nd^2/\epsilon)$, which is slower by at least a factor of (d/ϵ) compared to the fastest non-robust estimator. Motivated by this gap in the running time, in this work we want to resolve the following question:

> *Can we design a robust algorithm for learning Bayesian networks in the fixed-structure fully-observable setting that runs in nearly-linear time?*

1.1 Our Results and Contributions

We resolve this question affirmatively by proving Theorem 1.2. A Bayesian network is *c-balanced* if all its conditional probabilities are between c and $1-c$. Let m be the size of the *conditional probability table* and α be the *minimum parental configuration probability* (defined in Section 2).

Theorem 1.2 (informal statement)**.** *Consider an ϵ-corrupted set of $N = \widetilde{\Omega}(m/\epsilon^2)$ samples drawn from a d-dimensional Bayesian network P. Suppose P is c-balanced and has minimum parental configuration probability α, where both c and α are universal constants. We can compute a Bayesian network Q in time $\widetilde{O}(Nd)$ such that $d_{\mathrm{TV}}(P, Q) \leq \epsilon\sqrt{\ln(1/\epsilon)}$.* [1]

For simplicity, we stated our result in the very special case where both c and α are $\Omega(1)$. Our approach works for general values of α and c, where our error guarantee degrades gracefully as α and c gets smaller. A formal version of Theorem 1.2 is given as Theorem 4.1 in Section 4.

The sample complexity of our algorithm is optimal up to polylogarithmic factors. A sample complexity lower bound of $\Omega(m/\epsilon^2)$ holds even without corrupted samples (Canonne et al., 2017).

Our Contributions. We establish a more direct connection between robust mean estimation and robustly learning Bayesian networks. At a high level, we show that one can essentially reduce robust learning of Bayesian networks to robust mean estimation. This allows us to utilize recent advancements in robust mean estimation to get faster algorithms.

As a subroutine in our approach, we develop a nearly-linear time algorithm for robust mean estimation when the input is sparse. More specifically, our algorithm runs in time nearly linear in the total number of nonzero entries in the input, which may be of independent interest.

Our algorithm and analysis is considerably simpler than those in previous work. Consider binary product distribution as an example. Previous algorithm in Cheng et al. (2018) used the fact that the true covariance matrix should be diagonal (because each coordinate is independent), and tries to remove samples to make the empirical covariance matrix closer to a diagonal matrix. In contrast, in our approach, we show that it is sufficient to minimize the spectral norm of the empirical covariance matrix, regardless of whether it is close to being diagonal or not.

1.2 Related Work

Learning in the presence of outliers has been studied since the 1960s (Huber, 1964). After decades of work, many efficient robust estimators have been discovered (Tukey, 1975; Devroye & Györfi, 1985; Chen et al., 2018). However, these classic statistical estimators either has an error that scales polynomially with the dimension d, or is hard to compute in the worst-case.

Recent work (Diakonikolas et al., 2016; Lai et al., 2016) gave the first polynomial-time algorithms for a wide range of high-dimensional unsupervised learning tasks (e.g., mean and covariance estimation) that achieves dimension-independent error guarantees. After the dissemination of (Diakonikolas et al., 2016; Lai et al., 2016), the algorithmic question of designing efficient robust estimators in high-dimensions has attracted a lot of recent attention and has been studied extensively (see, e.g., Balakrishnan et al. (2017); Charikar et al. (2017); Diakonikolas et al. (2017a;b; 2018a); Steinhardt et al. (2018); Diakonikolas et al. (2018b;c); Hopkins & Li (2018); Kothari et al. (2018); Prasad et al. (2018); Diakonikolas et al. (2019a); Klivans et al. (2018); Diakonikolas et al. (2019b); Liu et al. (2020); Cheng et al. (2018; 2020); Zhu et al. (2020)). In particular, Cheng et al. (2018) gave the first polynomial-time algorithms for the robust learning of fixed-structure Bayesian networks.

[1] Throughout the paper, we use $\widetilde{O}(f)$ to denote $O(f \ \mathrm{polylog}(f))$.

While recent work gave polynomial-time robust algorithms for many fundamental tasks, these algorithms are often significantly slower than the fastest non-robust algorithms (e.g., sample average for mean estimation). The work of Cheng et al. (2019a) gave the first nearly-linear time algorithm for robust mean estimation and initiated the direction of designing robust estimators that are as efficient as their non-robust counterparts. Since then, there have been several works on obtaining faster asymptotic algorithms for robust high-dimensional estimation (Dong et al., 2019; Depersin & Lecué, 2019; Cheng et al., 2019b; Li & Ye, 2020; Cherapanamjeri et al., 2020).

Organization. In Section 2, we define our notations and provide some background on robust learning of Bayesian networks and robust mean estimation. In Section 3, we give an overview of our approach and highlight some of our key technical results. In Section 4, we present our algorithm for robust learning of Bayesian networks and prove our main result.

2 PRELIMINARIES

Bayesian Networks. Fix a d-node directed acyclic graph H whose nodes are labelled $[d] = \{1, 2, \ldots, d\}$ in topological order (every edge goes from a node with smaller index to one with larger index). Let $\mathrm{Parents}(i)$ be the parents of node i in H. A probability distribution P on $\{0,1\}^d$ is a *Bayesian network* (or *Bayes net*) with graph H if, for each $i \in [d]$, we have that $\Pr_{X\sim P}[X_i = 1 \mid X_1, \ldots, X_{i-1}]$ depends only on the values X_j where $j \in \mathrm{Parents}(i)$.

Conditional Probability Table. Let P be a Bayesian network with graph H. Let $\Gamma = \{(i, a) : i \in [d], a \in \{0,1\}^{|\mathrm{Parents}(i)|}\}$ be the set of all possible parental configurations. Let $m = |\Gamma|$. For $(i, a) \in \Gamma$, the *parental configuration* $\Pi_{i,a}$ is defined to be the event that $X(\mathrm{Parents}(i)) = a$. The conditional probability table $p \in [0,1]^m$ of P is given by $p_{i,a} = \Pr_{X\sim P}[X(i) = 1 \mid \Pi_{i,a}]$.

In this paper, we often index p as an m-dimensional vector. We use the notation p_k and the associated events Π_k, where each $k \in [m]$ stands for an $(i, a) \in \Gamma$ lexicographically ordered.

Notations. For a vector v, let $\|v\|_2$ and $\|v\|_\infty$ be the ℓ_2 and ℓ_∞ norm of v respectively. We write $\sqrt{v}$ and $1/v$ for the entrywise square root and entrywise inverse of a vector v respectively. For two vectors x and y, we write $x^\top y$ for their inner product, and $x \circ y$ for their entrywise product.

We use I to denote the identity matrix. For a matrix M, let M_i be the i-th column of M, and let $\|M\|_2$ be the spectral norm of M. For a vector $v \in \mathbb{R}^n$, let $\mathrm{diag}(v) \in \mathbb{R}^{n\times n}$ denote a diagonal matrix with v on the diagonal.

Throughout this paper, we use P to denote the ground-truth Bayesian network. We use d for the dimension (i.e., the number of nodes) of P, N for the number of samples, ϵ for the fraction of corrupted samples, and $m = \sum_{i=1}^{d} 2^{|\mathrm{Parents}(i)|}$ for the size of the conditional probability table of P. We use $p \in \mathbb{R}^m$ to denote the (unknown) ground-truth conditional probabilities of P, and $q \in \mathbb{R}^m$ for our current guess of p.

Let $G^\star$ be the original set of N uncorrupted samples drawn from P. After the adversary corrupts an ϵ-fraction of $G^\star$, let $G \subseteq G^\star$ be the remaining set of good samples, and B be the set of bad samples added by the adversary. The set of samples $S = G \cup B$ is given to the algorithm as input. Let $X \in \mathbb{R}^{N\times d}$ denote the sample matrix whose i-th column $X_i \in \mathbb{R}^d$ is the i-th input sample. Abusing notation, we sometimes also use X as a random variable (e.g., a sample drawn from P).

We use $\pi^P \in \mathbb{R}^m$ to denote the parental configuration probabilities of P. That is, $\pi^P_k = \Pr_{X\sim P}[X \in \Pi_k]$. For a set S of samples, we use $\pi^S \in \mathbb{R}^m$ to denote the *empirical* parental configuration probabilities over S: $\pi^S_k = \Pr_X[X \in \Pi_k]$ where X is uniformly drawn from S.

Balance and Minimum Configuration Probability. We say a Bayesian network P is c-balanced if all conditional probabilities of P are between c and $1 - c$. We use α for the minimum probability of parental configuration of P: $\alpha = \min_k \pi^P_k$.

In this paper, we assume that the ground-truth Bayesian network is c-balanced, and its minimum parental configuration probability α satisfies that $\alpha = \Omega\big((\epsilon\sqrt{\ln(1/\epsilon)})^{2/3} c^{-1/3})\big)$. Without loss of generality, we further assume that both c and α are given to the algorithm.

2.1 TOTAL VARIATION DISTANCE BETWEEN BAYESIAN NETWORKS

Let P and Q be two distributions supported on a finite domain D. For a set of outcomes A, let $P(A) = \Pr_{X\sim P}[X \in A]$. The total variation distance between P and Q is defined as

$$d_{\mathrm{TV}}(P, Q) = \max_{A\subseteq D} |P(A) - Q(A)| \ .$$

For two balanced Bayesian networks that share the same structure, it is well-known that the closeness in their conditional probabilities implies their closeness in total variation distance. Formally, we use the following lemma from Cheng et al. (2018), which upper bounds the total variation distance between two Bayesian networks in terms of their conditional probabilities.

Lemma 2.1 (Cheng et al. (2018))**.** *Let P and Q be two Bayesian networks that share the same structure. Let p and q denote the conditional probability tables of P and Q respectively. We have*

$$(d_{\mathrm{TV}}(P, Q))^2 \le 2\sum_k \sqrt{\pi_k^P \pi_k^Q} \frac{(p_k - q_k)^2}{(p_k + q_k)(2 - p_k - q_k)} \ .$$

2.2 EXPANDING THE DISTRIBUTION TO MATCH CONDITIONAL PROBABILITY TABLE

Lemma 2.1 states that to learn a known-structure Bayesian network P, it is sufficient to learn its conditional probabilities p. However, a given coordinate of $X \sim P$ may contain information about multiple conditional probabilities (depending on which parental configuration happens).

To address this issue, we use a similar approach as in Cheng et al. (2018). We expand each sample X into an m-dimensional vector $f(X, q)$, such that each coordinate of $f(X, q)$ corresponds to an entry in the conditional probability table. Intuitively, $q \in \mathbb{R}^m$ is our current guess for p, and initially we set q to be the empirical conditional probabilities. We use q to fill in the missing entries in $f(X, q)$ for which the parental configurations fail to happen.

Definition 2.2. *Let $f(X, q)$ for $\{0, 1\}^d \times \mathbb{R}^m \to \mathbb{R}^m$ be defined as follows:*

$$f(X, q)_{i,a} = \begin{cases} X_i - q_{i,a} & X \in \Pi_{i,a} \\ 0 & \text{otherwise} \end{cases}$$

When $X \sim P$ and $q = p$, the distribution of $f(X, p)$ has many good properties. Using the conditional independence of Bayesian networks, we can compute the first and second moment of $f(X, p)$ and show that $f(X, p)$ has subgaussian tails.

Lemma 2.3. *For $X \sim P$ and $f(X, p)$ as defined in Definition 2.2, we have*

(i) $\mathbb{E}(f(X, p)) = 0$. *(ii)* $\mathrm{Cov}[f(X, p)] = \mathrm{diag}(\pi^P \circ p \circ (1 - p))$.

(iii) For any unit vector $v \in \mathbb{R}^m$, we have $\Pr_{X\sim P}\left[|v^\top f(X, p)| \ge T\right] \le 2\exp(-T^2/2)$.

We defer the proof of Lemma 2.3 to Appendix A. A slightly stronger version of Lemma 2.3 was proved in Cheng et al. (2018), which discusses tail bounds for $f(X, q)$. For our analysis, Lemma 2.3 is sufficient.

For general values of q, we can similarly compute the mean of $f(X, q)$:

Lemma 2.4. *Let π^P denote the parental configuration of P. For $X \sim P$ and $f(X, q)$ as defined in Definition 2.2, we have $\mathbb{E}[f(X, q)] = \pi^P \circ (p - q)$.*

2.3 DETERMINISTIC CONDITIONS ON GOOD SAMPLES

To avoid dealing with the randomness of the good samples, we require the following deterministic conditions to hold for the original set $G^\star$ of N good samples (before the adversary's corruption).

We prove in Appendix A that these three conditions hold simultaneously with probability at least $1 - \tau$ if we draw $N = \Omega(m \log(m/\tau)/\epsilon^2)$ samples from P.

The first condition states that we can obtain a good estimation of p from $G^\star$. Let $p^{G^\star}$ denote the empirical conditional probabilities over $G^\star$. We have

$$\left\|\sqrt{\pi^P} \circ (p - p^{G^\star})\right\|_2 \le O(\epsilon) \ . \tag{1}$$

The second condition says that we can estimate the parental configuration probabilities π^P from any $(1-2\epsilon)$-fraction of $G^\star$. Formally, for any subset $T \subset G^\star$ with $|T| \geq (1-2\epsilon)N$, we have

$$\left\|\pi^T - \pi^P\right\|_\infty \leq O(\epsilon) \ . \tag{2}$$

The third condition is that the empirical mean and covariance of any $(1-2\epsilon)$-fraction of $G^\star$ are very close to the true mean and covariance of $f(X,p)$. Formally, for any subset $T \subset G^\star$ with $|T| \geq (1-2\epsilon)N$, we require the following to hold for $\delta_1 = \epsilon\sqrt{\ln 1/\epsilon}$ and $\delta_2 = \epsilon \ln(1/\epsilon)$:

$$\left\|\frac{1}{|T|}\sum_{i\in T} f(X_i,p)\right\|_2 \leq O(\delta_1) \ , \quad \left\|\frac{1}{|T|}\sum_{i\in T} f(X_i,p)f(X_i,p)^\top - \Sigma\right\|_2 \leq O(\delta_2) \ , \tag{3}$$

where $\Sigma = \mathrm{Cov}[f(X,p)] = \mathrm{diag}(\pi^P \circ p \circ (1-p))$.

2.4 Robust Mean Estimation and Stability Conditions

Robust mean estimation is the problem of learning the mean of a d-dimensional distribution from an ϵ-corrupted set of samples. As we will see in later sections, to robustly learn Bayesian networks, we repeatedly use robust mean estimation algorithms as a subroutine.

Recent work (Diakonikolas et al., 2016; Lai et al., 2016) gave the first polynomial-time algorithms for robust mean estimation with dimension-independent error guarantees. The key observation in Diakonikolas et al. (2016) is the following: if the empirical mean is inaccurate, then many samples must be far from the true mean in roughly the same direction. Consequently, these samples must alter the variance in this direction more than they distort the mean. Therefore, if the empirical covariance behaves as we expect it to be, then the empirical mean provides a good estimate to the true mean.

Many robust mean estimation algorithms follow the above intuition, and they require the following stability condition to work (Definition 2.5). Roughly speaking, the stability condition states that the mean and covariance of the good samples are close to that of the true distribution, and more importantly, this continues to hold if we remove any 2ϵ-fraction of the samples.

Definition 2.5 (Stability Condition (see, e.g., Diakonikolas & Kane (2019))). *Fix $0 < \epsilon < \frac{1}{2}$. Fix a d-dimensional distribution X with mean μ_X. We say a set S of samples is (ϵ,β,γ)-stable with respect to X, if for every subset $T \subset S$ with $|T| \geq (1-2\epsilon)|S|$, the following conditions hold:*

(i) $\left\|\frac{1}{|T|}\sum_{X\in T}(X-\mu_X)\right\|_2 \leq \beta$, *(ii)* $\left\|\frac{1}{|T|}\sum_{X\in T}(X-\mu_X)(X-\mu_X)^\top - I\right\|_2 \leq \gamma$.

Subsequent work (Cheng et al., 2019a; Dong et al., 2019; Depersin & Lecué, 2019) improved the runtime of robust mean estimation to nearly-linear time. In particular, we have the following lemma from Dong et al. (2019).

Lemma 2.6 (Robust Mean Estimation in Nearly-Linear Time (Dong et al., 2019)). *Fix a set of N samples $G^\star$ in $\mathbb{R}^d$. Suppose $G^\star$ is (ϵ,β,γ)-stable with respect to a d-dimensional distribution X with mean $\mu_X \in \mathbb{R}^d$. Let S be an ϵ-corrupted version of $G^\star$. Given as input S, ϵ, β, γ, these exists an algorithm that can output an estimator $\hat{\mu} \in \mathbb{R}^d$ in time $\widetilde{O}(Nd)$, such that with high probability, $\|\hat{\mu} - \mu_X\|_2 \leq O(\sqrt{\epsilon\gamma} + \beta + \epsilon\sqrt{\log 1/\epsilon})$.*

As we will see later, a black-box use of Lemma 2.6 does not give the desired runtime in our setting. Instead, we extend Lemma 2.6 to handle sparse input such that it runs in time nearly-linear in the number of non-zeros in the input (see Lemma 3.3).

3 Overview of Our Approach

In this section, we give an overview of our approach and highlight some of our key technical results.

To robustly learn the ground-truth Bayesian network P, it is sufficient to learn its conditional probabilities $p \in \mathbb{R}^m$. At a high level, we start with a guess $q \in \mathbb{R}^m$ for p and then iteratively improve our guess to get closer to p. For any $q \in \mathbb{R}^m$, we can expand the input samples into m-dimensional vectors $f(X,q)$ as in Definition 2.2. We first show that the expectation of $f(X,q)$ gives us useful information about $(p-q)$.

Recall that π^P is the parental configuration probabilities of P. By Lemma 2.4, we have

$$\mathbb{E}_{X\sim P}[f(X,q)] = \pi^P \circ (p-q) \ .$$

Note that if we had access to this expectation and the vector π^P, we could recover p immediately: we can set $q' = \mathbb{E}[f(X,q)] \circ (1/\pi^P) + q$ which simplifies to $q' = p$.

Note that since $\{X_i\}_{i\in S}$ is an ϵ-corrupted set of samples of P, we know that $\{f(X_i,q)\}_{i\in S}$ is an ϵ-corrupted set of samples of the distribution $f(X,q)$ (with $X \sim P$). Therefore, we can run robust mean estimation algorithms on $\{f(X_i,q)\}_{i\in S}$ to learn $\mathbb{E}[f(X,q)]$. It turns out a good approximation of $\mathbb{E}[f(X,q)]$ can help us improve our current guess q.

There are two main difficulties in getting this approach to work.

The first difficulty is that, to use robust mean estimation algorithms, we need to show that $f(X,q)$ satisfies the stability condition in Definition 2.5. This requires us to analyze the first two moments and tail bounds of $f(X,q)$. Consider the second moment for example. Ideally, we would like to have a statement of the form $\mathrm{Cov}[f(X,q)] \approx \mathrm{Cov}[f(X,p)] + (p-q)(p-q)^\top$, but this is false because we only have $f(X,p)_k - f(X,q)_k = (p-q)_k$ if the k-th parental configuration happens for X. Intuitively, the "error" $(p-q)$ is shattered into all samples where each sample only gives d out of m coordinates of $(p-q)$, and there is no succinct representation for $\mathrm{Cov}[f(X,q)]$.

The second difficulty is that $f(X,q)$ is m-dimensional. We cannot explicitly write down all the samples $\{f(X_i,q)\}_{i=1}^N$, because this takes time $\Omega(Nm)$, which could be much slower than our desired running time of $\widetilde{O}(Nd)$. Similarly, a black-box use of nearly-linear time robust mean estimation algorithms (e.g., Lemma 2.6) runs in time $\Omega(Nm)$, which is too slow.

In the rest of this section, we explain how we handle these two issues.

Stability Condition of $f(X,q)$. Because the second-order stability condition in Lemma 2.3 is defined with respect to I, we first scale the samples so that the covariance of $f(X,p)$ becomes I. Lemma 2.3 shows that $\mathrm{Cov}[f(X,p)] = \mathrm{diag}(\pi^P \circ p \circ (1-p))$. To make it close to I, we can multiply the k-th coordinate of $f(X,p)$ by $(\pi_k^P p_k(1-p_k))^{-1/2}$. However, we do not know the exact value of π^P or p, instead we use the corresponding empirical estimates π^S and q^S (see Algorithm 1).

Definition 3.1. *Let π^S and q^S denote the parental configuration probabilities and conditional means estimated over S. Let $s = 1/\sqrt{(\pi^S \circ q^S \circ (1-q^S))}$. Throughout this paper, for a vector $v \in \mathbb{R}^m$, we use $\hat{v} \in \mathbb{R}^m$ to denote $v \circ s$. In particular, we have $\hat{X}_i = X_i \circ s$ (and similarly $\hat{p}$, $\hat{q}$, $\hat{f}(x,q)$).*

Now we analyze the concentration bounds for $\hat{f}(X,q)$. Formally, we prove the following lemma.

Lemma 3.2. *Assume the conditions in Section 2.3 hold for the original set of good samples $G^\star$. Then, for $\delta_1 = \epsilon\sqrt{\log 1/\epsilon}$ and $\delta_2 = \epsilon \log(1/\epsilon)$, the set of samples $\left\{\hat{f}(X_i,q)\right\}_{i\in G^\star}$ is*

$$\left(\epsilon, O\Big(\frac{\delta_1}{\sqrt{\alpha c}} + \epsilon \left\|\hat{p}-\hat{q}\right\|_2\Big), O\Big(\frac{\delta_2}{\alpha c} + B + \sqrt{B}\Big)\right) \text{-stable,}$$

where $B = \|\sqrt{\pi^P} \circ (\hat{p}-\hat{q})\|_2^2$.

We provide some intuition for Lemma B.3 and defer its proof to Appendix B.

For the first moment, the difference between $\mathbb{E}[\hat{f}(X,q)]$ and the empirical mean of $\hat{f}(X,q)$ comes from several places. Even if $q = p$, we would incur an error of δ_1 from the concentration bound in Equation equation 3, which is at most $\delta_1(\alpha c)^{-1/2}$ after the scaling by s. Moreover, on average π_k^P fraction of the samples gives us information about $(\hat{p}-\hat{q})_k$. Since an ϵ-fraction of the samples are removed when proving stability, we may only have $(\pi_k^P - \epsilon)$-fraction instead, which introduces an error of $\epsilon \left\|\hat{p}-\hat{q}\right\|_2$. This is why the first-moment parameter is $\big(\delta_1(\alpha c)^{-1/2} + \epsilon \left\|\hat{p}-\hat{q}\right\|_2\big)$.

For the second moment, after the scaling, we have $\mathrm{Cov}[\hat{f}(X,p)] \approx I$. Ideally, we would like to prove $\mathrm{Cov}[\hat{f}(X,q)] \approx I + (\pi^P \circ (\hat{p}-\hat{q}))(\pi^P \circ (\hat{p}-\hat{q}))^\top$, but this is too good to be true. For two coordinates $k \neq \ell$, whether a sample gives information about $(\hat{p}-\hat{q})_k$ or $(\hat{p}-\hat{q})_\ell$ is not independent. We can upper bound the probability that both parental configurations happen by $\min(\pi_k^P, \pi_\ell^P)$. If they were independent we would have a bound of $\pi_k^P \pi_\ell^P$. The difference in these two upper bounds is intuitively why $\sqrt{\pi^P}$ appears in the second-moment parameter. See Appendix B for more details.

Robust Mean Estimation with Sparse Input. To overcome the second difficulty, we exploit the sparsity of the expanded vectors. Observe that each vector $f(X, q)$ is guaranteed to be d-sparse because exactly d parental configuration can happen (see Definition 2.2). The same is true for $\hat{f}(X, q)$ because scaling does not change the number of nonzeros. Therefore, there are in total $O(Nd)$ nonzero entries in the set of samples $\{\hat{f}(X, q)\}_{i \in S}$.

We develop a robust mean estimation algorithm that runs in time nearly-linear in the *number of nonzeros* in the input. Combined with the above argument, if we only invoke this mean estimation algorithm polylogarithmic times, we can get the desired running time of $\widetilde{O}(Nd)$.

Lemma 3.3. *Consider the same setting as in Lemma 2.6. There is an algorithm $\mathscr{O}_{mean}$ with the same error guarantee that runs in time $\widetilde{O}(\mathrm{nnz}(S) + N + m)$ where $\mathrm{nnz}(S)$ is the total number of nonzeros in S. That is, given an ϵ-corrupted version of an $(\epsilon, \beta, \gamma)$-stable set of N samples w.r.t. a d-dimensional distribution with mean μ_X, the algorithm $\mathscr{O}_{mean}$ outputs an estimator $\hat{\mu} \in \mathbb{R}^d$ in time $\widetilde{O}(\mathrm{nnz}(S)+N+m)$ such that with high probability, $\|\hat{\mu} - \mu_X\|_2 \leq O(\sqrt{\epsilon\gamma}+\beta+\epsilon\sqrt{\log 1/\epsilon})$.*

We prove Lemma 3.3 by extending the algorithm in Dong et al. (2019) to handle sparse input. The main computation bottleneck of recent nearly-linear time robust mean estimation algorithms (Cheng et al., 2018; Dong et al., 2019) is in using the matrix multiplicative weight update (MMWU) method. In each iteration of MMWU, a score is computed for each sample. Roughly speaking, this score indicates whether one should continue to increase the weight on the corresponding sample. Previous algorithms use the Johnson-Lindenstrauss lemma to approximate the scores for all N samples simultaneously. We show that the sparsity of the input vectors allows for faster application of the Johnson-Lindenstrauss lemma, and all N scores can be computed in time nearly-linear in $\mathrm{nnz}(S)$.

We defer the proof of Lemma 3.3 to Appendix C.

4 Robust Learning of Bayesian Networks in Nearly-Linear Time

In this section, we prove our main result. We present our algorithm (Algorithm 1) and prove its correctness and analyze its running time (Theorem 4.1).

Theorem 4.1. *Fix $0 < \epsilon < \epsilon_0$ where ϵ_0 is a sufficiently small universal constant. Let P be a c-balanced Bayesian network on $\{0, 1\}^d$ with known structure H. Let α be the minimum parental configuration probability of P. Assume $\alpha = \widetilde{\Omega}(\epsilon^{2/3}c^{-1/3})$.*

Let S be an ϵ-corrupted set of $N = \widetilde{\Omega}(m/\epsilon^2)$ samples drawn from P. Given H, S, ϵ, c, and α, Algorithm 1 outputs a Bayesian network Q in time $\widetilde{O}(Nd)$ such that, with probability at least $9/10$, $d_{\mathrm{TV}}(P, Q) \leq O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c})$.

The dependence on c and α in the error guarantee also appears in prior work (Cheng et al., 2018). Removing this dependence is an important technical question that is beyond the scope of this paper.

Theorem 4.1 follows from three key technical lemmas. At the beginning of Algorithm 1, we first scale all the input vectors as in Definition 3.1. We maintain a guess q for p and gradually move it closer to p. In our analysis, we track our progress by the ℓ_2-norm of $\pi^P \circ (\hat{p} - \hat{q})$.

Initially, we set q to be the empirical conditional mean over S. Lemma 4.2 proves that $\left\|\pi^P \circ (\hat{p} - \hat{q}')\right\|_2$ is not too large for our first guess. Lemma 4.3 shows that, as long as q is still relatively far from p, we can compute a new guess such that $\left\|\pi^P \circ (\hat{p} - \hat{q})\right\|_2$ decreases by a constant factor. Lemma 4.4 states that, when the algorithm terminates and $\left\|\pi^P \circ (\hat{p} - \hat{q})\right\|_2$ is small, we can conclude that the output Q is close to the ground-truth P in total variation distance.

In the following three lemmas, we consider the same setting as in Theorem 4.1 and assume the conditions in Section 2.3 hold.

Lemma 4.2 (Initialization)**.** *In Algorithm 1, we have*

$$\left\|\pi^P \circ (\hat{p} - \hat{q}^0)\right\|_2 \leq O(\epsilon\sqrt{d}/\sqrt{\alpha c}) \ .$$

Lemma 4.3 (Iterative Refinement)**.** *Fix an iteration t in Algorithm 1. Assume the robust mean estimation algorithm $\mathscr{O}_{mean}$ succeeds. If $\left\|\pi^P \circ (\hat{p} - \hat{q}^t)\right\|_2 \leq \rho^t$ and $\rho^t > O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c})$, then we have $\left\|\pi^P \circ (\hat{p} - \hat{q}^{t+1})\right\|_2 \leq c_1\rho^t$ for some universal constant $c_1 < 1$.*

Algorithm 1: Robustly Learning Bayesian Networks

Input : The dependency graph H of a c-balanced Bayesian network P with minimum parental configuration α, an ϵ-corrupted set S of $N = \widetilde{\Omega}(m/\epsilon^2)$ samples $\{X_i\}_{i=1}^N$ drawn from P, and the values of ϵ, c and α.

Output: A Bayesian network Q such that, with probability at least $9/10$, $d_{\mathrm{TV}}(P, Q) \leq O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c})$.

Compute the empirical probabilities π^S where $\pi^S(i, a) = \Pr_{X \in S}[\Pi_{i,a}]$;

Compute the empirical conditional probabilities q^S where $q^S(i, a) = \Pr_{X \in S}[X(i) = 1 \mid \Pi_{i,a}]$;

Compute the scaling vector $s = 1/\sqrt{(\pi^S \circ q^S \circ (1 - q^S))}$;

Let $T = O(\log d)$ and $q^0 = q^S$;

Let $\rho^0 = O(\epsilon\sqrt{d}/\sqrt{\alpha c})$. (We maintain upper bounds ρ^t s.t. $\left\|\pi^P \circ (\hat{p} - \hat{q}^t)\right\|_2 \leq \rho^t$ for all t);

for $t = 0$ **to** $T - 1$ **do**

  $\beta^t = O(\epsilon\rho^t/\alpha)$, $\gamma^t = O((\rho^t)^2/\alpha + \rho^t/\sqrt{\alpha})$;

  Solve a robust mean estimation problem. Let $\nu = \mathscr{O}_{mean}\big(\{\hat{f}(X_i, q^t)_{i \in S}\}, \epsilon, \beta^t, \gamma^t\big)$;

  $q^{t+1} = \nu \circ (1/s) \circ (1/\pi^S) + q^t$; $\rho^{t+1} = c_1\rho^t$;

return *the Bayesian network Q with graph H and conditional probabilities q^T*;

Lemma 4.4. *Let Q be a Bayesian network that has the same structure as P. Suppose that (1) P is c-balanced, (2) $\alpha = \Omega(r + \epsilon/c)$, and (3) $\left\|\pi^P \circ (\hat{p} - \hat{q})\right\|_2 \leq r/2$. Then we have $d_{\mathrm{TV}}(P, Q) \leq r$.*

We defer the proofs of Lemmas 4.2, 4.3, and 4.4 to Appendix D and we first prove Theorem 4.1.

Proof of Theorem 4.1. We first prove the correctness of Algorithm 1.

The original set of $N = \Omega(m \log(m/\epsilon)/\epsilon^2)$ good samples drawn from P satisfies the conditions in Section 2.3 with probability at least $1 - \frac{1}{20}$. With high probability, the robust mean estimation oracle $\mathscr{O}_{mean}$ succeeds in all iterations. For the rest of this proof, we assume the above conditions hold, which by a union bound happens with probability at least $9/10$.

From Lemma 4.2, we have the following condition on the initial estimate q^0.

$$\left\|\pi^P \circ (\hat{p} - \hat{q}^0)\right\|_2 = O(\epsilon\sqrt{d}/\sqrt{\alpha c}) \ .$$

We start with an upper bound ρ^0 of $\left\|\pi^P \circ (\hat{p} - \hat{q}^0)\right\|_2 = O(\epsilon\sqrt{d}/\sqrt{\alpha c})$. By Lemma 4.3, in each iteration round, if $\rho^t > O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c})$, we can obtain a new estimate q^{t+1} and an upper bound ρ^{t+1} on $\left\|\pi^P \circ (\hat{p} - \hat{q}^t)\right\|_2$ such that ρ^{t+1} is smaller than ρ^t by a constant factor. Hence after $O(\log(d))$ iterations, we can get a vector q^t such that

$$\left\|\pi^P \circ (\hat{p} - \hat{q}^t)\right\|_2 = O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c}) \ .$$

Let Q be the Bayesian network with conditional probability table q^t. The assumption that $\alpha = \widetilde{\Omega}(\epsilon^{2/3}c^{-1/3})$ allows us to apply Lemma 4.4 with $r = O(\epsilon\sqrt{\log(1/\epsilon)}/\sqrt{\alpha c})$, which gives the claimed upper bound on $d_{\mathrm{TV}}(P, Q)$.

Now we analyze the runtime of Algorithm 1. First, q^S and π^S can be computed in time $\widetilde{O}(Nd)$ because each sample only affects d entries of q. We do not explicitly write down $f(X, q)$. In each iteration, we solve a robust mean estimation problem with input $\left\{\hat{f}(X_i, q^t)\right\}_{i \in S}$, which takes time $\widetilde{O}(Nd)$. This is because there are N input vectors, each vector is d-sparse, and the robust mean estimation algorithm runs in time nearly-linear in the number of nonzeros in the input (Lemma 3.3). We can compute $q^{t+1} = \nu \circ (1/s) \circ (1/\pi^S) + q^t$ in time in time $O(m)$.

Since there are $O(\log d)$ iterations, the overall running time is

$$\widetilde{O}(Nd) + O(\log d)\left(\widetilde{O}(Nd) + O(m)\right) = \widetilde{O}(Nd) \ . \qquad \square$$

REFERENCES

P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *J. Mach. Learn. Res.*, 7:1743–1788, 2006.

A. Anandkumar, D. J. Hsu, F. Huang, and S. Kakade. Learning mixtures of tree graphical models. In *Proc. 26th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1061–1069, 2012.

S. Balakrishnan, S. S. Du, J. Li, and A. Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proc. 30th Annual Conference on Learning Theory (COLT)*, pp. 169–212, 2017.

G. Bresler. Efficiently learning Ising models on arbitrary graphs. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 771–782, 2015.

G. Bresler, E. Mossel, and A. Sly. Reconstruction of Markov random fields from samples: Some observations and algorithms. *SIAM J. Comput.*, 42(2):563–578, 2013.

G. Bresler, D. Gamarnik, and D. Shah. Structure learning of antiferromagnetic Ising models. In *NIPS*, pp. 2852–2860, 2014.

C. L. Canonne, I. Diakonikolas, D. M. Kane, and A. Stewart. Testing bayesian networks. In *Proc. 30th Annual Conference on Learning Theory (COLT)*, volume 65, pp. 370–448, 2017.

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

M. Chen, C. Gao, and Z. Ren. Robust covariance and scatter matrix estimation under Huber's contamination model. *The Annals of Statistics*, 46(5):1932–1960, 2018.

Y. Cheng, I. Diakonikolas, D. M. Kane, and A. Stewart. Robust learning of fixed-structure Bayesian networks. In *Proc. 32nd Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10304–10316, 2018.

Y. Cheng, I. Diakonikolas, and R. Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2755–2771, 2019a.

Y. Cheng, I. Diakonikolas, R. Ge, and D. P. Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In *Proc. 32nd Annual Conference on Learning Theory (COLT)*, pp. 727–757, 2019b.

Y. Cheng, I. Diakonikolas, R. Ge, and M. Soltanolkotabi. High-dimensional robust mean estimation via gradient descent. In *Proc. 37th International Conference on Machine Learning (ICML)*, 2020.

Y. Cherapanamjeri, S. Mohanty, and M. Yau. List decodable mean estimation in nearly linear time. *CoRR*, abs/2005.09796, 2020.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theor.*, 14(3):462–467, 1968.

S. Dasgupta. The sample complexity of learning fixed-structure Bayesian networks. *Machine Learning*, 29(2-3):165–180, 1997.

J. Depersin and G. Lecué. Robust subgaussian estimation of a mean vector in nearly linear time. *arXiv preprint arXiv:1906.03058*, 2019.

L. Devroye and L. Györfi. *Nonparametric Density Estimation: The L_1 View*. John Wiley & Sons, 1985.

I. Diakonikolas and D. M. Kane. Recent advances in algorithmic high-dimensional robust statistics. *CoRR*, abs/1911.05911, 2019.

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

I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Being robust (in high dimensions) can be practical. In *Proc. 34th International Conference on Machine Learning (ICML)*, pp. 999–1008, 2017a.

I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Proc. 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 73–84, 2017b.

I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robustly learning a Gaussian: Getting optimal error, efficiently. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2683–2702, 2018a.

I. Diakonikolas, D. M. Kane, and A. Stewart. List-decodable robust mean estimation and learning mixtures of spherical Gaussians. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1047–1060, 2018b.

I. Diakonikolas, D. M. Kane, and A. Stewart. Learning geometric concepts with nasty noise. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1061–1073, 2018c.

I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, J. Steinhardt, and A. Stewart. SEVER: A robust meta-algorithm for stochastic optimization. In *Proc. 36th International Conference on Machine Learning (ICML)*, pp. 1596–1606, 2019a.

I. Diakonikolas, W. Kong, and A. Stewart. Efficient algorithms and lower bounds for robust linear regression. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2745–2754, 2019b.

Y. Dong, S. B. Hopkins, and J. Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. In *Proc. 33rd Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6065–6075, 2019.

S. B. Hopkins and J. Li. Mixture models, robustness, and sum of squares proofs. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1021–1034, 2018.

P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, 03 1964.

F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, Incorporated, 2nd edition, 2007.

A. Klivans, P. Kothari, and R. Meka. Efficient algorithms for outlier-robust regression. In *Proc. 31st Annual Conference on Learning Theory (COLT)*, pp. 1420–1430, 2018.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

P. K. Kothari, J. Steinhardt, and D. Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1035–1046, 2018.

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

J. Li and G. Ye. Robust covariance estimation in nearly-matrix multiplication time. *arXiv preprint arXiv:2006.13312*, 2020.

L. Liu, Y. Shen, T. Li, and C. Caramanis. High dimensional robust sparse regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 411–421, 2020.

P. L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. In *NIPS*, pp. 2096–2104, 2012.

A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

N. P. Santhanam and M. J. Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Trans. Information Theory*, 58(7):4117–4134, 2012.

J. Steinhardt, M. Charikar, and G. Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *9th Innovations in Theoretical Computer Science Conference (ITCS)*, pp. 45:1–45:21, 2018.

J. W. Tukey. Mathematics and picturing of data. In *Proceedings of ICM*, volume 6, pp. 523–531, 1975.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

M. J. Wainwright, P. Ravikumar, and J. D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Proc. 20th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1465–1472, 2006.

B. Zhu, J. Jiao, and J. Steinhardt. Robust estimation via generalized quasi-gradients. *arXiv preprint arXiv:2005.14073*, 2020.