Trimmed Maximum Likelihood Estimation for Robust Learning in Generalized Linear Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of learning generalized linear models under adversarial corruptions. We analyze a classical heuristic called the *iterative trimmed maximum likelihood estimator* which is known to be effective against *label corruptions* in practice. Under label corruptions, we prove that this simple estimator achieves minimax near-optimal risk on a wide range of generalized linear models, including Gaussian regression, Poisson regression and Binomial regression. Finally, we extend the estimator to the much more challenging setting of *label and covariate corruptions* and demonstrate its robustness and optimality in that setting as well.

1 Introduction

Generalized linear models (GLMs) are an elegant framework for statistical modeling of data and are widely used in many applications [23, 13, 21]. A generalized linear model captures the relationship of labels (or observations) y to covariates $\mathbf{x}$ via the conditional density $f(y|\beta^\top \mathbf{x}) \propto \exp(y \cdot \beta^\top \mathbf{x} - b(\beta^\top \mathbf{x}))$. Here β is the parameter of the model. Parameter estimation in GLMs is done via the standard maximum likelihood estimation (MLE) paradigm and has been extensively studied [23]. While GLMs offer a mathematically tractable formulation for statistical modeling, real data rarely satisfies the generative process of a GLM and as a result there has been considerable interest in developing robust learning algorithms for GLMs [20, 22, 25]. The predominant way to model misspecification or adversarial corruptions in the data is Huber's ϵ-contamination model [16] and its recent extensions that allow for stronger adversaries [11]. These models assume that a small ϵ fraction of the data is corrupted by an adversary. Furthermore, the adversary can be restricted to either only corrupting the labels y, or be allowed to corrupt both the covariates $\mathbf{x}$ and the labels y for an ϵ fraction of the data.

Various algorithms have been proposed for robust estimation in generalized linear models. One line of work proposed computationally intractable algorithms that are based on non-convex M estimators [16, 20] or by running tournaments over an exponentially large search space [29]. Another line of work proposes polynomial time algorithms that either achieve sub-optimal error rates [25] or only apply to restricted settings such as the noise being heavy tailed [31]. In practical settings a simple heuristic namely the *iterative trimmed estimator* has been shown to work well under settings where only the labels are corrupted [26]. However, from a theoretical perspective the iterative trimmed MLE estimator has only been analyzed under restrictive settings such as when the underlying GLM is a Gaussian regression model.

Our key theoretical contribution in this work is a general analysis of the trimmed MLE estimator. In particular, we show that for a broad family of GLMs, and under adversarial corruptions of only the labels, not only does the iterative trimmed MLE estimator enjoy theoretical guarantees, it in fact achieve the minimax error rate! Next, we also consider the more challenging setting of corruptions to both covariates and labels. In this case, building upon the recent work of [7], we modify the iterative

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

trimmed MLE estimator to propose a polynomial time algorithm based on semi-definite programming that can simultaneously handle both covariate and label corruptions and nearly achieve minimax error rates. Below we state our main results.

Theorem 1.1 (Informal Theorem)**.** *Let $S_\epsilon = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be independent and identically distributed samples generated by a generalized linear model with sub-Gaussian x_i. Let an ϵ fraction of the labels be adversarially corrupted. Then, with high probability, the iterative trimmed MLE estimator when given as input S_ϵ provides the following guarantees:*

- *$O(\sigma\epsilon\log(1/\epsilon))$ parameter estimation error for the Gaussian regression model where σ^2 is the variance of Gaussian noise on y.*
- *$O(\epsilon\exp(\sqrt{\log(1/\epsilon)}))$ parameter estimation error for Poisson regression model.*
- *$O(\frac{1}{\sqrt{m}}\epsilon\sqrt{\log(m/\varepsilon)\log(1/\varepsilon)})$ parameter estimation error for the Binomial regression model with m trials.*
- *$O(\epsilon\log(1/\epsilon))$ parameter estimation error for a general class of smooth and continuous GLMs (includes the Gaussian regression model).*

Previous work [26] analyze the iterative trimmed maximum likelihood estimator under the Gaussian regression settings (least square) where only the labels are corrupted, and proved a $O(\sigma)$ ℓ_2-error bound for parameter estimation. Other iterative trimming approach such as [4] also achieves only achieve $O(\sigma)$ error in this setting. Our error bound of $O(\sigma\varepsilon\log(1/\varepsilon))$ significantly improved the dependency on the corruption level ε and nearly matches the minimax lower bound of $\sigma\varepsilon$ [15]. In all the generalized linear models studied in this work, we show that the iterative trimmed maximum likelihood estimator achieves $O(\varepsilon^{1-\delta})$ error for any $\delta > 0$, which matches the minimax lower bound $\Omega(\varepsilon)$ up to a sub-polynomial factor.

Next, we present our second main result that can simultaneously handle both covariate and label corruptions and nearly achieve minimax error rate.

Theorem 1.2 (Informal Theorem)**.** *Let $S_\epsilon = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be independent and identically distributed samples generated by a generalized linear model with sub-Gaussian x_i. Let an ϵ fraction of the labels and covariates be adversarially corrupted. Then there is a polynomial time algorithm based on semi-definite programming that takes S_ϵ as input and with high probability achieves the same parameter estimation recovery bounds as in Theorem 1.1 above.*

Our algorithm requires the covariance matrix of the covariate distribution being identity or known. Thus the error rate we get is incomparable to the results in the general covariance settings.

Outline of the paper. In Section 2 we discuss related work. We define preliminaries in Section 3 followed by the iterative trimmed MLE algorithm, formal results, and proof sketches for the label corruption case in Section 4. In Section B, we introduce our algorithm, results and proof sketches for the sample corruption case. We defer proofs to the Appendix.

2 Related Work

There is a large body of work both on the theoretical and practical aspects of generalized linear models. Here we discuss the works most relevant to the current setting that we study. Classical works in the Huber's contamination model present algorithms for general robust estimation that obtain near optimal error rates [16, 29]. However these algorithms are computationally inefficient as they requires enumeration over an exponentially large search space. Polynomial time algorithms for robust estimation in GLMs have also been proposed. The recent work of [31] propsoed a reweighted MLE estimator for dealing with settings where the covariates are heavy tailed. The work of [25] proposes a general procedure for robust gradient descent and shows that one can use this to design robust estimation algorithms for GLMs. However the bounds obtained have sub-optimal dependence on the error rate ϵ and also have a mild dimension dependence. In particular, the resulting algorithms cannot handle the case of constant corruption rates. In another line of work, a simple heuristic namely the trimmed MLE estimator has been shown to perform well under corruptions to observation [26]. There also exist theoretical guarantees for this heuristic under the Gaussian regression model in which case MLE reduces to standard linear least squares regression problem [26].

There is also a vast amount of literature in statistics, machine learning and theoretical computer science on algorithms that are robust to covariate corruptions. Minimax optimal but computationally inefficient robust estimators have been established in the works of [28, 29, 6, 15] for a variety of problems such as mean and covariance estimation. In recent years there has also been a line of work in designing computationally efficient algorithms for handling adversarial corruptions [19, 11, 5, 4, 7]. There is also a long line of work on designing robust algorithms for the linear least squares problem. This corresponds to the special case of the GLM being a Gaussian regression model [4, 3, 18, 8]. Another special case of GLMs that has been studied from a robustness perspective is the logistic regression model [14, 25]. Our general analysis of the iterative trimmed MLE estimator matches the best known error guarantees for this setting. The recent work of [10] also studies learning of GLMs under covariate corruptions. Building upon the works of [25, 12] the authors present nearly linear time algorithms that achieve an $O(\sqrt{\epsilon})$ error guarantee under bounded 4th moment assumption on the marginal of x. On the other hand, our main contribution is to show that a simple heuristic achieves minimax optimal rates for a variety of GLMs under observation corruptions, and can be easily modified to handle covariate corruptions as well.

3 Preliminaries

In this section, we formally introduce the robust generalized linear model studied in this paper. First we define the classical generalized linear model as follows.

Definition 3.1 (Generalized linear model)**.** *We say that $(\mathbf{x}, y)$ follows from a generalized linear model if there exists function $b(\cdot)$, function $c(\cdot)$ and scalar ϕ such that the probability density function of y equals*

$$f(y|\beta^\top \mathbf{x}) = c(y)\exp(y \cdot \beta^\top \mathbf{x} - b(\beta^\top \mathbf{x})).$$

The derivative $b'(\cdot)$ is called mean function since $\mathbb{E}[y|\beta^\top \mathbf{x}] = b'(\beta^\top \mathbf{x})$. The second order derivative $b''(\cdot)$ is called variance function since $\mathrm{Var}[y|\beta^\top \mathbf{x}] = b''(\beta^\top \mathbf{x})$

Here we provide three commonly used examples of generalized linear model studied in this paper.

Definition 3.2. *Commonly used examples of generalized liner model*

- ***Gaussian regression****: $b(\theta) = \frac{1}{2}\theta^2$, $c(y) = \frac{\exp(-y^2/2)}{\sqrt{2\pi}}$*
- ***Poisson regression****: $b(\theta) = \exp(\theta)$, $c(y) = \frac{1}{y!}$*
- ***Binomial regression****: Let m be the number of trials of the binomial distribution. $b(\theta) = m\log(1+\exp(\theta))$, $c(y) = \binom{m}{y}$*

We consider sub-Gaussian random design setting in this work, i.e., each covariate $\mathbf{x}_i$ is drawn i.i.d. from a sub-Gaussian distribution with zero mean and covariance Σ.

Definition 3.3 (Sub-Gaussian design)**.** *We assume that each covariate $\mathbf{x}_i$ is drawn independently from a zero mean, covariance Σ sub-Gaussian distribution with sub-Guassian norm 1, namely, $\mathbb{E}[\mathbf{x}_i] = 0$, $\mathbb{E}[\mathbf{x}_i\mathbf{x}_i^\top] = \Sigma$ and for all $\mathbf{v} \in \mathbb{R}^d$*

$$\Pr(\mathbf{v}^\top \mathbf{x}_i \geq t) \leq \exp(-t^2)$$

Having introduced the generation model of the good data, now we describe the corruption model where the adversary is allowed to corrupt a small fraction of the data points.

Definition 3.4 (Corruption model)**.** *We consider two different data generation model with ε fraction of adversarial corruptions:*

- ***Label corruption model****: Given n i.i.d. sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ generated by a generalized linear model. The adversary is allowed to inspect the sample, and replace a total of $\varepsilon \cdot n$ labels y_i with arbitrary values.*
- ***Sample corruption model:*** *Given n i.i.d. sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ generated by a generalized linear model. The adversary is allowed to inspect the sample, and replace a total of $\varepsilon \cdot n$ data points $(\mathbf{x}_i, y_i)$ with arbitrary values.*

We call the corrupted dataset $S = T \cup E$ where T contains the set of remaining uncorrupted data points, and E contains the set of data points that is controlled by the adversary.

The goal of our algorithm is recovering the underlying regression coefficient β^* from a set of corrupted examples under ℓ_2 error metric, i.e., $\|\hat{\beta} - \beta^*\|$. We assume $\|\beta^*\| \leq R$ for a constant R. For simplicity of the presentation, though out this paper, we assume $\Sigma = I_d$. We remark that our algorithm for *label corruption model* applies to the general covariance setting and is able to achieve small estimation error in terms of $\|\Sigma^{1/2}(\hat{\beta} - \beta^*)\|_2$ while the algorithm for *sample corruption model* only works with the knowledge of Σ.

4 Label Corruption

In this section, we formally describe our algorithm and proof-sketch for the label corruption setting.

4.1 Algorithm

We start with defining the trimmed maximum likelihood estimator, that will be used in our algorithm.

Definition 4.1 (Trimmed maximum likelihood estimator)**.** *Given a set of data points $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, define the trimmed maximum likelihood estimator as*

$$\hat{\beta}(S) = \min_{\beta} \min_{\hat{S} \subset S, |\hat{S}| = (1-\varepsilon)n} \sum_{(\mathbf{x}_i, y_i) \in \hat{S}} -\log f(y_i | \beta^\top \mathbf{x}_i)$$

In the setting of generalized linear model, the objective of trimmed maximum likelihood estimator is a biconvex problem in $\hat{S}$ and β but not jointly convex. The following alternating minimization algorithm is a simple heuristic to approximate the trimmed maximum likelihood estimator whose similar form has been studied in [26, 7].

Algorithm 1: Alternating minimization of trimmed maximum likelihood estimator

Input: Set of examples $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, ε, η, R

Output: $\hat{\beta}$

$S^{(0)} \leftarrow \arg\min_{T \subset [n]: |T| = (1-\varepsilon)n} \sum_{i \in T} |y_i|$;

$\hat{\beta}^{(1)} \leftarrow 0$;

for $t = 1$ **to** ∞ *do* **do**

  Choose $\hat{S}^{(t)} = \arg\min_{T \subset S^{(0)}: |T| = (1-2\varepsilon)n} \sum_{i \in T} -\log f(y_i | \langle \hat{\beta}^{(t)}, \mathbf{x}_i \rangle)$;

  Compute $\hat{\beta}^{(t+1)} = \arg\min_{\beta, \|\beta\| \leq R} \sum_{i \in \hat{S}^{(t)}} -\log f(y_i | \langle \beta, \mathbf{x}_i \rangle)$;

  if $\frac{1}{n} \sum_{i \in \hat{S}^{(t)}} -\log f(y_i | \langle \hat{\beta}^{(t+1)}, \mathbf{x}_i \rangle) > \frac{1}{n} \sum_{i \in \hat{S}^{(t)}} -\log f(y_i | \langle \hat{\beta}^{(t)}, \mathbf{x}_i \rangle) - \eta$ **then**

    Return $\beta^{(t)}$

The algorithm starts by naively pruning out $\varepsilon \cdot n$ data points with the largest labels. Then each round of the alternating minimization algorithm has two steps. In optimizing over S, we find the set $\hat{S}^{(t)}$ of size $(1-\varepsilon)n$ with the best likelihood. In optimizing over β, we find regression coefficient $\beta^{(t)}$ which maximize the likelihood on the current set of data $\hat{S}^{(t)}$. The algorithm terminate and output β when the likelihood no longer improves by more than η. It is clear that the algorithm terminate in $O(1/\eta)$ rounds when the log-likelihood is bounded. Since the original trimmed maximum likelihood estimator is a biconvex optimization problem in S, β which is not jointly convex, our algorithm does not guarantee to return a global optimal solution. Nonetheless, as we showed, it does return a first order stationary point which will be close to the true coefficient β^*.

We present the guarantee of our algorithm for Gaussian, Poisson and Binomial regression. The result for other GLM is deferred to Appendix A.4.

Theorem 4.2 (Gaussian regression with label corruption)**.** *Let $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be a set of data points generated by a Gaussian regression model with sub-Gaussian design, with ε_c-fraction of*

label corruption and $n = \Omega(\frac{d+\log(1/\delta)}{\varepsilon^2})$. *With probability* $1-\delta$, *Algorithm 1 with parameters* $\varepsilon = \varepsilon_c, \eta = \varepsilon_c^2, R = \infty$ *terminate within* $O(1/\varepsilon_c^2)$ *iterations, and output an estimate* $\hat{\beta}$ *such that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon \log(1/\varepsilon))$$

Theorem 4.3 (Poisson regression with label corruption)**.** *Let* $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ *be a set of data points generated by a Poisson regression model with sub-Gaussian design, with* ε_c*-fraction of label corruption and* $n = \Omega(\frac{d}{\varepsilon^2})$. *With probability* 0.99, *Algorithm 1 with parameters* $\varepsilon = 2\varepsilon_c, \eta = \varepsilon_c^2/(dn), R \geq \|\beta^*\|$ *terminate within* dn/ε_c^2 *iterations, and output an estimate* $\hat{\beta}$ *such that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon \exp(\sqrt{\log(1/\varepsilon)}))$$

Theorem 4.4 (Binomial regression with label corruption)**.** *Let* $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ *be generated by a Binomial regression model with Sub-Gaussian Design, with* ε_c*-fraction of label corruption and* $n = \Omega(\frac{d+\log(1/\delta)}{\varepsilon^2})$. *With probability* $1-\delta$, *Algorithm 1 with parameters* $\varepsilon = \varepsilon_c, \eta = \varepsilon_c^2, R \geq \|\beta^*\|$ *terminate within* m/ε_c^2 *iterations, and output an estimate* $\hat{\beta}$ *such that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon\sqrt{\frac{\log(m/\varepsilon)\log(1/\varepsilon)}{m}})$$

4.2 Proof Sketch

We provide an intuitive proof sketch for the above theorems (the full proofs are deferred to Section A in the Appendix)

The guarantee of our alternating minimization algorithm relies on two claims: First, the algorithm returns an approximate stationary point $\hat{\beta}$. Second, any approximate stationary point will be close to the true coefficient β^*. In this section, we will present high level intuition of the proof of the two claims.

Alternating minimization algorithm returns an approximate stationary point. First we define first order approximate stationary point as follows. Let $\hat{\beta} \in \mathbb{R}^d$ be a regression coefficient vector and $\hat{S}$ contains the set of datapoints of size $(1-\varepsilon)n$ with the largest log-likelihood under $\hat{\beta}$. We call $\hat{\beta}$ a γ-approximate stationary point if

$$\frac{1}{n}\sum_{i\in\hat{S}} \nabla_\beta \log f(y_i|\langle\hat{\beta}, \mathbf{x}_i\rangle)^\top \frac{(\beta^* - \hat{\beta})}{\|\beta^* - \hat{\beta}\|} \leq \gamma$$

i.e., the gradient of the log-likelihood projected along the $\beta^* - \hat{\beta}$ direction is small. Our goal is to show when the algorithm terminate, that is when $\hat{\beta}$ can not be improved by more than η, the gradient along the $\beta^* - \hat{\beta}$ must be small. This is clear where the empirical log-likelihood function is smooth, simply because if the gradient is large, one can improve the log-likelihood by more than η which will result in a contradiction. The smoothness (norm of the Hessian matrix) of the empirical log-likelihood in generalized linear model is directly related to the range of $b''(\theta)$. In particular, $b''(\theta)$ is bounded for Gaussian regression and Binomial regression.

However, the problem arise for Poisson regression where $b''(\theta) = \exp(\theta)$ becomes extremely large for large θ, which results in non-smooth curvature for empirical log-likelihood. We overcome this difficulty by leveraging the special property of function $b(\theta) = \exp(\theta)$ in the Poisson regression setting. Observing that the derivative $b'(\theta)$ is equal to second order derivative $b''(\theta)$ for Poisson regression, the gradient along the $\beta^* - \hat{\beta}$ direction can not be small when the second order derivative along the $\beta^* - \hat{\beta}$ direction gets large, which will result in a more than η improvement of the log-likelihood by moving toward β^* and a contradiction. Therefore, the second order derivative along the $\beta^* - \hat{\beta}$ direction is small, and we are back to the smooth objective function setting.

Any approximate stationary point will be close to the true coefficient. Let us first write down the γ-approximate stationary condition for generalized linear model as

$$\frac{1}{n}\sum_{i\in\hat{S}} \nabla_\beta \log f(y_i|\langle\hat{\beta}, \mathbf{x}_i\rangle)^\top(\beta^* - \hat{\beta}) = \frac{1}{n}\sum_{i\in\hat{S}}(y_i - b'(\hat{\beta}^\top\mathbf{x}_i))(\beta^* - \hat{\beta})^\top\mathbf{x}_i \leq \gamma\|\beta^* - \hat{\beta}\|$$

Recall that T contains the set of uncorrupted data points, and E contains the set of data points that is controlled by the adversary. Split $\hat{S}$ into $\hat{S} \cap T$ and $\hat{S} \cap E$, and rearrange the terms we get

$$\frac{1}{n} \sum_{i \in \hat{S} \cap T} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))(\beta^* - \hat{\beta})^\top \mathbf{x}_i \leq -\frac{1}{n} \sum_{i \in \hat{S} \cap E} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))(\beta^* - \hat{\beta})^\top \mathbf{x}_i + \gamma \|\beta^* - \hat{\beta}\|.$$

To obtain an upper bound on $\|\beta^* - \hat{\beta}\|$, we will prove a lower bound in terms of $\|\beta^* - \hat{\beta}\|$ on the left hand side, and an upper bound terms of $\|\beta^* - \hat{\beta}\|$ on the right hand side. Finally we will combine the upper and lower bound into an upper bound on $\|\beta^* - \hat{\beta}\|$.

Lower bound on the LHS. Note that $\hat{S} \cap T$ contains uncorrupted data points. The high level intuition is that since mean of y_i is $b'({\beta^*}^\top \mathbf{x}_i)$ and $b'(\cdot)$ is monotone, $\left(y_i - b'({\beta^*}^\top \mathbf{x}_i)\right)(\beta^* - \hat{\beta})^\top \mathbf{x}_i$ should be roughly $\left((\beta^* - \hat{\beta})^\top \mathbf{x}_i\right)^2$, and $\left((\beta^* - \hat{\beta})^\top \mathbf{x}_i\right)^2$ should be proportional to $\|\beta^* - \hat{\beta}\|^2$ given enough samples. More formally, we will decompose the LHS as

$$\begin{aligned}
&\frac{1}{n} \sum_{i \in \hat{S} \cap T} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))(\beta^* - \hat{\beta})\mathbf{x}_i \\
=&\frac{1}{n} \sum_{i \in \hat{S} \cap T} \left(y_i - b'({\beta^*}^\top \mathbf{x}_i)\right)(\beta^* - \hat{\beta})^\top \mathbf{x}_i + \frac{1}{n} \sum_{i \in \hat{S} \cap T} \left(b'({\beta^*}^\top \mathbf{x}_i) - b'(\hat{\beta}^\top \mathbf{x}_i)\right)(\beta^* - \hat{\beta})^\top \mathbf{x}_i.
\end{aligned}$$

The first term contains a $(1 - \varepsilon)$ fraction of uncorrupted examples sampled from a zero mean distribution with certain tail bound, e.g. sub-exponential for Gaussian and Binomial regression, k-th momment bound for Poisson regression.

Therefore, we can apply *resilience* property to bound the first term. Over this work, we heavily utilize the *resilience* property of the sample set that is drawn from "nice" distributions. Take sample mean as an example, *resilience* [27, 30] dictates that given a large enough sample set $S = \{\mathbf{x}_i\}_{i=1}^n$, the sample mean of any large enough subset of S will be close to each other. Specifically, under sub-Gaussian distribution, for any $T \subset S, |T| \geq (1 - \varepsilon)n$, it holds that $\|\frac{1}{|T|} \sum_{i \in T} \mathbf{x}_i - \frac{1}{|S|} \sum_{i \in S} \mathbf{x}_i\| \leq \varepsilon \sqrt{\log(1/\varepsilon)}$. Resilience property applies to sub-exponential and k-th moment bounded distribution as well, and this gives us a way to control the behavior of any subset of good data.

For the second term, we prove that $\frac{1}{n} \sum_{i \in \hat{S} \cap T} b(\beta^\top \mathbf{x}_i)$ is a strongly convex function again using the resilience property, which implies $\frac{1}{n} \sum_{i \in \hat{S} \cap T} \left(b'({\beta^*}^\top \mathbf{x}_i) - b'(\hat{\beta}^\top \mathbf{x}_i)\right)(\beta^* - \hat{\beta})^\top \mathbf{x}_i = \Omega(\|\beta^* - \hat{\beta}\|^2)$

Upper bound on the RHS. To upper bound $-\frac{1}{n} \sum_{i \in \hat{S} \cap E} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))(\beta^* - \hat{\beta})^\top \mathbf{x}_i$, we will prove an upper bound on $\sqrt{\frac{1}{n} \sum_{i \in \hat{S} \cap E} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))^2}$ and $\sqrt{\frac{1}{n} \sum_{i \in \hat{S} \cap E} ((\beta^* - \hat{\beta})^\top \mathbf{x}_i)^2}$, then apply Cauchy-Schwarz inequality. The key difficulty is bounding $\sqrt{\frac{1}{n} \sum_{i \in \hat{S} \cap E} (y_i - b'(\hat{\beta}^\top \mathbf{x}_i))^2}$, as it contains corrupted data points controlled by an adversary, which does not follow any good property possessed by the good stochastic data. However, since $\hat{S}$ contains $(1 - \varepsilon)n$ datapoints with the largest log-likelihood under $\hat{\beta}$, we can argue that

$$\sum_{i \in \hat{S} \cap E} -\log f(y_i | \langle \hat{\beta}, \mathbf{x}_i \rangle) \leq \sum_{i \in T \setminus \hat{S}} -\log f(y_i | \langle \hat{\beta}, \mathbf{x}_i \rangle)$$

since otherwise one can replace the data point in $\hat{S} \cap E$ by the ones in $T \setminus \hat{S}$ to form a new set with better likelihood than $\hat{S}$. This gives us an upper bound on the negative log-likelihood of y_i in $\hat{S} \cap E$. Therefore we adopt a two step approach to upper bound the $\sqrt{\frac{1}{n} \sum_{i \in \hat{S} \cap E} ((\beta^* - \hat{\beta})^\top \mathbf{x}_i)^2}$. First we prove an upper bound on the negative log-likelihood on $T \setminus \hat{S}$, which becomes a negative log-likelihood bound on $\hat{S} \cap E$ immediately. Second we turn the negative log-likelihood bound intro a square error bound.

The two steps varies drastically for different regression models. For the first step of upper bounding the negative log-likelihood, in Gaussian regression we use the resilience property of the quadratic

form of sub-Gaussian random variable. In Poisson regression, we leverage the resilience property of $y_i\mathbf{x}_i$. In Binomial regression, since the distribution only has support size m, we directly analyze the resilience of the negative log-likelihood. For the second step of turning the log-likelihood bound to a quadratic bound, we get the bound trivially in Gaussian regression since Gaussian likelihood is indeed quadratic. For Poisson and Binomial setting, we have to build a proxy function which lower bound the negative log-likelihood function $-\log f(y_i|\langle\hat{\beta}, \mathbf{x}_i\rangle)$ to connect it to quadratic function.

4.3 Proof for Poisson Regression

As an illustrative example, we show how the above proof sketch can be used for Poisson regression to formally prove Theorem 4.3. (Proofs for the other GLMs are deferred to Section A in the Appendix)

Lemma 4.5 (Approximate stationary point close to β^* for Poisson regression)**.** *Given as set of datapoints $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ generated by a Poisson regression model with ε-fraction of label corruption, and the largest εn labels removed. Let $\hat{\beta}$ be a $\max(\varepsilon, \varepsilon^2/\|\beta^* - \hat{\beta}\|)$-stationary point and $\|\hat{\beta}\| \leq R$. Given that $n = \Omega(\frac{d}{\varepsilon^2})$, with probability 0.99, it holds that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon \exp(\Theta(\sqrt{\log(1/\varepsilon)})))$$

Since $b''(\theta) = \exp(\theta)$ is unbounded for Poisson regression, the following lemma (proved in the appendix) shows that alternating minimization algorithm still return an approximate stationary point

Lemma 4.6 (Algorithm 1 finds an approximate stationary point for Poisson regression)**.** *Given as set of datapoints $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ generated by a Poisson model with ε_c-fraction of corruption. Assuming that $n = \Omega(\frac{d+\log(1/\delta)}{\varepsilon^2})$, then with probability $1-\delta$, the output of Algorithm 1 with input parameters $\varepsilon = 2\varepsilon_c, R \geq \|\beta^*\|, \eta = \varepsilon^2/(dn)$, is a $\max(\varepsilon, \frac{2\varepsilon^2}{\|\beta^* - \hat{\beta}\|})$-approximate stationary point.*

Proof of Theorem 4.3. Lemma 4.6 implies the output of Algorithm 1 is a $\max(\varepsilon, \frac{\varepsilon^2}{\|\beta^* - \hat{\beta}\|})$ approximate stationary point. Lemma 4.5 then implies that $\|\hat{\beta} - \beta^*\| = O(\varepsilon \exp(\sqrt{\log(1/\varepsilon)}))$. To bound the number of iterations, we need an upper bound on the negative log-likelihood on $\beta = 0$, and a uniform lower bound on the negative log-likelihood. The initial negative log-likelihood is upper bounded by $\frac{1}{n}\sum_{i\in\hat{S}^{(1)}} \log(y_i!) + 1 \leq O(\mathbb{E}[y_i^2] + 1) = O(1)$ where $S^{(1)}$ contains the smallest $(1-\varepsilon)n$ labels. Trivially, there is a 0 lower bound on the negative log-likelihood for Poisson distribution. Therefore, the algorithm will terminate in dn/ε_c^2 iterations. □

5 Result for Sample Corruption Model

The learning problem becomes much harder in the presence of label and covariate corruption, since it becomes much harder to tell whether a data point is corrupted by simply looking at the likelihood of y_i. From a technical level, the resilience condition we leveraged on covariate $\mathbf{x}_i$ in set E breaks down when there is covariate corruption. Luckily, we are able to restore the resilience property by adding a positive semidefinite (PSD) constraint when choosing the set $\hat{S}^{(t)}$. Specifically, if the covariate distribution has identity (or known) covariance and sub-Gaussian tail, one can add the PSD constraint $\frac{1}{p}\sum w_i\mathbf{x}_i\mathbf{x}_i^\top \preceq (1 + \varepsilon\log(1/\varepsilon)) \cdot I$ to the ε corrupted data set, and the resulting weighted data set $\{w_i\mathbf{x}_i\}_{i=1}^n$ will have the same resilient condition as an uncorrupted data set. Similar approaches have been applied be applied before in robust mean estimation [9] and robust regression [7].

To adapt our alternating minimization algorithm to the sample corruption model, we replace the step of choosing the best fit subset $\hat{S}^{(t)}$ by a SDP program, which intuitively tries to find the set with the best likelihood and has bounded covariance at the same time. This is a covering/packing SDPs which can be solved in nearly linear time [24, 2].

5.1 Algorithm

The guarantee of Algorithm 2 is formalized in the following theorems. Our approach can be extends to other generalized linear models, as we showed in Appendix B.

Algorithm 2: Alternating minimization for trimmed maximum likelihood estimator in sample corruption model

Input: Set of examples $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}, \varepsilon, \eta, R$

Output: $\hat{\beta}$

$\hat{\beta}^{(1)} \leftarrow 0$;

$S^{(0)} \leftarrow \arg\min_{T \subset [n]: |T| = (1-\varepsilon)n} \sum_{i \in T} |y_i|$;

for $t = 1$ **to** ∞ *do* **do**

Let $\hat{\mathbf{w}}^{(t)}$ be the solution of the following SDP

$$\begin{aligned} \arg\min_{\mathbf{w}} \sum_{i=1}^{n} & -w_i \log f(y_i | \langle \hat{\beta}^{(t)}, \mathbf{x}_i \rangle) \\ \text{s.t. } & 0 \le w_i \le 1 \ \ \forall i \in S^{(0)} \\ & w_i = 0 \ \ \forall i \notin S^{(0)} \\ & \|\mathbf{w}\|_1 = (1-2\varepsilon)n \\ & \frac{1}{n} \sum_{i \in [n]} w_i \mathbf{x}_i \mathbf{x}_i^\top \preceq (1 + \varepsilon \log(1/\varepsilon)) \cdot I \end{aligned}$$

Compute $\hat{\beta}^{(t+1)} = \arg\min_{\beta, \|\beta\| \le R} \sum_{i=1}^{n} -\hat{w}_i^{(t)} \log f(y_i | \langle \beta, \mathbf{x}_i \rangle)$;

if $\sum_{i=1}^{n} -\hat{w}_i^{(t)} \log f(y_i | \langle \hat{\beta}^{(t+1)}, \mathbf{x}_i \rangle) > \sum_{i=1}^{n} -\hat{w}_i^{(t)} \log f(y_i | \langle \hat{\beta}^{(t)}, \mathbf{x}_i \rangle) - \eta$ **then**

Return $\beta^{(t)}$

Theorem 5.1 (Gaussian regression with sample corruption)**.** *Given as set of datapoints $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ generated by a Gaussian regression model with sub-Gaussian design, ε_c-fraction of sample corruption and $n = \Omega(\frac{d + \log(1/\delta)}{\varepsilon^2})$. With probability $1 - \delta$, Algorithm 2 with parameters $\varepsilon = \varepsilon_c, \eta = \varepsilon_c^2, R = \infty$ terminate within $O(1/\varepsilon_c^2)$ iterations, and output an estimate $\hat{\beta}$ such that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon \log(1/\varepsilon))$$

Theorem 5.2 (Binomial regression with sample corruption)**.** *Given as set of datapoints $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ generated by a Binomial regression model with ε_c-fraction of label corruption and $n = \Omega(\frac{d + \log(1/\delta)}{\varepsilon^2})$. With probability $1 - \delta$, Algorithm 2 with parameters $\varepsilon = \varepsilon_c, \eta = \varepsilon_c^2, R \ge \|\beta^*\|$ terminate within m/ε_c^2 iterations, and output an estimate $\hat{\beta}$ such that*

$$\|\hat{\beta} - \beta^*\| = O(\varepsilon \sqrt{\frac{\log(m/\varepsilon) \log(1/\varepsilon)}{m}})$$

5.2 Proof Sketch

The proof is very similar to the label corruption setting, except that instead of applying resilience condition to subset of T, we need the same condition on the weighted subset of T. For example replacing $\frac{1}{|\hat{S} \cap T|} \sum_{i \in \hat{S} \cap T} \mathbf{x}_i$ by $\frac{1}{\sum_{i \in T} w_i} \sum_{i \in T} w_i \mathbf{x}_i$. Luckily, all the resilience condition applies to the weighted subset as well.

Compared to the label corruption setting, specifically there are two places we need to change in the argument.

1. Since $\mathbf{x}_i, i \in E$ is now controlled by the adversary, we can no longer bound $\sqrt{\frac{1}{n} \sum_{i \in \hat{S} \cap E} ((\beta^* - \hat{\beta})^\top \mathbf{x}_i)^2}$ by the resilience property of sub-Gaussian samples. Instead, we leverage the fact that corrupted sample with small covariance is also resilient.
2. Previously, T is trivially a feasible solution in the step of choosing $\hat{S}$. Now we need to argue additionally that $\frac{1}{n} \sum_{i \in T} \mathbf{x}_i \mathbf{x}_i^\top \preceq (1 + \varepsilon \log(1/\varepsilon)) \cdot I$

6 Conclusion

In this paper, we provided a general theoretical analysis showing that a simple and practical heuristic namely the iterative trimmed MLE estimator achieves minimax optimal error rates under label corruptions for a wide class of generalized linear models (GLMs). We also designed algorithms based on semi-definite programming (SDP) to make the estimator robust to covariate corruptions as well. While our SDP based algorithm runs in polynomial time, it would be interesting to explore more practical variants that can handle covariate corruptions. It would also be interesting to study whether our techniques can be extended to design robust algorithms for more general exponential families beyond GLMs.

References

[1] Thomas D Ahle. Sharp and simple bounds for the raw moments of the binomial and poisson distributions. *Statistics & Probability Letters*, 182:109306, 2022.

[2] Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive sdp solver. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1824–1831. SIAM, 2016.

[3] Kush Bhatia, Prateek Jain, Parameswaran Kamalaruban, and Purushottam Kar. Consistent robust regression. *Advances in Neural Information Processing Systems*, 30, 2017.

[4] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. *Advances in neural information processing systems*, 28, 2015.

[5] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60, 2017.

[6] Mengjie Chen, Chao Gao, and Zhao Ren. Robust covariance matrix estimation via matrix depth. *arXiv preprint arXiv:1506.00691*, 2015.

[7] Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Online and distribution-free robustness: Regression and contextual bandits with huber contamination. *arXiv preprint arXiv:2010.04157*, 2020.

[8] Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In *International Conference on Machine Learning*, pages 774–782. PMLR, 2013.

[9] Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the thirtieth annual ACM-SIAM symposium on discrete algorithms*, pages 2755–2771. SIAM, 2019.

[10] Yeshwanth Cherapanamjeri, Efe Aras, Nilesh Tripuraneni, Michael I Jordan, Nicolas Flammarion, and Peter L Bartlett. Optimal robust linear regression in nearly linear time. *arXiv preprint arXiv:2007.08137*, 2020.

[11] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

[12] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR, 2019.

[13] Annette J Dobson and Adrian G Barnett. *An introduction to generalized linear models*. Chapman and Hall/CRC, 2018.

[14] Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. Robust logistic regression and classification. *Advances in neural information processing systems*, 27, 2014.

[15] Chao Gao. Robust regression via mutivariate regression depth. *Bernoulli*, 26(2):1139–1170, 2020.

[16] Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

[17] Arun Jambulapati, Jerry Li, and Kevin Tian. Robust sub-gaussian principal component analysis and width-independent schatten packing. *Advances in Neural Information Processing Systems*, 33:15689–15701, 2020.

[18] Adam Klivans, Pravesh K Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*, pages 1420–1430. PMLR, 2018.

[19] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

[20] Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in Neural Information Processing Systems*, 24, 2011.

[21] Peter McCullagh and John A Nelder. *Generalized linear models*. Routledge, 2019.

[22] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical science*, 27(4):538–557, 2012.

[23] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[24] Richard Peng and Kanat Tangwongsan. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. In *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pages 101–108, 2012.

[25] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[26] Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR, 2019.

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

[28] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

[29] Yannis G Yatracos. Rates of convergence of minimum distance estimators and kolmogorov's entropy. *The Annals of Statistics*, 13(2):768–774, 1985.

[30] Banghua Zhu, Jiantao Jiao, and Jacob Steinhardt. Generalized resilience and robust statistics. *arXiv preprint arXiv:1909.08755*, 2019.

[31] Ziwei Zhu and Wenjing Zhou. Taming heavy-tailed features by shrinkage. In *International Conference on Artificial Intelligence and Statistics*, pages 3268–3276. PMLR, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] (See Section 6)

(c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]