Sever: A Robust Meta-Algorithm for Stochastic Optimization

Ilias Diakonikolas¹, Gautam Kamath², Daniel M. Kane³, Jerry Li⁴, Jacob Steinhardt⁵, Alistair Stewart¹

(alphabetical order)

¹USC  ²Waterloo  ³UCSD  ⁴MSR AI  ⁵Berkeley
Main question: can you learn a good classifier from poisoned training data?
DEFENDING AGAINST DATA POISONING

Main question: can you learn a good classifier from poisoned training data?

Given a labeled training set, where an (unknown) $\epsilon$-fraction of them are adversarially corrupted, can we learn a model which achieves good accuracy on a clean test set?
DEFENDING AGAINST DATA POISONING

Main question: can you learn a good classifier from poisoned training data?

Example: Training an SVM with 3% poisoned data
DEFENDING AGAINST DATA POISONING

Main question: can you learn a good classifier from poisoned training data?

Example: Training an SVM with 3% poisoned data

Against known defenses, the test error can go up to 30%!

[Koh-Steinhardt-Liang ’18]
DEFENDING AGAINST DATA POISONING

Main question: can you learn a good classifier from poisoned training data?

Example: Training an SVM with 3% poisoned data

Lots of work on related problems:


Against known defenses, the test error can go up to 30%!
OUR RESULTS

We present a framework for robust stochastic optimization

- **Strong theoretical guarantees** against strong adversarial models
- **Outperforms benchmark defenses** on state-of-the-art data poisoning attacks
- Works well in **high dimensions**
- Works with **black-box access** to any learner for any stochastic optimization task
Idea: Until termination:
1. train black box learner to find approximate minima of empirical risk on corrupted training set,
2. then run outlier detection method on the gradients of the loss functions at ERM to remove suspected outliers
**SEVER**

**Idea:** Until termination:
1. train black box learner to find approximate minima of empirical risk on corrupted training set,
2. then run outlier detection method on the gradients of the loss functions at ERM to remove suspected outliers
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?

We exploit a novel connection to robust mean estimation.
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?

We exploit a novel connection to robust mean estimation

Filtering [DKKLMS16, DKKLMS17]: Given a set of points $X_1, \ldots, X_n$ drawn from a “nice” distribution, but where an $\varepsilon$-fraction are corrupted, there is a linear time algorithm which either:
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?

We exploit a novel connection to robust mean estimation

Filtering [DKKLMS16, DKKLMS17]: Given a set of points $X_1, \ldots, X_n$ drawn from a “nice” distribution, but where an $\varepsilon$-fraction are corrupted, there is a linear time algorithm which either:

1. Certifies that the true mean is close to the empirical mean of the corrupted dataset
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?

We exploit a novel connection to **robust mean estimation**

**Filtering \([\text{DKKLMS16, DKKLMS17}]\):** Given a set of points \(X_1, \ldots, X_n\) drawn from a “nice” distribution, but where an \(\varepsilon\)-fraction are corrupted, there is a linear time algorithm which either:

1. Certifies that the true mean is close to the empirical mean of the corrupted dataset
2. Removes more bad points than good points
FILTERING AND ROBUST MEAN ESTIMATION

How should we detect outliers from the gradients?

We exploit a novel connection to robust mean estimation

Filtering [DKKLMS16, DKKLMS17]: Given a set of points $X_1, ..., X_n$ drawn from a “nice” distribution, but where an $\varepsilon$-fraction are corrupted, there is a linear time algorithm which either:

1. Certifies that the true gradient of the loss function is close to 0
2. Removes more bad points than good points
GUARANTEES

Theorem (informal): Suppose we have a distribution \( \mathcal{D} \) over convex functions \( f \), and \( \text{Cov} [\nabla f (\theta)] \ll \sigma^2 I \). Suppose we have \( f_1 (\theta), f_2 (\theta), ..., f_n (\theta) \) drawn from \( \mathcal{D} \), where \( \varepsilon \)-fraction of them are adversarial. Under mild assumptions on \( \mathcal{D} \), then given enough samples, SEVER outputs a \( \hat{\theta} \) so that w.h.p.

\[
\bar{f}(\hat{\theta}) - \min_{\theta} f (\theta) < O \left( \sqrt{\sigma^2 \varepsilon} \right).
\]

Can also give results for non-convex objectives

Sample complexity / runtime are polynomial but not super tight

For GLMs (e.g. SVM, regression), we obtain tight(er) bounds
EMPIRICAL EVALUATION: REGRESSION
EMPIRICAL EVALUATION: SVM

SVM: Strongest attacks against `gradientCentered` on Enron

SVM: Strongest attacks against `loss` on Enron

SVM: Strongest attacks against `SEVER` on Enron

Test Error

Outlier Fraction epsilon
CONCLUSIONS

Main question: can you learn a good classifier from poisoned data?

Sever is a **meta-algorithm** for robust stochastic optimization based on connections to robust mean estimation.

Interested? See poster #143 this evening!