Training Set Debugging Using Trusted Items

Xuezhou Zhang

University of Wisconsin-Madison
Joint work with

Jerry Zhu

Steve Wright
Why debug the training data?

- Real-world data is messy!
- Mislabeling from Mechanical Turkers
- Historical biases
- Data poisoning attack
Why debug the training data?

- Real-world data is messy!
Why debug the training data?

- Real-world data is messy!
- Mislabeling from
  - Mechanical Turkers
  - historical biases
  - data poisoning attack
  - ...

What can we do?

▶ If you have a training set with "wrong" labels, we can automatically find them.

▶ Q: Is it just outlier detection?

▶ A: No! Incorrect labels can come in the form of systematic bias.
If you have a training set with "wrong" labels, we can automatically find them.
What can we do?

- If you have a training set with "wrong" labels, we can automatically find them.
- Q: Is it just outlier detection?
What can we do?

- If you have a training set with "wrong" labels, we can automatically find them.
- Q: Is it just outlier detection?
- A: No! Incorrect labels can come in the form of systematic bias.
Hogwarts Alumni

![Diagram showing two points on a graph. One point represents Hermione with a muggle-born education and the other point represents Gregory with a pure-blood education.](image)
Hired by the Ministry of Magic?

- yes
- no
Data contain historical biases

- Learned vs. ideal decision boundary

(RBF kernel logistic regression)
Data contain historical biases

- Impossible to detect without additional information!
- What can we do?
Trusted items

- obtained by expensive vetting
- insufficient to learn from
Debugging Using Trusted Items (DUTI)

- propose training label bugs
- flipping them makes re-trained model agree with trusted items
This is not our goal

Just to learn a better model (Robust Statistics):

\[
\min_{\theta \in \Theta} \ell(X, Y, \theta) + \lambda \|\theta\|
\]

s.t. \( \theta(\tilde{X}) = \tilde{Y} \)

or equivalently, with some \( \gamma >> 1 \),

\[
\min_{\theta \in \Theta} \ell(X, Y, \theta) + \gamma \ell(\tilde{X}, \tilde{Y}, \theta) + \lambda \|\theta\|
\]
This is our goal

To identify bugs and fix them (and learn a better model):

\[
\min_{Y', \hat{\theta}} \| Y - Y' \|
\]

s.t. \( \theta(\tilde{X}) = \tilde{Y} \)

\( \hat{\theta} = \arg\min_{\theta \in \Theta} \ell(X, Y', \theta) + \lambda \| \theta \| \)
Input / output to our debugger

Input:
1. dirty training set \((X, Y)\)
2. trusted items \((\tilde{X}, \tilde{Y})\)
3. the learner

Output:
1. \(Y'\)
2. confidence
Input / output to our debugger

Input:
1. dirty training set \((X, Y)\)
2. trusted items \((\tilde{X}, \tilde{Y})\)
3. the learner

Output:
1. \(Y'\)
2. confidence
Conceptual Formulation

\[
\min_{Y'} \|Y' - Y\|
\]
Conceptual Formulation

$$\min_{Y'} \|Y' - Y\|$$

s.t. $\hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y'_i, \theta) + \lambda \|\theta\|^2$
Conceptual Formulation

\[
\min_{Y'} \quad \|Y' - Y\| \\
\text{s.t.} \quad \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i', \theta) + \lambda \|\theta\|^2 \\
\hat{\theta}(\tilde{X}) = \tilde{Y}
\]
Conceptual Formulation

\[
\begin{align*}
\min_{Y'} & \quad \|Y' - Y\| \\
\text{s.t.} & \quad \hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i', \theta) + \lambda \|\theta\|^2 \\
& \quad \hat{\theta}(\tilde{X}) = \tilde{Y}
\end{align*}
\]

Difficult!

- combinatorial
- bilevel optimization
Combinatorial to continuous relaxation
Combinatorial to continuous relaxation

step 1. label to probability simplex

\[ y'_i \rightarrow \delta_i \in \Delta \]
Combinatorial to continuous relaxation

step 1. label to probability simplex

\[ y'_i \rightarrow \delta_i \in \Delta \]

step 2. counting to probability mass

\[ \|Y' - Y\| \rightarrow \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_i, y_i) \]
Combinatorial to continuous relaxation

step 1. label to probability simplex

\[ y'_i \rightarrow \delta_i \in \Delta \]

step 2. counting to probability mass

\[ \|Y' - Y\| \rightarrow \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i}) \]

step 3. soften postcondition

\[ \hat{\theta}(\tilde{X}) = \tilde{Y} \rightarrow \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \theta) \]
Continuous now, but still bilevel

\[
\arg\min_{\delta \in \Delta^n, \hat{\theta}} \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \hat{\theta}) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i})
\]

s.t. \[\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2\]
Removing the lower level problem

\[ \hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2 \]
Removing the lower level problem

\[ \hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2 \]

**step 1.** Assuming strong convexity, \( \hat{\theta}(\delta) \) is a unique function of \( \delta \)

\[ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \nabla_{\theta} \ell(x_i, j, \theta) + 2\lambda \theta = 0 \]
Removing the lower level problem

\[ \hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2 \]

step 1. Assuming strong convexity, \( \hat{\theta}(\delta) \) is a unique function of \( \delta \)

\[ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \nabla_{\theta} \ell(x_i, j, \theta) + 2\lambda \theta = 0 \]

step 2. Plug implicit function \( \theta(\delta) \) into upper level problem

\[ \arg\min_{\delta} \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \theta(\delta)) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i}) \]
Removing the lower level problem

\[ \hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \| \theta \|^2 \]

**step 1.** Assuming strong convexity, \( \hat{\theta}(\delta) \) is a unique function of \( \delta \)

\[ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \nabla_{\theta} \ell(x_i, j, \theta) + 2\lambda\theta = 0 \]

**step 2.** plug implicit function \( \theta(\delta) \) into upper level problem

\[ \arg\min_{\delta} \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \theta(\delta)) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i}) \]

**step 3.** compute gradient \( \nabla_{\delta} \) with implicit function theorem
Harry Potter Toy Example

data

DUTI

influence function

[Koh & Liang 2017]

nearest neighbor

label noise detection

[Nearest Neighbor 2015]

average PR

[data DUTI influence function [Koh & Liang 2017]]
Real Data Experiments

Adult Income

German Loan

MNIST flag accuracy

MNIST fix accuracy
Extension I: Debugging the ML pipeline

$\text{data } (X, Y) \rightarrow \text{learner } \ell \rightarrow \text{parameters } \lambda \rightarrow \text{model } \hat{\theta}$

$\hat{\theta} = \arg\min_{\theta \in \Theta} \ell(X, Y, \theta) + \lambda \|\theta\|$
Extension II: General Postconditions

\[ \Psi(\hat{\theta}) \]

Examples:

- “the learned model must correctly predict an important item \((\tilde{x}, \tilde{y})\)”
  \[ \hat{\theta}(\tilde{x}) = \tilde{y} \]

- “the learned model must satisfy \textbf{individual fairness}”
  \[ \forall x, x', |p(y = 1 \mid x, \hat{\theta}) - p(y = 1 \mid x', \hat{\theta})| \leq L\|x - x'\| \]
Extension III: Debugging Guarantee

- Theorem (Po-Ling Loh, Xiaomin Zhang)

If we solve the following problem to debug OLS,

$$\hat{\theta}, \hat{\delta} = \arg \min_{\theta, \delta} \left\{ \frac{1}{2n} \|y - X\theta - \delta\|_2^2 + \frac{\gamma}{2m} \|\tilde{Y} - \tilde{X}\theta\|_2^2 + \lambda \|\delta\|_1 \right\},$$

under mild conditions on $X$, $\tilde{X}$, and with sufficiently large $\lambda$, it is guaranteed that $\text{supp}(\hat{\delta}) \subseteq \text{supp}(\delta^*)$, and

$$\|\hat{\delta} - \delta^*\|_\infty \leq B(\lambda, X, \tilde{X}).$$
Applications

- Defending against data poisoning attack
- Repairing machine learning unfairness
- Interpretable explanation of test errors
Application I: Defending against data poison attack
Application I: Defending against data poison attack

- Motivating Example (ridge regression)
Application I: Defending against data poison attack

- Motivating Example (ridge regression)
- Attacker’s goal is to predict $y^*$ on $x^*$:

$$\min_{\delta \in \mathbb{R}^n} \|\delta\|_2$$

s.t. $$x^* \top \left[ X \top X + \lambda I \right]^{-1} X \top (y - \delta) = y^*$$
Application I: Defending against data poison attack

▶ Motivating Example (ridge regression)
▶ Attacker’s goal is to predict $y^*$ on $x^*$:

$$
\begin{align*}
&\min_{\delta \in \mathbb{R}^n} \|\delta\|_2 \\
&\text{s.t.} \quad x^*\top [X\top X + \lambda I]^{-1} X\top (y - \delta) = y^*
\end{align*}
$$

▶ Defender’s goal is to fix the prediction on $x^*$ to be $y^{**}$:

$$
\begin{align*}
&\min_{\delta \in \mathbb{R}^n} \|\delta\|_2 \\
&\text{s.t.} \quad x^*\top [X\top X + \lambda I]^{-1} X\top (y - \hat{\delta} + \delta) = y^{**}
\end{align*}
$$
Application I: Defending against data poison attack

Simple calculation shows that both the attacker’s and the defender’s problem is equivalent to:

\[ x^*^T [X^T X + \lambda I]^{-1} X^T \delta = y^{**} - y^* \]
Application I: Defending against data poison attack

Simple calculation shows that both the attacker’s and the defender’s problem is equivalent to:

\[ x^\top [X^\top X + \lambda I]^{-1} X^\top \delta = y^{**} - y^* \]

Message: By identifying the attacker’s target, the defender can exactly recover the attack!
Application II: Repairing Machine Learning Unfairness

- **Message:** The learned model is unfair, if and only if the training data is unfair (w.h.p.).

- **Debugging**

\[
\min_{Y'} \| Y' - Y \|
\]

s.t. \[
\hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y'_i, \theta) + \lambda \| \theta \|^2
\]

\forall x, x', |p(y = 1 \mid x, \hat{\theta}) - p(y = 1 \mid x', \hat{\theta})| \leq L \| x - x' \|
Application III: Interpretable explanation of test errors
Thanks for listening!