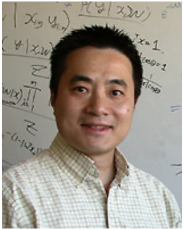
Training Set Debugging Using Trusted Items

Xuezhou Zhang

University of Wisconsin-Madison

- Joint work with



Jerry Zhu

Steve Wright

Why debug the training data?

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?

What can we do?

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

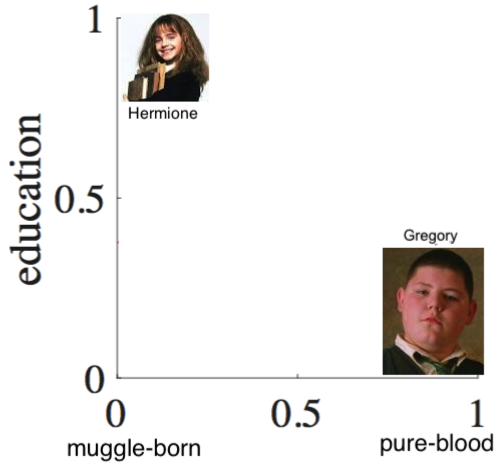
What can we do?

- 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.

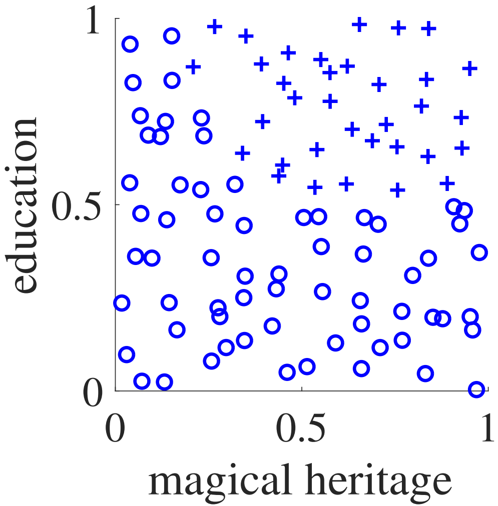
Hogwarts Alumni



Hired by the Ministry of Magic?

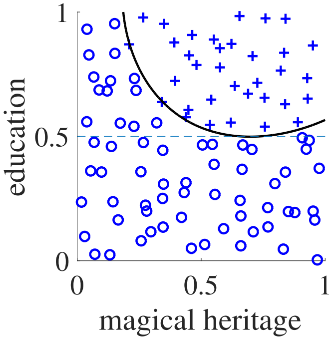
+ yes

o no



Data contain historical biases

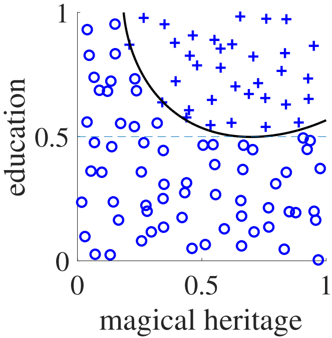
- Learned vs. ideal decision boundary



(RBF kernel logistic regression)

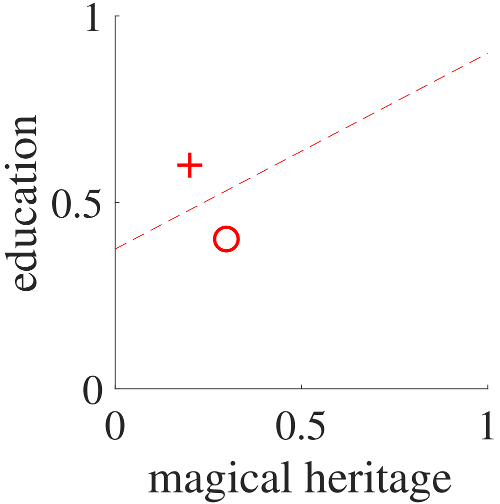
Data contain historical biases

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



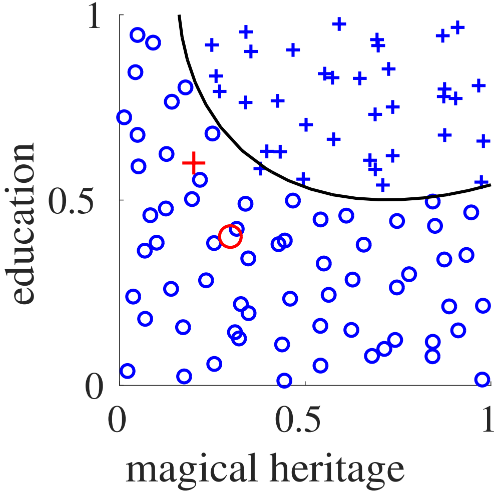
Trusted items

- 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

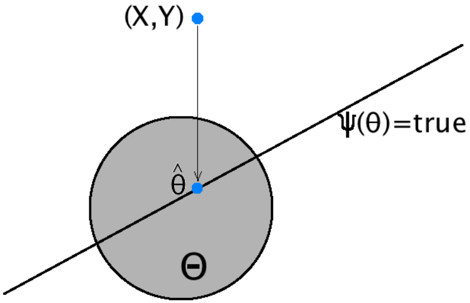
Just to learn a better model (Robust Statistics):

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

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

or equivalently, with some $\gamma >> 1$,

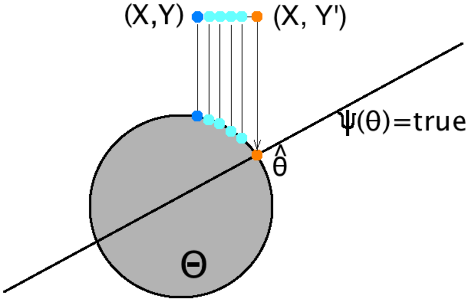
$$\min_{\theta \in \Theta} \quad \ell(X, Y, \theta) + \gamma \ell(\tilde{X}, \tilde{Y}, \theta) + \lambda \|\theta\|$$



This is our goal

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

$$\begin{aligned}
\min_{Y', \hat{\theta}} \quad & \|Y - Y'\| \\
\text{s.t.} \quad & \theta(\tilde{X}) = \tilde{Y} \\
& \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \ell(X, {\color{red}Y'}, \theta) + \lambda \|\theta\|
\end{aligned}$$



Input / output to our debugger

Input:

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

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

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

Conceptual Formulation

$$
\begin{aligned}
\min_{Y'} \quad & \|Y' - Y\| \\
\text{s.t.} \quad & \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i', \theta) + \lambda \|\theta\|^2
\end{aligned}
$$

Conceptual Formulation

$$\begin{aligned} \min_{Y'} \quad & \|Y' - Y\| \\ \text{s.t.} \quad & \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, {\color{red} y_i'}, \theta) + \lambda \|\theta\|^2 \\ & \hat{\theta}(\tilde{X}) = \tilde{Y} \end{aligned}$$

Conceptual Formulation

$$\begin{aligned} \min_{Y'} \quad & \|Y' - Y\| \\ \text{s.t.} \quad & \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, {\color{red} y_i'}, \theta) + \lambda \|\theta\|^2 \\ & \hat{\theta}(\tilde{X}) = \tilde{Y} \end{aligned}$$

Difficult!

- combinatorial
- bilevel optimization

Combinatorial to continuous relaxation

Combinatorial to continuous relaxation

step 1. label to probability simplex

$$y_i' \to \delta_i \in \Delta$$

Combinatorial to continuous relaxation

step 1. label to probability simplex

$$y_i' \to \delta_i \in \Delta$$

step 2. counting to probability mass

$$\|Y' - Y\| \to \frac{1}{n}\sum_{i=1}^{n}(1 - \delta_{i,y_i})$$

Combinatorial to continuous relaxation

step 1. label to probability simplex

$$y_i' \to \delta_i \in \Delta$$

step 2. counting to probability mass

$$\|Y' - Y\| \to \frac{1}{n}\sum_{i=1}^{n}(1 - \delta_{i,y_i})$$

step 3. soften postcondition

$$\hat{\theta}(\tilde{X}) = \tilde{Y} \to \frac{1}{m}\sum_{i=1}^{m}\ell(\tilde{x}_i, \tilde{y}_i, \theta)$$

Continuous now, but still bilevel

$$
\begin{aligned}
\underset{\delta \in \Delta^n, \hat{\theta}}{\operatorname{argmin}} \quad & \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \hat{\theta}) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i}) \\
\text{s.t.} \quad & \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2
\end{aligned}
$$

Removing the lower level problem

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2$$

Removing the lower level problem

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2$$

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

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

Removing the lower level problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2$$

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

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

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

$$\underset{\delta}{\operatorname{argmin}} \qquad \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \theta(\delta)) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i})$$

Removing the lower level problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \ell(x_i, j, \theta) + \lambda \|\theta\|^2$$

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

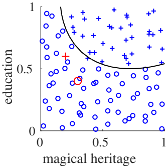
$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} \nabla_{\theta} \ell(x_i, j, \theta) + 2\lambda\theta = 0$$

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

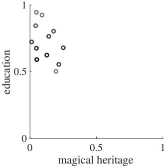
$$\underset{\delta}{\operatorname{argmin}} \qquad \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{x}_i, \tilde{y}_i, \theta(\delta)) + \gamma \frac{1}{n} \sum_{i=1}^{n} (1 - \delta_{i,y_i})$$

step 3. compute gradient ∇_{δ} with implicit function theorem

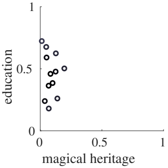
Harry Potter Toy Example



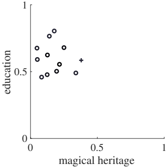
data



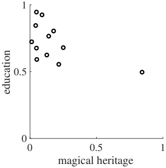
DUTI



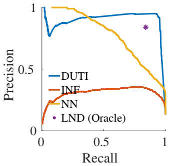
influence function [Koh & Liang 2017]



nearest neighbor

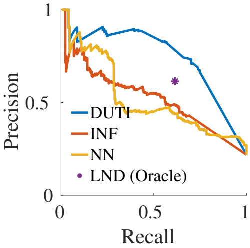


label noise detection [Bhadra & Hein 2015]

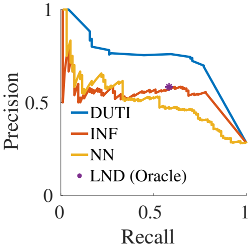


average PR

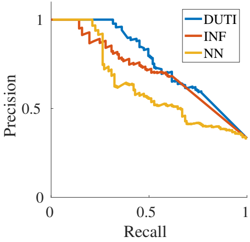
Real Data Experiments



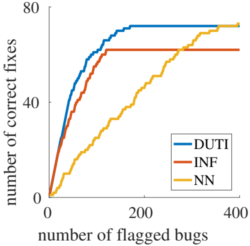
Adult Income



German Loan



MNIST flag accuracy



MNIST fix accuracy

Extension I: Debugging the ML pipeline

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

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \ell(X, Y, \theta) + \lambda \|\theta\|$$

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 **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 λ, it is guaranteed that $\mathrm{supp}(\hat{\delta}) \subseteq \mathrm{supp}(\delta^*)$, and

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

Applications

- Defending against data poisoning attack
- Repairing machine learning unfairness
- Interpretable explaination 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^*:

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

Application I: Defending against data poison attack

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

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

- Defender's goal is to fix the prediction on x^* to be y^{**}:

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

Application I: Defending against data poison attack

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

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

Application I: Defending against data poison attack

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

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

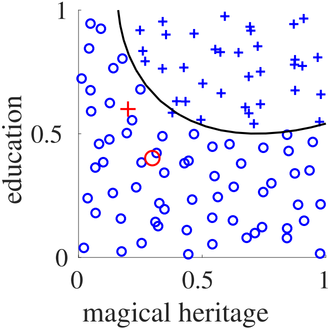
- Message: By identifying the attacker's target, the defender can exactly recover the attack!

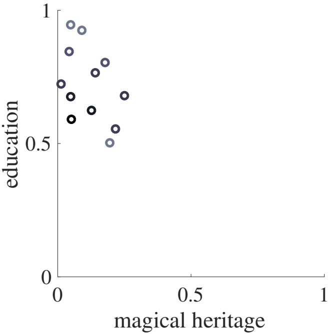
Application II: Repairing Machine Learning Unfairness

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

$$
\begin{aligned}
\min_{Y'} \quad & \|Y' - Y\| \\
\text{s.t.} \quad & \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i', \theta) + \lambda \|\theta\|^2 \\
& \forall x, x', |p(y = 1 \mid x, \hat{\theta}) - p(y = 1 \mid x', \hat{\theta})| \leq L \|x - x'\|
\end{aligned}
$$

Application III: Interpretable explaination of test errors





Thanks for listening!