Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners

WISCONSIN
UNIVERSITY OF WISCONSIN – MADISON

Shike Mei and Xiaojin Zhu

{mei, jerryzhu}@cs.wisc.edu

aaai 2015

Take-Home Message

- **We play "white hat hackers".**
- **We optimally poison the training set to mislead machine learners to specific wrong models.**
- **This is done via a bilevel optimization framework and KKT conditions.**

Identifying Attacks by the KKT Conditions

For convex and regular objective O_L and continuous search space $\mathbb{D}$ (e.g. continuous features space) , we reduce the framework to a single-level constrained optimization problem via the Karush–Kuhn–Tucker (KKT) conditions of the lower-level problem

$$\min_{D\in\mathbb{D},\theta,\lambda,\mu} \quad O_A(D,\theta)$$

KKT conditions

$$\text{s.t.} \quad \partial_\theta\left(O_L(D,\theta)+\lambda^\top \mathbf{g}(\theta)+\mu^\top \mathbf{h}(\theta)\right)=\mathbf{0}$$
$$\lambda_i g_i(\theta)=0,\ i=1\ldots m$$
$$\mathbf{g}(\theta)\le \mathbf{0},\ \mathbf{h}(\theta)=\mathbf{0},\ \lambda\ge \mathbf{0}.$$

We optimize training data D by projected gradient descent. In the t-th iteration, we update the data from $D^{(t)}$ to $D^{(t+1)}$ by

$$D^{(t+1)}=\text{Proj}_{\mathbb{D}}\left(D^{(t)}+\alpha_t\nabla_D O_A(D,\theta^{(t)})\Big|_{D=D^{(t)}}\right),$$

where $\nabla_D O_A(D,\theta) \;=\; \nabla_\theta O_A(D,\theta)^\top \frac{\partial\theta}{\partial D}$.

We assume that $\nabla_\theta O_A(D,\theta)$ can be easily calculated. To calculate $\frac{\partial\theta}{\partial D}$, we denote $i=(\theta,\lambda,\mu)$ and calculate $\frac{\partial i}{\partial D}$ by the implicit function theorem

$$\frac{\partial i}{\partial D}=-\left[\frac{\partial \mathbf{f}}{\partial\theta}\Big|\frac{\partial \mathbf{f}}{\partial\boldsymbol{\lambda}}\Big|\frac{\partial \mathbf{f}}{\partial\boldsymbol{\mu}}\right]^{-1}\left(\frac{\partial \mathbf{f}}{\partial D}\right).$$

Where $f=\mathbf{0}$ represents the equality constraints in KKT conditions and

$$\mathbf{f}(D,\theta,\lambda,\mu)=\begin{pmatrix}\partial_\theta\left(O_L(D,\theta)+\lambda^\top\mathbf{g}(\theta)+\mu^\top\mathbf{h}(\theta)\right)\\ \lambda_i g_i(\theta),\ i=1\ldots m\\ \mathbf{h}(\theta)\end{pmatrix}.$$

Bilevel Training-Set Attack Framework by Machine Teaching

Bilevel Framework

$$\min_{D\in\mathbb{D},\hat\theta_D} \quad O_A(D,\hat\theta_D) \qquad \text{Upper-level: attacker}$$

$$\text{s.t.} \quad \hat\theta_D\in\text{argmin}_{\theta\in\Theta}\ O_L(D,\theta) \quad \text{s.t. } \mathbf{g}(\theta)\le\mathbf{0},\ \mathbf{h}(\theta)=\mathbf{0}. \qquad \text{Lower-level: learner}$$

- Using **bilevel** optimization to unify the attacker's goal and the learner's response (learner's objective function O_L).
- Closely related to **machine teaching**, which focuses on maximally influencing/educating a human learner by designing the optimal training set/lesson.
- Bilevel problem is **NP-hard** in general, but for a broad family of attack settings we have efficient solutions by using the KKT conditions.

Attacker

$\mathbb{D}$	Search space of feasible manipulations, e.g. data poisoned within budget
$O_A(D,\hat\theta_D)$	Overall attacker objective function, i.e. $O_A(D,\hat\theta_D)=R_A(\hat\theta_D)+E_A(D,D_0)$
$R_A(\hat\theta_D)$	Attacker risk function, e.g. $R_A(\hat\theta_D)=\lVert\hat\theta_D-\theta^*\rVert$
$E_A(D,D_0)$	Attacker effort function, e. g. $E_A(D,D_0)=\lVert X-X_0\rVert_F$

Learner

D	Training Data
Θ	The hypothesis space
$\Omega(\theta)$	Empirical risk function
$R_L(D,\theta)$	Regularizer
g and h	*Constraint functions, can be nonlinear*
$\hat\theta_D$	The learned model

$$\hat\theta_D\in\text{argmin}_{\theta\in\Theta} \quad \underbrace{R_L(D,\theta)+\lambda\Omega(\theta)}_{O_L(D,\theta)}$$
$$\text{s.t.} \quad g_i(\theta)\le 0,\ i=1\ldots m$$
$$h_i(\theta)=0,\ i=1\ldots p$$

Examples

SVM Learner

$$O_L(D,\mathbf{w},b,\xi) = \frac{1}{2}\lVert\mathbf{w}\rVert_2^2+C\sum_i\xi_i$$
$$g_i = 1-\xi_i-y_i(\mathbf{x}_i^\top\mathbf{w}+b)$$
$$g_{i+n} = -\xi_i$$

We convert it to the corresponding KKT conditions (for w_j)

$$w_j-\alpha_i\sum_i\mathbb{I}_1(1-y_i(\mathbf{x}_i^\top\mathbf{w}+b)\ge 0)y_ix_{ij}=0$$

The attacker wants to make the learned weight close to the target weight w^* by risk $R_A(\hat{\mathbf{w}}_D)=\frac{1}{2}\lVert\hat{\mathbf{w}}_D-\mathbf{w}^*\rVert_2^2$ and to minimally modify features by attacker's effort $E_A(D,D_0)=\frac{\lambda}{2}\lVert\mathbf{X}-\mathbf{X}_0\rVert_F^2$ Combining them we get the KKT single-level framework

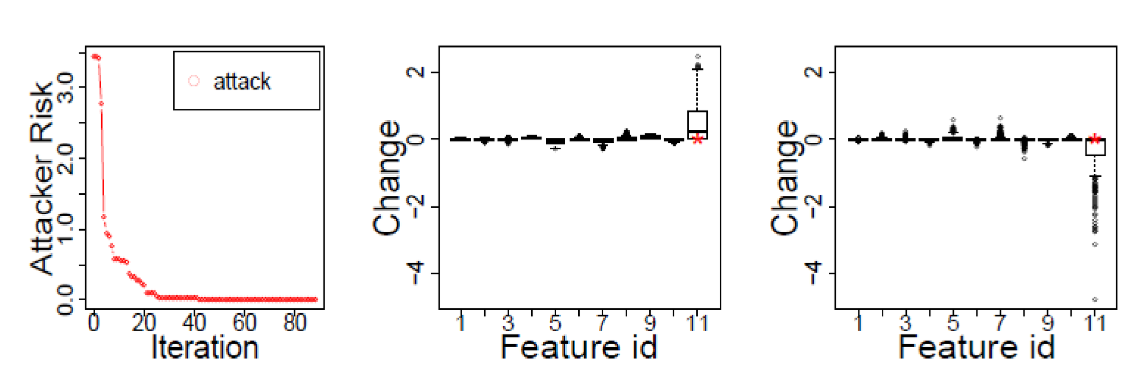
$$\min_{D\in\mathbb{D},\mathbf{w}} \quad \frac{1}{2}\lVert\mathbf{w}-\mathbf{w}^*\rVert_2^2+\frac{\lambda}{2}\lVert\mathbf{X}-\mathbf{X}_0\rVert_F^2$$
$$\text{s.t.} \quad w_j-\alpha_i\sum_i\mathbb{I}_1(1-y_i(\mathbf{x}_i^\top\mathbf{w}+b)\ge 0)y_ix_{ij}=0.$$

Experiment

Learning task: given features of wine the learner should classify good/bad wine.

Target weight: only correlated with feature "alcohol" (the 11-th feature).

Attack behavior is mainly increasing/decreasing the 11-th feature for good/bad data.



(a) attacker risk R_A (b) feature changes on positive data (c) feature changes on negative data

Logistic Regression Learner

$$O_L(D,\mathbf{w},b)=\sum_i\log\left(1+\exp(-y_i\hat h_i)\right)+\frac{\mu}{2}\lVert\mathbf{w}\rVert_2^2$$

KKT Conditions

$$\sum_i-(1-\sigma(y_i\hat h_i))y_ix_{ij}+\mu w_j = 0.$$

The attacker has the same risk and effort function as in SVM. Combining the risk, effort and the KKT conditions we get

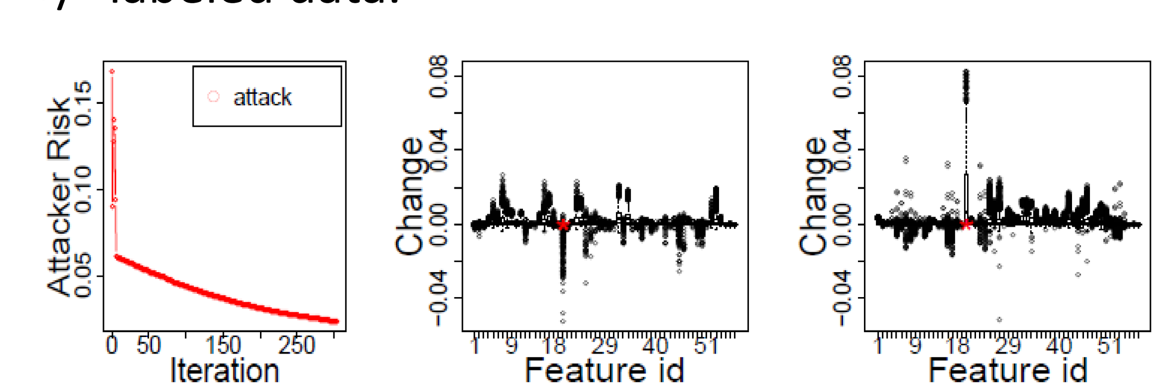
$$\min_{D\in\mathbb{D},\mathbf{w}} \quad \frac{1}{2}\lVert\mathbf{w}-\mathbf{w}^*\rVert_2^2+\frac{\lambda}{2}\lVert\mathbf{X}-\mathbf{X}_0\rVert_F^2$$
$$\text{s.t.} \quad w_j-\alpha_i\sum\mathbb{I}_1(1-y_i(\mathbf{x}_i^\top\mathbf{w}+b)\ge 0)y_ix_{ij}=0.$$

Experiment

Learning task: given word frequencies in emails, the learner should classify them as spam/not spam.

Target weight: The attacker wants to make the weight on feature "credit frequency" close to zero with minimal change of other weights. So we set feature "credit frequency" in training data to zero and refer the learned weight as the target weight.

The "credit" feature was increased/decreased for +/- labeled data.



(a) attacker risk R_A (b) feature changes on positive data (c) feature changes on negative data

Ordinary Least Square (OLS)

Ordinary Least Square (OLS) the KKT conditions is the objective it self

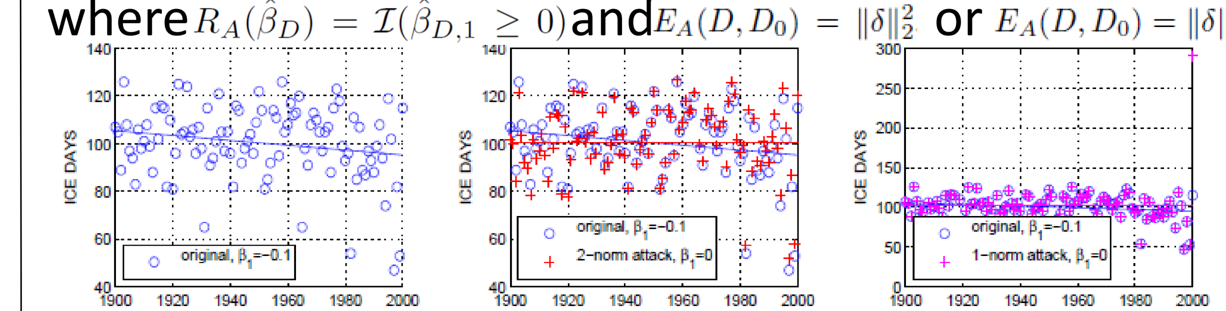
$$O_L(D,\beta)=\lVert\mathbf{y}-X\beta\rVert^2.$$

Learning task: learn the trend of #frozen days of Lake Mendota. **Attack goal**: hide the lake warming trend. Different attacker effort functions lead to different attack behaviors.

The attacker wants to make the first dimensional weight $\hat\beta_{D,1}$ non-negative and to minimize response modification δ (measured by l1 and l2 norm)

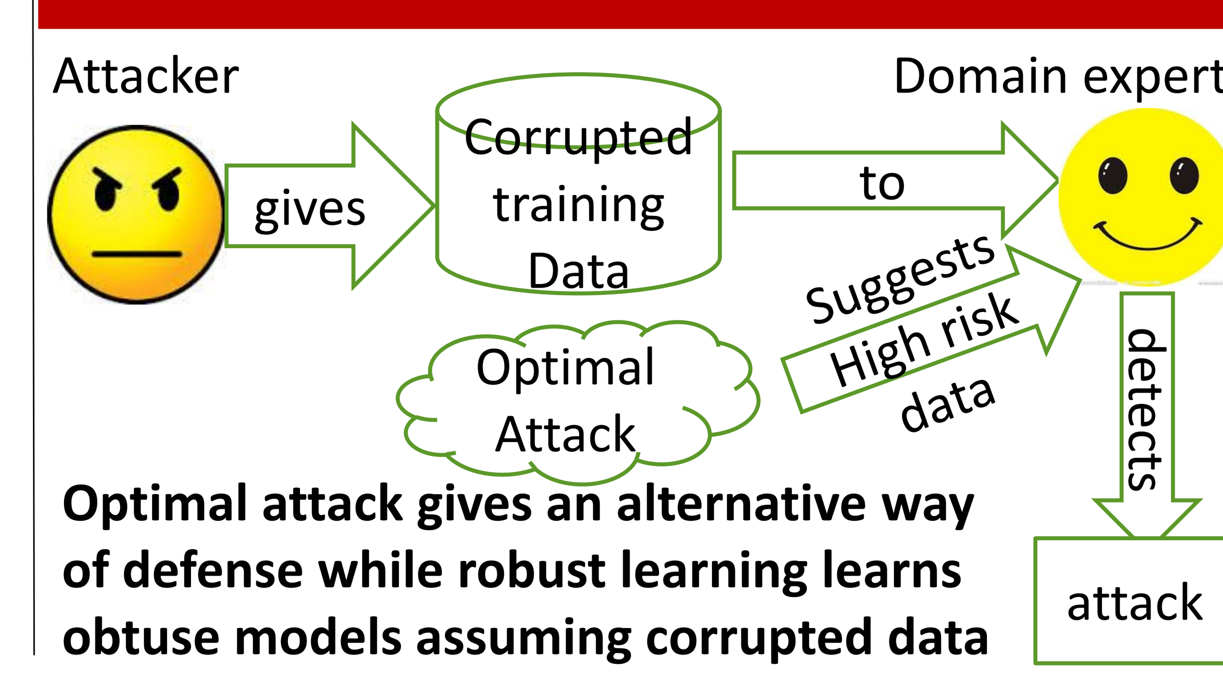
$$\min_\delta R_A(\hat\beta_D)+E_A(\delta)$$

where $R_A(\hat\beta_D)=\mathcal{I}(\hat\beta_{D,1}\ge 0)$ and $E_A(D,D_0)=\lVert\delta\rVert_2^2$ or $E_A(D,D_0)=\lVert\delta\rVert_1$



OLS on D_0 — ℓ_2 attacker effort — ℓ_1 attacker effort

Defense



Optimal attack gives an alternative way of defense while robust learning learns obtuse models assuming corrupted data